Lecture Notes in Artificial Intelligence 4029

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Leszek Rutkowski Ryszard Tadeusiewicz
Lotfi A. Zadeh Jacek Zurada (Eds.)

Artificial Intelligence and Soft Computing – ICAISC 2006

8th International Conference
Zakopane, Poland, June 25-29, 2006
Proceedings

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Leszek Rutkowski
Częstochowa University of Technology
Armii Krajowej 36, 42-200 Częstochowa, Poland
E-mail: lrutko@kik.pcz.czest.pl

Ryszard Tadeusiewicz
AGH University of Science and Technology
Mickiewicza 30, 30-059 Kraków, Poland
E-mail: rtad@agh.edu.pl

Lotfi A. Zadeh
University of California
Dept. of Electrical Engineering and Computer Sciences, and
Berkeley Initiative in Soft Computing (BISC)
Berkeley, CA 94720-1776, California, USA
E-mail: zadeh@eecs.berkeley.edu

Jacek Zurada
University of Louisville
Department of Electrical Engineering
Louisville, KY 40292, USA
E-mail: jmzurada02@louisville.edu

Library of Congress Control Number: 2006928069

CR Subject Classification (1998): I.2, F.4.1, F.1, F.2, I.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-35748-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-35748-3 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11785231 06/3142 5 4 3 2 1 0

Preface

This volume constitutes the proceedings of the 8th Conference on Artificial Intelligence and Soft Computing, ICAISC 2006, held in Zakopane, Poland in June 25-29, 2006. The conference was organized by the Polish Neural Network Society in cooperation with the Academy of Humanities and Economics in Łódź, the Department of Computer Engineering at the Czestochowa University of Technology, and the IEEE Computational Intelligence Society – Poland Chapter. The previous conferences took place in Kule (1994), Szczyrk (1996), Kule (1997) and Zakopane (1999, 2000, 2002, 2004) and attracted a large number of papers and internationally recognized speakers: Lotfi A. Zadeh, Shun-ichi Amari, Daniel Amit, Piero P. Bonissone, Zdzislaw Bubnicki, Andrzej Cichocki, Wlodzislaw Duch, Jerzy Grzymala-Busse, Kaoru Hirota, Janusz Kacprzyk, Laszlo T. Koczy, Soo-Young Lee, Robert Marks, Evangelia Micheli-Tzanakou, Erkki Oja, Witold Pedrycz, Sarunas Raudys, Enrique Ruspini, Jorg Siekman, Roman Slowinski, Ryszard Tadeusiewicz, Shiro Usui, Ronald Y. Yager, Syozo Yasui and Jacek Zurada. The aim of this conference is to build a bridge between traditional artificial intelligence techniques and recently developed soft computing techniques. It was pointed out by Lotfi A. Zadeh that "Soft Computing (SC) is a coalition of methodologies which are oriented toward the conception and design of information/intelligent systems. The principal members of the coalition are: fuzzy logic (FL), neurocomputing (NC), evolutionary computing (EC), probabilistic computing (PC), chaotic computing (CC), and machine learning (ML). The constituent methodologies of SC are, for the most part, complementary and synergistic rather than competitive". This volume presents both traditional artificial intelligence methods and soft computing techniques. Our goal is to bring together scientists representing both traditional artificial intelligence approaches and soft computing techniques. The volume is divided into eight parts:

- Neural Networks and Their Applications
- Fuzzy Systems and Their Applications
- Evolutionary Algorithms and Their Applications
- Rough Sets
- Classification and Clustering
- Image Analysis and Robotics
- Bioinformatics and Medical Applications
- Various Problems of Artificial Intelligence

The conference attracted a total of 400 submissions from 41 countries and after the review process, 128 papers were accepted for publication in this volume. I would like to thank our participants, invited speakers and reviewers of the papers for their scientific and personal contribution to the conference. I also thank Alfred Hofmann editor-in-chief of Lecture Notes in Computer Science/Artificial Intelligence and the rest of Springer's LNCS team for their cooperation in the

preparation of this volume. Finally I thank my co-workers Łukasz Bartczuk, Piotr Dziwiński, Marcin Gabryel, Marcin Korytkowski and Rafał Scherer for their enormous efforts to make the conference a very successful event.

June 2006

Leszek Rutkowski
President of the Polish Neural Network Society

Organization

ICAISC 06 was organized by the Polish Neural Network Society in cooperation with the Academy of Humanities and Economics in Łódź, the Department of Computer Engineering at the Czestochowa University of Technology, and the IEEE Computational Intelligence Society – Poland Chapter.

Chairpersons

Honorary chairmen	Lotfi Zadeh (USA)
	Zdzislaw Bubnicki (Poland)
	Zdzisław Pawlak (Poland)
	Jacek Żurada (USA)
General chairman	Leszek Rutkowski (Poland)
Co-chairmen	Włodzisław Duch (Poland)
	Janusz Kacprzyk (Poland)
	Józef Korbicz (Poland)
	Ryszard Tadeusiewicz (Poland)

International Program Committee

Robert Babuska, Netherlands
Bernadette Bouchon-Meunier, France
Juan Luis Castro, Spain
Yen-Wei Chen, Japan
Andrzej Cichocki, Japan
Krzysztof Cios, USA
Oscar Cordon, Spain
Bernard De Baets, Belgium
Juan José González de la Rosa, Spain
Nabil Derbel, Tunisia
David Elizondo, UK
David B. Fogel, USA
Adam Gaweda, USA
Jerzy W. Grzymala-Busse, USA
Petr Hajek, Czech Republic
Saman Halgamuge, Australia
Rainer Hampel, Germany
Yoichi Hayashi, Japan
Francisco Herrera, Spain
Kaoru Hirota, Japan
Mo Jamshidi, USA
Robert John, UK
Nikola Kasabov, New Zealand
Okyay Kaynak, Turkey
Vojislav Kecman, New Zealand
Etienne Kerre, Belgium
Frank Klawonn, Germany
Laszlo Koczy, Hungary
Rudolf Kruse, Germany
Boris V. Kryzhanovsky, Russia
Adam Krzyzak, Canada
Vera Kurkova, Czech Republic
Soo-Young Lee, Korea
Zhi-Qiang Liu, Hong Kong
Kurosh Madani, France
Luis Magdalena, Spain
Jerry M. Mendel, USA
Radko Mesiar, Slovakia
Zbigniew Michalewicz, Australia
Sudip Misra, Canada

Detlef D. Nauck, Germany
Erkki Oja, Finland
Witold Pedrycz, Canada
Vincenzo Piuri, Italy
Danil Prokhorov, USA
Sarunas Raudysm, Lithuania
Vladimir Redko, Russia
Raul Rojas, Germany
Imre J. Rudas, Hungary
Norihide Sano, Japan
Rudy Setiono, Singapore
Peter Sincak, Slovakia
Tomasz G. Smoliski, USA
Ron Sun, USA
Hideyuki Takagi, Japan
Yury Tiumentsev, Russia
Vicenc Torra, Spain
Burhan Turksen, Canada
Shiro Usui, Japan
Lipo Wang, Singapore
Michael Wagenknecht, Germany
Jun Wang, Hong Kong
Bernard Widrow, USA
Bogdan M. Wilamowski, USA
Donald C. Wunsch, USA
Ronald R. Yager, USA
John Yen, USA

Polish Program Committee

Jarosław Arabas
Leon Bobrowski
Leonard Bolc
Tadeusz Burczyński
Andrzej Cader
Wojciech Cholewa
Paweł Cichosz
Ewa Dudek-Dyduch
Andrzej Dzieliński
Roman Galar
Marian Bolesław Gorzałczany
Krzysztof Grąbczewski
Zygmunt Hasiewicz
Zdzisław Hippe
Adrian Horzyk
Andrzej Janczak
Norbert Jankowski
Jerzy Józefczyk
Tadeusz Kaczorek
Wadysław Kamiński
Jacek Kluska
Leonid Kompanets
Przemysław Korohoda
Jacek Koronacki
Witold Kosiński
Jan M. Kościelny
Zdzisaw Kowalczuk
Krzysztof Krawiec
Roman Kulikowski
Juliusz Kulikowski
Marek Kurzyński
Halina Kwaśnicka
Antoni Ligęza
Jacek Łęski
Bohdan Macukow
Witold Malina
Krzysztof Malinowski
Jacek Mańdziuk
Andrzej Materka
Zbigniew Mikrut
Wojciech Moczulski
Antoine Naud
Edward Nawarecki
Antoni Niederliński
Robert Nowicki
Andrzej Obuchowicz
Marek R. Ogiela
Stanisaw Osowski
Andrzej Pieczyński
Andrzej Piegat
Lech Polkowski
Anna M. Radzikowska
Ewaryst Rafajłowicz
Danuta Rutkowska

Robert Schaefer
Paweł Sewastianow
Andrzej Skowron
Ewa Skubalska-Rafajłowicz
Roman Słowiński
Czesław Smutnicki
Jerzy Stefanowski
Paweł Strumiłło
Eulalia J. Szmidt
Jerzy Świątek
Tomasz Walkowiak
Sławomir Wiak
Maciej Wygralak
Roman Wyrzykowski
Sławomir Zadrożny
Jerzy S. Zieliński

Organizing Committee

Rafał Scherer, Secretary
Łukasz Bartczuk, Organizing Committee Member
Piotr Dziwiński, Organizing Committee Member
Marcin Gabryel, Organizing Committee Member
Marcin Korytkowski, Databases and Internet Submissions

Reviewers

Ajith Abraham
Rafał Adamczak
Rafal A. Angryk
Jarosław Arabas
Tomasz Babczyński
Andrzej Bargiela
Łukasz Bartczuk
Leon Bobrowski
Piotr Boguś
Bernadette Bouchon-Meunier
Tadeusz Burczyński
Bohdan S. Butkiewicz
Krzysztof Cetnarowicz
Maiga Chang
Yen-Wei Chen
Wojciech Cholewa
Michał Choraś
Ryszard Choraś
Andrzej Cichocki
Paweł Cichosz
Krzysztof Cios
Oscar Cordon
Boguslaw Cyganek
Ireneusz Czarnowski
Juan José González de la Rosa
Nabil Derbel
Grzegorz Dobrowolski
Włodzisław Duch
Ewa Dudek-Dyduch
Ludmiła Dymowa
Andrzej Dzieliński
Piotr Dziwiński
David Elizondo
Mariusz Flasiński
David Fogel
Marcin Gabryel
Roman Galar
Adam Galuszka
Adam Gawęda
Mariusz Giergiel
Fernando Gomide
Marian Gorzałczany
Krzysztof Grąbczewski
Wlodzimierz Greblicki
Maciej Grzenda
Jerzy Grzymala-Busse
Saman Halgamuge
Zygmunt Hasiewicz
Yoichi Hayashi
Francisco Herrera
Zdzisław Hippe
Kaoru Hirota
Adrian Horzyk
Edward Hrynkiewicz
Andrzej Janczak
Szymon Jaroszewicz
Władysław Kamiński
Iwona Karcz-Duleba
Vojislav Kecman
Etienne Kerre
Frank Klawonn
Jacek Kluska
Laszlo Koczy
Leonid Kompanets
Jacek Konopacki
Józef Korbicz
Przemyslaw Korohoda
Jacek Koronacki
Korytkowski Marcin
Marcin Korzeń
Witold Kosiński
Kościelny Jan M.
Zdzisław Kowalczuk

Jaroslaw Kozlak
Krzysztof Krawiec
Boris V. Kryzhanovsky
Adam Krzyzak
Juliusz Kulikowski
Vera Kurkova
Jan Kusiak
Antoni Ligęza
Zhi-Qiang Liu
Jacek Łęski
Bohdan Macukow
Kurosh Madani
Luis Magdalena
Witold Malina
Krzysztof Malinowski
Jacek Mańdziuk
Urszula Markowska-Kaczmar
Andrzej Materka
Jerry M. Mendel
Zbigniew Michalewicz
Zbigniew Mikrut
Wojciech Moczulski
Wojciech Mokrzycki
Antoine Naud
Edward Nawarecki
Antoni Niederliński
Robert Nowicki
Andrzej Obuchowicz
Marek R. Ogiela
Erkki Oja
Osowski Stanisław
Krzysztof Patan
Miroslaw Pawlak
Witold Pedrycz
Andrzej Pieczyński
Andrzej Piegat
Zbigniew Pietrzykowski
Jakub Piskorski
Vincenzo Piuri
Agata Pokropińska
Danil Prokhorov
Piotr Prokopowicz
Andrzej Przybył
Anna Radzikowska
Ewaryst Rafajłowicz
Artur Rataj
Sarunas Raudys
Vladimir Redko
Izabela Rejer
Leszek Rolka
Leszek Rutkowski
Norihide Sano
Robert Schaefer
Rafał Scherer
Rudy Setiono
Paweł Sevastjanov
Władysław Skarbek
Andrzej Skowron
Krzysztof Skrzypczyk
Ewa Skubalska-Rafajłowicz
Tomasz Smolinski
Czesław Smutnicki
Aleksander Sokołowski
Jerzy Stefanowski
Magdalena Stobinska
Ewa Straszecka
Barbara Strug
Paweł Strumiłło
Ron Sun
Jakub Swacha
Andrzej Szalas
Eulalia Szmidt
Piotr Szymak
Przemysław Śliwiński
Grażyna Ślusarczyk
Roman Śmierzchalski
Hideyuki Takagi
Yury Tiumentsev
Vicenc Torra
Bogdan Trawinski
Mariusz Urbański
Shiro Usui
Michael Wagenknecht
Anna Walaszek-Babiszewska
Tomasz Walkowiak
Jun Wang
Sławomir Wiak
Bernard Widrow
Bogdan M. Wilamowski
Marcin Witczak
Marek Wojciechowski
Michal Wozniak
Robert Wrembel
Maciej Wygralak
Sławomir Zadrożny
Jerzy Stanisław Zieliński

Table of Contents

Neural Networks and Their Applications

Multichannel Data Aggregation by Layers of Formal Neurons
Leon Bobrowski 1

Estimation of Distribution Algorithm for Optimization of Neural Networks for Intrusion Detection System
Yuehui Chen, Yong Zhang, Ajith Abraham 9

Neural Network Implementation in Reprogrammable FPGA Devices – An Example for MLP
Marek Gorgoń, Mateusz Wrzesiński 19

A New Approach for Finding an Optimal Solution and Regularization by Learning Dynamic Momentum
Eun-Mi Kim, Jong Cheol Jeong, Bae-Ho Lee 29

Domain Dynamics in Optimization Tasks
Boris Kryzhanovsky, Bashir Magomedov 37

Nonlinear Function Learning by the Normalized Radial Basis Function Networks
Adam Krzyżak, Dominik Schäfer 46

Sum and Product Kernel Regularization Networks
Petra Kudová, Terezie Šámalová 56

Chaotic Cellular Neural Networks with Negative Self-feedback
Wen Liu, Haixiang Shi, Lipo Wang, Jacek M. Zurada 66

An Efficient Nonlinear Predictive Control Algorithm with Neural Models and Its Application to a High-Purity Distillation Process
Maciej Ławryńczuk, Piotr Tatjewski 76

Creativity of Neural Networks
Urszula Markowska-Kaczmar, Katarzyna Czeczot 86

Speed Up of the SAMANN Neural Network Retraining
Viktor Medvedev, Gintautas Dzemyda 94

Application of Neural Networks in Chain Curve Modelling
Andrzej Piegat, Izabela Rejer, Marek Mikolajczyk 104

RBF Nets in Faults Localization
Ewaryst Rafajłowicz 113

A Hypertube as a Possible Interpolation Region of a Neural Model
Izabela Rejer, Marek Mikolajczyk 123

RBF Neural Network for Probability Density Function Estimation and Detecting Changes in Multivariate Processes
Ewa Skubalska-Rafajłowicz 133

Fast Orthogonal Neural Networks
Bartłomiej Stasiak, Mykhaylo Yatsymirskyy 142

AI Methods in Solving Systems of Interval Linear Equations
Nguyen Hoang Viet, Michal Kleiber 150

A Fast and Numerically Robust Neural Network Training Algorithm
Youmin Zhang 160

Fuzzy Systems and Their Applications

On Interpretation of Non-atomic Values and Induction of Decision Rules in Fuzzy Relational Databases
Rafal A. Angryk 170

A Genetic-Programming-Based Approach for the Learning of Compact Fuzzy Rule-Based Classification Systems
Francisco Jos Berlanga, Mara Jos del Jesus, Mara Jos Gacto, Francisco Herrera 182

Performance Evaluation of Fuzzy-Neural HTTP Request Distribution for Web Clusters
Leszek Borzemski, Krzysztof Zatwarnicki 192

Fuzzy Approach to Correlation Function
Bohdan S. Butkiewicz 202

A Method for Designing Flexible Neuro-fuzzy Systems
Krzysztof Cpalka 212

Deterministic Annealing Integrated with ε-Insensitive Learning in Neuro-fuzzy Systems
Robert Czabański 220

Transformation Lemma on Analytical Modeling Via Takagi-Sugeno Fuzzy System and Its Applications
Jacek Kluska 230

Combining Logical-Type Neuro-fuzzy Systems
Marcin Korytkowski, Robert Nowicki, Leszek Rutkowski, Rafał Scherer 240

On Fuzzy Number Calculus and Some Application
Witold Kosiński 250

Combination of Fuzzy TOPSIS and Fuzzy Ranking for Multi Attribute Decision Making
Mohammad Reza Mehregan, Hossein Safari 260

Flow Graphs and Decision Tables with Fuzzy Attributes
Alicja Mieszkowicz-Rolka, Leszek Rolka 268

Elements of the Type-2 Semantics in Summarizing Databases
Adam Niewiadomski, Michał Bartyzel 278

Isolines of Statistical Information Criteria for Relational Neuro-fuzzy System Design
Agata Pokropińska, Robert Nowicki, Rafał Scherer 288

Adjusting Software-Intensive Systems Developed by Using Software Factories and Fuzzy Features
Silva Robak, Andrzej Pieczyński 297

Boosting Ensemble of Relational Neuro-fuzzy Systems
Rafał Scherer 306

An Application of Intuitionistic Fuzzy Set Similarity Measures to a Multi-criteria Decision Making Problem
Eulalia Szmidt, Janusz Kacprzyk 314

Evolutionary Algorithms and Their Applications

Additive Sequential Evolutionary Design of Experiments
Balazs Balasko, Janos Madar, Janos Abonyi 324

A New Inter-island Genetic Operator for Optimization Problems with Block Properties
Wojciech Bożejko, Mieczysław Wodecki 334

Multiobjective Design Optimization of Electrostatic Rotary Microactuators Using Evolutionary Algorithms
Paolo Di Barba, Sławomir Wiak 344

Evolutionary Learning of Mamdani-Type Neuro-fuzzy Systems
Marcin Gabryel, Leszek Rutkowski 354

Study of Objective Functions in Fuzzy Job-Shop Problem
Inés González-Rodríguez, Camino R. Vela, Jorge Puente 360

Scheduling with Memetic Algorithms over the Spaces of Semi-active and Active Schedules
Miguel A. González, Camino R. Vela, Ramiro Varela 370

Chaos Detection with Lyapunov Exponents in Dynamical System Generated by Evolutionary Process
Iwona Karcz-Dulęba 380

Improving Evolutionary Multi-objective Optimization Using Genders
Zdzislaw Kowalczuk, Tomasz Bialaszewski 390

Evolutionary Learning of Linear Trees with Embedded Feature Selection
Marek Krętowski, Marek Grześ 400

Solving the Balanced Academic Curriculum Problem with an Hybridization of Genetic Algorithm and Constraint Propagation
Tony Lambert, Carlos Castro, Eric Monfroy, Frédéric Saubion 410

A Graph-Based Genetic Algorithm for the Multiple Sequence Alignment Problem
Heitor S. Lopes, Guilherme L. Moritz 420

Improved Multi-Objective Diversity Control Oriented Genetic Algorithm
Theera Piroonratana, Nachol Chaiyaratana 430

Directional Distributions and Their Application to Evolutionary Algorithms
Przemysław Prętki, Andrzej Obuchowicz 440

Adaptive Inertia Weight Particle Swarm Optimization
Zheng Qin, Fan Yu, Zhewen Shi, Yu Wang 450

Estimation of the Evolution Speed for the Quasispecies Model: Arbitrary Alphabet Case
Vladimir Red'ko, Yuri Tsoy 460

Partitioning of VLSI Circuits on Subcircuits with Minimal Number of Connections Using Evolutionary Algorithm
Adam Słowik, Michał Białko 470

Genetic Approach to Modeling of a Dispatcher in Discrete Transport Systems
Tomasz Walkowiak, Jacek Mazurkiewicz 479

Rough Sets

Interactive Analysis of Preference-Ordered Data Using Dominance-Based Rough Set Approach
Jerzy Błaszczyński, Krzysztof Dembczyński, Roman Słowiński 489

Additive Preference Model with Piecewise Linear Components Resulting from Dominance-Based Rough Set Approximations
Krzysztof Dembczyński, Wojciech Kotłowski, Roman Słowiński 499

Induction of Decision Rules Using Minimum Set of Descriptors
Andrzej Dominik, Zbigniew Walczak 509

Comparison of Information Theoretical Measures for Reduct Finding
Szymon Jaroszewicz, Marcin Korzeń 518

Rough Approximation Operations Based on IF Sets
Anna Maria Radzikowska 528

Relationships Between Concept Lattice and Rough Set
Hong Wang, Wen-Xiu Zhang 538

Classification and Clustering

Extended SMART Algorithms for Non-negative Matrix Factorization
Andrzej Cichocki, Shun-ichi Amari, Rafal Zdunek, Raul Kompass, Gen Hori, Zhaohui He 548

MAICS: Multilevel Artificial Immune Classification System
Michal Bereta, Tadeusz Burczynski 563

Selection of Prototype Rules: Context Searching Via Clustering
Marcin Blachnik, Włodzisław Duch, Tadeusz Wieczorek 573

Committee Machine for Road-Signs Classification
Bogusław Cyganek 583

Cluster Analysis Via Dynamic Self-organizing Neural Networks
Marian B. Gorzałczany, Filip Rudziński 593

Learning Vector Quantization Classification with Local Relevance Determination for Medical Data
Barbara Hammer, Thomas Villmann, Frank-Michael Schleif, Cornelia Albani, Wieland Hermann 603

Genetically Evolved Trees Representing Ensembles
Ulf Johansson, Tuve Löfström, Rikard König, Lars Niklasson 613

Sequential Classification Via Fuzzy Relations
Marek Kurzynski, Andrzej Zolnierek 623

Attention Improves the Recognition Reliability of Backpropagation Network
Zbigniew Mikrut, Agata Piaskowska 633

An Accurate MDS-Based Algorithm for the Visualization of Large Multidimensional Datasets
Antoine Naud 643

The Multi-Agent System for Prediction of Financial Time Series
Šarūnas Raudys, Indre Zliobaite 653

Visualization of Single Clusters
Frank Rehm, Frank Klawonn, Rudolf Kruse 663

Dynamic Data Condensation for Classification
Dymitr Ruta 672

Handwriting Recognition Accuracy Improvement by Author Identification
Jerzy Sas 682

Adaptive Potential Active Hypercontours
Arkadiusz Tomczyk, Piotr S. Szczepaniak 692

KIDBSCAN: A New Efficient Data Clustering Algorithm
Cheng-Fa Tsai, Chih-Wei Liu 702

Image Analysis and Robotics

Localization and Extraction of the Optic Disc Using the Fuzzy Circular Hough Transform
Marianne Blanco, Manuel G. Penedo, Noelia Barreira, Marta Penas, Maria Jose Carreira 712

Object Recognition for Obstacle Avoidance in Mobile Robots
José M. Bolanos, Wilfredis Medina Meléndez, Leonardo Fermín, José Cappelletto, Gerardo Fernández-López, Juan C. Grieco 722

Gait Synthesis and Modulation for Quadruped Robot Locomotion Using a Simple Feed-Forward Network
Jose Cappelletto, Pablo Estevez, Wilfredis Medina, Leonardo Fermin, Juan M. Bogado, Juan C. Grieco, Gerardo Fernandez-Lopez 731

A Two-Stage Fuzzy Filtering Method to Restore Images Contaminated by Mixed Impulse and Gaussian Noises
Jyh-Yeong Chang, Shih-Mao Lu 740

Determination of the Optimal Seam-Lines in Image Mosaicking with the Dynamic Programming (DP) on the Converted Cost Space
Jaechoon Chon, Hyongsuk Kim 750

Symmetry-Based Salient Points Detection in Face Images
Michał Choraś, Tomasz Andrysiak 758

Cellular Neural Networks and Dynamic Enhancement for Cephalometric Landmarks Detection
Daniela Giordano, Rosalia Leonardi, Francesco Maiorana, Concetto Spampinato 768

Adaptive Critic Neural Networks for Identification of Wheeled Mobile Robot
Zenon Hendzel 778

A New Chromatic Color Image Watermarking and Its PCA-Based Implementation
Thai Duy Hien, Zensho Nakao, Kazuyoshi Miyara, Yasunori Nagata, Yen Wei Chen 787

Human Identification Based on Fingerprint Local Features
Maciej Hrebień, Józef Korbicz 796

Genetic Graph Programming for Object Detection
Krzysztof Krawiec, Patryk Lijewski 804

Selective Motion Analysis Based on Dynamic Visual Saliency Map Model
Inwon Lee, Sang-Woo Ban, Kunihiko Fukushima, Minho Lee 814

Efficient Ant Reinforcement Learning Using Replacing Eligibility Traces
SeungGwan Lee, SeokMi Hong 823

Face Recognition Using Correlation Between Illuminant Context
Mi Young Nam, Battulga Bayarsaikhan, Phill Kyu Rhee 833

An Efficient Face and Eye Detector Modeling in External Environment
Mi Young Nam, Eun Jin Koh, Phill Kyu Rhee 841

Keypoints Derivation for Object Class Detection with SIFT Algorithm
Krzysztof Slot, Hyongsuk Kim 850

Gray Image Contrast Enhancement by Optimal Fuzzy Transformation
Roman Vorobel, Olena Berehulyak 860

Non-negative Matrix Factorization with Quasi-Newton Optimization
Rafal Zdunek, Andrzej Cichocki 870

Bioinformatics and Medical Applications

Active Mining Discriminative Gene Sets
Feng Chu, Lipo Wang 880

A Novel Approach to Image Reconstruction from Discrete Projections Using Hopfield-Type Neural Network
Robert Cierniak 890

Leukemia Prediction from Gene Expression Data—A Rough Set Approach
Jianwen Fang, Jerzy W. Grzymala-Busse 899

Random Forest of Dipolar Trees for Survival Prediction
Małgorzata Krętowska 909

Interpretation of Medical Images Based on Ontological Models
Juliusz L. Kulikowski 919

Fuzzy Logic in Stuttering Therapy
Halina Kwasnicka, Blazej Zak 925

Using Most Similarity Tree Based Clustering to Select the Top Most Discriminating Genes for Cancer Detection
Xinguo Lu, Yaping Lin, Xiaolin Yang, Lijun Cai, Haijun Wang, Gustaph Sanga 931

Nonambiguous Concept Mapping in Medical Domain
Paweł Matykiewicz, Włodzisław Duch, John Pestian 941

Feature Selection and Ranking of Key Genes for Tumor Classification: Using Microarray Gene Expression Data
Srinivas Mukkamala, Qingzhong Liu, Rajeev Veeraghattam, Andrew H. Sung 951

Cognitive Analysis in Diagnostic DSS-Type IT Systems
Lidia Ogiela, Ryszard Tadeusiewicz, Marek R. Ogiela 962

Interpretability of Bayesian Decision Trees Induced from Trauma Data
Derek Partridge, Vitaly Schetinin, Dayou Li, Timothy J. Coats, Jonathan E. Fieldsend, Wojtek J. Krzanowski, Richard M. Everson, Trevor C. Bailey 972

The Greatest and the Least Eigen Fuzzy Sets in Evaluation of the Drug Effectiveness Levels
Elisabeth Rakus-Andersson 982

Cardiac Ventricle Contour Reconstruction in Ultrasonographic Images Using Bayesian Constrained Spectral Method
Tomasz Sołtysiński, Krzysztof Kałużynski, Tadeusz Pałko 988

A Model of a Diagnostic Rule in the Dempster-Shafer Theory
Ewa Straszecka 998

DNA Fragment Assembly by Ant Colony and Nearest Neighbour Heuristics
Wannasak Wetcharaporn, Nachol Chaiyaratana, Sissades Tongsima 1008

Various Problems of Artificial Intelligence

Application of Bayesian Confirmation Measures for Mining Rules from Support-Confidence Pareto-Optimal Set
Roman Slowinski, Izabela Brzezinska, Salvatore Greco 1018

Cognitive Analysis Techniques in Business Planning and Decision Support Systems
Ryszard Tadeusiewicz, Lidia Ogiela, Marek R. Ogiela 1027

PERT Based Approach to Performance Analysis of Multi–Agent Systems
Tomasz Babczyński, Jan Magott 1040

Rule-Based Automated Price Negotiation: Overview and Experiment
Costin Bădică, Maria Ganzha, Marcin Paprzycki 1050

A New Version of the Fuzzy-ID3 Algorithm
Łukasz Bartczuk, Danuta Rutkowska 1060

New Interpolation Method with Fractal Curves
Andrzej Cader, Marcin Krupski 1071

Integrating Lookahead and Post Processing Procedures with ACO for Solving Set Partitioning and Covering Problems
Broderick Crawford, Carlos Castro 1082

Learning Algorithms for Scheduling Using Knowledge Based Model
Ewa Dudek-Dyduch, Tadeusz Dyduch 1091

Knowledge Representation of Pedestrian Dynamics in Crowd: Formalism of Cellular Automata
Ewa Dudek-Dyduch, Jarosław Wąs 1101

Algorithm for Generating Fuzzy Rules for WWW Document Classification
Piotr Dziwiński, Danuta Rutkowska 1111

A Possibilistic-Logic-Based Information Retrieval Model with Various Term-Weighting Approaches
Janusz Kacprzyk, Katarzyna Nowacka, Sławomir Zadrożny 1120

Sketch of Autopoietic Essence of *Computing* and *KnowledgeWorking*
Leonid Kompanets 1130

Apply the Particle Swarm Optimization to the Multidimensional Knapsack Problem
Min Kong, Peng Tian 1140

Self-stabilizing Algorithms for Graph Coloring with Improved Performance Guarantees
Adrian Kosowski, Łukasz Kuszner 1150

A Novel Modeling Methodology: Generalized Nets
Maciej Krawczak 1160

A Hierarchical Particle Swarm Optimization for Solving Bilevel Programming Problems
Xiangyong Li, Peng Tian, Xiaoping Min 1169

Methods of Artificial Intelligence in Blind People Education
Bohdan Macukow, Wladyslaw Homenda 1179

Neural Networks and the Estimation of Hands' Strength in Contract Bridge
Krzysztof Mossakowski, Jacek Mańdziuk 1189

Mining Travel Resources on the Web Using L-Wrappers
Elvira Popescu, Amelia Bădică, Costin Bădică 1199

A New Evaluation Method for E-Learning Systems
Krzysztof Przybyszewski 1209

Parameter Estimation of Systems Described by the Relation by Maximum Likelihood Method
Jerzy Świątek 1217

A Distributed Learning Control System for Elevator Groups
Tomasz Walczak, Paweł Cichosz 1223

Author Index 1233

Multichannel Data Aggregation by Layers of Formal Neurons*

Leon Bobrowski[1,2]

[1] Faculty of Computer Science, Białystok Technical University
[2] Institute of Biocybernetics and Biomedical Engineering, PAS, Warsaw, Poland

Abstract. Principles of separable aggregation of multichannel (multisource) data sets by parallel layers of formal neurons are considered in the paper. Each data set contains such feature vectors which represent objects assigned to one of a few categories.The term multichannel data sets means that each single object is characterised by data obtained through different information channels and represented by feature vectors in a different feature space. Feature vectors from particular feature spaces are transformed by layers of formal neurons what results in the aggregation of some feature vectors. The postulate of separable aggregation is aimed at the minimization of the number of different feature vectors under the condition of preserving the categories separabilty.

1 Introduction

The processing and aggregation of multichannel (multisource) data sets by hierachical networks of formal neurons is considered in the paper. This problem could be related to modelling of information processing and aggregation by different sensor modules in a nervous system such as vision, hearing, smell or touch.

The neural networks similar to multilayer perceptrons with a hierarchical structure of parallel sublayers are taken into consideration ([1], [2]). Designing hierarchical networks for the purpose of multichannel data aggregation is analysed in the paper. The term designing a neural network means here a choice of a neural network structure (e.g. the number of layers and the number of elements in particular layers) and the weights of connections from elements of a lower layer to the elements of a next, higher layer.

Data sets are often divided into subsets (learning subsets) related to particular classes. A basic principle of optimising the neural structure could be based on the postulate of separable data aggregation [3]. Transformation of the learning subsets by the separable layer of formal neurons allows one to decrease number of different feature vectors while preserving the class separability. The ranked and the dipolar strategies of designing separable layers of formal neurons have been proposed ([3], [4]). These strategies have been implemented through the

* This work was partially supported by the W/II/1/2006 and SPUB-M (COST 282) grants from the Białystok University of Technology and by the 16/St/2006 grant from the Institute of Biocybernetics and Biomedical Engineering PAS.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1–8, 2006.
© Springer-Verlag Berlin Heidelberg 2006

minimisation of the convex and piecewise linear (*CPL*) criterion functions. In this paper the general principles of separable processing in parallel neural structures or in a family of fetaure subspaces are analysed.

2 Separable Learning Sets

Let us assume that each of the m analysed objects O_j $(j = 1, \ldots, m)$ can be represented as the so called feature vector $\boldsymbol{x}_j = [x_{j1}, \ldots, x_{jn}]^T$, or as a point in the n-dimensional *feature space* $F[n]$ ($\boldsymbol{x}_j \in F[n]$). The components (*features*) x_i of the vector $\boldsymbol{x}$ are numerical results of a variety of examinations of a given object O. The feature vectors $\boldsymbol{x}$ can be of mixed, qualitative-quantitative type ($x_i \in \{0, 1\}$ or $x_i \in R$).

We assume that the database contains the descriptions $\boldsymbol{x}_j(k)$ of m objects $O_j(k)$ $(j = 1, \ldots, m)$ labelled in accordance with their *category* (*class*) ω_k $(k = 1, \ldots, K)$. The learning set $C(k)$ contains m_k feature vectors $\boldsymbol{x}_j(k)$ assigned to the k-th category ω_k

$$C_k = \{\boldsymbol{x}_j(k)\} \ (j \in J_k) \tag{1}$$

where J_k is the set of indices j of the feature vectors $\boldsymbol{x}_j(k)$ belonging to the class ω_k.

Definition 1. *The learning sets C_k (1) are* separable *in the feature space $F[n]$, if they are disjoined in this space ($C_k \cap C_k = \emptyset$, if $k \neq k'$).*

It means that the feature vectors $\boldsymbol{x}_j(k)$ and $\boldsymbol{x}_{j'}(k')$ belonging to different learning sets C_k and $C_{k'}$ cannot be equal:

$$(k \neq k') \Rightarrow (\forall j \in J_k) \text{ and } (\forall j' \in J_{k'})) \ \boldsymbol{x}_j(k) \neq \boldsymbol{x}_{j'}(k') \tag{2}$$

We are also considering the separation of the sets C_k (1) by the hyperplanes $H(\boldsymbol{w}_k, \theta_k)$ in the n-dimensional feature space $F[n]$

$$H(\boldsymbol{w}_k, \theta_k) = \{\boldsymbol{x} : \boldsymbol{w}_k^T \boldsymbol{x} = \theta_k\} \tag{3}$$

where $\boldsymbol{w}_k = [w_{k1}, \ldots, w_{kn}]^T \in R^n$ is the weight vector, $\theta_k \in R^1$ is the threshold, and $\boldsymbol{w}_k^T \boldsymbol{x}$ is the inner product.

Definition 2. *The learning sets (1) are* linearly separable *in the feature space $F[n]$ if each of the sets C_k can be fully separated from the sum of the remaining sets C_i by some hyperplane $H(\boldsymbol{w}_k, \theta_k)$ (3):*

$$\begin{array}{lllll} (\forall k \in \{1, \ldots, K\}) & (\exists \boldsymbol{w}_k, \theta_k) & (\forall \boldsymbol{x}_j(k) \in C_k) & & \boldsymbol{w}_k^T \boldsymbol{x}_j(k) \geq \theta_k \\ & \text{and} & (\forall \boldsymbol{x}_j(k) \in C_i), i \neq k) & & \boldsymbol{w}_k \boldsymbol{x}_j(k) < \theta_k \end{array} \tag{4}$$

In accordance with the relation (4), all vectors $\boldsymbol{x}_j(k)$ belonging to the learning set C_k are situated on the positive side ($\boldsymbol{w}_k^T \boldsymbol{x}_j(k) \geq \theta_k$) of the hyperplane $H(\boldsymbol{w}_k, \theta_k)$ (3) and all feature vectors $\boldsymbol{x}_j(i)$ from the remaining sets C_i are situated on the negative side (($\boldsymbol{w}_k^T \boldsymbol{x}_j(k) < \theta_k$) of this hyperplane.

3 Data Processing by a Layer of Formal Neurons

The formal neuron $NF(\boldsymbol{w}, \theta)$ can be defined by the activation function $r_t(\boldsymbol{w}, \theta; \boldsymbol{x})$

$$r = r_t(\boldsymbol{w}, \theta; \boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{w}^T\boldsymbol{x} \geq \theta \\ 0 & \textbf{if } \boldsymbol{w}^T\boldsymbol{x} < \theta \end{cases} \tag{5}$$

where $\boldsymbol{w} = [w_1, \ldots, w_n]^T \in R^n$ is the weight vector, θ is the positive threshold ($\theta > 0$), and r is the output signal.

The formal neuron $NF(\boldsymbol{w}, \theta)$ is activated ($r = 1$) by the vector $\boldsymbol{x}$ if and only if this vector is situated on the positive side of the hyperplane $H(\boldsymbol{w}, \theta)$ (3). It means that the relation $\boldsymbol{w}^T\boldsymbol{x} \geq \theta$ holds.

The layer of L formal neurons $NF(\boldsymbol{w}_k, \theta_k)$ transforms feature vectors $\boldsymbol{x}$ into the output vectors $\boldsymbol{r} = [r_1, \ldots, r_L]^T$ with L binary components $r_i \in \{0, 1\}$ which are determined by the equation $\boldsymbol{r} = r_t(\boldsymbol{w}_i, \theta_i; \boldsymbol{x})$ (8):

$$r = r(\boldsymbol{W}; \boldsymbol{x}) = [r_t(\boldsymbol{w}_1, \theta_1; \boldsymbol{x}), \ldots, r_t(\boldsymbol{w}_L, \theta_L; \boldsymbol{x})]^T \tag{6}$$

where $\boldsymbol{W} = [\boldsymbol{w}_1^T, \theta_1, \ldots, \boldsymbol{w}_L^T, \theta_L]^T$ is the vector of the layer parameters.

The transformed learning sets C_k (1) are obtained from the vectors $\boldsymbol{r}_j(k) = \boldsymbol{r}(\boldsymbol{W}; \boldsymbol{x}_j(k))$

$$C_k = \{\boldsymbol{r}_j(k)\} \ (j \in J_k) \tag{7}$$

We are examining such properties of the transformation (6) by a neural layer which assure the separability (2) or the linear separability (4) of the sets C_k (10). This property can linked to the concept of mixed dipole separation [3].

Definition 3. *A pair of feature vectors $(\boldsymbol{x}_j(k), \boldsymbol{x}_{j'}(k'))$ $(\boldsymbol{x}_j(k) \neq \boldsymbol{x}_{j'}(k'), j < j')$ constitutes a* mixed dipole *if and only if the vectors $\boldsymbol{x}_j(k)$ and $\boldsymbol{x}_{j'}(k')$ belong to different classes ω_k $(k \neq k')$. Similarly, a pair of different feature vectors from the same class ω_k constitutes a* clear dipole *$(\boldsymbol{x}_j(k), \boldsymbol{x}_{j'}(k))$.*

Definition 4. *The formal neuron $NF(\boldsymbol{w}_k, \theta_k)$ (5)* separates *(*divides*) the dipole $(\boldsymbol{x}_j(k), \boldsymbol{x}_{j'}(k'))$ if* only one *feature vector from this pair activates the neuron $(r = 1)$.*

Lemma 1. *The transformed sets C_k (7) are separable (2) if and only if each mixed dipole $(\boldsymbol{x}_j(k), \boldsymbol{x}_{j'}(k'))$ constituted by elements $\boldsymbol{x}_j(k)$ of the sets C_k (1) is divided by at least one neuron $NF(\boldsymbol{w}_l, \theta_l)$ (5) of the neural layer (6).*

The proof of this Lemma can be found in the reference [3].

Definition 5. *The layer of formal neurons $NF(\boldsymbol{w}_k, \theta_k)$ (5) performs the* separable aggregation *of the learning sets C_k if and only if the sets C_k (7) are separable (2) and each of transformed vectors $\boldsymbol{r}_j(k) = \boldsymbol{r}(\boldsymbol{W}; \boldsymbol{x}_j(k))$ (6) is different from zero $(\boldsymbol{r}_j(k) \neq \boldsymbol{0})$.*

$$(\forall k \in \{1, \ldots, K\}) \ (\forall \boldsymbol{x}_j(k) \in C_k) \quad \boldsymbol{r}(\boldsymbol{W}; \boldsymbol{x}_j(k)) \neq 0 \tag{8}$$

Lemma 2. *The condition (8) is fulfilled if and only if each element $\boldsymbol{x}_j(k)$ of the sets C_k (1) activates ($r = 1$) at least one neuron $NF(\boldsymbol{w}_k, \theta_k)$ (5) of the layer.*

In accordance with the lemma assumption each feature vector $\boldsymbol{x}_j(k)$ (1) should be situated on the positive side of at least one hyperplane $H(\boldsymbol{w}_k, \theta_k)$ (3). The thesis of the lemma can be proved directly on the base of the relations (5), (6) and (8).

4 Convex and Piecewise Linear Criterion Function (*CPL*)

The procedure of the separable layer design can be based on a sequence of minimisation of the of the convex and piecewise linear (*CPL*) criterion functions $\Psi_l(\boldsymbol{w}, \theta)$ [3], [4]. It is convenient to define the functions $\Psi_l(\boldsymbol{w}, \theta)$ by using the positive G_l^+ and the negative G_l^- sets of the feature vectors $\boldsymbol{x}_j$ (1).

$$G_l^+ = \{\boldsymbol{x}_j\} \ (j \in J_l^+) \text{ and } G_l^- = \{\boldsymbol{x}_j\} \ (j \in J_l^-) \tag{9}$$

Each element $\boldsymbol{x}_j$ of the set G_l^+ defines the positive penalty function $\varphi_j^+(\boldsymbol{w}, \theta)$

$$\varphi_j^+(\boldsymbol{w}, \theta) = \begin{cases} 1 - \boldsymbol{w}^T \boldsymbol{x}_j + \theta & if \ \ \boldsymbol{w}^T \boldsymbol{x}_j - \theta \leq 1 \\ 0 & if \ \ \boldsymbol{w}^T \boldsymbol{x}_j - \theta > 1 \end{cases} \tag{10}$$

Similarly, each element $\boldsymbol{x}_j$ of the set G_l^- defines the negative penalty function $\varphi_j^-(\boldsymbol{w}, \theta)$

$$\varphi_j^-(\boldsymbol{w}, \theta) = \begin{cases} 1 + \boldsymbol{w}^T \boldsymbol{x}_j - \theta & if \ \ \boldsymbol{w}^T \boldsymbol{x}_j - \theta \leq -1 \\ 0 & if \ \ \boldsymbol{w}^T \boldsymbol{x}_j - \theta > -1 \end{cases} \tag{11}$$

The penalty function $\varphi_j^+(\boldsymbol{w}, \theta)$ is aimed at situating the vector $\boldsymbol{x}_j$ ($\boldsymbol{x}_j \in G_l^+$) on the positive side of the hyperplane $H(\boldsymbol{w}, \theta)$ (3). Similarly, the function $\varphi_j^-(\boldsymbol{w}, \theta)$ should situate the vector $\boldsymbol{x}_j$ ($\boldsymbol{x}_j \in G_l^-$) on the negative side of this hyperplane.

The criterion function $\Psi_l(\boldsymbol{w}, \theta)$ is the weighted sum of the above penalty functions

$$\Psi_l(\boldsymbol{w}, \theta) = \sum_{j \in J_l^+} \alpha_j^+ \varphi_j^+(\boldsymbol{w}, \theta) + \sum_{j \in J_l^-} \alpha_j^- \varphi_j^-(\boldsymbol{w}, \theta) \tag{12}$$

where α_j^+ ($\alpha_j^+ > 0$) and α_j^- ($\alpha_j^- > 0$) are positive parameters (*prices*).

The criterion function $\Psi_l(\boldsymbol{w}, \theta)$ belongs to the family of the convex and piecewise linear (*CPL*) criterion functions. Minimization of the function $\Psi_l(\boldsymbol{w}, \theta)$ allows one to find paramerters $(\boldsymbol{w}_l^*, \theta_l^*)$ which define the l-th optimal neuron $NF(\boldsymbol{w}_l^*, \theta_l^*)$ (5).

$$\Psi_l^* = \Psi_l(\boldsymbol{w}_l^*, \theta_l^*) = \min \Psi_l(\boldsymbol{w}, \theta) \geq 0 \tag{13}$$

The basis exchange algorithms which are similar to linear programming allow one to find the minimum of the criterion function $\Psi_l(\boldsymbol{w}, \theta)$ efficiently even in a case of large, multidimensional data sets G_l^+ and G_l^- (8) [5].

5 Designing Separable Layers of Formal Neurons

The separable layer of formal neurons $NF(\boldsymbol{w}_l, \theta_l)$ (5) preserves the separability (2) of the learning sets C_k (1) during the transformation (6). In accordance with the Lemma 1, the necessary condition for the separability of the neural layer is the division of each mixed dipole $(\boldsymbol{x}_j(k), \boldsymbol{x}_{j'}(k'))$ by at least one element $NF(\boldsymbol{w}_l, \theta_l)$ of this layer. Another necessary condition (8) for the separability of the neural layer is specified in the Lemma 2. In accordance with this lemma each feature vector $\boldsymbol{x}_j(k)$ should activate at least one neuron $NF(\boldsymbol{w}_k, \theta_k)$ (5) of the layer.

The multistage procedures of separable layer designing has been proposed [3],[4], [5]. One of these procedures has been called the dipolar and is based on the successive divisions of all mixed dipoles. During the l-th stage of this procedure the l-th neuron $NF(\boldsymbol{w}_l, \theta_l)$ (5) is designed in such a manner that as many as possible mixed and undivided yet dipoles $(\boldsymbol{x}_j(k), \boldsymbol{x}_{j'}(k'))$ become divided. The l-th neuron $NF(\boldsymbol{w}_l, \theta_l)$ (5) is designed for dipole division by minimisation of the criterion function $\Psi_l(\boldsymbol{w}, \theta)$ (17) defined on special data sets G_l^+ and G_l^- (8). The procedure is stopped when all mixed dipoles $(\boldsymbol{x}_j(k), \boldsymbol{x}_{j'}(k'))$ are divided.

The choice of the positive G_l^+ and the negative G_l^- sets (9) of the feature vectors $\boldsymbol{x}_j$ (1) is crucial for the implementation of the dipolar procedure. Let us introduce for this purpose the below symbols, which allow for definition of the positive and negative sets separately for each category ω_k :

$$\begin{array}{l} G_l^+(k) = \{\boldsymbol{x}_j(k)\}\ (j \in J_l^+(k)) \quad \text{and} \\ G_l^-(k) = \{\boldsymbol{x}_{j'}(k')\}\ (j' \in J_l^-(k),\ k' \neq k) \end{array} \tag{14}$$

where $G_l^+(k)$ is the set of such feature vectors $\boldsymbol{x}_j(k)$ from the k-th set C_k (1) (category ω_k) that are supposed to be situated on the positive side of the hyperplane $H(\boldsymbol{w}_l(k), \theta_l(k))$ (3) during the l-th stage. Similarly, $G_l^-(k)$ is a set of such feature vectors $\boldsymbol{x}_{j'}(k')$ from other sets $C_{k'}(k' \neq k)$ which are supposed to be located on the negative side of $H(\boldsymbol{w}_l(k), \theta_l(k))$.

We are assuming here that each category ω_k has its own sublayer S_k of the formal neurons $NF(\boldsymbol{w}_l(k), \theta_l(k))$ (5). The positive $G_l^+(k)$ and the negative $G_l^-(k)$ sets (14) are expected to contain elements $\boldsymbol{x}_j(k)$ and $\boldsymbol{x}_{j'}(k')$ of such mixed dipoles $(\boldsymbol{x}_j(k), \boldsymbol{x}_{j'}(k'))$ which have not yet been divided by the first elements $NF(\boldsymbol{w}_{l'}(k), \theta_{l'}(k))$ $(0 < l' < l)$ of the k-th sublayer S_k related to the category ω_k.

Example 1. (*dipolar layer*): Let us choose the sets $G_l^+(k)$ and $G_l^-(k)$ (14) related during first stage $(l = 1)$ to the k-th category ω_k and the sets C_k (1) in the manner below:

$$G_1^+(k) = C_k \text{ and } G_1^-(k) = \bigcup_{k \neq k'} C_{k'} \tag{15}$$

The minimization (13) of the criterion function $\Psi_1(\boldsymbol{w}, \theta)$ (12) allows one to find the first neuron $NF(\boldsymbol{w}_1^*(k), \theta_1^*(k))$ (5) of the k-th sublayer S_k and the separating hyperplane $H(\boldsymbol{w}_1^*(k), \theta_1^*(k))$ (3).

The sets $G^+_{l+1}(k)$ and $G^-_{l+1}(k)$ (14) during the $(l+1)$-th stage are defined in an iterative manner

$$\begin{array}{l} G^+_{l+1}(k) = G^+_l(k) - R^+_k(\boldsymbol{w}^*_l(k), \theta^*_l(k)) \quad \text{and} \\ G^-_{l+1}(k) = G^-_l(k) - R^-_{k'}(\boldsymbol{w}^*_l(k), \theta^*_l(k)) \end{array} \tag{16}$$

where $R^+_k(\boldsymbol{w}^*_l(k), \theta^*_l(k))$ is the set of such elements $\boldsymbol{x}_j(k)$ of $G^+_l(k)$ which are situated on the positive side of the hyperpalane $H(\boldsymbol{w}^*_l(k), \theta^*_l(k))$ i.e. $\boldsymbol{w}^*_l(k)^T \boldsymbol{x}_j(k) > \theta^*_l(k)$. Similarly, $R^-_{k'}(\boldsymbol{w}^*_l(k), \theta^*_l(k))$ is the set of such elements $\boldsymbol{x}_j(k')$ of $G^-_l(k)$ which are situated on the negative side of $H(\boldsymbol{w}^*_l(k), \theta^*_l(k))$ i.e. $\boldsymbol{w}^*_l(k)^T \boldsymbol{x}_j(k') < \theta^*_l(k)$. □

The choice of the sets $G^+_{l+1}(k)$ and $G^-_{l+1}(k)$ in accordance with the rule (16) realizes the principle that such mixed dipoles $(\boldsymbol{x}_j(k), \boldsymbol{x}_{j'}(k'))$ which have been divided in the sublayer S_k are omitted from further consideration and the rule (8) is reinforced. One can see that after a finite number of stages all mixed dipoles $(\boldsymbol{x}_j(k), \boldsymbol{x}_{j'}(k'))$ will be divided in this way.

6 Multichannel Data Sets

Let us consider now the situation when the same objects O_j are characterised by data obtained from L different channels. Such a situation may exist for example in modelling the nervous system, when a given object is perceived simultaneously through the vision, hearing and touch sensors.

Models of such a situation could include the decomposition of the n - dimensional feature space $F[n]$ into L feature subspaces (channels) $F_l[n_l]$ $(l = 1, \ldots, L)$

$$F[n] = F[n_l] \cup F[n_2] \cup \ldots \cup F[n_L] \tag{17}$$

where $F_l[n_l] \cap F_{l'}[n_{l'}] = \emptyset$ if $l' \neq l$.

Feature vectors $\boldsymbol{x}_j[n_l]$ in the n_l - dimensional subspace $F_l[n_l]$ are obtained from $\boldsymbol{x}_j[n]$ (1) in the result of the feature reduction (selection):

$$\boldsymbol{x}_j[n_l] = [x_{j1}, \ldots, x_{jn_l}]^T \text{ where } \boldsymbol{x}_j[n_l] \in F_l[n_l] \tag{18}$$

In the result of the feature space $F[n]$ reduction to the n_l - dimensional subspace $F_l[n_l]$ (one channel), each learning set C_k (1) is replaced by the set $C'_{kl}[n_l]$

$$C'_{kl}[n_l] = \{\boldsymbol{x}_j[n_l]\} \quad (j \in J_k) \tag{19}$$

The set $C'_{kl}[n_l]$ contains such reduced vectors $\boldsymbol{x}_j[n_l]$ (18) $(\boldsymbol{x}_j[n_l] \in F_l[n_l])$ which belong to the class ω_k.

The feature space reduction to one channel $F_l[n_l]$ can result in a loss of the sets C_k (1) separabilty (2). It means that the sets $C'_{kl}[n_l]$ (19) can overlap in the n_l - dimensional subspace $F_l[n_l]$:

$$(\exists l \in \{1, \ldots, L\})\ (\exists k \neq k')\ C'_{kl}[n_l] \cap C'_{k'l}[n_l] \neq \emptyset \tag{20}$$

This property results from the equality of some reduced vectors $\boldsymbol{x}_j[n_l]$ (18) ($\boldsymbol{x}_j[n_l] \in F_l[n_l]$)

$$(\exists l \in \{1,\ldots,L\})\ (\exists k \neq k')\ (\exists j \neq j')\ \ \boldsymbol{x}_j[n_l] = \boldsymbol{x}_{j'}[n_l] \tag{21}$$

where $\boldsymbol{x}_j[n_l] \in \omega_k$ and $\boldsymbol{x}_{j'}[n_l] \in k'$.

Data aggregation could be performed in accordance with rule (21). It means that, in the result of the feature space reduction to one channel $F_l[n_l]$, some different feature vectors $\boldsymbol{x}_j(k)$ and $\boldsymbol{x}_{j'}(k')$ are equalized and treated as a single one. Feature vectors $\boldsymbol{x}_j(k)$ and $\boldsymbol{x}_{j'}(k')$ can also be made equal as a result of their transformations (6) by layers of formal neurons $NF(\boldsymbol{w}_m(k), \theta_m(k))$ (5).

Designing postulate: We are interested in designing such a layer of formal neurons $NF(\boldsymbol{w}_m(k), \theta_m(k))$ (5) operating in a particular feature subspace $F_l[n_l]$ ($l = 1,\ldots,L$) which allows one to preserve the separability (2) of the learning sets C_k (1) during data aggregation.

Example 2. Let us consider designing a dipolar layer in the l-th feature subspace $F_l[n_l]$ in a manner similar to one described in Example 1. In order to take into account the possible equality (21) of the vectors $\boldsymbol{x}_j[n_l]$ in the feature space $F_l[n_l]$, the sets $G_1^+(k)$ and $G_1^-(k)$ (15) are defined in the manner below:

$$G_1^+(k) = \{\boldsymbol{x}_j[n_l]\ :\ j \in J_k\ (1)\} \tag{22}$$

$$G_1^-(k) = \{\boldsymbol{x}_{j'}[n_l]\ :\ (\forall \boldsymbol{x}_j[n_l] \in G_1^+(k))\ \boldsymbol{x}_{j'}[n_l] \neq \boldsymbol{x}_j[n_l]\} \tag{23}$$

The negative set $G_1^-(k)$ (23) contains only such vectors $\boldsymbol{x}_{j'}[n_l]$ ($\boldsymbol{x}_{j'}[n_l] \in F_l[n_l]$) which are different from any element $\boldsymbol{x}_j[n_l]$ of the positive set $G_1^+(k)$ (22).

The minimization (13) of the criterion function $\Psi(\boldsymbol{w}, \theta)$ (12) with the sets $G_1^+(k)$ (22) and $G_1^-(k)$ (23) allows one to find the first neuron $NF(\boldsymbol{w}_1^*(k), \theta_1^*(k))$ (5) of the k-th sublayer S_k and the separating hyperplane $H(\boldsymbol{w}_1^*(k), \theta_1^*(k))$ (3) in the l-th feature subspace $F_l[n_l]$.

The sets $G_{l+1}^+(k)$ and $G_{l+1}^-(k)$ during the $(l+1)$-th stage can be defined in an iterative manner by the rules (16). The sets $G_{l+1}^+(k)$ and $G_{l+1}^-(k)$ allows one to find the next separating hyperplanes $H(\boldsymbol{w}_l^*(k), \theta_l^*(k))$ (3) and the succesive neurons $NF(\boldsymbol{w}_l^*(k), \theta_l^*(k))$ (5) of the k-th sublayer S_k.

One can see that after a finite number of such stages all mixed dipoles $(\boldsymbol{x}_j[n_l], \boldsymbol{x}_{j'}[n_l])$ constituted from different elements $\boldsymbol{x}_j[n_l]$ and $\boldsymbol{x}_{j'}[n_l]$ ($\boldsymbol{x}_j[n_l] \neq \boldsymbol{x}_{j'}[n_l]$)) will be divided. The *separable sublayer* S_k is designed in this way. □

The sublayer S_k in the l-th feature subspace $F_l[n_l]$ transforms the learning sets C_k (1) into the sets $C_{k'}$ (7). It can be proved that the separable sublayer S_k allows one to preserve the learning sets C_k (1) separability (2). Designing a postulate can be fulfilled through the procedure described in Example 2.

7 Concluding Remarks

Separable aggregation of the learning sets C_k (1) can be realized through their transformations by layers of formal neurons $NF(\boldsymbol{w}_m(k), \theta_m(k))$ (5) operating

in particular feature subspaces $F_l[n_l]$ $(l = 1, \ldots, L)$. Separable transformations allow one to decrease number of different feature vectors in particular learning sets C_k under the condition of preserving the separability (2) of these sets. The dipolar strategy of the separable layers design has been described in the paper. Particular attention has been paid to applications of the dipolar strategy to designing separable layers in feature subspaces $F_l[n_l]$.

The decomposition (17) of the feature space $F[n]$ into subspaces $F_l[n_l]$ can be implicated not only by a multichannel sensor structure but also by the high dimensionality of the problem. The dimensionality of the feature space $F[n]$ can be so high (e.q. millions of pixels in graphical objects $\boldsymbol{x}_j(k)$ (1)) that implementation of computational procedures becomes practically impossible. There are still open problems both theoretical as well as implementational, related to decomposition (17) of the feature space $F[n]$ and to the designing of separable neural layers in particular feature subspaces $F_l[n_l]$.

References

1. Rosenblatt F.: Principles of neurodynamics, Spartan Books, Washington, 1962
2. Duda O.R. and Hart P.E., Stork D.G. : Pattern Classification, J. Wiley, New York, 2000
3. Bobrowski L.: 'Piecewise-Linear Classifiers, Formal Neurons and Separability of the Learning Sets', Proceedings of ICPR96, pp. 224-228, Wienna, Austria, 1996
4. Bobrowski L.: Eksploracja danych oparta na wypukych i odcinkowo-liniowych funkcjach kryterialnych (Data mining based on convex and piecewise linear (*CPL*) criterion functions) (in Polish), Technical University Białystok, 2005.
5. Bobrowski L.: 'Design of piecewise linear classifiers from formal neurons by some basis exchange technique', Pattern Recognition, 24(9), pp. 863-870, 1991

Estimation of Distribution Algorithm for Optimization of Neural Networks for Intrusion Detection System

Yuehui Chen[1], Yong Zhang[2], and Ajith Abraham[1,3]

[1] School of Information Science and Engineering
Jinan University, Jinan 250022, P.R. China
yhchen@ujn.edu.cn
[2] School of Control Science and Engineering
Jinan University, Jinan 250022, P.R. China
cse_zhangy@ujn.edu.cn
[3] School of Computer Science and Engineering
Chung-Ang University, Seoul, Republic of Korea
ajith.abraham@ieee.org

Abstract. An Intrusion Detection System (IDS) is a program that analyzes what happens or has happened during an execution and tries to find indications that the computer has been misused. An IDS does not eliminate the use of preventive mechanism but it works as the last defensive mechanism in securing the system. This paper evaluates the performances of Estimation of Distribution Algorithm (EDA) to train a feed-forward neural network classifier for detecting intrusions in a network. Results are then compared with Particle Swarm Optimization (PSO) based neural classifier and Decision Trees (DT). Empirical results clearly show that evolutionary computing techniques could play an important role in designing real time intrusion detection systems.

1 Introduction

Attacks on the nation's computer infrastructures are becoming an increasingly serious problem. Computer security is defined as the protection of computing systems against threats to confidentiality, integrity, and availability [1]. Confidentiality (or secrecy) means that information is disclosed only according to policy, integrity means that information is not destroyed or corrupted and that the system performs correctly, availability means that system services are available when they are needed. Computing system refers to computers, computer networks, and the information they handle. Security threats come from different sources such as natural forces (such as flood), accidents (such as fire), failure of services (such as power) and people known as intruders. There are two types of intruders: the external intruders who are unauthorized users of the machines they attack, and internal intruders, who have permission to access the system with some restrictions. The traditional prevention techniques such as user authentication, data encryption, avoiding programming errors and firewalls are

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 9–18, 2006.
© Springer-Verlag Berlin Heidelberg 2006

used as the first line of defense for computer security. If a password is weak and is compromised, user authentication cannot prevent unauthorized use, firewalls are vulnerable to errors in configuration and ambiguous or undefined security policies. They are generally unable to protect against malicious mobile code, insider attacks and unsecured modems. Programming errors cannot be avoided as the complexity of the system and application software is changing rapidly leaving behind some exploitable weaknesses. Intrusion detection is therefore required as an additional wall for protecting systems. Intrusion detection is useful not only in detecting successful intrusions, but also provides important information for timely countermeasures. Intrusion detection is classified into two types: misuse intrusion detection and anomaly intrusion detection. Misuse intrusion detection uses well-defined patterns of the attack that exploit weaknesses in system and application software to identify the intrusions. Anomaly intrusion detection identifies deviations from the normal usage behavior patterns to identify the intrusion.

We have two options to secure the system completely, either prevent the threats and vulnerabilities which come from flaws in the operating system as well as in the application programs or detect them and take some action to prevent them in future and also repair the damage. It is impossible in practice, and even if possible, extremely difficult and expensive, to write a completely secure system. Transition to such a system for use in the entire world would be an equally difficult task. Cryptographic methods can be compromised if the passwords and keys are stolen. No matter how secure a system is, it is vulnerable to insiders who abuse their privileges. There is an inverse relationship between the level of access control and efficiency. More access controls make a system less user-friendly and more likely of not being used. An Intrusion Detection system is a program (or set of programs) that analyzes what happens or has happened during an execution and tries to find indications that the computer has been misused. An Intrusion detection system does not eliminate the use of preventive mechanism but it works as the last defensive mechanism in securing the system. Data mining approaches are a relatively new technique for intrusion detection. There are a wide variety of data mining algorithms drawn from the fields of statistics, pattern recognition, machine learning, and databases. Previous research of data mining approaches for intrusion detection model identified several types of algorithms as useful techniques. Classification is one of the data mining algorithms, which have been investigated as a useful technique for intrusion detection models.

Various intelligent paradigms namely Neural Networks [2], Support Vector Machine [3], Neuro-Fuzzy systems [4], Linear Genetic Programming [5], Flexible Neural Tree [6][7][8], ensemble of intelligent paradigms [22][23] and Decision Trees [10] have been used for intrusion detection. Various data mining techniques have been applied to intrusion detection because it has the advantage of discovering useful knowledge that describes a user's or program's behavior from large audit data sets.

This paper proposes an EDA based evolutionary neural network classifier for detecting intrusions. The weights, bias and flexible activation function parameters are optimized by EDA algorithm. Results are then compared with Particle Swarm Optimization (PSO) based neural classifier and Decision Trees (DT). The rest of paper is organized as follows. A simple introduction to neural networks is given in Section 2. The EDA and PSO based neural networks training algorithms are presented in Section 3. Some simulation results and comparisons are provided in Sections 4. Finally in Section 5 we present some conclusions and future works.

2 Neural Networks

A typical three-layer feedforward neural network consists of an input layer, a hidden layer and an output layer. The nodes are connected by weights and output signals, which are a function of the sum of the inputs to the node modified by a simple nonlinear activation function. The usually used activation function is the sigmoid function with threshold defined as

$$f(\sum_{i=1}^{n} w_i x_i - \theta) = \frac{1}{1 + exp(-(\sum_{i=1}^{n} w_i x_i - \theta))} \tag{1}$$

where x_i is the input to the node and w_i is the corresponding input weight, θ is a value which is usually called the threshold, n is the number of the inputs to the node. In this study, an flexible activation functions at hidden and output layers is selected. Some flexible activation functions shown in Table 1.

The output of a node is scaled by the connecting weight and is fed forward as an input to the nodes in the next layer of the network. The input layer plays no computational role but merely serves to pass the input vector to the network. The input layer and the hidden layer are connected by weights and likewise the hidden layer and output layer also have connection weights. The network has the ability to learn through training. The training requires a set of training data, i.e., a series of input and associated output vectors. During the training, the network is repeatedly presented with the training data and the weights and thresholds in the network are adjusted from time to time till the desired inputCoutput mapping occurs.

Table 1. The flexible activation functions

Gaussian Function	$f(x,a,b) = exp(-\frac{(x-a)^2}{b^2})$
Unipolar sigmoid function	$f(x,a) = \frac{2\lvert a\rvert}{1+e^{-2\lvert a\rvert x}}$
Bipolar sigmoid function	$f(x,a) = \frac{1-e^{-2xa}}{a(1+e^{-2xa})}$
Nonlocal radial coordinates	$f(x,a,b) = (b^2 + \lVert x-a\rVert^2)^{-\alpha} (\alpha > 0)$
General multi quadratics	$f(x,a,b) = (b^2 + \lVert x-a\rVert^2)^{\beta} (0 < \beta < 1)$
Thin-plate s-spline function	$f(x,a,b) = (b\lVert x-a\rVert)^2 \ln(b\lVert x-a\rVert)$

3 Neural Network Training by EDA and PSO

3.1 Estimation of Distribution Algorithm (EDA)

(EDA) [16][17][18][19][20][21] is a new class of EAs. EDA directly extracts the global statistical information about the search space from the search so far and builds a probabilistic model of promising solutions. New solutions are sampled from the model thus built. Several EDAs [18][19][20][21] have been proposed for the global continuous optimization problem. These algorithms are very promising, but much work needs to be done to improve their performances. An efficient evolutionary algorithm should make use of both the local information of solutions found so far and the global information about the search space. The local information of solutions found so far can be helpful for exploitation, while the global information can guide the search for exploring promising areas. The search in EDAs is mainly based on the global information, but DE on the distance and direction information which is a kind of local information. Therefore, it is worthwhile investigating whether combining DE with EDA could improve the performance of the DE algorithm and EDA.

One of the major issues in EDAs is how to select parents. A widely-used selection method in EDA is the truncation selection. In the truncation selection, individuals are sorted according to their objective function values. Only the best individuals are selected as parents.

Another major issue in EDAs is how to build a probability distribution model p(x). In EDAs for the global continuous optimization problem, the probabilistic model p(x) can be a Gaussian distribution [11], a Gaussian mixture [12][13], a histogram [14], or a Gaussian model with diagonal covariance matrix (GM/DCM) [12].

GM/DCM is used in our algorithm. In GM/DCM, the joint density function of the k-th generation is written as follows:

$$p_k(x) = \prod_{i=1}^{n} N(x_i; \mu_i^k, \sigma_i^k) \tag{2}$$

where

$$N(x_i; \mu_i^k, \sigma_i^k) = \frac{1}{\sqrt{2\pi}\sigma_i} exp(-\frac{1}{2}(\frac{x_i - \mu_i}{\sigma_i})^2) \tag{3}$$

In (2), the n-dimensional joint probability distribution is factorized as a product of n univariate and independent normal distributions. There are two parameters for each variable required to be estimated in the k-th generation: the mean, μ_i^k, and the standard deviation, σ_i^k. They can be estimated as follows:

$$\hat{\mu}_i^k = \bar{x}_i^k = \frac{1}{M} \sum_{j=1}^{M} x_{ji}^k \tag{4}$$

$$\hat{\sigma}_i^k = \sqrt{\frac{1}{M} \sum_{j=1}^{M} (x_{ji}^k - \bar{x}_i^k)^2} \tag{5}$$

Implementation of EDA for NN classifier. Before describing the details of EDA for training NN classifier, the issue of coding is presented. Coding concerns the way the weights, bias and the flexible activation function parameters of NN are represented by individuals. A float point coding scheme is adopted here. For NN coding, suppose there are M nodes in hidden layer and one node in output layer and n input variables, then the number of total weights is $n * M + M * 1$, the number of thresholds is $M + 1$ and the number of flexible activation function parameters is $M + 1$, therefore the total number of free parameters in a NN to be coded is $n * M + M + 2(M + 1)$. These parameters are coded into an individual or particle orderly.

Let $Pop(t)$ be the population of solutions at generation t. EDAs work in the following iterative way.

S1 Selection. Select M promising solutions from Pop(t) to form the parent set Q(t) by truncation selection method;
S2 Modeling. Build a probabilistic model p(x) based on the statistical information extracted from the solutions in Q(t);
S3 Sampling. Sample new solutions according to the constructed probabilistic model p(x);
S4 Replacement. Partly replace solutions in Pop(t) by the sampled new solutions to form a new population Pop(t + 1).

3.2 Parameter Optimization with PSO

The Particle Swarm Optimization (PSO) conducts searches using a population of particles which correspond to individuals in evolutionary algorithm (EA). A population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector $\mathbf{x_i}$. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a velocity vector $\mathbf{v_i}$. At each time step, a function f_i representing a quality measure is calculated by using $\mathbf{x_i}$ as input. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved so far in a vector $\mathbf{p_i}$. Furthermore, the best position among all the particles obtained so far in the population is kept track of as $\mathbf{p_g}$. In addition to this global version, another version of PSO keeps track of the best position among all the topological neighbors of a particle. At each time step t, by using the individual best position, $\mathbf{p_i}$, and the global best position, $\mathbf{p_g}(\mathbf{t})$, a new velocity for particle i is updated by

$$\mathbf{v_i}(\mathbf{t+1}) = \mathbf{v_i}(\mathbf{t}) + c_1\phi_1(\mathbf{p_i}(\mathbf{t}) - \mathbf{x_i}(\mathbf{t})) + c_2\phi_2(\mathbf{p_g}(\mathbf{t}) - \mathbf{x_i}(\mathbf{t})) \tag{6}$$

where c_1 and c_2 are positive constant and ϕ_1 and ϕ_2 are uniformly distributed random number in [0,1]. The term $\mathbf{v_i}$ is limited to the range of $\pm\mathbf{v_{max}}$. If the velocity violates this limit, it is set to its proper limit. Changing velocity this way enables the particle i to search around its individual best position, $\mathbf{p_i}$, and global best position, $\mathbf{p_g}$. Based on the updated velocities, each particle changes its position according to the following equation:

$$\mathbf{x_i}(\mathbf{t+1}) = \mathbf{x_i}(\mathbf{t}) + \mathbf{v_i}(\mathbf{t+1}). \tag{7}$$

A neural network classifier trained by PSO algorithm with flexible bipolar sigmoid activation functions at hidden layer were constructed for the breast-cancer data set. The issue of coding is similar with the one used in EDA-NN discussed above.

The simple loop of the proposed training algorithm for neural network is as follows.

S1 Initialization. Initial population is generated randomly. The learning parameters c_1 and c_2 in PSO should be assigned in advance.
S2 Evaluation. The objective function value is calculated for each particle.
S3 Modification of search point. The current search point of each particle is changed using Eqn.(2) and Eqn.(1).
S4 if maximum number of generations is reached or no better parameter vector is found for a significantly long time (100 steps), then stop, otherwise goto step **S2**;

3.3 Decision Tree Classification

For comparison purpose, a Decision tree induction is one of the classification algorithms in data mining. The Classification algorithm is inductively learned to construct a model from the pre-classified data set. Each data item is defined by values of the attributes. Classification may be viewed as mapping from a set of attributes to a particular class. The Decision tree classifies the given data item using the values of its attributes. The decision tree is initially constructed from a set of pre-classified data. The main approach is to select the attributes, which best divides the data items into their classes. According to the values of these attributes the data items are partitioned. This process is recursively applied to each partitioned subset of the data items. The process terminates when all the data items in current subset belongs to the same class. A node of a decision tree specifies an attribute by which the data is to be partitioned. Each node has a number of edges, which are labeled according to a possible value of the attribute in the parent node. An edge connects either two nodes or a node and a leaf. Leaves are labeled with a decision value for categorization of the data.

Induction of the decision tree uses the training data, which is described in terms of the attributes. The main problem here is deciding the attribute, which will best partition the data into various classes. The ID3 algorithm [12] uses the information theoretic approach to solve this problem. Information theory uses the concept of entropy, which measures the impurity of a data items. The value of entropy is small when the class distribution is uneven, that is when all the data items belong to one class. The entropy value is higher when the class distribution is more even, that is when the data items have more classes. Information gain is a measure on the utility of each attribute in classifying the data items. It is measured using the entropy value. Information gain measures the decrease of the weighted average impurity (entropy) of the attributes compared with the impurity of the complete set of data items. Therefore, the attributes with the largest information gain are considered as the most useful for classifying the data items.

To classify an unknown object, one starts at the root of the decision tree and follows the branch indicated by the outcome of each test until a leaf node is reached. The name of the class at the leaf node is the resulting classification. Decision tree induction has been implemented with several algorithms. Some of them are ID3 [12] and later on it was extended into C4.5 [13] and C5.0. Another algorithm for decision trees is CART [14]. Of particular interest to this work is the C4.5 decision tree algorithm. C4.5 avoids over fitting the data by determining a decision tree, it handles continuous attributes, is able to choose an appropriate attribute selection measure, handles training data with missing attribute values and improves computation efficiency. C4.5 builds the tree from a set of data items using the best attribute to test in order to divide the data item into subsets and then it uses the same procedure on each sub set recursively. The best attribute to divide the subset at each stage is selected using the information gain of the attributes.

4 Simulation Studies

4.1 The Data Set

The data for our experiments was prepared by the 1998 DARPA intrusion detection evaluation program by MIT Lincoln Lab. The data set contains 24 attack types that could be classified into four main categories namely *Denial of Service (DOS)*, *Remote to User (R2L)*, *User to Root (U2R)* and *Probing*. The original data contains 744 MB data with 4,940,000 records. The data set has 41 attributes for each connection connection record plus one class label. Some features are derived features, which are useful in distinguishing normal from attacks. These features are either nominal or numeric. Some features examine only the connection in the past two seconds that have the same destination host as the current connection, and calculate statistics related to protocol behavior, service, etc. These called same host features. Some features examine only the connections in the past two seconds that have same service as the current connection and called same service features. Some other connection records were also stored by destination host, and features were constructed using a window of 100 connections to the same host instead of a time window. These called host-based traffic features. R2L and U2R attacks don't have any sequential patterns like DOS and Probe because the former attacks have the attacks embedded in the data packets whereas the later attacks have many connections in a short amount of time. So some features that look for suspicious behavior in the data packets like number of failed logins are constructed and these are called contents features. The data for our experiments contains randomly generated 11982 records having 41 features [9].

This data set has five different classes namely *Normal DOS*, *R2L*, *U2R* and *Probe*. The training and test comprises of 5092 and 6890 records respectively. All the IDS models were trained and tested with the same set of data. As the data set has five different classes we performed a 5-class binary classification. The *normal* data belongs to class 1, *Probe* belongs to class 2, *DOS* belongs to class 3, *U2R* belongs to class 4 and *R2L* belongs to class 5.

4.2 Intrusion Detection by EDA-NN

A neural network classifier with structure {41-8-1} trained by EDA with flexible bipolar sigmoid activation functions were constructed using the training data sets and then the neural network classifier was used on the test data set to detect the different types of attacks. All the input variables were used for the experiments. Table 2 depicts the detection performance of EDA-NN for test data set.

4.3 Intrusion Detection by PSO-NN

For comparison purpose, a neural network classifier with structure {41-8-1} trained by PSO and with flexible bipolar sigmoid activation functions were also constructed using the same training data sets and then the neural network classifier was used on the test data set to detect the different types of attacks. All the input variables were used for the experiments. Table 2 depicts the detection performance of PSO-NN for test data set.

4.4 Intrusion Detection by DT

The important variables were also decided by their contribution to the construction of the decision tree. Variable rankings were generated in terms of percentages. We eliminated the variables that had 0.00% rankings and considered only the primary splitters or surrogates. This resulted in a reduced 12 variable data

Table 2. Detection performance using EDA-NN, PSO-NN and DT classification models for test data set

Attack Class	EDA-NN	PSO-NN	DT
Normal	**97.58%**	95.69%	82.32%
Probe	**95.57%**	95.53%	94.83%
DOS	**97.76%**	90.41%	77.10%
U2R	99.90%	**100%**	99.83%
R2L	**98.90%**	98.10%	94.33%

Table 3. Comparison of false positive rate (fp) and true positive rate (tp) for EDA-NN, PSO-NN and DT classifiers for test data set

Attack	EDA-NN		PSO-NN		DT	
Class	fp(%)	tp(%)	fp(%)	tp(%)	fp(%)	tp(%)
Normal	0.29	99.64	4.3	88.70	29.66	99.60
Probe	0.02	56.57	0.40	37.15	0.24	31.00
DOS	3.94	98.86	3.68	89.38	72.10	97.63
U2R	0.01	52.00	0.05	55.81	0.022	59.26
R2L	0.08	87.39	0.17	86.63	0.022	30.73

set with x_2, x_4, x_5, x_{11}, x_{22}, x_{23}, x_{24}, x_{27}, x_{30}, x_{31}, x_{32}, x_{34} as variables. The detection performance of the DT by using the original 41 variable data set is shown in Table 2.

The achieved true positive and false positive rates using 41 input variables by the EDA-NN, PSO-NN and DT algorithms are depicted in Table 3.

5 Conclusions

In this paper, we have illustrated the importance of evolutionary algorithm and neural networks based techniques for modeling intrusion detection systems. EDA-NN classification accuracy is grater than 95% for all the considered attack types (classes) and achieved good true positive and false positive rates. It is to be noted that for real time intrusion detection systems EDA and neural networks would be the ideal candidates because of its simplified implementation.

Acknowledgment

This research was partially supported the Natural Science Foundation of China under contract number 60573065, and The Provincial Science and Technology Development Program of Shandong under contract number SDSP2004-0720-03.

References

1. R.C. Summers, Secure Computing: Threats and Safeguards. McGraw Hill, New York, 1997.
2. M. Debar, D. Becke, and A. Siboni. "A Neural Network Component for an Intrusion Detection System". *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, 1992.
3. S. Mukkamala, A.H. Sung and A. Abraham, "Intrusion Detection Using Ensemble of Soft Computing Paradigms", *Advances in Soft Computing, Springer Verlag, Germany*, pp. 239-248, 2003.
4. K. Shah, N. Dave, S. Chavan, S. Mukherjee, A. Abraham and S. Sanyal, "Adaptive Neuro-Fuzzy Intrusion Detection System", *IEEE International Conference on ITCC'04*, Vol. 1, pp. 70-74, 2004.
5. A. Abraham, Evolutionary Computation in Intelligent Web Management, Evolutionary Computing in Data Mining, A. Ghosh and L. Jain (Eds.), Studies in Fuzziness and Soft Computing, Springer Verlag Germany, Chapter 8, pp. 189-210, 2004.
6. Y. Chen, B. Yang and J. Dong, "Nonlinear System Modeling via Optimal Design of Neural Trees", *International Journal of Neural Systems*, 14(2):125-137, 2004.
7. Y. Chen, B. Yang, J. Dong, and A. Abraham, "Time-series Forecasting using Flexible Neural Tree Model", *Information Science*, 174(3-4): 219-235, 2005.
8. Y. Chen, A. Abraham, "Feature Selection and Intrusion Detection using Hybrid Flexible Neural Tree", ISNN-05, LNCS 3498, pp. 439-444, 2005.
9. KDD cup 99, `http://kdd.ics.uci.edu/database/kddcup99/kddcup.data_10_percent.gz`

10. S. Chebrolu, A. Abraham, J. P. Thomas, Feature Detection and Ensemble Design of Intrusion Detection Systems. *Computers and security*, Vol. 24/4, pp. 295-307, 2005.
11. Barbara D., Couto J., Jajodia S. and Wu N., ADAM: A Testbed for Exploring the Use of Data Mining in Intrusion Detection. *SIGMOD Record*, 30(4), pp. 15-24, 2001.
12. J. R. Quinlan. Induction of Decision Trees. Machine Learning, 1:81-106, 1986.
13. J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.
14. Brieman L., Friedman J., Olshen R., and Stone C., Classification of Regression Trees. Wadsworth Inc., 1984.
15. D. Joo, T. Hong, I. Han, The neural network models for IDS based on the asymmetric costs of false negative errors and false positive errors, *Expert Systems with Applications*, Vol. 25, pp. 69-75, 2003.
16. A. Ochoa, H. Muhlenbein, M. Soto, A Factorized Distribution Algorithm Using Single Connected Bayesian Networks, *PPSN*, 787-796, 2000.
17. M. Pelikan, D. E. Goldberg and E. Cantu-Paz, BOA: The Bayesian Optimization Algorithm, *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO99)*, I, 525-532, 1999.
18. S. Rudlof and M. Koppen, Stochastic Hill-Climbing with Learning by Vectors of Normal Distributions, Nagoya, Japan, 1996.
19. P. Larranaga and J. A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer Academic Publishers, 2001.
20. P. A. N. Bosman and D. Thierens, Expanding from Discrete to Continuous EDAs: The IDEA, *In Proceedings of Parallel Problem Solving from Nature*, PPSN-VI, 767-776, 2000.
21. S. Tsutsui, M. Pelikan, and D. E. Goldberg, Evolutionary Algorithm Using Marginal Histogram Models in Continuous Domain, *In Proceedings of the 2001 Genetic and Evolutionary Computation Conference Workshop*, 230-233, San Francisco, CA, 2001.
22. S. Mukkamala, A.H. Sung, A. Abraham, Intrusion detection using an ensemble of intelligent paradigms, *Journal of Network and Computer Applications*, 28(2): 167-182, 2005.
23. S. Chebrolu, A. Abraham, J.P. Thomas, Feature deduction and ensemble design of intrusion detection systems, *Computers and Security*, 24(4): 295-307, 2005.

Neural Network Implementation in Reprogrammable FPGA Devices – An Example for MLP*

Marek Gorgoń and Mateusz Wrzesiński

AGH University of Science and Technology, Biocybernetic Laboratory,
Department of Automatics, Al. Mickiewicza 30,
30-059 Kraków, Poland
mago@agh.edu.pl, mmtww@o2.pl

Abstract. In the present paper an implementation of multilayer perceptron (MLP) in a new generation SRAM-FPGA device is discussed. The presented solution enables easy realization of MLP with arbitrary structure and calculation accuracy. The solution is based on utilization of the structural parallelism of the FPGA device and economical realization of individual neurons. A flexible and effective method has been applied for approximation of the nonlinear activation function by a series of linear segments. Selection mechanisms have been also introduced for a compromise between the amount of logical resources used and the network operation speed. Therefore the presented solution can be applied both for implementation of big networks in small FPGA devices and for implementation working in real time, for which high operation speed is required.

1 Introduction

The implementation of artificial neural networks in FPGA devices has been the subject of many research papers. In principle two groups of solutions can be distinguished: universal type realizations [6], [7] and dedicated neural networks for specific applications [1], [2], [3]. Several solutions are concentrated on the neuron model [8]. Other publications discuss the network implementations providing the possibility of network's learning by making use of dynamic reconfiguration of the FPGA device [4], [5]. The proposed solutions mostly exhibit the scientific research features, with limited prospects for mass (large-scale) realization. The decreasing prices of FPGA devices with constantly increasing capacities stimulates the search for economically effective solutions for applications in home appliances. Depending on the specific nature of the application the networks with specific geometries are applied (understood as the number of layers and the

* This work was supported from the found for Scientific Research in years 2003-2006. We wish to thank the Celoxica University Program and Xilinx University Program for software donation.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 19–28, 2006.
© Springer-Verlag Berlin Heidelberg 2006

number of neurons in particular layers). The requirements regarding the precision and speed of the calculations may vary, and so may the available hardware resources.

Therefore universal solutions are required, which offer arbitrary choice of geometry for the implemented network and the arbitrary calculation precision, still preserving the required efficiency. It should be taken into account that in some applications high operation speed is expected, which should be ensured by parallel operation of considerable hardware resources, while in some other it is quite the opposite - limited resources are used even if it compromises the operation speed.

In the present paper a method is presented for implementation of multilayer perceptron (MLP) in FPGA devices of the Virtex and Spartan-II series (Xilinx), which fulfils the above-mentioned requirements.

The proposed solution is scalable. This means that starting from simple, replicable elements one can choose arbitrary geometry and calculation precision as well as the amount of resources that are available for a that specific implementation. At the same time the attention has been also focused on achieving the highest possible efficiency, independent of the choice of the above-mentioned parameters. The efficiency is regarded as the relation of the speed and precision of the calculations to the number of logical resources used (measured by the number of blocks in the FPGA matrix). The calculation speed has been taken as the number of synaptic connections realized in a given time unit.

2 Data Format in Reconfigurable Computing Systems

Reconfigurable Computing Systems based on reprogrammable FPGA devices, allow for designing of hundreds processing elements working in parallel, so they are particularly useful for neural nets implementation. However in FPGA's the fixed-point data format is preferred. A floating-point unit consumes huge number of logical resources and considerably reduces computation speed. In most cases the involved function, e.g. sigmoidal function, contains an exponential element, the calculation of which usually requires quite a developed system. Taking into account the neural net implementation in a reprogrammable device one should consider simplified model of a neuron and apply fixed-point algorithm. On the other hand a coarse approximation by simpler functions, easier to implement, may disturb the correct operation of the network. For a given application, minimal fixed-point data representation is required, which preserves a recognition level similar to the one achieved in floating-point algorithm. In consequence some approximations of the nonlinear activation function have to be investigated.

In the presented neural processor an intermediate solution has been applied, lying between the simple arithmetic and the function tabulation. The activation function has been approximated as a series of maximum 64 linear segments. In practice quite limited number of segments (approx. 10-100) leads to sufficiently accurate approximation. As an example a net consisting of 2 layers has been implemented for handwritten digit recognition. The original sigmoidal function has been approximated by three linear segments.

An attempt has been made to determine how the recognition effectiveness will be depend on the representation of the weight coefficients and what slope value of the central section of the sigmoidal function, responsible for the active part of the activation function, will ensure the recognition rate value similar to one achieved with real sigmoidal function.

The results listed in Table 1 show, that there is a critical resolution of weight coefficients, below which the network rapidly decreases its recognition quality, while the sensitivity to the slope angle of the central section is negligible (with the exception of the 1/16 value, for which the function is almost flat and it actually never works in the saturation region).

Table 1. Recognition rates (in %) as a function of resolution of the weight coefficients and slope of the central section of activatioin function (the nearest: 1/4 ; 6)

Activation function approx	Bit resolution of weight coefficients											
Slope of the central section	3	4	5	**6**	7	8	9	10	11	12	13	14
Threshold (∞)	10	54	81	**91**	94	94	92	91	91	91	91	91
4	10	58	81	**93**	95	95	93	92	92	92	92	92
2	12	62	82	**93**	95	94	93	93	93	93	93	93
1	11	69	83	**95**	96	95	95	93	93	93	93	93
1/2	10	75	87	**96**	96	95	94	93	93	93	93	93
1/4 (nearest)	**19**	**84**	**95**	**96**	**96**	**95**	**95**	**95**	**95**	**95**	**95**	**95**
1/8	36	81	97	**96**	97	96	95	95	95	95	95	95
1/(16)	30	33	26	**28**	29	29	28	30	32	32	32	32

The realized numerical experiment, in spite of the simplicity of its assumptions, definitely shows that the fixed-point hardware implementation, can relatively well substitute for the floating-point network model.

3 Implementation of Numerical Tasks

From the mathematical point of view MLP is a function mapping one vector space into another vector space. Such a function is a composition of transformations realized by individual layers i.e. simple perceptrons. Therefore there are two most important numerical tasks. First - called a synaptic function - consists of summation of series of products. In the algebraic sense it is a matrix multiplication by a vector. The second task consists of calculation of the activation function values for the obtained sums. For an M-layer perceptron the tasks should be realized interchangeably M times. The results obtained in one stage are used in the next stage. In typical applications MLP is used as a tool for classification. Finding the number of the recognized class and checking the reliability of the recognition are treated as a separate, third numerical task.

The proposed MLP implementation method is based - generally speaking - on the application of specialized subsystems for realization of particular, elementary tasks of the system. The main goal of the design was to enable independent, parallel and thus fluent operation of individual subsystems, without any forced halts. Therefore a pipelined data flow organization and proper buffering of intermediate results has been introduced. The internal BlockRAM memories - available in Virtex/Spartan series devices - have been used as data buffers. The BlockRAM memories are very fast, two-gate static memories of relatively large capacities. Because of the fast and easy access the application of that memory resulted in an efficient data flow between individual subsystems - and what follows - high calculation efficiency of the whole system. BlockRAM memories are particularly well fit for application as data buffers, which can be at the same time being written by one block and being read by another block.

In the designed system each of the three numerical tasks mentioned above are realized by a separate, specialized subsystem - as shown in Figure 1:

– synaptic function,
– activation function,
– analysis of the output vectors.

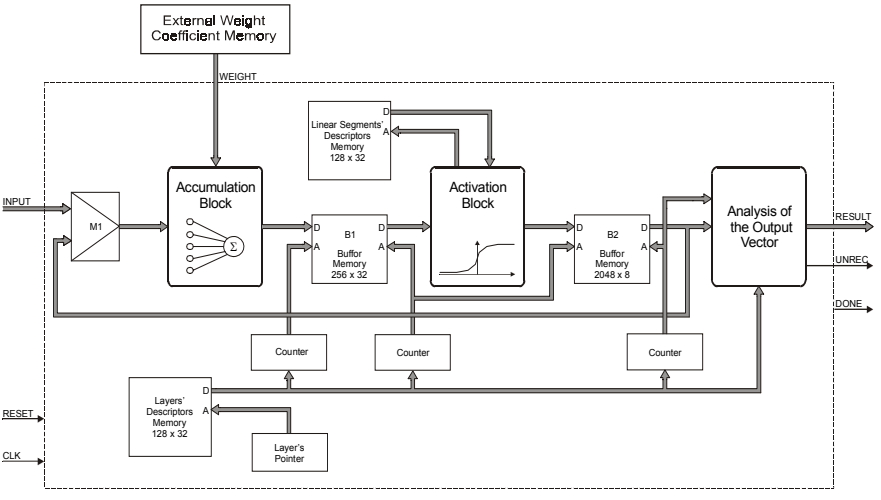


Fig. 1. Block diagram of calculation system

Computations for a following layers are performed sequentially on the same hardware structure (the loop and the multiplexer M1 – Figure 1). Descriptors of activations functions and layers' descriptors are stored in the BRAM memories, and are loaded during FPGA configuration. The synaptic block is connected to an external SRAM memory, from which the weight values are fetched. Layer's pointer and counters are responsible for system synchronization.

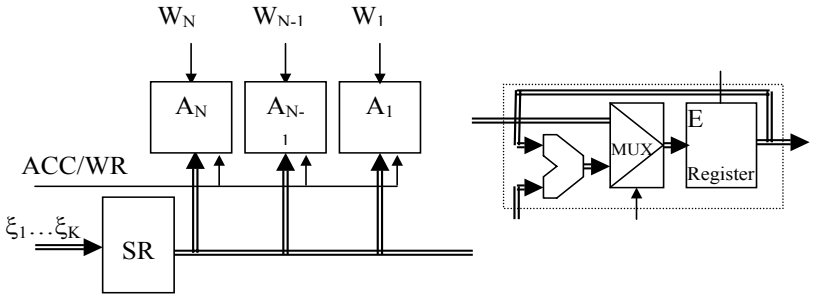


Fig. 2. Synaptic block: a) connection of the serial register with the set of accumulators, b) structure of an individual accumulator

Implementation of the Synaptic Function. Architecture of the Accumulation Block is presented on Figure 2a, and the architecture of single accumulator on Figure 2b.

For a layer consisting of N neurons, to which K inputs are connected, the realization of synaptic function requires the calculation of N sums $h_1, \ldots, h_N$ of the product sequence of length K

$$h_n = \sum_k w_{n,k}\xi_k \tag{1}$$

where ξ_k is the element of the vector subject to transformation (input layer) and $w_{n,k}$ are values of the weight coefficients. The realization of the synaptic function consists of repeated operations of multiplication and accumulation. In the presented solution serial multiplication has been applied, which includes addition of properly weighted multiples of one factor controlled by the values of individual bits in the second factor. That both the multiplication of input values by the weight coefficients and the accumulation of consecutive products are realized by the same element - adder coupled to a register. The system consists of a shift register SR and a set of accumulator registers $A_1, \ldots, A_N$. The number of accumulators is equal to the number of neurons in a layer and it determines the level of parallelization of the calculations. The accumulator implementation requires the number of logic cells (LC) equal to its resolution. Because each LC contains only one register further reduction of the accumulator size is not possible. Thus a kind of minimal hardware representation for a single neuron has been achieved.

Implementation of the Activation Function. As has been explained in the Chapter 2 the activation function is approximated by a series of linear segments. It's shape is determined by the table of records defining individual segments. The individual i-th record contains the following information:

- h_i – the h coordinate of the segment's start point
- O_i – the O coordinate of the segment's start point
- a_i – the slope of the segment.

Because the set of intervals covers the whole h domain in a continuous way, therefore the i-th segment of the set covers the interval $\langle h_i, h_{i+1})$. The realized approximation of the activation function can be formally written as:

$$O(h) = O_i + a_i(\Delta h), \qquad \Delta h = h - h_i \tag{2}$$

where i satisfies the following inequality:

$$h_i \leq h < h_{i+1} \ . \tag{3}$$

The relations (2) and (3) define the calculation tasks of the activation block. The first task consists of finding the segment covering the interval containing the h value - i.e. the solution of inequality (3) with respect to i. The second tasks comprises the reading data from the record describing the required i-th segment and executing the calculations according to (2).

The search for the proper segment is realized by a method similar to the successive approximation used in AD converters. It has been assumed that the total number of segments is an integer power of 2. Individual bits of the required i value are determined in succession, starting from the MSB bit, in consecutive clock cycles. The whole procedure looks as follows. Let's assume that the approximation uses 64 segments. In the first cycle the h value is compared with h_{32}. If it is smaller it has been determined that $i \in [0, 31]$, otherwise $i \in [32, 63]$. Thus the MSB bit of the number i has been determined. In the next cycle the h value is compared - depending on the previous result - with h_{16} or h_{48}, and as a result the next bit is determined. The total number of clock cycles required for determination of the number i satisfying the inequality (3) is given by the base-2 logarithm of the total number of segments.

As has been mentioned at the beginning of the present chapter the h values are signed numbers, encoded in a shifted binary code (the $100\ldots0$ binary value denote zero). Such encoding allows the comparison of unsigned numbers, what makes the comparator design much simpler. It is obvious that the h_i parameters must be encoded in the same way.

The second task - defined by the (2) - is realized as a serial multiplication:

$$O(\Delta h) = O_i + \sum_{j=0}^{Q-1} 2^{-j} z_i \Delta h^{e_j - j} \tag{4}$$

where

$$a_i = 2^{-e_j} z_i \ . \tag{5}$$

The Δh^n symbol denotes the n-th bit of the Δh value. The slope coefficient for a given segment is stored in the table of records as a floating point number - its value is defined by two integer numbers: z_i, e_i. The Δh value can not be greater than the width of the interval. Therefore the calculations realized according to (4) will be correct if the e_i value will be not less than the number of the most significant bit in Δh, which can takes the value 1:

$$e_i > \log_2(h_{i+1} - h) - 1 \ . \tag{6}$$

It is obvious that in order to obtain the most accurate representation of the a_i coefficient the lowest value of e_i, satisfying the (6), should be taken.

The resolution of O_i and z_i parameters should be the same as the resolution of the calculated output value O (therefore the same as for the ξ values). However the resolution of the sum values h can be much higher, because of the resolution excess in the accumulators, resulting from addition of many numbers with resolution of ξ values. If a layer is connected to, let's say, 1000 inputs than the resolution of accumulators should be by 10 bits higher than the resolution of ξ values. In a typical activation function the dominant part of the h domain lies in the saturation region. The width of the active region (near zero or near the middle of the interval) is rather small compared to the width of the whole interval. In the active region one should expect a great number of short segments with high slope coefficients, while in the saturation region small number of segments with slope coefficients sometimes several orders of magnitude smaller. The chosen realization of the calculations determined by (4) ensures the preservation of constant multiplication accuracy, independent of the slope coefficient values. The multiplication resolution Q should obviously correspond with the resolution of the calculated output value O. The calculation time - when the single accumulator implementation is used - is equal to $Q + 1$ clock cycles.

The separate subsystems realizing the two tasks of the activation block operate in complete independence. The total calculation time for one value is equal to the sum of the operation times of both subsystems. However because of the pipelined data flow the throughput is limited only by the operation time of the slower one. Higher throughput of the activation block is the necessary condition for ensuring a continuous operation of the synaptic block - as has been shown in the previous chapter.

An arbitrary number of segments can be taken, accordingly to the required approximation accuracy and the size of available memory. The search for the required segment record is realized in logarithmic time. Because of the typical shape of the activation function - steep active region and flat saturation regions - a special mechanism has been introduced, which preserves constant calculation accuracy for wide variation of segment slope coefficients. There is also a possibility to apply various shapes of the activation function for individual perceptron layers. The only modification required are separate tables describing the individual function shapes.

Implementation of the Output Vector Analysis. As mentioned finding the number of the recognized class and checking the reliability of the recognition are treated as a separate numerical task. Formally it consists of finding the number c of the greatest element of the y vector, according to (7) and testing the condition (8) - its fulfillment informs that the classification is reliable. In the (7) and (8) the C symbol denotes the size of vector y, or the number of classes recognized.

$$c: \; y_c = \max_{n \in \langle 1, C \rangle} y_n \tag{7}$$

$$n \in \langle 1, C \rangle \wedge n \neq c \Rightarrow y_c \geq y_n + \Delta \; . \tag{8}$$

The first phase described by (7) starts when the activation block writes the first element of the y vector in the B2 memory (Fig. 1) and it lasts until the activation block finishes its work. In the second phase the elements of the y vector are read once again from the B2 memory and compared to the determined maximum value, in order to check whether the condition (8) has been satisfied.

4 System Configuration

According to what has been declared in the introduction the presented solution should enable the MLP implementations of arbitrary geometry. Additionally depending on the required calculation speed it should allow the attribution of various amounts of hardware resources for a given implementation. In the present chapter different versions of the system are discussed in the context of the abovementioned problems.

In Chapter 3 the synaptic block has been described under the assumption that the number of accumulators is equal to number of neurons. However sometimes a situation may occur that the number of neurons is so high that the respective number of accumulators can not be fitted in the attributed FPGA part. In such a situation the calculations for such layer can be divided into several stages. If the synaptic block contains P accumulators and the layer includes N neurons ($N > P$) in the first stage the sums $h_1 \ldots h_P$ are calculated, then in the second stage $h_{P+1} \ldots h_{2P}$ etc., until all the sums are done. Each of the stages consists of the accumulation phase and the writing phase, during which the calculated group of sums is being written to the proper location in the buffer memory B1 (Fig.1). If the number of neurons is not integer multiple of the accumulators number then in the last stage some of the accumulators are not employed. During each stage the synaptic block reads from the B2 (Fig.1) memory the full set of values $x_1 \ldots x_k$. Therefore the contents of that memory cannot be modified before the last accumulation phase is completed. Only after its completion the activation block can start its operation.

If the number of accumulators is equal to the number of neurons, the total time of synaptic calculations (the accumulation phase) is equal to $(K + 1) * R$ clock cycles (K denotes the number of inputs and R the weight values resolution). In some cases, particularly for small layers with great number of inputs, it turns out that such calculation time is too long and the implemented accumulators do not exhaust all the resources available. Thus the situation is quite opposite to the one described above. Then several accumulators can be attributed to work to one neuron. Each of the neurons calculates a partial sum of a given h_n sum. During the writing phase these partial results are added and then the final result is written to the memory. Synaptic block with two accumulators attributed to one neuron requires the application of two shift registers SR instead of one and an additional adder for adding the partial sums in the writing phase. Synaptic block with a multiplicated accumulator number fetches not one but several x_i values every R cycles. However it may turn out that the capacity of a single activation block is not sufficient and several such blocks, working in parallel, have to be applied.

In Chapter 3 it was assumed that one synaptic block and one activation block realize in sequence the calculations for consecutive MLP layers. An alternative way of implementation is to attribute separate blocks for each layer. In that concept the individual layers work in pipelined mode. When a given layer completes the calculations it transfers the results to the next layer and fetches new input data.

The first method - with single synaptic and activation blocks - is optimal in situations when a short response time is expected but the data sent for processing do not arrive with high frequency. The basic problem is related to the fact that each layer may have a different number of neurons while the synaptic block contains always the same number of accumulators. Therefore for the bigger layers the process should be divided into stages while for the smaller layers - on the contrary - each sum should be distributed onto several accumulators.

The second method of MLP implementation - with separate blocks for each layer - requires much more hardware resources, but it offers proportionally higher throughput of the system. The method may be very useful in situations when the data for processing arrive at the well-known and constant frequency. Then the number of accumulators in individual synaptic blocks should by chosen in a way which results in processing times matched to the actual frequency of data arrival.

5 Results and Conclusion

As a results of the realization of implementation with 10 neurons (with 32-bit wide accumulation elements) in a medium-sized Virtex FPGA device, only 598 LC has been used, what comprises 10% of the whole device resources. The analysis of time reports of the implementation has shown, that the maximum clock rate of the FPGA device was 50 MHz. It has been estimated that the maximum total number of accumulators, that can be implemented in the XCV1000 device is 759.

The environment of reconfigurable devices is well suited for implementation of neural networks. The presented solution provides an example of the universal method of multi-layer perceptron (MLP) implementation. The possibility of free choice of the number of accumulators allows both the implementations of very large networks in relatively small devices or the application of large hardware resources for achievement of high calculation speed by their massive parallelization. The method used for activation function calculations allows the selection of its practically arbitrary shape and approximation accuracy. The universality of the solution is also supported by the fact that a decrease in the calculation accuracy is directly translated to an increase of the operation speed and smaller consumption of resources. The above statement concerns both the synaptic function, where serial multiplication has been applied with operation time proportional to the weight values resolution, and the activation function, where the number of segments and multiplication precision - thus the approximation accuracy - directly translates to the calculation time and the amount of memory occupied. That fact allows the full utilization of calculation capabilities of the implemented

logic for achieving the calculation accuracy, which is actually required in a given application. At the present stage such a solution is applied to networks trained in advance (i.e. networks with fixed and known values of synaptic weights), thus the learning algorithm is not implemented.

Thus the presented method can provide a good starting point for many practical applications of neural networks and also provide a good basis for algorithms for automated generation of the system structure starting from the set of neural network parameters provided by the user.

References

1. Chorti, A. et al.: *An electronic system for simulation of neural networks with a micro-second real time constraint*. American Institute of Physics Conference Proceedings, No. 583 USA, (2001), pp. 76–79
2. Djones L., Braun A., Bodgan M., Gerlach J., Rosenstiel W.: *Synthesis of Embedded SystemC Design: A Case Study of Digital Neural Networks*. Proceedings of the Design, Automation and Test in Europe Conference and Exhibition Designers' Forum (DATE'04). IEEE CS 1530-1591/04. IEEE Comput. Soc. Los Alamitos (2004), pp. 30248–30253
3. Freeman M., Austin J.: *Designing a Binary Neural Network Co-processor*. Proceedings of the 2005 8th Euromicro Conference on Digital System Design (DSD'05). IEEE CS 0-7695-2433-8/05. IEEE Comput. Soc. Los Alamitos (2005), pp. 223–227
4. James-Roxby, P., Blodget, B.: *Adapting constant multipliers in neural network implementation*. Proceedings of IEEE Symposium on FCCM. IEEE CS No. PR00871. IEEE Comput. Soc. Los Alamitos (2000), pp. 335–336
5. James-Roxby, P., Blodget, B.: *A study of high performance reconfigurable constant coefficient multiplier implementations*. SPIE Proceedings, Vol. 4212 USA, (2000), pp. 150–161
6. Restrepo, H. F. et al.: *A networked FPGA-based hardware implementation of a neural network application*. Proceedings of IEEE Symposium on FCCM. IEEE CS No. PR00871. IEEE Comput. Soc. Los Alamitos, CA, USA (2000), pp. 337–338
7. Vitabile S., Conti V., Gennaro F., Sorbello F.: *Efficient MLP Digital Implementation on FPGA*. Proceedings of the 2005 8th Euromicro Conference on Digital System Design (DSD'05). IEEE 0-7695-2433-8/05. IEEE Comput. Soc. Los Alamitos (2005), pp. 218–222
8. Wang-Qinruo; Yi-Bo; Xie - Yun, Liu-Bingru: *The hardware structure design of perceptron with FPGA implementation*. SMC'03-Conference-Proceedings. IEEE Cat. -No.03CH37483, Vol.1. IEEE, Piscataway, NJ (2003), pp. 762–767

A New Approach for Finding an Optimal Solution and Regularization by Learning Dynamic Momentum

Eun-Mi Kim[1], Jong Cheol Jeong[2], and Bae-Ho Lee[3]

[1] Dept. of Computer Engineering,
Chonnam National University, Republic of Korea
koreaeunmi@yahoo.com
[2] Dept. of Electrical Engineering & Computer Science,
The University of Kansas, USA
korjcjeong@yahoo.com
[3] Dept. of Computer Engineering,
Chonnam National University, Republic of Korea
bhlee@chonnam.ac.kr

Abstract. Regularization and finding optimal solution for the classification problems are well known issue in the machine learning, but most of researches have been separately studied or considered as a same problem about these two issues. However, it is obvious that these approaches are not always possible because the evaluation of the performance in classification problems is mostly based on the data distribution and learning methods; therefore this paper suggests a new approach to simultaneously deal with finding optimal regularization parameter and solution in classification and regression problems by introducing dynamically rescheduled momentum with modified SVM in kernel space.

1 Introduction

Regularization were introduced to solve ill-posed problems[9], and now days many researches show that regularization is one of the useful method for finding optimal solution in classification[2,3,4,6,11]. However, it is not always true that well regularized problems results optimal solution for the given problems because the theoretical base of regularization is for reorganizing the program into the well-posed problem. This means it cannot always guarantee to find optimal solution, but it can give high possibility to get the best result of the given ill-posed problem. Conversely, the best regularization must result from the optimal solution of a given ill-posed problem because the optimal solution is what most studies finally want from the regularization.

From this point of view, this paper introduces a new method to get an optimal solution with an optimal regularization parameter by learning dynamically scheduled momentum. To apply the new method, the modified Support Vector Machine (SVM)[4,11], which was basically based on the Structural Risk Minimization principle and guarantees the lowest true error[9,10], has been introduced, and Generalize

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 29–36, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Cross-Validation(GCV) and L-Curve are used for comparing performances. GCV and L-Curve were introduced as a method of finding the well defined regularization parameter[2]. Although these methods have an excellent algorithm and high efficiency to solve given problems, both methods have an unrecoverable defect that these algorithms need more than two processing steps to solve problems. This is the reason why GCV and L-Curve rarely used in practical classification problems; therefore the Bias Learning(BL) method[4,11] has been introduce to solve this problem in GCV and L-Curve algorithm. BL is using the kernel method[4,11] which is modified from sequential learning method and Mercer's theorem[3], and this can overcome the defect in GCV and L-Curve by simultaneously learning regularization parameter within the learning process of pattern weights[4,11].

However, the bias learning method is based on RBF neural networks; therefore, this method also has a potential possibility to be exposed to the classical machine learning problems such as, the stagnation in local optimum and over fitting; therefore, this paper shows that dynamic momentum can effectively solve those problems that the bias learning method has. The dynamic momentum is defined by both learning performance and epoch which are related with limited reciprocal proportion, and the momentum value is actively changed by a learning rate.

In short, the dynamic momentum method is a well scheduled momentum which reflects its learned feature vectors into the present learning state with preventing the convergence oscillation effect. This paper is dealing with two different types of data set to evaluate the dynamic momentum and compare the performance of classification and regression among GCV, L-Curve and the dynamic momentum.

In the first phase, the experiment shows the results of the comparing error scales between classification and regression problems. The second experiment is conducted under artificially composed ill-posed problem by Gaussian distribution. Finally this paper proves that the suggested dynamic momentum finds the regularization parameter more precisely than others.

2 Modified SVM

The original SVM[3,4,7,10,11] is defined as followed by

$$\begin{aligned} L(\alpha) &= b\sum_{i=1}^{N}\alpha_i - \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j \mathbf{x}_i^T\mathbf{x}_j \\ subject\,to\ \ (1)\ &\sum_{i=1}^{N}\alpha_i d_i = 0 \quad (2)\, 0 \le \alpha_i \le C \end{aligned} \tag{1}$$

Where b is margin, is weighting vector, d is the destination of training data, which is used to expressed with positive class and negative class, and x is input vector.

The modified SVM[4,11] is defined as followed by

$$L(\alpha) = b\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j \mathbf{y}_i^T\mathbf{y}_j \ \ subject\,to\ \ 0 \le \alpha_i \le C \tag{2}$$

As shown above the equation is almost same except above equation changed original vector $\mathbf{x}$ into augmented input vector $\mathbf{y}$, which is composed by input vector and the bias in the RBF neural networks and the second restriction in the equation (1) has been omitted. This is very important because the omitted equation condition makes possible that SMV can be applied to sequential learning methods by substituting kernel K[3,4,11] for inner product pattern $\mathbf{y}_i^T\mathbf{y}_j$.

3 Regularization

The cost function of the regularization to solve an ill-posed problem is defined by following equations [3,4,11].

$$J(\alpha,\lambda) = \frac{1}{2}\varepsilon^2 + \lambda\frac{1}{2}\alpha^t\mathbf{K}\alpha \qquad \varepsilon \;=\; \mathbf{d} - \mathbf{K}^t\alpha \tag{3}$$

Where α is weight vector, λ is regularization parameter, ε is error, and $\mathbf{K}$ is the kernel $\mathbf{K}$ which is from $\mathbf{y}_i^T\mathbf{y}_j$.

To get minimized cost value, the Lagrange multiplier is defined as followed by

$$\begin{aligned} L(\alpha,\varepsilon;\beta) &= \tfrac{1}{2}\varepsilon^2 + \lambda\tfrac{1}{2}\alpha^t\mathbf{K}\alpha + \sum_{i=1}^{N}\beta_i[d_i - \mathbf{K}(i,:)\alpha - \varepsilon_i] \\ \tfrac{\partial L}{\partial\alpha} &= \lambda\mathbf{K}\alpha - \sum_{i=1}^{N}\beta_i\mathbf{K}(i,:) = 0, \quad \lambda\alpha = \beta \\ \tfrac{\partial L}{\partial\varepsilon_i} &= \varepsilon + \textstyle\sum\beta_i(-1) = 0, \;\; \varepsilon \;=\; \beta = \lambda\alpha \\ \tfrac{\partial L}{\partial\alpha} &= -\lambda(\lambda\alpha - \mathbf{K}\alpha + \mathbf{d}) = 0, \;\; (\mathbf{K} + \lambda\mathbf{I})\alpha = \mathbf{d} \end{aligned} \tag{4}$$

Where $\mathbf{K}$(i,:) indicates the vector of i row's every column in kernel $\mathbf{K}$, and the regularization parameter has to be big enough to make possible to get the inverse matrix of $(\mathbf{K} + \lambda\mathbf{I})$. Therefore, above equation can be rewritten as followed by

$$\alpha_{reg} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{d}, \qquad \mathbf{d} = g_k(\mathbf{y}) = \mathbf{K}\alpha_{reg}. \tag{5}$$

4 Bias Learning

The structure of regularization RBF network is under-determined nonlinear system[3].

$$(\mathbf{K} + \lambda\mathbf{I})\alpha = \mathbf{d} \tag{6}$$

The relationship among the bias, a regularization parameter, and weight α, is defined as followed by. b

$$\begin{aligned} \mathbf{d} = \mathbf{K}\,\alpha + \lambda\,\alpha \;&= \mathbf{K}\,\alpha + \xi \;= \mathbf{K}\,\alpha + \bar{\xi} + (\xi - \bar{\xi}) \;= \mathbf{K}\,\alpha + \alpha_0 + \varepsilon \\ \alpha_0 = \bar{\xi} \equiv \tfrac{\sum\xi_i}{N} &= \tfrac{\lambda\sum\alpha_i}{N}, \qquad \lambda = \tfrac{\alpha_0}{\sum\alpha_i}N \end{aligned} \tag{7}$$

The solution for the problem of the pure regularization networks is following as[3].

$$F_\lambda(\mathbf{y}) = \alpha_{reg}^t\mathbf{K}(:,\mathbf{y}) \tag{8}$$

The solution of the modified regularization networks is as follows.

$$F_\lambda(\mathbf{y}) = \alpha_{reg}^t \mathbf{K}(:, \mathbf{y}) + \alpha_0 \tag{9}$$

The weight vector and the bias are simultaneously learning using augmented kernel in the algorithm. The pattern matrix is augmented to learn the bias as follows.

$$\mathbf{H}^t \omega = \mathbf{d}, \quad \mathbf{H} \equiv [\mathbf{K}\ ones(N, 1)], \quad \omega \equiv \begin{bmatrix} \alpha \\ \alpha_0 \end{bmatrix} \tag{10}$$

Where $[\mathbf{K}\ ones(N, 1)]$ is augmented matrix with initializing the last column with 1s. Therefore, the BL is sequential learning method with augmented kernel and weight matrix, so the regularization parameter, which is used to externally set by the user, can be learned while the weight vectors are learning by input patterns.

5 Dynamic Momentum

The basic concept of dynamic momentum scheduling is that the size of momentum value is getting decreased from initial state to convergence state. To apply this concept in the algorithm, the momentum has to be satisfied with certain conditions that the scale of the value of the momentum cannot exceed initialized momentum value, and the momentum value has to be regulated with smaller value than initial value[4,11].

$$\mathbf{M} = \frac{m(k+1)}{\tau} \qquad if \ \ \mathbf{M} > m \ \ then \ \ \mathbf{M} = 0; \quad \tau = \tau^2; \tag{11}$$

In (11), dynamic momentum M is automatically initialized, and reorganizes next epoch tolerance τ as changing its value to τ^2 when the value M is bigger than the upper bound m; therefore, M is continuously learning its momentum value under given conditions. In conclusion, compared with existing static momentum scheduling which choose its momentum value by user externally, and it cannot be changed during its learning time, dynamic momentum can find momentum value actively as to be affected by learning epoch into given scheduling method[4,6,7,8,11].

6 Experiment Environment

This experiment is conducted by two steps with using different data set. At the first step, we use IRIS data which is widely known as standard data for classification problem. Second, the gauss data which is a model of the singular matrix is used for exposing suggested method into ill-posed and more complex problems. The IRIS data is composed of three classes, Setosa, Versicolour, and Virginica. The Setosa can be completely separated from other two classes, so complete linear classification is possible, but the Versicolour and Virginica cannot

completely separate each other with either linear or nonlinear classification. The number of data of each class is 50, and each data is organized with four attributes, sepal length, sepal width, petal length, petal width. For this experiment, we used the Versicolour and Virginica in which the linear classification is impossible. At the second phase, Gaussian distribution data which is composed by 500 training data and 1000 testing data. This experiment prove that the GCV, L-Curve, and kernel method can be applied to big enough and general problems with performing good regularization and shows that regularization is essential on ill-posed problem or singular system. From now on, we attach K, BL and DM at the end of the name of kernel learning method which is applied to Kernel method, Bias Learning, and Dynamic Momentum respectively. In addition RMS indicates Root Mean Square error. For instance, KLMSBL is that Least Mean Squared (LMS) procedure is applied into Bias Learning in Kernel methods[1,4,6,11], and KLMSBLDM is applied into both Bias Learning and Dynamic Momentum. The learning weight vector in the classical LMS procedure in sequential learning is followed by[1,4,11]:

$$\alpha(i+1) = \alpha(i) + \eta(\cdot)\frac{d_i b - \alpha^t K(:,i)}{\|K(:,i)\|^2} K(:,i). \tag{12}$$

RMS error are calculated as followed by

RMS Error $= \sqrt{\frac{\|\mathbf{d}-\mathbf{K}\alpha\|^2}{N}}$, RMS Error with Bias $= \sqrt{\frac{\|\mathbf{d}-(\mathbf{K}\alpha+\alpha_0)\|^2}{N}}$.

7 Experiment Results

At the first phase experiment using Iris data, each kernel method is learning its regularization parameter. A word, Inverse, is added next to the name of kernel method to distinguish between the value from pure learning methods and from regularization parameters.

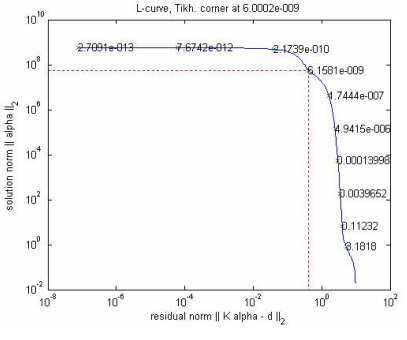


Fig. 1. L-Curve(Iris)

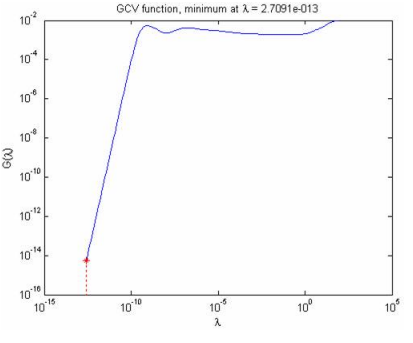


Fig. 2. GCV(Iris)

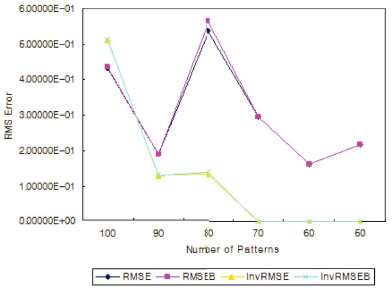


Fig. 3. KLMSBLDM(training)

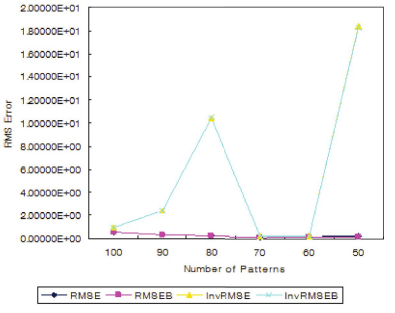


Fig. 4. KLMSBL(training)

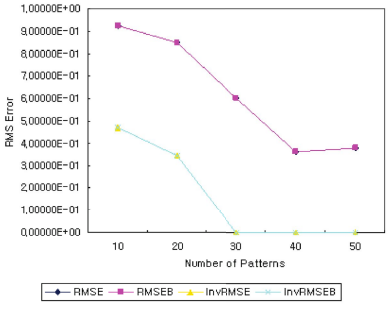


Fig. 5. KLMSBLDM(testing)

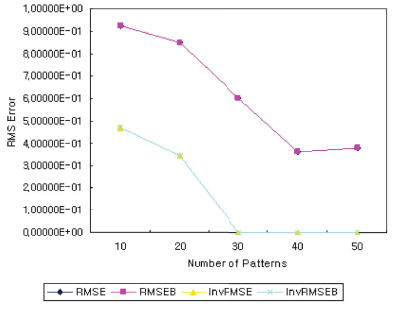


Fig. 6. KLMSBL(testing)

Fig.3 shows the graph which describes RMS errors under KLMSBLDM that applied dynamic momentum with changing the number of training data, and Fig.4 shows KLMSBL which does not apply dynamic momentum.

In Fig.3, although the values of both RMSE, which does not include a bias term, and RMSEB, which includes a bias term within learning weight vectors, are similar, RMS errors in the method using matrix inverse with regularization parameter are mostly lower than those of other methods.

In Fig.4, RMS errors in RMSE and RMSEB are similar each other, but compared with these values, InvRMSE and InvRMSEB have large values. These results show that the learning methods which do not include DM found local optimum value upon typical patterns or overly fitted in training data, but KLMSBLDM found globally optimized value. Therefore, the learning methods are ineffective without dynamic momentum although those methods are applied in the BL which is learning a regularization parameter. As a result, dynamic momentum is necessary to improve the precision of a regularization parameter.

Fig.5 shows that the RMS errors in learning methods and Regularization method are proportionally increased or decreased, but in Fig.6, those errors of KLMSBL appears big differences between the RMS errors in the learning methods and those values in Regularization methods. This results re same as what training data appeared.

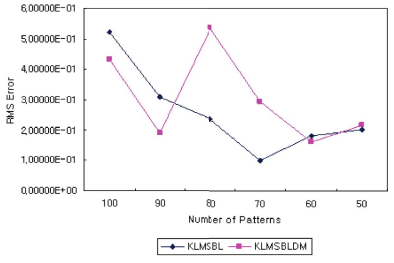


Fig. 7. RMS Error(training)

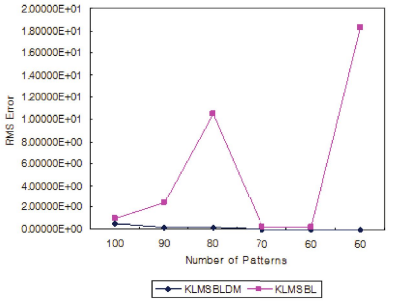


Fig. 8. Inverse RMS Error(training)

In short, compared with the Fig.5, the Fig.6 shows that the results of the given learning methods and the using regularization parameters are rarely related; therefore, these results also prove that learning regularization parameter is ineffectively adapted with learning weight vectors without DM.

Fig.7 and 8 which show the graph about RMS errors under learning KLMSBLDM and KLMSBL that separately learned weight vectors and regularization parameters by training data and testing data, also verify that this relationship between the RMS error in pure learning methods and those results in using the regularization parameter.

The result of KLMSBLDM shows that the methods using the regularization parameter with matrix inverse have proportionally similar graph shape with the graph patterns of RMS errors which resulted from learning methods under training and testing data, but those results appeared on KLMSBL have less relation than those of KLMSBLDM.

The second experiment with Gaussian distribution data, which represents complex and ill-posed system, shows every algorithm has equivalent performance both classification and regression problem except KLMSBL and KLMSBL Inverse[6].

Same as the result of the first experiment, the second experiment also shows that both RMS errors of KLMSBL between the result of pure learning and Inverse using regularization parameter are very different from those of other methods; therefore, this experiment proves that KLMSBL is overly treated against training data same as previous experiment did. Namely although the results of the classification problems seems to be stable with 82.6% and 83% same as others, it is just because of the condition of the applied data set. In fact, compared with the RMS error of pure learning, the inverse method using regularization parameter has a bigger error than other methods; therefore, this situation also shows that KLMSBL is overly treated against training data set. In summary, this second experiment proves that KLMSBL has been overly learned against input patterns, and also shows that the suggested dynamic momentum is stable to help both improving the classification performance and reducing the regression error rate.

8 Conclusions

Upon two experiments to evaluate classification performance and regression precision using L-Curve, GCV, and KLMS which is one of the kernel methods, the first and second experiment using Iris data and gauss data respectively proved that KLMS has potential problems of classical Neural Networks, which is well known for the convergence of the local value and over fitting, by compared two results between pure learning performance and calculating Inverse with regularization parameter. In addition, the experiments prove that the suggested dynamic moment effectively help to improve both classification performance and reducing the regression errors when KLMSBL is applied into dynamic momentum. The second experiment especially shows that the dynamic momentum helps to find more optimized regularization parameter than KLMSBL does upon a generalized problem.

Acknowledgements. This work was supported by the RRC-HECS, CNU under grant R12-1998-032-08005-0, and NURI-CEIA, CNU.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification second edition , John Wiley & Sons, Inc. New York, NY, (2001)
2. Hansen, P.C.: Regularization Tools, A Matlab Package for Analysis and Solution of Discrete Ill-Posed Problems. Version 3.1 for Matlab 6.0, (2001)
3. Haykin, S.: Neural Networks, A comprehensive Foundation Second Edition by Prentice-Hall, Inc, (1999)
4. Jeong, J.C.: A New Learning Methodology for Support Vector Machine and Regularization RBF Neural Networks, Thesis for the degree of the master of engineering, Department of Computer Engineering Graduate School, Yosu National University, Republic of Korea (2002)
5. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features, Proceedings of ECML-98, 10th European Conference on Machine Learning, (1998)
6. Kim, E.M, Park, S.M, Kim, K.H., Lee, B.H.: An effective machine learning algorithm using momentum scheduling, Hybrid Intelligent Systems, Japan, (2004) 442-443
7. Mangasarian O.L., Musicant, D.R.,: Active Set Support Vector Machine Classification, Neural Information Processing Systems 2000 (NIPS 2000), T. K. Lee, T. G. Dietterich and V. Tresp, editors, MIT Press, (2001), 577-583
8. Platt, J.C.: Fast Training of Support Vector Machines Using Sequential Minimal Optimization, In Advances in Kernels Methods : Support Vector Learning, ed., MIT Press, Cambridge, (1998)
9. Tikhonov, A.N.: On solving incorrectly posed problems and method of regularization, Doklady Akademii Nauk USSR, vol.151, (1963), 501-504
10. Vapnik, V.: Statistical learning theory, John Wiley and Sons, New York, (1998)
11. Yoo, J.H., Jeong, J.C.: Sparse Representation Learning of Kernel Space Using the Kernel Relaxation Procedure", Journal of Fuzzy Logic and Intelligent Systems v.11, no.9, (2002) 817-821

Domain Dynamics in Optimization Tasks

Boris Kryzhanovsky and Bashir Magomedov

Institute for optical-neural technologies RAS, Vavilova st. 44/2, 199333, Moscow, Russia
kryzhanov@mail.ru, bashir.magomedov@gmail.com
http://www.iont.ru

Abstract. A new type of dynamics of Hopfiled model - the domain dynamics - is proposed for using in optimization tasks. It is shown that this kind of dynamic allows one to find more deep minima of the energy than the standard asynchronous dynamics. It is important that the number of calculation does not rise when we replace standard spin dynamics by the domain one.

1 Introduction

The dynamics of well-known spin models of neural networks [1-9] consists in aligning of each spin along the direction of the local field.

In paper [10] we proposed a new type of neural network, which was called *the domain neural network*. Its dynamics is defined by overturns of domains. Each domain is a group of strongly coupled spins. Overturn of a domain means the simultaneous changing of orientations of all the *k* spins constituting the domain. In this paper we want to show that the domain dynamics can be efficiently used in optimization problems. The point is that this model allows us to find minima on the energy surface that are deeper than the ones obtained with the aid of the Hopfield model.

2 Domain Dynamics

Let us examine a system of *N* spins, which take the values $s_i = \pm 1$, where $i = 1,2,...,N$. The behavior of the system is described by the energy function

$$E = -\tfrac{1}{2}\sum_{i=1}^{N} J_{ij} s_i s_j \tag{1}$$

where J_{ij} is the Hebbian matrix [2],

$$J_{ij} = (1-\delta_{ij})\sum_{\mu=1}^{M} s_i^{(\mu)} s_j^{(\mu)} \tag{2}$$

based on the set of randomized binary patterns $S_\mu = (s_1^{(\mu)}, s_2^{(\mu)}, ..., s_N^{(\mu)})$, $\mu = 1,...,M$. The local field acting on the i^{th} spin is calculated according to the usual rule: $h_i = -\partial E / \partial s_i$.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 37 – 45, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Let us define the domain neural network. We suppose that the system of N spins is divided into groups each of which contains k spins. Each group is a domain. In the domain the spins are strongly coupled, and when the domain turns over, all the spins in the domain change their signs simultaneously. Thus, our system consists of N/k domains. When the state of the system is changing due to overturns of domains only, its dynamics is called *the domain dynamics*. From physical point of view the behavior of the domain network is determined by stability of domains in the local field. The given domain turns over, if as a result the energy of the system decreases. For example, let us examine the first domain, i.e. the group of coupled spins whose numbers are $1 \le r \le k$. To define the domain stability, let us write down its energy (the sum of the energies of all k spins constituting the domain) in the form of two terms. The first term is the intrinsic energy of the domain that is the energy of interaction of the spins of the domain. The second term is the energy of interaction of the given domain with other domains of the system (E_{int}) , i.e. the energy of interaction of the spins belonging to the given domain with spins of all other domains:

$$E_{\text{int}} = -\sum_{r=1}^{k}\sum_{j=k+1}^{N} J_{rj} s_r s_j \tag{3}$$

Evidently, the domain stability is defined completely by the sign of the interaction energy E_{int}. The value and the sign of the intrinsic energy of the domain are of no importance, since they do not change when the domain turns over. Consequently, the domain dynamics of the network is defined as follows. If at the time t inequality $E_{\text{int}}(t) > 0$ is fulfilled, then the domain turns over at the next step, i.e. it transforms to the state $s_r(t+1) = -s_r(t), \quad \forall r = 1,...,k$, with the negative interaction energy $E_{\text{int}}(t+1) < 0$. If $E_{\text{int}}(t) < 0$, then the domain is stable and at the next step its state is the same: $s_r(t+1) = s_r(t), \quad \forall r = 1,...,k$. Under the described dynamics the energy of the system as a whole decreases, and, consequently, the algorithm converges after finite number of steps. It should be stressed that a domain can overturn even if each of its spins is in the stable state, i.e. each spin is directed along the its own local field.

3 Efficiency of the Domain Algorithm

The properties of the domain neural network are described in [10]. In particular, the memory capacity of this type of neural network is shown to be k times greater than that of the conventional Hopfield network. We will not dwell on these properties because they have to do with the use of the domain network in an associative memory system rather than optimization algorithms. Instead, we will show that the efficiency of the domain algorithm is k^2 times higher than in the conventional spin dynamics.

Using the double indexing of spins, we can rewrite (1) as:

$$E = -\tfrac{1}{2}\sum_{n=1}^{N_D}\sum_{m=1}^{N_D}\sum_{\alpha=1}^{k}\sum_{\beta=1}^{k} J_{n\alpha,m\beta}\, s_{n\alpha}\, s_{m\beta} \tag{4}$$

where n, m are the domain numbers, and α, β are the spin numbers in these domains. The new indices are related to the indices of expression (1) as $i = k(n-1)+\alpha$, $j = k(m-1)+\beta$.

Redirection of a domain means a simultaneous change of direction of all constituent spins. Therefore, all k spin variables of the n-th domain can be represented as $s_{n\alpha} = d_n s^0_{n\alpha}$, $\forall \alpha = 1,...,k$, $\forall n = 1,...,N_D$. Here $s^0_{n\alpha} = s_{n\alpha}(t=0)$ is the original direction of a spin, and $d_n = d_n(t)$ is the domain variable that characterizes the behavior of a domain with time (the initial value of the variable is $d_n(t=0) = 1$). With these variables expression (5) takes the form:

$$E = E_0 - \tfrac{1}{2}\sum_{\substack{n,m=1 \\ n\neq m}}^{N_D} A_{nm}\, d_n\, d_m \tag{5}$$

where

$$A_{nm} = \sum_{\alpha=1}^{k}\sum_{\beta=1}^{k} J_{n\alpha,m\beta}\, s^0_{n\alpha}\, s^0_{m\beta} \quad , \quad E_0 = -\tfrac{1}{2}\sum_{n=1}^{N_D} A_{nn} \, . \tag{6}$$

We see that introduction of domain variable $d_n(t)$ allowed us to reduce the problem to the Hopfield model with matrix A_{nm}. The new notation brings the above-described domain behavior to the following: local field $h^A_n = -\partial E / \partial d_n$ is computed and the domain is directed along this field $d_n = \mathbf{sgn}(h^A_n)$. It can easily be found that $E_{\mathbf{int}} = d_n h^A_n$. This algorithm allows a k^2 times reduction in computations because the dimensionality of matrix A_{nm} is k times smaller than that of the origin matrix J_{ij}.

4 Optimization Capabilities of the Domain Neural Network

Replacement of the spin dynamics by the domain behavior implies an increased spacing of landscape traversal. The increase makes it possible to bypass minor local minima, which usually block the Hopfield network. By way of example a dotted line is used in Figure 1 to depict a landscape. The initial state of the neural network is point A. When the domain dynamics is used, the neural network converges to point B, skipping small local minima.

The conventional spin dynamics of the Hopfield model is usually compared to the travel of a material point across a landscape. In these terms, the domain network dynamics can be regarded as movement of a large object which does not react to small surface imperfections. Following up the analogy, one can expect the object to perceive the surface trend and travel towards deepest minima.

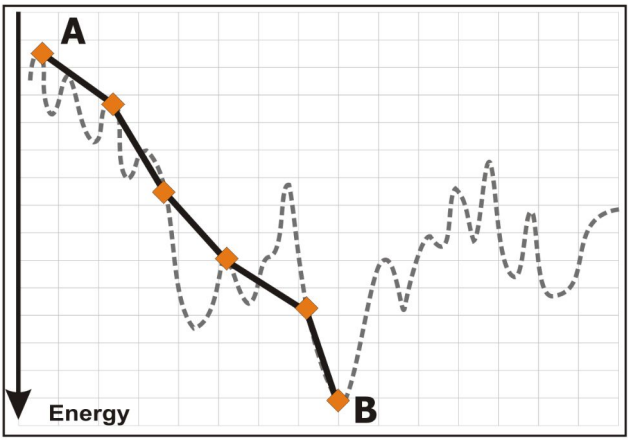


Fig. 1. Travel across a landscape. The dotted line represent the spin dynamics, and the solid line the domain descent. It is seen from the example that after redirection of 5 domain, the neural network converges from point A to B, skipping small local minima.

4.1 Experiment

Below a few experiments are described that we carried out to prove that the domain network dynamics helps a system to reach deeper minima with higher probability. As optimization capabilities of a neural network are strongly dependent on what kind of the interconnection matrix is used, matrices that had different spectral density of minima (from sparsely scattered minima to almost continuous spectrum) spectrum were investigated (see Figures 2 to 4). To model such spectra, we used Hebbian rule with different load parameters ($M/N = 0.05, 0.1, 0.2, 0.5, 1$) to form the matrices. Patterns are generated in a random way: the vector components are either +1 or -1 with probability ½. Besides, we used generalized Hebbian rule

$$J_{ij} = \sum_{\mu=1}^{M} w_\mu s_i^{(\mu)} s_j^{(\mu)} \tag{7}$$

to build the matrix and generate an energy surface with pronounced minima. Here $w_\mu \in (0,1)$ are the random statistical weights of the patterns obeying a uniform distribution. With such interconnection matrices, the energy relief (1) becomes more complex: different vectors embedded in the matrix correspond to local minima with different minima.

We used a two-stage optimization process. At first we generated a random configurational vector whose components were grouped in domains of size $k \geq 2$. Then the network converged to a local minimum with the help of the domain dynamics algorithm. Next, the network was deprived of rigid interconnects that grouped spins in domains, that is to say, parameter k was set to 1, and the descent continued using conventional spin dynamics of the Hopfield network. The fact that the domain neural net usually held unsatisfied spins after reaching a local minimum

necessitated the use of the method that combined domain and spin dynamics. By orienting these spins along the local field direction, we were able to significantly decrease the energy. Further, term "domain dynamics" will be used to denote just this kind of combined behavior.

Optimization efficiency of four types of algorithms (domain algorithms with $k = 2, 5, 10$ and spin behavior with $k = 1$) was determined. The interconnection matrix and initial states of the network were the same for all algorithms.

A few various parameters were observed to evaluate the optimization efficiency of the domain dynamics. Firstly, we recorded the energy of the minima that the system reached as a result of domain (E_d) and spin (E_s) dynamics. Secondly, we computed the probability of a) the domain dynamics bringing the system to a deeper minimum (W^+), b) the domain and spin dynamics leading the system to the same minimum (W^0), c) the domain dynamics bringing the system in a shallower minimum (W^-). Thirdly, we calculated the profit:

$$\delta E = 100\% \times (E_d - E_s) / E_s \tag{8}$$

which allows us to evaluate the depth of minima that the domain network finds.

5×10^4 random starts were made for each interconnection matrix. Over 500 of 100×100 matrices were investigated.

4.2 Experimental Results

The results of the experiments can be divided into three groups according to the type of matrix: matrices with a sparse (line) spectrum of minima, matrices with a combined (line-continuous) spectrum and those with a continuous spectrum.

Matrices with a sparse spectrum. The typical spectrum and distribution of minima for such matrices is given in Figure 2a. Matrices with a small number of minima give this kind of spectrum, a wide minima-free area lying between deep and shallow minima. This sort of matrices were modeled with the help of Hebbian rule (8) with the small load parameter $M / N = 0.05$.

It was said earlier that finding minima with these matrices involves simultaneous use of conventional spin dynamics and three types of domain dynamics (with $k = 2, 5, 10$). All of the algorithms proved good in finding deepest minima.

With sparse matrices, all of the dynamics bring the system to the deepest minimum with almost the same probability. For example, the spin dynamics put the system to the deepest minimum in 279 cases of 1000, and the domain dynamics ($k = 2$) in 280. The deepest minimum is not called global because there is chance that it was not been found despite of a large number of runs (5×10^4).

However, the efficiency can be evaluated differently. Let us consider the experimental results presented in Table 1. The first column carries W^+ and δE for the case when the domain dynamics brought the network to a deeper minimum of energy than the spin dynamics ($E_d < E_s$, $\delta E > 0$). The second column gives W^0 and δE for the case when both algorithms put the system in the same minimum

Table 1. Line-spectrum matrix ($M / N = 0.05$)

	$E_d < E_s$	$E_d = E_s$	$E_d > E_s$
	W^+ δE	W^0 δE	W^- δE
k =2	**0.27** 30.5%	**0.58** 0	**0.15** -19.9%
k =5	**0.30** 28.2%	**0.48** 0	**0.22** -24.8%
k =10	**0.30** 27.4%	**0.46** 0	**0.24** -24.5%

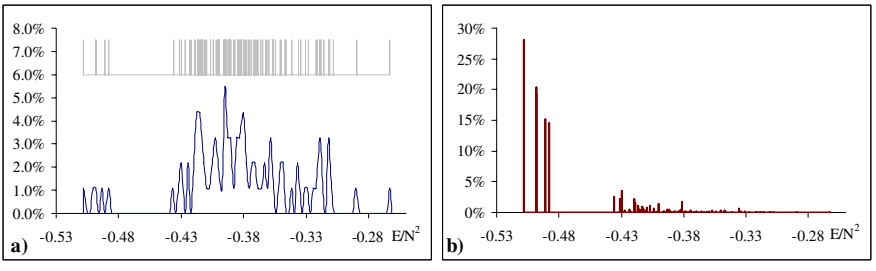


Fig. 2. Sparse-spectrum matrix ($M / N = 0.05$). a) The solid line denotes the spectral density of minima (the number of minima at length ΔE). The upper part of the figure shows the spectrum of minima distribution – each vertical line corresponds to a particular minimum. The Y-axis presents spectral density in % and the X-axis is the normalized values of energy minima E / N^2 . b) Probability of domain-algorithm system finding a minimum with energy E (k=2). The Y-axis is the probability of finding a particular minimum (%) and the X-axis is the normalized values of energy minima.

($E_d = E_s, \delta E = 0$). The third column bears W^- and δE for the case when the spin algorithm gave a deeper minimum ($E_d > E_s, \delta E < 0$). The rows correspond to different variants of the domain algorithm (with different k).

For example, the domain algorithm with $k = 2$ brought the network to a deeper minimum with probability $W^+ = 0.27$, the minimum being deeper by $\delta E = 30.5\%$ than in using the spin algorithm. At the same time, the spin dynamics gave better results in 15 cases of 100 with a 19.9% difference in depth ($\delta E = -19.9\%$).

Matrices with a combined spectrum. This kind of matrices were modeled with the help of Hebbian generalized rule (8), the load parameter being $M / N = 0.2$. A typical spectrum and distribution density of minima are given in Figure 3a.

Though there are areas of deep and shallow minima in this case too, it is not so pronounced. One can see a denser collection of local minima in the middle of the spectrum (Figure 3a).

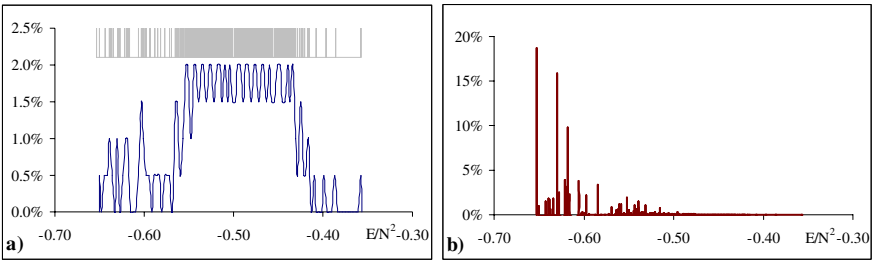


Fig. 3. The combined-spectrum matrix ($M / N = 0.2$). The legend is the same as in Figure 2.

All the algorithms coped with the problem of finding the deepest minimum. For example, the spin algorithm brought the system to a state with the sought-for energy in 49 cases of 100. As expected, the domain algorithm gave better results (Table 2), and the probability of the network reaching the deepest energy minimum was 18.5%, 18.9%, and 18.6% for $k = 2$, 5 and 10 correspondingly.

We see that the values of probability and profit are always higher in the left column than in the right one. Clear that the domain network comes to the lowest-energy state more often for any *k*.

Continuous-spectrum matrices. Hebbian generalized rule (8) with load parameter $M / N = 1$ was used to model this kind of matrix. The typical spectrum and distribution density of minima are given in Figure 4a.

This sort of matrix holds a lot of minima, which increases the probability of optimization algorithms bringing the system to a shallow minimum. Our experiments showed that even in this case the domain algorithm puts the net to lower energy more often than conventional spin algorithm. Yet the difference in results is not so noticeable (see Table 3). The same is true for profit δE , the average value of which is no greater than 8% in this case.

Three matrices of each type were chosen to present the results of the experiments with the algorithms in question. In fact, more than 100 matrices of each type were considered, letting alone matrices of mixed types (those were modeled using (8) and load parameters $M / N = 0.1$ and $M / N = 0.5$), 500 in all.

Table 2. Line-continuous spectrum matrix ($M / N = 0.2$)

	$E_d < E_s$	$E_d = E_s$	$E_d > E_s$
	W^+ δE	W^0 δE	W^- δE
k =2	**0.28** 22.8%	**0.53** 0	**0.19** -17.2%
k =5	**0.35** 22.8%	**0.42** 0	**0.23** -17.6%
k =10	**0.33** 32.4%	**0.42** 0	**0.25** -22.7%

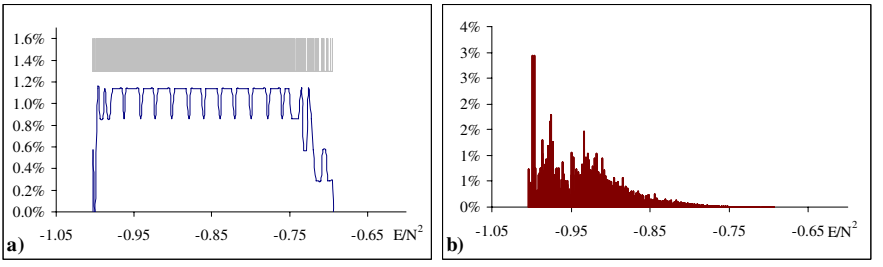


Fig. 4. The combined-spectrum matrix ($M / N = 1$). The legend is the same as in Figure 2.

Table 3. Continuous-spectrum matrix (M/N=1)

k	$E_d < E_s$	$E_d = E_s$	$E_d > E_s$
	W^+ δE	W^0 δE	W^- δE
k =2	**0.50** 6.4%	**0.04** 0	**0.46** -5.7%
k =5	**0.51** 5.4%	**0.06** 0	**0.43** -4.8%
k =10	**0.47** 7.8%	**0.07** 0	**0.46** -6.8%

It should be noted that though W^+ and δE for each of those 500 matrices differed from the values given above, the general behavior remained nearly the same. W^+ and δE for better performance of the domain network always exceeded corresponding values of W^- and δE for better performance of the Hopfield model. Sometimes, the domain network outperformed the conventional Hopfield model by $\delta E = 48\%$, and the probability of finding the deepest minimum exceeded 0.7.

It is seen (Figures 2b, 3b, 4b) that the system reaching one of deep minima or the deepest was a frequent result for all types of matrix under investigation. This agrees well with theoretical evaluations and experimental results of paper [11] stating that the probability of finding a minimum increase with the depth of the minimum.

5 Conclusion

The results of application of the domain network to solving optimization problems allow us to draw the following conclusions:

a) The domain network usually reaches deeper energy minima than the Hopfield network. The rate of the system falling in a deeper minimum depends on the type of the interconnection matrix and domain dimension; however, it is always higher than in the Hopfield network.

b) Energy profit in the cases when the domain net reaches deeper minima always exceeds the loss in the case of the domain system coming to shallower minima than the Hopfield net. The energy depth profit of the domain net is particularly noticeable when the energy landscape has a wide difference between shallow and deep minima.

c) The domain algorithm outperforms the Hopfield model not only when there are pronounced minima, but also when $M / N \geq 1$, i. e., when the difference between the depths of minima is very small. The experiment showed that the domain algorithm could detect this difference and find deepest minima with 5% profit.

d) The efficiency of the domain network is k^2 times higher than that of the Hopfield net. The advantage becomes particularly noticeable with increased dimensionality.

All these observations are true for all of the domain nets being investigated ($k = 2, 5, 10$). The same results were obtained in experiments with matrices whose elements were random numbers obeying the Gaussian distribution with the zero mean.

We see that the domain algorithm allows faster solution of optimization problems, requires lower computational power and yields better results than the conventional algorithm of random search.

Acknowledgements. The work was supported by Russian Basic Research Foundation (the project 04-07-90038) and the program "Intellectual computer systems" (the project 2.45).

References

1. Hopfield, J.: Neural Networks and physical systems with emergent collective computational abilities. Proc.Nat.Acad.Sci.USA. v.79 (1982) 2554-2558.
2. Hebb, D.: The Organization of Behavior. New York: Wiley, 1949.
3. Palm, G., Sommer, F.: Information capacity in recurrent McCulloch-Pitts networks with sparsely coded memory states. Network v.3 (1992) 177-186
4. Abu-Mostafa Y. S., St. Jacques J. Information capacity of the Hopfield model. IEEE Transactions on Information Theory vol.31 (4), (1985) 461-64.
5. Cohen M. A., Grossberg S. G. Absolute stability of global pattern formation and parallel memory storage by compatitive neural networks. IEEE Transactions on Systems, Man and Cybernetics v. 13 (1983) 815-26.
6. Grossberg S. The adapptive brain, vol. 1 and 2. Amsterdam: North-Holland. 1987.
7. Amit D.J., Gutfreund H. & Sompolinsky H. Statistical mechanics of neural networks near saturation. Annal of Physics, v.173, (1987) 30-67.
8. Kinzel W. Learning and pattern recognition in spin glass models. 2 Physik B, v. 60 (1985) 205-213.
9. Kryzhanovsky B.V., Mikaelian A.L. An associative memory capable of recognizing strongly correlated patterns. Doklady Mathematics, v.67, No.3 (2003) 455-459.
10. Kryzhanovsky B., Magomedov B., Mikaelian A.: Domain Model of a Neural Network. Doklady Mathematics, v. 71, No. 2 (2005) 310-314.
11. Kryzhanovsky B.V., Magomedov B.M., and Mikaelyan A.L.: A Relation between the Depth of a Local Minimum and the Probability of Its Detection in the Generalized Hopfield Model. Doklady Mathematics, v. 72, No. 3 (2005) pp. 986-991.

Nonlinear Function Learning by the Normalized Radial Basis Function Networks

Adam Krzyżak[1] and Dominik Schäfer[2]

[1] Department of Computer Science and Software Engineering
Concordia University
Montréal, Canada H3G 1M8
krzyzak@cs.concordia.ca
and
Institute of Control Engineering
Technical University of Szczecin
70-313 Szczecin, Poland
[2] Dominik Schäfer
Fachbereich Mathematik
Universität Stuttgart
D-70569 Stuttgart, Germany
schaefdk@mathematik.uni-stuttgart.de**

Abstract. We study strong universal consistency and the rates of convergence of nonlinear regression function learning algorithms using normalized radial basis function networks. The parameters of the network including centers, covariance matrices and synaptic weights are trained by the empirical risk minimization. We show the rates of convergence for the networks whose parameters are learned by the complexity regularization.

1 Introduction

Multilayer perceptrons (MLP) and radial basis function (RBF) are the most popular neural networks used in practice. All these networks turned out to be very useful in many applications such as interpolation, classification, data smoothing and regression estimation, prediction, data mining, etc. Convergence analysis and various properties of MLP and RBF networks can be found among others in Anthony and Bartlett [1], Cybenko [3], Devroye et al. [4], Györfi et al. [8], Ripley [22], and Hornik et al. [9] (MLP) and in Györfi et al. [8], Krzyżak et al. [11], Krzyżak and Linder [12], Moody and Darken [17], and Park and Sandberg [18] (RBF).

** Part of this research was carried out when the first author was with the Technical University of Szczecin on sabbatical leave from Concordia University. The second author is presently at Würtembergische Versicherung AG, Stuttgart, Germany. The research of the first and the second author was supported by NSERC, the Alexander von Humboldt Foundation, and by the College of Graduates "Parallel and Distributed Systems", University of Stuttgart, respectively.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 46–55, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In the present paper we study normalized radial basis function (NRBF) networks in the context of nonlinear function learning (estimation of regression). Specifically, we consider the class of NRBF networks $\mathcal{F} = \mathcal{F}(k, \ell, L, R, B)$ with one hidden layer of k nodes ($k \in \mathbb{N}, L \geq \ell \geq 0, R, B > 0$), i.e. the class of functions of the form

$$\mathcal{F} = \left\{ f(x) = \frac{\sum_{i=1}^{k} w_i K\left((x-c_i)^T A_i (x-c_i)\right)}{\sum_{i=1}^{k} K\left((x-c_i)^T A_i (x-c_i)\right)} =: \frac{\sum_{i=1}^{k} w_i K_{c_i,A_i}(x)}{\sum_{i=1}^{k} K_{c_i,A_i}(x)} \right\}, \quad (1)$$

where $K : \mathbb{R}_0^+ \to \mathbb{R}^+$ is a left-continuous, decreasing *kernel*, $c_1, ..., c_k \in \mathbb{R}^d$ are vector *centers* with $\|c_i\| \leq R$ for all $i = 1, ..., k$ ($\|\cdot\|$ is the Euclidean norm), $A_1, ..., A_k$ are symmetric, positive definite, real $d \times d$-matrices each of which satisfies the eigenvalue conditions $\ell \leq \lambda_{min}(A_i) \leq \lambda_{max}(A_i) \leq L$, where $\lambda_{min}(A_i)$ and $\lambda_{max}(A_i)$ are the minimal and the maximal eigenvalue of A_i, respectively, and $w_1, ..., w_k \in \mathbb{R}$ are the *output weights* satisfying boundedness condition $|w_i| \leq B$ for all $i = 1, ..., k$.. Matrix A_i determines the shape of the *receptive field* about the center c_i. When $A_i = \frac{1}{\sigma_i^2} I$, the shape is a hyperspherical ball with radius σ_i. When $A_i = diag[\sigma_{i,1}^{-2}, \ldots, \sigma_{i,d}^{-2}]$, then the shape of the receptive field is an elliptical ball with each axis coinciding with a coordinate axis; the lengths of the axes of the ellipsoid are determined by $\sigma_{i,1}, \ldots, \sigma_{i,d}$. When A_i is non-diagonal but symmetric, we have $A_i = R_i^T D_i R_i$ where D_i is a diagonal matrix which determines the shape and size of the receptive field and R_i is a rotation matrix which determines the orientation of the receptive field. In this paper the kernel is fixed, whereas network parameters $w_i, c_i, A_i, i = 1, \ldots, k$ are learned from the data. Although $\mathcal{F}(k, \ell, L, R, B)$ also depends on K we will suppress it in the notation for the sake of simplicity. Throughout the paper we use the convention $0/0 = 0$.

NRBF networks were introduced by Moody and Darken [17] and Specht [21] as modifications of standard RBF networks defined by

$$f(x) = \sum_{i=1}^{k} w_i K((x-c_i)^T A_i (x-c_i)) + w_0. \quad (2)$$

Kernel K plays an important role in NRBF network. The common choices of kernels include the following [8] (see Figure 1):

- $K(x) = I_{\{x \in [0,1]\}}$ (window)
- $K(x) = \max\{(1-x^2), 0\}$ (truncated parabolic)
- $K(x) = e^{-x^2}$ (Gaussian)
- $K(x) = e^{-x}$ (exponential).

Other kernels include:

- $K(x) = \frac{1}{\sqrt{x^2+1}}$ (inverse multiquadratic)

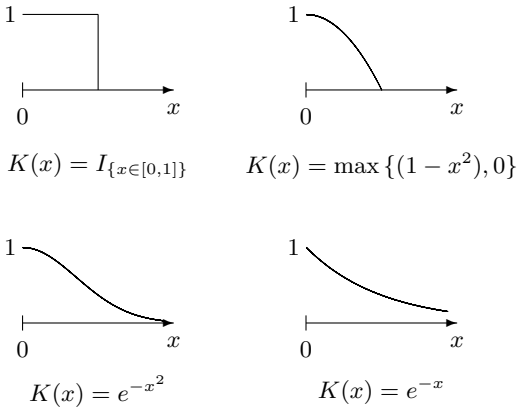


Fig. 1. Window, truncated parabolic, Gaussian, and exponential kernels

- $K(x) = \max\{(1-x), 0\}$ (triangular)
- $K(x) = \max\{\cos\frac{\pi}{2}x, 0\}$ (trigonometric).

Radial basis networks naturally appear in a variety of applications involving smoothing splines (cf. Duchon [5]), interpolation using multiquadrics, shifted surface splines or thin-plate splines (see Girosi et al. [7]), and regression analysis (see, e.g., Györfi et al. [8] and Krzyżak et al. [11]). In [11] the authors consider RBF regression estimates and classifiers based on minimizing the empirical risk and prove universal consistency results. The rate of convergence of RBF-based regression estimates using complexity regularization is studied in Krzyżak and Linder [12]. Practical and theoretical advantages and disadvantages of RBF and NRBF networks are discussed by Shorten and Murray-Smith [20]. They point out that radial coefficients $K_{c_i,A_i}(x)/\sum_{i=1}^{k} K_{c_i,A_i}(x)$ of weights $w_i, i = 1,\ldots,k$ in (1) in NRBF networks sum to one at every point of the input space, thus their response is highly desirable both from computational and theoretical point of view.

In the following we study the application of NRBF networks nonlinear in function learning. Suppose that random variables X and Y take values in $\mathbb{R}^d$ and $\mathbb{R}$, respectively. The task is to find a measurable function $f : \mathbb{R}^d \to \mathbb{R}$ such that $f(X)$ is a good approximation of Y in the mean squared error sense. In particular, if $\mathbf{E}|Y|^2 < \infty$, our goal is to find a measurable function r minimizing the L_2-risk, that is,

$$J^* = \inf_{f:\mathbb{R}^d\to\mathbb{R}} \mathbf{E}|f(X) - Y|^2 = \mathbf{E}|r(X) - Y|^2.$$

The solution of this minimization problem is given by the regression function $r(x) = \mathbf{E}[Y|X = x]$. The regression function, however, can only be computed if the distribution of (X, Y) is known. Otherwise we have to rely on estimates of r.

In order to estimate r without making any assumption about the distribution of (X, Y), i. e., in nonparametric case, we assume that a training set $D_n := \{(X_1, Y_1),...,(X_n, Y_n)\}$ of independent, identically distributed samples of (X, Y) is given, where D_n is independent of (X, Y). The problem of determining the specific values of parameters from the training sequence $\mathcal{D}_n$ is called *learning* or *training*. The most common parameter learning strategies are:

- cluster input vectors $X_i (i = 1, \ldots, n)$ and set center vectors c_i $(i = 1, \ldots, k)$ to cluster centers. Remaining parameters are determined by minimizing the empirical L_2 risk on D_n. If the elements of the covariance matrices A_i $(i = 1, \ldots, k)$ are chosen arbitrarily then finding the output weights w_i $(i = 1, \ldots, k)$ by the least squares method is an easy linear problem
- choose from D_n a random k-element subset

$$D'_n = \{(X'_1, Y'_1), \ldots, (X'_k, Y'_k)\}$$

of samples and assign $X'_i \to c_i, Y'_i \to w_i$ $(i = 1, \ldots, k)$. The elements of the covariance matrices $A_i, i = 1, \ldots, k$ are chosen arbitrarily. This method reduces NRBF network to the kernel regression estimate

$$f(x) = \frac{\sum_{i=1}^{k} Y'_i K\left((x - X'_i)^T A_i (x - X'_i)\right)}{\sum_{i=1}^{k} K\left((x - X'_i)^T A_i (x - X'_i)\right)}$$

(see [8] for the detailed analysis of the kernel regression estimates)
- choose all the parameters of the network by minimizing the empirical L_2 risk.

The second strategy has been adopted by Xu et al. [25] in their study of convergence of NRBF networks. Computationally, the last strategy is the most costly, however, it is the most general approach and thus it is the method of choice in the present paper.

The method of the empirical risk minimization obtains an estimate $\hat{f}_n$ of r by selecting the parameter vector which minimizes the residual sum of squares over a suitable class $\mathcal{F}_n$ of functions. In other words, we use the training sequence to choose an estimator $\hat{f}_n \in \mathcal{F}_n$ minimizing the *empirical* L_2-risk (mean residual sum of squares)

$$J_n(f) = \frac{1}{n} \sum_{j=1}^{n} |f(X_j) - Y_j|^2,$$

or equivalently,

$$J_n(\hat{f}_n) \leq J_n(f) \quad \text{for all } f \in \mathcal{F}_n.$$

The performance of the regression estimator $\hat{f}_n$ is measured by

$$J(\hat{f}_n) = \mathbf{E}[|\hat{f}_n(X) - Y|^2 | D_n].$$

In this framework an estimator $\hat{f}_n$ is called *strongly (weakly) consistent* if it asymptotically attains the minimal L_2-risk J^* almost surely (in probability), i.e., if

$$J(\hat{f}_n) - J^* \to 0 \quad \text{with prob. 1} \quad \text{(in probability)} \quad \text{as } n \to \infty. \tag{3}$$

Observe that $J(\hat{f}_n) - J^* \to 0$ if and only if

$$\mathbf{E}[|\hat{f}_n(X) - Y|^2 | D_n] - \mathbf{E}|r(X) - Y|^2 = \mathbf{E}[|\hat{f}_n(X) - r(X)|^2 | D_n] \to 0,$$

thus the consistency of $\hat{f}_n$ is tantamount to L_2-consistency of the regression function estimate $\hat{f}_n$.

The empirical risk minimization has been a very popular approach to function learning in the machine learning literature. When the minimization is carried out over a rich (and complex) families $\mathcal{F}_n$ of candidate functions, the resulting estimate usually overfits the data and does not generalize well, i.e., it is not likely to perform well for a new data that is independent of the training set. Different measures of complexity of $\mathcal{F}_n$ have been used for different purposes, but they are all related to the cardinality of a finite subset representing the whole family in a certain sense. Examples are metric entropy (Kolmogorov and Tihomirov [10]), VC-dimension (Vapnik and Chervonenkis [24]) and random covering numbers (Pollard [19]). Based on these measures, asymptotic properties of the method of empirical risk minimization were studied among others by Vapnik [24] and Györfi et al. [8]. The class $\mathcal{F}_n$ of candidate functions should clearly allow the statistician to find good approximations for a multitude of target functions. Therefore, one generally needs to increase the size of the candidate family as the size of the training set increases. However, a good trade-off should also be maintained between the complexity of the candidate family and the training data size to avoid overfitting. The idea of using nested candidate classes which grow in a controlled manner with the size of the training data is Grenander's method of sieves [4]. This approach was successfully applied to pattern recognition by Devroye et al. [4], to regression estimation by Györfi et al. [8] and Lugosi and Zeger [15], and by Faragó and Lugosi [6] in the neural network framework.

In this paper we apply empirical risk minimization to obtain consistent NRBF networks-based regression estimates. We also apply complexity regularization (see section 3) to derive the rate of convergence of the estimates. First consistency results for regression estimates based on NRBF networks were obtained by Xu et al. [25] under a restrictive assumption that the centers are placed at the data points, the covariance matrices are chosen according to some specified rules and only the output weights are learned by minimizing the residual sum of squares. Much more flexibility and generality is obtained by learning all these parameters from the data. This approach is used in [14] to establish consistency of NRBFs. The present paper discusses both universal consistency and the ates of convergence the nonlinear regression learning algorithms derived from NRBF networks when network parameters (centers, receptive fields and output weights) are learned by the empirical risk minimization and by complexity regularization (number of hidden neurons).

2 Consistency of the Algorithm

We will limit the choice of kernels to:

- *Window kernels.* These are kernels for which there exists some $\delta > 0$ such that $K(v) \notin (0, \delta)$ for all $v \in \mathbb{R}_0^+$. The classical naive kernel $K(v) = \mathbf{1}_{[0,1]}(v)$ is a member of this class.
- *Kernels with unbounded support*, i.e. $K(v) > 0$ for all $v \in \mathbb{R}_0^+$. The most famous example from this class is $K(v) = \exp(-v)$. Then $K(x^T x)$ is the classical Gaussian kernel.

Let the parameters $k_n \in \mathbb{N}, L_n, R_n$ and $B_n > 0$ tend to ∞ as $n \to \infty$. Note that the $\mathcal{F}(k_n, 0, L_n, R_n, B_n)$ are not nested as n increases. We therefore consider the nested classes

$$\mathcal{F}_n := \bigcup_{k=1}^{k_n} \mathcal{F}(k, 0, L_n, R_n, B_n).$$

The condition on the minimal admissible eigenvalue is not used here.

Suppose i.i.d. observations $D_n := \{(X_1, Y_1),...,(X_n, Y_n)\}$ are available. Let the estimate $\hat{f}_n \in \mathcal{F}_n$ of the regression function $r(\cdot) = \mathbf{E}[Y|X = \cdot]$ be chosen by the empirical risk minimization, i.e.,

$$\hat{f}_n(\cdot) = \arg\min_{f \in \mathcal{F}_n} \frac{1}{n} \sum_{i=1}^{n} |f(X_i) - Y_i|^2. \tag{4}$$

This estimation strategy yields consistent estimators when a window type kernel or a kernel with unbounded support is used [14]:

Theorem 1. *Suppose* $k_n, L_n, R_n, B_n \to \infty \quad (n \to \infty)$ *in such a way that*

(a) for window type kernel K*:*

$$\frac{B_n^4 k_n}{n} \log B_n^2 k_n \to 0.$$

(b) for kernel K *with unbounded support:*

$$\frac{B_n^4 k_n}{n} \log \frac{B_n^2}{K(4R_n^2 L_n)} \to 0.$$

Then the estimate $\hat{f}_n$ *defined by (4) is strongly consistent for any distribution of* (X, Y) *with* $\mathbf{E}Y^2 < \infty$ *and* $\|X\| \le Q < \infty$ *almost surely, if*

$$B_n^4 \le \frac{n}{(1+\beta)\log n}$$

for some $\beta > 0$ *and all sufficiently large* n*.*

In case (a) one may choose $L_n = R_n = \infty$ for all $n \geq 1$, which means that learning is unrestricted in choosing the best parameters $c_i, A_i, i = 1, \ldots, k$ and, the condition $\|X\| \leq Q < \infty$ almost surely is not necessary. For the proof of Theorem 1 and additional details we refer the reader to [14].

Examples. For the naive kernel NRBF with $K(v) = \mathbf{1}_{[0,1]}(v)$,

$$B_n^4 k_n = O(n^q) \quad \text{for } q \in (0,1)$$

suffices for weak consistency. For Gaussian kernel NRBFs, $K(v) = \exp(-v)$,

$$B_n^4 k_n R_n^2 L_n = O(n^q) \quad \text{for } q \in (0,1)$$

is sufficient for convergence.

3 Rate of Convergence of the Algorithm

We apply the principle of complexity regularization to establish the rate of convergence of NRBF networks regression function estimation algorithm. This approach enables us to automatically adapt to the smoothness of the regression function and to adapt the structure of the network (the number of the hidden neurons) to the data. The complexity regularization principle was introduced by Vapnik and Chervonenkis [23] and Vapnik [24] in pattern recognition as structural risk minimization. It was applied to regression estimation by Barron, Birgé, and Massart [2] and to investigation of the rate of convergence of RBF networks regression estimate by Krzyżak and Linder [12]. Lugosi and Nobel [16] studied complexity regularization with a data-dependent penalty.

We start with defining a measure of complexity of a class of functions. Let $\mathcal{F}$ be a class of real-valued functions on $\mathbb{R}^d$. Let $x_1^n = (x_1, ..., x_n) \in \mathbb{R}^{dn}$ and $\epsilon > 0$. We say a class $\mathcal{G}$ of real-valued functions on $\mathbb{R}^d$ is an ϵ-cover of $\mathcal{F}$ in x_1^n, if for each $f \in \mathcal{F}$ there exists a $g \in \mathcal{G}$ such that

$$\frac{1}{n} \sum_{j=1}^{n} |f(x_j) - g(x_j)| \leq \epsilon.$$

The *covering number* $\mathbb{N}(\epsilon, \mathcal{F}, x_1^n)$ is the smallest integer m such that an ϵ-cover $\mathcal{G}$ of $\mathcal{F}$ in x_1^n exists with cardinality $|\mathcal{G}| = m$. Covering numbers are useful in studying convergence of algorithms learned from random data (see [19], [4], [8] for their properties and applications). We can now define our regularized NRBF regression estimate. Consider the following class of NRBF networks:

$$\mathcal{F}_k = \left\{ \frac{\sum_{i=1}^{k} w_i K\left((x - c_i)^T A_i (x - c_i)\right)}{\sum_{i=1}^{k} K\left((x - c_i)^T A_i (x - c_i)\right)} : \sum_{i=1}^{k} |w_i| \leq \beta_n \right\}. \tag{5}$$

The learning algorithm will be defined in two steps.

1. Let
$$m_{n,k} = \arg\min_{f \in \mathcal{F}_k} \frac{1}{n} \sum_{i=1}^{n} |f(X_i) - Y_i|^2.$$
Hence $m_{n,k}$ minimizes the empirical L_2 risk for n training samples over $\mathcal{F}_k$. (We assume the existence of such a minimizing function for each k and n.)
2. Define the complexity penalty of the kth class for n training samples as any nonnegative number $pen_n(k)$ satisfying
$$pen_n(k) \geq 2568 \frac{\beta_n^4}{n} \cdot (\log \mathbb{N}(1/n, \mathcal{F}_{n,k}) + t_k), \tag{6}$$
where $\mathbb{N}(\epsilon, \mathcal{F}_{n,k})$ is almost sure uniform upper bound on the random covering numbers $\mathbb{N}(\epsilon, \mathcal{F}_{n,k}, X_1^n)$ and the nonnegative constants $t_k \in \mathbb{R}_+$ satisfy Kraft's inequality $\sum_{k=1}^{\infty} e^{-t_k} \leq 1$. The coefficients t_k may be chosen as $t_k = 2 \log k + t_0$ with $t_0 \geq \log \left(\sum_{k \geq 1} k^{-2} \right)$.

   The penalized empirical L_2 risk is defined for each $f \in \mathcal{F}_k$ as
$$\frac{1}{n} \sum_{i=1}^{n} |f(X_i) - Y_i|^2 + pen_n(k).$$
Our estimate m_n is then defined as the $m_{n,k}$ minimizing the penalized empirical risk over all classes
$$m_n = m_{n,k^*}, \tag{7}$$
where
$$k^* = \arg\min_{k \geq 1} \left(\frac{1}{n} \sum_{i=1}^{n} |m_{n,k}(X_i) - Y_i|^2 + pen_n(k) \right).$$

The following covering number bounds for $\mathcal{F} = \mathcal{F}(k, \ell, L, R, B)$ needed in establishing the rates are quoted from [14].

(a) Let K be a *window type kernel* that does not attain values in $(0, \delta)$. Then
$$\mathbb{N}(\epsilon, \mathcal{F}, x_1^n) \leq 2^k \cdot \left(\frac{4B}{\epsilon} + 1 \right)^k \cdot \left(\frac{16eBk}{\epsilon\delta} \right)^{2k(d^2+d+2)}.$$

(b) Let K be a *kernel with unbounded support*. If $\|x_i\| \leq Q$ $(i = 1, ..., n)$, we have
$$\mathbb{N}(\epsilon, \mathcal{F}, x_1^n) \leq 2^k \cdot \left(\frac{4B}{\epsilon} + 1 \right)^k \cdot \left(\frac{16eB}{\epsilon \cdot K((R+Q)^2 L)} \right)^{2k(d^2+d+2)}.$$

The next theorem presents the rate of convergence in the class of convex closure $\overline{\mathcal{F}}$ of $\mathcal{F} = \bigcup_k \mathcal{F}_k$ in $L_2(\mu)$.

Theorem 2. *Let $1 \leq L < \infty, n \in \mathbb{N}$, and let $L \leq \beta_n < \infty$. Suppose, furthermore, that $|Y| \leq L < \infty$ a.s. Let K be a window type kernel or a kernel with unbounded support. Assume that the penalty satisfies (6) for some t_k such that $\sum_{k=1}^{\infty} e^{-t_k} \leq 1$. If $r \in \overline{\mathcal{F}}$ then the NRBF regression estimate with parameters learned by the complexity regularization (7) satisfies for n sufficiently large*

$$\mathbf{E} \int |m_n(x) - m(x)|^2 \mu(dx) = O\left(\beta_n^2 \left(\frac{\log(\beta_n n)}{n}\right)^{1/2}\right). \tag{8}$$

Note that if $\beta_n < const < \infty$ then

$$pen(k) = O\left(\frac{k \log n}{n}\right)$$

and the rate in (8) becomes

$$\mathbf{E} \int |m_n(x) - r(x)|^2 \mu(dx) = O\left(\sqrt{\frac{\log n}{n}}\right).$$

Characterization of $\overline{\mathcal{F}}$ is an interesting open problem. A similar problem for RBF regression estimates has been considered in [8, chapter 17.3].

4 Conclusion

We considered convergence and rates of the normalized radial basis function networks in nonlinear function estimation problem. We established convergence for a class of window and unbounded support radial functions using empirical risk minimization. The rates of convergence were obtained for the networks with all parameters (centers, receptive fields, output weights and the number of hidden neurons) learned automatically from the data by means of complexity regularization.

References

1. M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, Cambridge, 1999.
2. A. R. Barron, L. Birgé and P. Massart, Risk bounds for model selection via penalization. *Probability Theory and Related Fields*, vol. 113, pp. 301–413, 1999.
3. G. Cybenko, Approximations by superpositions of sigmoidal functions, *Mathematics of Control, Signals, and Systems*, vol. 2, pp. 303–314, 1989.
4. L. Devroye, L. Györfi and G. Lugosi, *Probabilistic Theory of Pattern Recognition*, Springer-Verlag, New York, 1996.
5. J. Duchon, Sur l'erreur d'interpolation des fonctions de plusieurs variables par les D^m-splines, *RAIRO Anal. Numér.*, vol. 12, no. 4, pp. 325–334, 1978.
6. A. Faragó and G. Lugosi, Strong universal consistency of neural network classifiers, *IEEE Trans. on Information Theory*, vol. 39, pp. 1146–1151, July 1993.

7. F. Girosi, M. Jones, and T. Poggio, Regularization theory and neural network architectures, *Neural Computation*, vol. 7, pp. 219–267, 1995.
8. L. Györfi, M. Kohler, A. Krzyżak and H. Walk, *A Distribution-Free Theory of Nonparametric Regression*, Springer-Verlag, New York, 2002.
9. K. Hornik, S. Stinchocombe and H. White, Multilayer feed-forward networks are universal approximators, *Neural Networks*, vol. 2, pp. 359–366, 1989.
10. A. N. Kolmogorov and V. M. Tihomirov, ϵ-entropy and ϵ-capacity of sets in function spaces. *Translations of the American Mathematical Society*, vol. 17, pp. 277–364, 1961.
11. A. Krzyżak, T. Linder and G. Lugosi, Nonparametric estimation and classification using radial basis function nets and empirical risk minimization, *IEEE Trans. Neural Networks*, vol. 7, no. 2, pp. 475–487, March 1996.
12. A. Krzyżak and T. Linder, Radial basis function networks and complexity regularization in function learning, *IEEE Trans. Neural Networks* , vol. 9, no. 2, pp. 247–256, March 1998.
13. A. Krzyżak and H. Niemann, Convergence and rates of convergence of radial basis functions networks in function learning, *Nonlinear Analysis*, vol. 47, pp. 281–292, 2001.
14. A. Krzyżak and D. Schäfer, Nonparametric regression estimation by normalized radial basis function networks, *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 1003-1010, March 2005.
15. G. Lugosi and K. Zeger, Nonparametric estimation via empirical risk minimization, *IEEE Trans. on Information Theory*, vol. 41, pp. 677–687, May 1995.
16. G. Lugosi and A. Nobel, Adaptive model selection using empirical complexities. *Annals of Statistics*, vol. 27, pp. 1830–1864, 1999.
17. J. Moody and J. Darken, Fast learning in networks of locally-tuned processing units, *Neural Computation*, vol. 1, pp. 281–294, 1989.
18. J. Park and I. W. Sandberg, Universal approximation using Radial-Basis-Function networks, *Neural Computation*, vol. 3, pp. 246–257, 1991.
19. D. Pollard *Convergence of Stochastic Processes*, Springer-Verlag, New York, 1984.
20. R. Shorten and R. Murray-Smith, Side effects of normalising radial basis function networks, *International Journal of Neural Systems*, vol. 7, pp. 167-179, 1996.
21. D. F. Specht, Probabilistic neural networks, *Neural Networks*, vol. 3, pp. 109-118, 1990.
22. B. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, 1996.
23. V. N. Vapnik and A. Ya. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities, *Theory of Probability and its Applications*, vol. 16, pp. 264–280, 1971.
24. V. N. Vapnik, *Estimation of Dependences Based on Empirical Data*, 2nd ed., Springer-Verlag, New York, 1999.
25. L. Xu, A. Krzyżak and A. L. Yuille, On radial basis function nets and kernel regression: approximation ability, convergence rate and receptive field size, *Neural Networks*, vol. 7, pp. 609–628, 1994.

Sum and Product Kernel Regularization Networks

Petra Kudová* and Terezie Šámalová**

Institute of Computer Science, Academy of Sciences of CR,
Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
{petra, terka}@cs.cas.cz

Abstract. We study the problem of learning from examples (i.e., supervised learning) by means of function approximation theory. Approximation problems formulated as regularized minimization problems with kernel-based stabilizers exhibit easy derivation of solution, in the shape of a linear combination of kernel functions (one-hidden layer feed-forward neural network schemas). Based on Aronszajn's formulation of sum of kernels and product of kernels, we derive new approximation schemas – Sum Kernel Regularization Network and Product Kernel Regularization Network. We present some concrete applications of the derived schemas, demonstrate their performance on experiments and compare them to classical solutions. For many tasks our schemas outperform the classical solutions.

1 Introduction

The problem of *learning from examples* (also called *supervised learning*) is a subject of great interest at present. The need for a good supervised learning technique stems from a wide range of application areas, covering various approximation, classification, and prediction tasks.

In this paper we study one learning technique – *Regularization Networks* (RN). RNs are feed-forward neural networks with one hidden layer. They benefit from a good theoretical background, their architecture has been proved to be the solution of the problem of learning from examples formulated as regularized minimization problem (see [1,2,3,4]).

In [5] the first author has shown that performance of the RN learning depends significantly on the choice of kernel function. Moreover the choice of kernel function always depends on particular task. Different kernel functions are suitable for different data types, but we often have to deal with heterogeneous data, in the sense that different attributes differ in type or quality, or that the character of data differs in different parts

* Supported by the Program "Information Society" under project 1ET100300414, by the European Commission's Research Infrastructures activity of the Stucturing the European Research Area programme, contract number RII3-CT-2003-506079 (HPC-Europa) and also by the Institutional Research Plan AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Appplications".

** Supported by the Program "Information Society" under project 1ET100300517 and also by the Institutional Research Plan AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Appplications".

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 56–65, 2006.
© Springer-Verlag Berlin Heidelberg 2006

of the input space. Therefore we propose network architectures using composite types of kernels that might better reflect the heterogeneous character of data.

In the following section we briefly describe the RN learning technique. In Sec. 3 we introduce two types of composite kernels: *Product Kernel* and *Sum Kernel*, and explain their mathematical justification. In Sec. 4 we demonstrate proposed network architectures on experiments.

2 Learning with Regularization Networks

Our problem can be formulated as follows. We are given a set of examples (pairs) $z = \{(\boldsymbol{x}_i, y_i) \in R^d \times R\}_{i=1}^N$ that was obtained by random sampling of some real function f, generally in the presence of noise (see Fig. 1). Our goal is to recover the function f from data, or to find the best estimate of it.

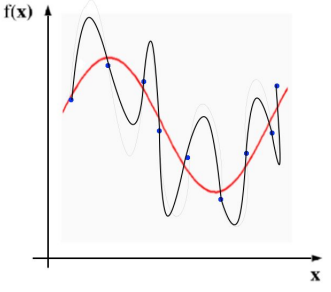


Fig. 1. Learning from examples

This problem is generally ill-posed. Thus we impose additional (regularization) conditions on the solution [3]. This is typically some a-priori knowledge, or smoothness constraints. The solution has to minimize a functional that is composed of the data part and the "smoothness" part:

$$\mathcal{F}(f) = \mathcal{E}_z(f) + \gamma \Phi(f), \tag{1}$$

where $\mathcal{E}_z$ is the error functional depending on the data z, Φ is the regularization part (also called *stabilizer*), and $\gamma \in \mathbb{R}^+$ is *the regularization parameter*, that gives trade-off between the two parts of minimization functional.

Reproducing Kernel Hilbert Spaces (RKHSs) (studied by Aronszajn [6]) represent a mathematical tool that can help us to deduce existence, uniqueness and even form of the solution of (1).

Let $K : \Omega \times \Omega \to \mathbb{R}$ (for $\Omega \subseteq \mathbb{R}^d$) be a symmetric, positive semi-definite function and $\mathcal{H}$ be the (unique) RKHS defined by the kernel K, with norm $\|.\|_K$. (That is, $\mathcal{H}$ is a Hilbert space of functions $\Omega \to \mathbb{R}$ generated by functions $\{K(x, x_0), x_0 \in \Omega\}$ and the norm is given by the scalar product $\langle K(x, x_0), K(x, x_1)\rangle_K = K(x_0, x_1)$.) We let the stabilizer be $\Phi(f) = \|f\|_K^2$, choose classical mean square error for the first part (any convex differentiable error functional would work here) and get

$$\mathcal{F}(f) = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 + \gamma\|f\|_K^2. \tag{2}$$

Existence and uniqueness of the solution of (2) was shown in [7]. Derivation of the shape of the solution, known as the Representer theorem, has been shown already in [3] but without taking advantage of RKHS. Papers [2], [4] deal with the problem as well, for the kernel case see [7]. All the proofs are based on the idea that a minimum of a function can exist in an interior point only if first derivative equals zero.

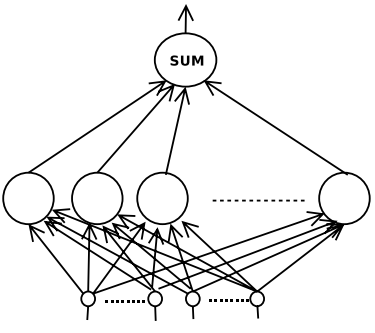


Fig. 2. Regularization Network schema

Employing the Representer theorem we obtain the solution in the form:

$$f(x) = \sum_{i=1}^{N} w_i K(x_i, x), \tag{3}$$

where x_i are the data points and $K(\cdot,\cdot)$ the corresponding kernel. The weights w_i are given by the linear system

$$(N\gamma I + K)w = y, \quad \text{where } K_{ij} = K(x_i, x_j) \tag{4}$$

Such function corresponds to a neural network with one hidden layer, called *Regularization Network* (see Fig. 2). Its learning algorithm (see Fig. 3) fixes the firsts arguments of the kernels (called centers) to the given data points, and computes the output weights w_i as a solution of linear system (4). The regularization parameter γ and the kernel function K are supposed to be known in advance.

3 Product Kernel and Sum Kernel Regularization Networks

The most commonly used kernels are based on Gaussian functions (the kernel being $K(x,y) = e^{-\|x-y\|^2/d^2}$), other symmetric positive definite function may be used as well. In this section we introduce methods to produce more sophisticated instances of kernels from simpler ones: *Sum Kernels* and *Product Kernels*; we also show a mathematical justification for such kernels (for proofs see [8]).

Input: Data set $\{x_i, y_i\}_{i=1}^N \subseteq \mathbb{R}^n \times \mathbb{R}$
Output: Regularization network.

1. Set the centers of kernels:

$$\forall i \in \{1, \ldots, N\} : c_i \leftarrow x_i$$

2. Compute the values of weights:

$$(N\gamma I + K)w = y,$$

$$K_{i,j} = K(c_i, x_j), \quad y = (y_1, \ldots, y_N), \quad \gamma > 0.$$

Fig. 3. RN learning algorithm

First we will consider a sum of RKHSs. A sum of reproducing kernels was proposed and basic properties proved in [6]. We use the theory to infer regularized neural networks.

For $i = 1, 2$ let F_i be an RKHS of functions on Ω, let K_i be the corresponding kernels and $\|.\|_i$ the corresponding norms. Consider the following space of all couples $\{f_1, f_2\}$ on Ω: $H = \{\{f_1, f_2\} \mid f_1 \in F_1, f_2 \in F_2\}$. The metric will be given by $\|\{f_1, f_2\}\|^2 = \|f_1\|_1^2 + \|f_2\|_2^2$.

We have to deal with duplicities. Consider the class F_0 of all functions f belonging to $F_1 \cap F_2$. We define $H_0 := \{\{f, -f\}; f \in F_0\}$. It is a closed subspace of H and thus we can write $H = H_0 \oplus H'$, where H' is the complementary subspace to H_0. Now to every element $\{f', f''\}$ of H there corresponds a function $f(x) = f'(x) + f''(x)$. This is a linear correspondence transforming H into a linear class F of functions on Ω. Elements of H_0 are precisely those transformed into the zero function and thus the correspondence between H' and F is one-to-one and has an inverse (for every $f \in F$ we obtain one $\{g'(f), g''(f)\}$). We define metric on F by

$$\|f\|^2 = \|\{g'(f), g''(f)\}\|^2 = \|g'(f)\|_1^2 + \|g''(f)\|_2^2.$$

Theorem 1. *([6]) Let F_i (for $i = 1, 2$) be RKHSs and K_i and $\|.\|_i$ the corresponding kernels and norms. Let F be defined as above with norm $\|f\|^2 = \|\{g'(f), g''(f)\}\|^2 = \|g'(f)\|_1^2 + \|g''(f)\|_2^2$. Then*

$$K(x, y) = K_1(x, y) + K_2(x, y) \tag{5}$$

is the kernel corresponding to F.

The claim holds also for F defined as class of all functions $f = f_1 + f_2$ with $f_i \in F_i$ and norm $\|f\|^2 = \min(\|f_1\|_1^2 + \|f_2\|_2^2)$ minimum taken for all decompositions $f = f_1 + f_2$ with f_i in F_i.

The kernel that can be expressed as (5) we call a *Sum Kernel*. We will present two types of Sum Kernels. First suppose that a-priori knowledge or analysis of data suggests to look for a solution as a sum of two functions (for example data is generated from function influenced by two sources of different frequencies). Then we can use a

kernel summed of two parts (employing Theorem 1) corresponding to high and low frequencies, a simple example is the sum of two Gaussians of different widths:

$$K(x,y) = K_1(x,y) + K_2(x,y) = e^{-(\frac{\|x-y\|}{d_1})^2} + e^{-(\frac{\|x-y\|}{d_2})^2}. \tag{6}$$

Since we operate in an RKHS we can employ the Representer theorem and obtain a solution in the form of

$$f_0(x) = \sum_{i=1}^{N} w_i \left(e^{-\left(\frac{\|x-x_i\|}{d_1}\right)^2} + e^{-\left(\frac{\|x-x_i\|}{d_2}\right)^2} \right).$$

The second task would be to approximate data with different distribution in the input space. We can divide the space to several subsets $A_1, \ldots, A_s$ and choose (possibly different) kernels K_i for each A_i. Then we obtain a kernel as a sum of kernels K_i restricted to corresponding sets:

$$K(x,y) = \begin{cases} K_i(x,y) & x,y \in A_i \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Now we will derive a product of kernels. We will deal with the product of RKHSs. For $i = 1, 2$ let F_i be an RKHS on Ω_i, and K_i the corresponding kernel. Consider the following set of functions on $\Omega = \Omega_1 \times \Omega_2$: $F' = \left\{ \sum_{k=1}^{n} f_{1,k}(x_1) f_{2,k}(x_2) \mid n \in \mathbb{N}, f_{1,k} \in F_1, f_{2,k} \in F_2 \right\}$. Clearly, F' is a vector space, but not a complete one. To make it complete, we first define a scalar product on F'. Let $f, g \in F'$ be expressed as $f(x_1,x_2) = \sum_{k=1}^{n} f_{1,k}(x_1) f_{2,k}(x_2)$, $g(x_1,x_2) = \sum_{j=1}^{m} g_{1,j}(x_1) g_{2,j}(x_2)$. We define $\langle f, g \rangle = \sum_{k=1}^{n} \sum_{j=1}^{m} \langle f_{1,k}, g_{1,j} \rangle_1 \langle f_{2,k}, g_{2,j} \rangle_2$, where $\langle \cdot, \cdot \rangle_i$ denotes the scalar product in F_i. It is a routine to check that this definition does not depend on the particular form in which f and g are expressed and that the properties of scalar product are satisfied. We define a norm on F' by $\|f\| = \sqrt{\langle f, f \rangle}$. Finally, let F be the completion of F'. It can be shown [6] that the completion exists not only as an abstract Hilbert space but that F is in fact a space of functions on Ω. We call F the product of F_1 and F_2 and write $F = F_1 \otimes F_2$.

Theorem 2. *([6]) For $i = 1, 2$ let F_i be an RKHS on Ω_i with kernel K_i. Then the product $F = F_1 \otimes F_2$ on $\Omega_1 \times \Omega_2$ is an RKHS with kernel given by*

$$K((x_1,x_2),(y_1,y_2)) = K_1(x_1,y_1) K_2(x_2,y_2), \tag{8}$$

where $x_1, y_1 \in \Omega_1$, $x_2, y_2 \in \Omega_2$.

A kernel obtained as a product of another two kernel functions (8) is called a *Product Kernel*. Product Kernels might be useful if a-priori knowledge of data suggests to look for the solution as a member of a product of two function spaces.

Generally, it is possible to combine different types of kernels, for example for heterogeneous data, where individual attributes are of different types and different kernels are suitable for them.

RN with Sum Kernels, respectively Product Kernels, in the hidden layer (see Fig. 4) we call a *Sum Kernel Regularization Network* (SKRN), resp. *Product Kernel Regularization Network* (PKRN).

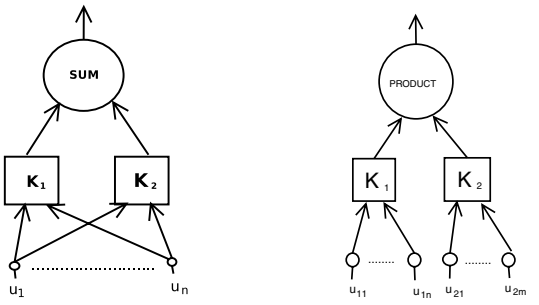


Fig. 4. a) Sum Unit b) Product Unit

4 Experimental Results

The goal of our experiments was to demonstrate the performance of the proposed PKRN and SKRN, and compare them with a classical RN with Gaussian kernels.

For the comparison we have chosen the Proben1 data repository (see [9]) containing both approximation and classification tasks, listed in Tab. 1. In addition we applied the PKRN on a real-life task, the prediction of the flow rate on the Czech river Ploučnice.

Table 1. Overview of Proben1 tasks. Number of inputs (n), number of outputs (m), number of samples in training and testing sets (N_{train},N_{test}). Type of task: approximation or classification.

Task name	n	m	N_{train}	N_{test}	**Type**
cancer	9	2	525	174	class
card	51	2	518	172	class
diabetes	8	2	576	192	class
flare	24	3	800	266	approx
glass	9	6	161	53	class
heartac	35	1	228	75	approx
hearta	35	1	690	230	approx
heartc	35	2	228	75	class
heart	35	2	690	230	class
horse	58	3	273	91	class
soybean	82	19	513	170	class

As Product and Sum Kernels we used products and sums of two Gaussian functions with different widths. For each task and network we first estimated the explicit parameters of the learning algorithm, i.e., the regularization parameter γ and the width(s) of Gaussians. Parameters with the lowest cross-validation error on the training set were chosen and used to learn the network on the whole training set. Then the error on the testing set was evaluated, as a measure of real performance of the network. The error was always normalised:

$$E = 100\frac{1}{N}\sum_{i=1}^{N}||y_i - f(x_i)||^2. \tag{9}$$

The LAPACK library [10] was used for linear system solving.

The resulting errors achieved by RN, PKRN, and two types of SKRN on tasks from Proben1 are compared in Tab. 2 (SKRN_1 is a network with Sum Kernels of the first

type (6), SKRN_2 uses Sum Kernels of the second type (7)). E_{train} and E_{test} stands for the value of the normalised error function (9) on the training set and testing set, respectively.

We can see that all types of RNs achieved comparable results in terms of errors on testing set, best results on most tasks were achieved by SKRN_1.

The SKRN_1 showed an interesting behavior on several data sets. The error on the training set is almost zero (rounded to zero) and still the generalization ability of the network is good, i.e., the error on the testing set is not high. This is caused by the fact that the chosen kernel consists of two Gaussians, one being very narrow (see Fig. 4). The diagonal of the matrix K from (4) is dominant and so the regularization member is not needed.

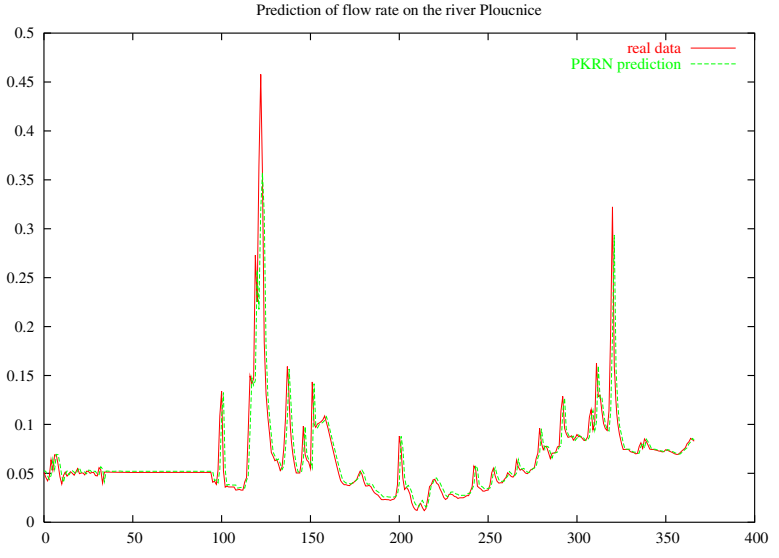


Fig. 5. Prediction of the flow rate by PKRN

In case of SKRN_2 we divided each data set into two or three disjoint sets, on each of them only one kernel was active (see (7)). This enables us to replace the larger linear system by two (resp. three) smaller ones for individual subsets, which significantly decreases the time requirements (see Fig. 6), and in addition makes parallelization possible.

The applicability of PKRN on real life problems was demonstrated on the prediction of the flow rate. Our goal is to predict the current flow rate from the flow rate and total rainfall from the previous date , i.e. we are approximating function $f : \Re \times \Re \rightarrow \Re$. The data set contains 1000 training samples and 367 testing samples.

Table 3 shows that the PKRN outperforms the so called *conservative predictor* (CP). CP is a predictor saying that the value will be the same as it was yesterday, and in spite of its simplicity it is very successful on some tasks, including this one.

The prediction on the testing set made by PKRN is displayed on Fig. 5.

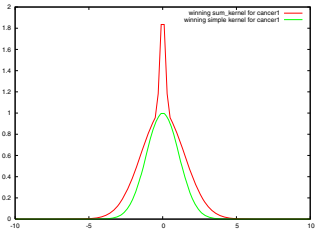
Table 2. Comparisons of errors on training and testing set for RN with Gaussian kernels and SKRN and PKRN

	RN		$SKRN_1$		PKRN		$SKRN_2$	
Task	E_{train}	E_{test}	E_{train}	E_{test}	E_{train}	E_{test}	E_{train}	E_{test}
cancer1	2.28	**1.75**	0.00	1.77	2.68	1.81	2.11	1.93
cancer2	1.86	3.01	0.00	**2.96**	2.07	3.61	1.68	3.37
cancer3	2.11	2.79	0.00	**2.73**	2.28	2.81	1.68	2.95
card1	8.75	**10.01**	8.81	10.03	8.90	10.05	8.55	10.58
card2	7.55	**12.53**	0.00	12.54	8.11	12.55	7.22	13.03
card3	6.52	12.35	6.55	**12.32**	7.01	12.45	6.22	12.86
diabetes1	13.97	16.02	14.01	**16.00**	16.44	16.75	12.92	16.66
diabetes2	14.00	**16.77**	13.78	16.80	15.87	18.14	13.64	17.33
diabetes3	13.69	16.01	13.69	**15.95**	16.31	16.62	12.85	16.34
flare1	0.36	0.55	0.35	**0.54**	0.36	**0.54**	0.35	0.59
flare2	0.42	0.28	0.44	**0.26**	0.42	0.28	0.41	0.28
flare3	0.38	0.35	0.42	**0.33**	0.40	0.35	0.38	0.34
glass1	3.37	6.99	2.35	**6.15**	2.64	7.31	2.56	6.78
glass2	4.32	7.93	1.09	**6.97**	2.55	7.46	3.27	7.29
glass3	3.96	7.25	3.04	**6.29**	3.31	7.26	3.48	6.44
heart1	9.61	**13.66**	0.00	13.91	9.56	13.67	9.51	13.79
heart2	9.33	13.83	0.00	**13.82**	9.43	13.86	8.52	14.31
heart3	9.23	15.99	0.00	**15.94**	9.15	16.06	8.30	16.75
hearta1	3.42	4.38	0.00	**4.37**	3.47	4.39	3.20	4.45
hearta2	3.54	4.07	3.51	**4.06**	3.28	4.29	3.17	4.34
hearta3	3.44	4.43	0.00	4.49	3.40	4.44	3.37	**4.40**
heartac1	4.22	**2.76**	0.00	3.26	4.22	**2.76**	3.68	3.37
heartac2	3.50	3.86	0.00	**3.85**	3.49	3.87	2.99	3.97
heartac3	3.36	**5.01**	3.36	**5.01**	3.26	5.18	3.14	5.13
heartc1	9.99	16.07	0.00	**15.69**	10.00	16.08	6.50	16.07
heartc2	12.70	**6.13**	0.00	6.33	12.37	6.29	11.06	6.69
heartc3	8.79	12.68	0.00	12.38	8.71	12.65	9.91	**11.74**
horse1	7.35	**11.90**	0.20	**11.90**	14.25	12.45	7.66	12.62
horse2	7.97	15.14	2.84	**15.11**	12.24	15.97	6.84	15.70
horse3	4.26	**13.61**	0.18	14.13	9.63	15.88	8.56	15.24
soybean1	0.12	0.66	0.11	0.66	0.13	0.86	0.12	**0.64**
soybean2	0.24	**0.50**	0.25	0.53	0.23	0.71	0.19	0.54
soybean3	0.23	0.58	0.22	**0.57**	0.21	0.78	0.15	0.72

Table 3. Flow rate errors by PKRN and CP

	PKRN	**CP**
E_{train}	0.057	0.093
E_{test}	0.048	0.054

Table 4. Chosen kernels for cancer1 task



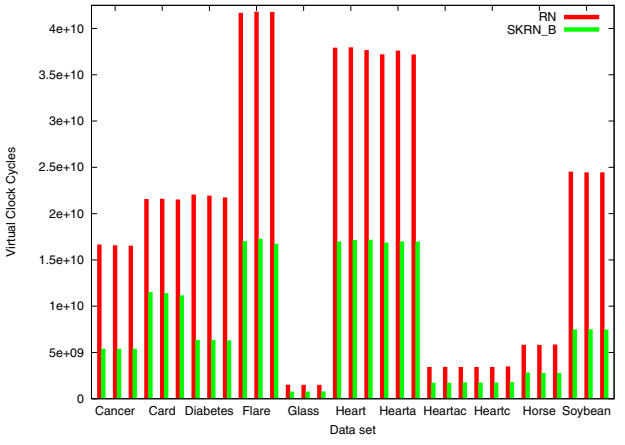


Fig. 6. Time (in clock cycles, using the PAPI library [11]) needed for one run of learning algorithm for RN and SKRN$_B$

5 Conclusion

We have shown how to use results of Aronszajn on sums and products of RKHSs to obtain the Sum and Product Regularization Networks.

We compared proposed PKRN and SKRN to classical RN on benchmarks. All methods gave comparable results, though our SKRN achieved lowest errors in most cases. We also demonstrated how SKRN can be used to decrease the time requirements for larger data sets. In addition, we demonstrated the performance of PKRN on prediction of a flow rate on the river Ploučnice and showed that it performs better than the Conservative Predictor.

We showed that our algorithms are a vital alternative to classical RNs. We can benefit from them in situations where some knowledge of the character of data is available or if we can expect that for some groups of inputs different kernel functions are suitable.

Our future work should be focused on the application of other types of kernel functions (not only Gaussian functions), and the automated choice of the optimal kernel function for a given problem.

For proofs, details of the learning algorithm, a more detailed description of our experiments, and further results see [8].

References

1. Cucker, F., Smale, S.: On the mathematical foundations of learning. Bulletin of the American Mathematical Society **39** (2001) 1–49
2. Girosi, F.: An equivalence between sparse approximation and support vector machines. Technical report, Massachutesetts Institute of Technology (1997) A.I. Memo No. 1606.
3. Girosi, F., Jones, M., Poggio, T.: Regularization theory and Neural Networks architectures. Neural Computation **2** (1995) 219–269

4. Poggio, T., Smale, S.: The mathematics of learning: Dealing with data. Notices of the AMS **50** (2003) 536–544
5. Kudová, P.: Learning with kernel based regularization networks. In: Information Technologies - Applications and Theory. (2005) 83–92
6. Aronszajn, N.: Theory of reproducing kernels. Transactions of the AMS **68** (1950) 337–404
7. Šidlofová, T.: Existence and uniqueness of minimization problems with fourier based stabilizers. In: Proceedings of Compstat, Prague. (2004)
8. Šámalová, T., Kudová, P.: Sum and product kernel networks. Technical report, Institute of Computer Science, AS CR (2005)
9. Prechelt, L.: PROBEN1 – a set of benchmarks and benchmarking rules for neural network training algorithms. Technical Report 21/94, Universitaet Karlsruhe (1994)
10. LAPACK: Linear algebra package (`http://www.netlib.org/lapack/`)
11. PAPI: Performance appl. prog. interface (`http://icl.cs.utk.edu/papi/`)

Chaotic Cellular Neural Networks with Negative Self-feedback

Wen Liu[1,2], Haixiang Shi[2], Lipo Wang[1,2], and Jacek M. Zurada[2,3]

[1] College of Information Engineering,
Xiangtan University, Xiangtan, Hunan, China
[2] School of Electrical and Electronic Engineering,
Nanyang Technology University,
Block S1, 50 Nanyang Avenue, Singapore 639798
[3] Computational Intelligence Laboratory/Lutz Hall,
Room 439 Electrical and Computer Engineering Department
University of Louisville, Louisville, KY 40292
{liuw0004, pg02782641, elpwang}@ntu.edu.sg,
jmzura02@louisville.edu

Abstract. We propose a new model of Chaotic Cellular Neural Networks (C-CNNs) by introducing negative self-feedback into the Euler approximation of the continuous CNNs. According to our simulation result for the single neuron model, this new C-CNN model has richer and more flexible dynamics, compared to the conventional CNN with only stable dynamics. The hardware implementation of this new network may be important for solving a wide variety of combinatorial optimization problems.

1 Introduction

Many modern systems developed these years are based on artificial intelligence and soft computing, where artificial neural networks play an important role, together with fuzzy logic, evolutionary computation, and chaos. Artificial neural networks have been widely studied in various areas since the end of the $20th$ century.

Chua and Yang proposed a circuit architecture, called cellular neural networks (CNNs) [1] in 1988, which possess the ability to do parallel signal processing in real time. Compared with general neural networks, CNNs are much more amenable to Very Large Scale Integration (VLSI) implementation according to its neighbor interactive property. Some rather promising applications of CNNs in image processing, communication systems and optimization problem have been reported in [2,3,4,5,6,7,8].

Grassi [9] designed a discrete-time cellular neural network (DTCNN) which is globally asymptotically stable to behave as associative memories. Fantacci et al [10] exploited CNNs' capability to account for the optimization problem by implementing of a cell scheduling algorithm. They also showed that for a class of optimization problems, the performance of CNNs is compared to Hopfield neural networks (HNN). Nakaguchi et al [11] proposed an architecture based on the Hysteresis CNN

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 66–75, 2006.
© Springer-Verlag Berlin Heidelberg 2006

and applied it on the N-Queen problem, a classic combinatorial optimization problem. The result showed the effectiveness of this network architecture.

Chaos have been widely investigated by not only mathematicians and physicists, but also engineers, economists, and scientists from various discipline. Chaotic dynamics have several special characteristics, such as a sensitivity to initial conditions, determinism as the system function is well defined, and long term unpredictability.

Using chaotic dynamics to solve combinatorial optimization problems (COP) has been studied widely after the work of Nozawa [12], Chen and Aihara [13]. Nozawa modified Hopfield network by adding negative self-feedback connection, analyzed one-dimensional maps for the single neuron model under different control parameters, and demonstrated the existence of chaos. Two engineering applications were also presented to show the effectiveness of the new model. Chen and Aihara proposed a neural network model with transient chaos (TCNN), which acted better in searching for globally optimal or near-optimal solutions compared with the conventional Hopfield-type neural network.

Combinatorial optimization problems is a branch of optimization problem where feasible solutions are discrete. The objective is to find the optimal possible solution. Some found that chaotic dynamics are more effective for solving optimization problems than stochastic dynamics [14,15]. Bucolo et al investigated the effect of chaos on helping order to arise from disorder and studied the subject "*Does chaos work better than noise?*" [16]. They did several experiments for different applications to compare chaos and noise, and asserted that although a general judgment have not been formulated, chaotic dynamic are often better than random signals.

Hayakawa [17] emphasized on the effects of chaos in neural network dynamic through a simple model for the traveling salesman problem (TSP). Aihara [18] discussed chaotic dynamics from an engineering point of view. The paper recounted the development and prospective of chaos engineering. He [19] added decaying chaotic noise which was generated by the logistic map to the discrete-time continuous-output Hopfield neural network. The simulation on the TSP showed better results in terms of searching ability and iteration steps compared to Chen and Aihara's model [13].

Based on the efficient search ability with chaos and the mature technology of VLSI implementation of CNNs, this paper proposes a new *chaotic cellular neural network* (C-CNN) with negative self-feedback. This new model share the best feature of both world: It has complex dynamics so possesses higher ability to search for optimal solutions for optimization problems; And furthermore, the local interconnection feature of CNNs made it especially suited for large scale analog implementation.

The complex dynamics in CNNs have already been reported in the past years. Civalleri and Gilli [20] reviewed and discussed the stability of the original CNN model, including complete stability, stability almost everywhere and global asymptotic stability. Zou and Nossek observed a chaotic attractor in a two-cell non-autonomous CNN [21]. They also presented bifurcation phenomena and chaotic attractor in autonomous CNNs but with space variant templates [22].

Gilli [23] analyzed a delayed cellular neural network (DCNN) which consist of 2 cells and then presented strange attractors generated by different delays. Gilli et al [24] explored the dynamic behavior in autonomous space-invariant CNNs. They showed limit cycle and chaotic attractor in a CNN composed by 9 (3×3) cells. Petras et al [25] found that boundary conditions will affect the stability of CNNs when off-diagonal template elements have opposite-signs.

In this paper, we add negative self-feedbacks into the Euler approximation of the continuous CNN model and demonstrate a variety of dynamic behaviors, including fixed points, periodic oscillations, and chaos, thereby creating a new chaotic CNN.

This paper is organized as follows. We propose the new model C-CNN in Section 2. The stability analysis of the network is presented in Section 3. The Simulation result of the new model is presented in Section 4. Finally, we conclude this paper in Section 5.

2 The Continuous Cellular Neural Network

Consider an original $M \times N$ cellular neural network proposed in [1], the r-neighborhood of neuron (i, j) $1 \leq i \leq M, 1 \leq j \leq N$ are defined as $N_r(i, j) = \{(k, l)|max\{|k - i|, |l - j| \leq r\}$, $1 \leq k \leq M, 1 \leq l \leq N$.

The system equation can be described as follows:

$$C\frac{dx_{ij}(t)}{dt} = -\frac{1}{R_x}x_{ij}(t) + \sum_{C(k,l)\in N_r(i,j)} A(i,j;k,l)y_{kl}(t) + \sum_{C(k,l)\in N_r(i,j)} B(i,j;k,l)u_{kl} + I \tag{1}$$

Where x_{ij}, y_{ij}, u_{ij} is the internal state, output and input of neuron (i, j) respectively. A and B are two matrices consisting of feedback synaptic weights and input control parameters. C is a linear capacitor and R_x is a linear resistor. I is an independent voltage source functioned as a constant bias.

The activation function of neuron (i, j) is piecewise linear function:

$$y_{ij}(t) = \frac{1}{2}(|x_{ij}(t) + 1| - |x_{ij}(t) - 1|) \tag{2}$$

Constraint conditions are:

$$|x_{ij}(0)| \leq 1, \qquad 1 \leq i \leq M; 1 \leq j \leq N.$$

$$|u_{ij}| \leq 1, \qquad 1 \leq i \leq M; 1 \leq j \leq N.$$

The network model described above is proven to be stable [1,26] when the space templates are symmetry i.e. $A(i, j; k, l) = A(k, l; i, j)$ and the self-feedback $A(i, j; i, j) > 1$. Any cell in a cellular neural network is connected only to its neighboring cells. This CNN property of nearest neighbor interactions makes CNNs much more amenable to VLSI implementation compared to general neural networks.

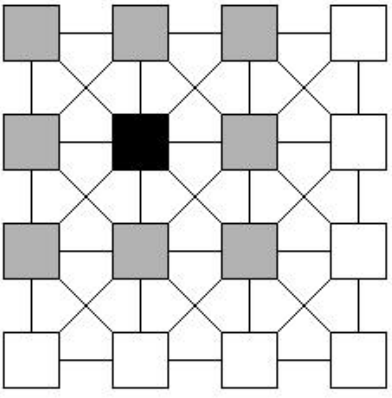


Fig. 1. An 4-by-4 network. (shaded squares are the 1-neighborhood cells of the black cell.)

3 Chaotic Cellular Neural Networks

Usually there are positive self-feedbacks in the CNN, i.e., $A(i,j,i,j) \geq 0$. Let us change it to negative by adding a negative self-feedback and take the difference equation version by Euler's method [27], then the model becomes

$$\begin{aligned} x_{ij}(t+1) &= x_{ij}(t) + f(x_{ij}(t))\Delta t \\ &= (1 - \frac{\Delta t}{CR_x})x_{ij}(t) + \frac{\Delta t}{C}\sum_{C(k,l)\in N_r(i,j)} A(i,j;k,l)y_{kl}(t) \\ &\quad + \frac{\Delta t}{C}\sum_{C(k,l)\in N_r(i,j)} B(i,j;k,l)u_{kl} + \frac{\Delta t}{C}I - z\left[y_{ij}(t) - I_0\right] \\ &= p\, x_{ij}(t) - z\left[y_{ij}(t) - I_0\right] + \sum_{C(k,l)\in N_r(i,j)} a(i,j;k,l)y_{kl}(t) \\ &\quad + \sum_{C(k,l)\in N_r(i,j)} b(i,j;k,l)u_{kl} + i \end{aligned} \tag{3}$$

$$y_{ij}(t) = \frac{1}{2}\left(|x_{ij}(t)+1| - |x_{ij}(t)-1|\right) \tag{4}$$

where

$$\begin{aligned} x_{ij} &= \text{Internal state of neuron } (i,j), \\ y_{ij} &= \text{Output of neuron } (i,j), \\ u_{ij} &= \text{Input of neuron } (i,j), \\ A(i,j;k,l) &= \text{The output feedback parameter,} \end{aligned}$$

$$
\begin{aligned}
B(i,j;k,l) &= \text{The input control parameter,}\\
C(i,j) &= \text{The neuron } (i,j),\\
N_r(i,j) &= \{(k,l)|max\{|k-i|,|l-j|\} \le r,\\
&\quad 1 \le k \le M, 1 \le l \le N\}\\
&\quad \text{The r-neighborhood of neuron } (i,j),\\
I &= \text{An independent voltage source,}\\
p &= 1 - \frac{\Delta t}{CR_x},\\
z &= \text{Self-feedback connection weight,}\\
I_0 &= \text{A positive bias factor,}\\
C &= \text{A linear capacitor,}\\
R_x &= \text{A linear resistor.}
\end{aligned}
$$

This new C-CNN has two new terms compared to the conventional CNN. The first term $-z\,[y_{ij}(t) - I_0]$ is related to the negative self-feedback with a bias I_0. The second term is $p\,x_{ij}(t)$, where p is determined by the time step Δt in Euler's method.

In order to compare the new model with the continuous time CNN, Eqn. (3)-(4) can be rewritten in forms of differential equations as follows:

$$
\begin{aligned}
\frac{dx_{ij}(t)}{dt} = &-z(y_{ij}(t) - I_0) - x_{ij}(t) + \sum_{C(k,l)\in N_r(i,j)} A(i,j;k,l)y_{kl}(t)\\
&+ \sum_{C(k,l)\in N_r(i,j)} B(i,j;k,l)u_{kl} + I
\end{aligned} \tag{5}
$$

$$
y_{ij}(t) = \frac{1}{2}(|x_{ij}(t)+1| - |x_{ij}(t)-1|). \tag{6}
$$

4 Stability of the New C-CNN Model

Based on the energy function used in [1], we use a computational energy function as below:

$$
\begin{aligned}
E(t) = &-\frac{1}{2}\sum_{(i,j)}\sum_{(k,\,l)}[a(i,j;k,l) - z\delta_{ik}\delta_{jl}]\,y_{ij}(t)\,y_{kl}(t)\\
&-\sum_{(i,j)}\sum_{(k,l)} b(i,j;k,l)\,y_{ij}(t)\,u_{kl} - \sum_{(i,j)}[i + zI_0]\,y_{ij}(t)\\
&+\frac{1-p}{2}\sum_{(i,j)} y_{ij}(t)^2
\end{aligned} \tag{7}
$$

The change in the energy function between two time steps:

$$\begin{aligned}\Delta E = &-\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)}[a(i,j;k,l) - z\delta_{ik}\delta_{jl}]\,\Delta y_{ij}(t)\,\Delta y_{kl}(t)\\ &-\sum_{(i,j)}\sum_{(k,l)}[a(i,j;k,l) - z\delta_{ik}\delta_{jl}]\,y_{kl}(t)\Delta y_{ij}(t)\\ &-\sum_{(i,j)}\Delta y_{ij}(t)\left[\sum_{(k,l)} b(i,j;k,l)\,u_{kl} + i + zI_0\right]\\ &+\frac{1-p}{2}\sum_{(i,j)}\Delta y_{ij}(t)\,[y_{ij}(t+1) + y_{ij}(t)] \end{aligned} \tag{8}$$

Substituting the cell circuit equation (3) into equation (8), we obtain

$$\begin{aligned}\Delta E = &-\frac{1}{2}\sum_{(i,j)}\sum_{(k,l)}[a(i,j;k,l) - z\delta_{ik}\delta_{jl}]\,\Delta y_{ij}(t)\Delta y_{kl}(t)\\ &-\sum_{(i,j)}\Delta y_{ij}(t)[x_{ij}(t+1) - p\,x_{ij}(t)]\\ &+\frac{1-p}{2}\sum_{(i,j)}\Delta y_{ij}(t)[y_{ij}(t+1) + y_{ij}(t)] \end{aligned} \tag{9}$$

According to the output function (4)

$$y_{ij}(t) = x_{ij}(t), \text{ when } |x_{ij}(t)| < 1$$
$$\Delta y_{ij}(t) = 0, \text{ when } |x_{ij}(t)| \geq 1$$

then

$$\Delta E = -\frac{1}{2}\sum_{|x_{ij}|<1}\sum_{|x_{kl}|<1}\{a(i,j;k,l) + [-z+1+p]\delta_{ik}\delta_{jl}\}\;\Delta y_{ij}(t)\Delta y_{kl}(t) \tag{10}$$

Therefore $\Delta E \leq 0$, or the network is stable according to Lyapunov Theorem [28], if the matrix $\{a(i,j;k,l) + [-z+1+p]\delta_{ik}\delta_{jl}\}$ is positive-definite.

Hence a sufficient stability condition for the new model is

$$a(i,j;i,j) - z > -(1+p) \tag{11}$$

The new C-CNN model converges to a stable state when we choose a proper value for self-feedback parameter z that satisfies inequality (11).

5 Simulation Result

Deriving from (3)-(4), we obtain the single neuron model:

$$x(t+1) = -z(y(t) - I_0) + p\,x(t) + A\,y(t) + B\,u + I \tag{12}$$

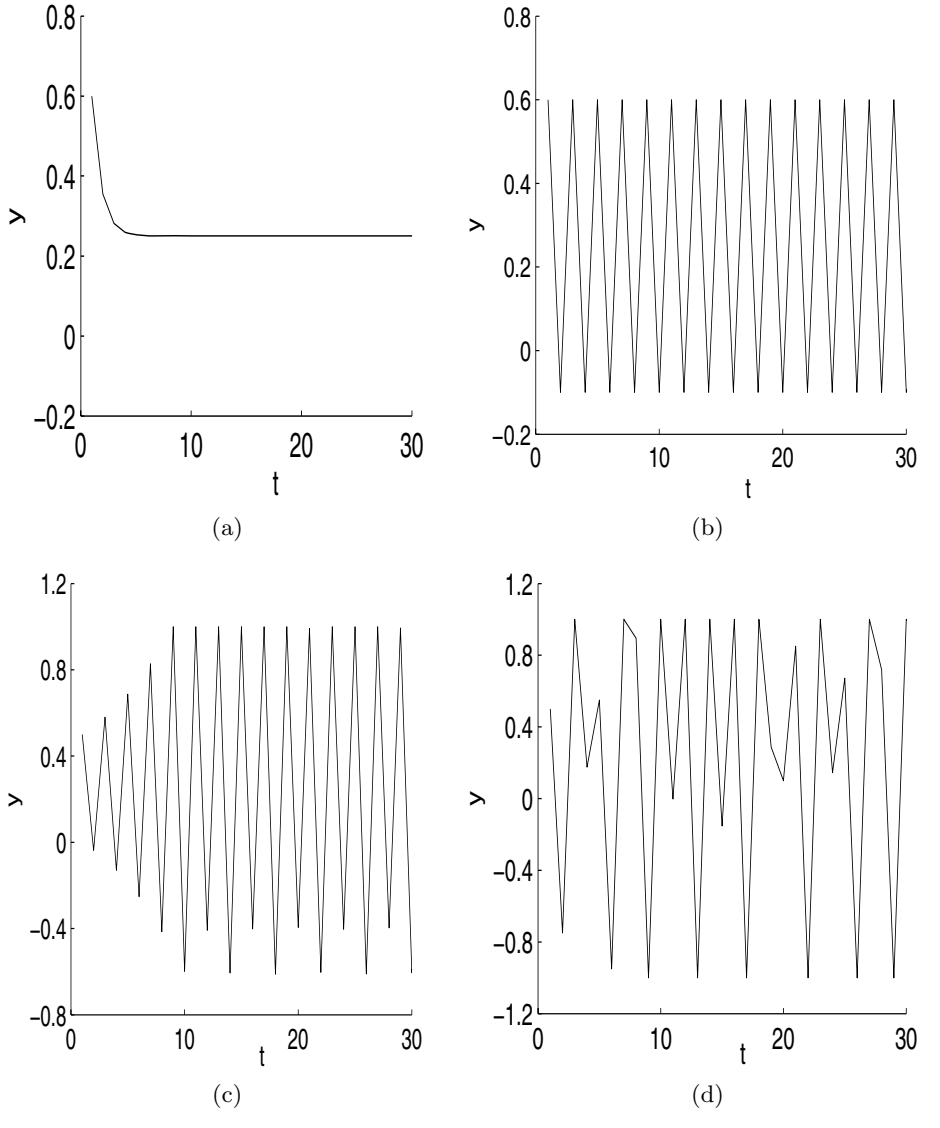


Fig. 2. Same template values can produce various dynamics with different self-feedback for C-CNN (a) stable dynamic with z=0.7; (b) period-2 oscillations with z=2; (c) period-3 oscillations with z=2.15; (d) chaotic dynamics with z=5.

$$y(t) = \frac{1}{2}\left(|x(t)+1| - |x(t)-1|\right) \tag{13}$$

In the following, we vary only the value of z to investigate the dynamics of the network while other parameters are fixed in Eqn.(12) as $p = 0.999, A = 0.1, Bu + I = 0, I_0 = 0.25$. Fig.2 shows the time evolutions for $y(t)$ when $z = 0.7, 2.0, 2.15, 5$, respectively, and the initiative value for neuron state x is randomly generated in $[-1, 1]$.

We observe the output of the neuron in 30 iteration steps. When $z < 2$, there is only one stable equilibrium for the output. When $z = 2$ the process begins oscillating between two different points (0.6 and -0.1) without converging to either (period-2 oscillations). When $z = 2.15$ the output oscillates between three different points (1, -0.4 and -0.6) forming a period-3 output - "Period-3 implies chaos" [29]. When z is larger, the output has chaotic dynamics.

6 Conclusion

When we add negative self-feedbacks, our discrete *Chaotic Cellular Neural Networks* (C-CNNs) have various complex dynamics (i.e. limit cycles, bifurcation processes and chaotic attractors) depending on the magnitude of the self-feedback. The stability condition is proved. The new C-CNN model take advantage of both efficient search for global optimal with chaos and the mature technology of VLSI implementation of CNNs. In future work, we will apply this new model to solve various combinatorial optimization problems.

References

1. Chua, L.O., Yang, L.: Cellular neural networks: Theory. IEEE Transactions on Circuits and Systems I **35**(10) (1988) 1257–1272
2. Chua, L.O., Yang, L.: Cellular neural networks: Applications. IEEE Transactions onCircuits and Systems I **35**(10) (1988) 1273–1290
3. Manganaro, G., Pineda de Gyvez, J.: One-dimensional discrete-time cnn with multiplexed template-hardware. IEEE Transactions on Circuits and Systems I **47**(5) (2000) 764–769
4. Bang, S.H., Sheu, B.J., Chou, E.Y.: A hardware annealing method for optimal solutions on cellular neural networks. IEEE Transactions on Circuits and Systems **43**(6) (1996) 409–421
5. Caponetto, R., Fortuna, L., Occhipinti, L., Xibilia, M.G.: Sc-cnns for chaotic signal applications in secure communication systems. International Journal of Neural Systems **13**(6) (2003) 461–468
6. Takahashi, N., Otake, T., Tanaka, M.: The template optimization of discrete time cnn for image compression and reconstruction. In: IEEE International Symposium on Circuits and Systems, ISCAS. (2002) 237–240
7. Bise, R., Takahashi, N., Nishi, T.: An improvement of the design method of cellular neural networks based on generalized eigenvalue minimization. IEEE Transactions on Circuits and Systems I **50**(12) (2003) 1569–1574

8. Wang, S., Wang, M.: A new detection algorithm (nda) based on fuzzy cellular neural networks for white blood cell detection. IEEE Transactions on information technology in biomedicine **10**(1) (Jan. 2006) 5–10
9. Grassi, G.: On discrete-time cellular neural networks for associative memories. IEEE Transactions on Circuits and Systems **48**(1) (Jan. 2001) 107–111
10. Fantacci, R., Forti, M., Pancani, L.: Cellular neural network approach to a class of communication problems. IEEE Transactions Circuits and Systems I **46**(12) (1999) 1457–1467
11. Nakaguchi, T., Omiya, K., Tanaka, M.: Hysteresis cellular neural networks for solving combinatorial optimization problems. In: Proc. of CNNA 2002. (2002) 539–546
12. Nozawa, H.: A neural-network model as a globally coupled map and applications based on chaos. Chaos **2**(3) (1992) 377–386
13. Chen, L.N., Aihara, K.: Chaotic simulated annealing by a neural network model with transient chaos. Neural Networks **8**(6) (1995) 915–930
14. Wang, L.P., Li, S., Tian, F.Y., Fu, X.J.: A noisy chaotic neural network for solving combinatorial optimization problems: stochastic chaotic simulated annealing. IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics **34**(5) (2004) 2119–2125
15. He, Z., Zhang, Y., Wei, C., Wang, J.: A multistage self-organizing algorithm combined transiently chaotic neural network for cellular channel assignment. Vehicular Technology, IEEE Transactions on **51**(6) (2002) 1386 – 1396
16. Bucolo, M., Caponetto, R., Fortuna, L., Frasca, M., Rizzo;, A.: Does chaos work better than noise? Circuits and Systems Magazine, IEEE **2**(3) (2002) 4–19
17. Hayakawa, Y., Marumoto, A., Sawada, Y.: Effects of the chaotic noise on the performance of a neural network model for optimization problems. Physical review E **51**(4) (1995) R2693CR2696
18. Aihara, K.: Chaos engineering and its application to parallel distributed processing with chaotic neural networks. Proceedings of the IEEE **90**(5) (2002) 919–930
19. He, Y.: Chaotic simulated annealing with decaying chaotic noise. Neural Networks, IEEE Transactions on **13**(6) (2002) 1526 – 1531
20. Civalleri, P.P., Gilli, M.: On stability of cellular neural networks. Journal of VLSI signal processing **23** (1999) 429–435
21. Zou, F., Nossek, J.A.: A chaotic attractor with cellular neural networks. IEEE Transactions on Circuits and Systems I **38**(7) (1991) 811–812
22. Zou, F., Nossek, J.A.: Bifurcation and chaos in cellular neural networks. IEEE Transactions on Circuits and Systems I **40**(3) (1993) 166–173
23. Gilli, M.: Strange attractors in delayed cellular neural networks. IEEE Transactions on Circuits and Systems I **40**(11) (1993) 849–853
24. Gilli, M., Biey, M., Civalleri, P., Checco, P.: Complex dynamics in cellular neural networks. In: Proc. of IEEE International Symposium on Circuits and Systems. (2001) 45–48
25. Petras, I., P.Checco, Gilli, M., Roska, T., biey, M.: On the effect of boundary condition on cnn dynamics: Stability and instability; bifurcation processes and chaotic phenomena. In: Proc. of ISCAS 2003. (2003) 590–592
26. Li, X., Ma, C., Huang, L.: Invariance principle and complete stability for cellular neural networks. IEEE Transactions on Circuits and Systems II **53**(3) (2006) 202–206

27. Nozawa, H.: Solution of the optimization problem using the neural-network model as a globally coupled map. In: Towards the Harnessing of Chaos. M. Yamaguti (1994) 99–114
28. Haykin, S. In: Neural Networks-A comprehensive Foundation (second edition). Prentice Hall International Inc., Hamilton, Canada (1999)
29. Li, T., Yorke, J.: Period-3 implies chaos. Am. Math. Monthly **82** (1975) 985–992

An Efficient Nonlinear Predictive Control Algorithm with Neural Models and Its Application to a High-Purity Distillation Process

Maciej Ławryńczuk and Piotr Tatjewski

Institute of Control and Computation Engineering,
Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
Tel.: +48 22 660-73-97
lawrynczuk@ia.pw.edu.pl, tatjewski@ia.pw.edu.pl

Abstract. This paper is concerned with a computationally efficient (sub-optimal) nonlinear model-based predictive control (MPC) algorithm and its application to a high-purity high-pressure ethylene-ethane distillation column. A neural model of the process is used on-line to determine the local linearisation and the nonlinear free response. In comparison with general nonlinear MPC technique, which hinges on non-convex optimisation, the presented structure is far more reliable and less computationally demanding because it results in a quadratic programming problem, whereas its closed-loop control performance is similar.

1 Introduction

Model Predictive Control (MPC) is recognised as the only advanced control technique (i.e. more advanced than the well known PID approach) which has been very successful in practical applications [2], [7], [10], [13], [14], [15]. More specifically, MPC algorithms can take into account constraints imposed on both process inputs (manipulated variables) and outputs (controlled variables), which usually decide on quality, economic efficiency and safety. Furthermore, MPC techniques are very efficient in multivariable process control. The underlaying idea of MPC is relatively easy to explain to engineering and operator staff, which is of fundamental importance when it comes to introducing new techniques into industrial practice.

The paper describes the computationally efficient MPC with Nonlinear Prediction and Linearisation (MPC-NPL) algorithm with feedforward neural network models and its application to a distillation process. The algorithm gives good closed-loop performance and, unlike the nonlinear MPC techniques, which hinge on nonlinear optimisation, it uses on-line only the quadratic programming approach.

The paper is organised as follows. In Section 2 various MPC approaches are shortly discussed in light of both computational load and closed-loop performance. Next, in Section 3, the MPC-NPL algorithm with feedforward neural networks is detailed. In Section 4 simulation results are discussed and the paper is summarised in Section 5.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 76–85, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Model Predictive Control Algorithms

Although a number of different MPC techniques have been developed over the years, the main idea (i.e. the explicit application of a process model, the receding horizon and optimisation of a cost function) is always the same [7], [15]. At each consecutive sampling instant k a set of future control increments is calculated

$$\Delta \boldsymbol{u}(k) = [\Delta u(k|k)\ \Delta u(k+1|k) \dots \Delta u(k+N_u-1|k)]^T \tag{1}$$

It is assumed that $\Delta u(k+p|k) = 0$ for $p \geq N_u$, where N_u is the control horizon. The objective is to minimise the differences between the predicted values of the output $\hat{y}(k+p|k)$ and the reference trajectory $y^{ref}(k+p|k)$ over the prediction horizon N. The following quadratic cost function is usually used

$$J(k) = \sum_{p=1}^{N} (y^{ref}(k+p|k) - \hat{y}(k+p|k))^2 + \lambda \sum_{p=0}^{N_u-1} (\Delta u(k+p|k))^2 \tag{2}$$

Typically, $N_u < N$, which decreases the dimensionality of the optimisation problem and leads to smaller computational load. Only the first element of the determined sequence (1) is applied to the process, the control law is then

$$u(k) = \Delta u(k|k) + u(k-1) \tag{3}$$

At next sampling instant, $k+1$, the prediction is shifted one step forward and the whole procedure is repeated.

If the constraints are taken into account, future control increments are determined as the solution to the following optimisation problem (assuming hard output constraints [7], [15])

$$\begin{aligned} &\min_{\Delta u(k|k) \dots \Delta u(k+N_u-1|k)} \{J(k)\} \\ &u^{\min} \leq u(k+p|k) \leq u^{\max} \quad p = 0, \dots, N_u - 1 \\ &-\Delta u^{\max} \leq \Delta u(k+p|k) \leq \Delta u^{\max} \quad p = 0, \dots, N_u - 1 \\ &y^{\min} \leq \hat{y}(k+p|k) \leq y^{\max} \quad p = 1, \dots, N \end{aligned} \tag{4}$$

Predicted output values over the prediction horizon are calculated using a dynamic model of the process. The choice of the model (linear or nonlinear, if nonlinear – fundamental or black-box) is crucial. This decision affects not only the possible control accuracy but also the computational load and reliability of the whole control policy. MPC algorithms based on linear models have been usually applied in practice [10], [13], [15], since the predictions $\hat{y}(k+p|k)$ can be expressed as a linear combination of decision variables, which means that the optimisation problem (4) is a quadratic programming one [7], [14], [15]. Unfortunately, when the process exhibits severe nonlinearity, such an approach is likely to result in poor closed-loop control performance, even instability. In general, a nonlinear model used for prediction purposes leads to a non-quadratic,

non-convex and even multi-modal optimisation problem. For such problems there are no sufficiently fast and reliable optimisation algorithms, i.e. those which would be able to determine the global optimal solution at each sampling instant and within predefined time limit as it is required in on-line control. Gradient based optimisation techniques may terminate in local minima while global ones substantially increase the computational burden, yet they still give no guarantee that the global solution is found [8].

In order to overcome the problems inevitable in MPC with nonlinear optimisation, a few alternatives have been suggested. For example, affine nonlinear models of neural structure result in a quadratic programming problem [3]. An interesting idea is to approximate the whole MPC algorithm by a neural network, which is trained off-line [11]. Yet another option is to use a combination of a neural steady-state model and a simplified nonlinear second order quadratic dynamic model [12]. The resulting optimisation task is not convex. Nevertheless, the model is relatively simple, the approach is reported to be successful in many industrial applications.

Bearing in mind all the aforementioned computational difficulties typical of nonlinear MPC, a straightforward idea is to use a linearisation-based MPC techniques, in which only a quadratic programming problem is solved on-line. When compared to MPC algorithms with full nonlinear optimisation, they are sub-optimal, but in most practical applications the accuracy is sufficient [2], [5], [6], [10], [14], [15]. The main issue is the choice of the process model structure, since it decides on the controller's performance and accuracy. Fundamental (first-principles) models, although potentially very precise, are usually not suitable for on-line control because they are very complicated (vast number of equations.) The MPC-NPL algorithm described in the paper [5], [6], [14] uses feedforward neural networks, who are able to approximate precisely nonlinear behaviour of technological processes, have relatively small number of parameters and simple structure [1].

3 MPC-NPL Algorithm with Neural Models

Let the single-input single-output (SISO) process under consideration be described by the following nonlinear discrete-time equation

$$y(k) = g(u(k-\tau), \ldots, u(k-n_B), y(k-1), \ldots, y(k-n_A)) \tag{5}$$

where $g : \Re^{n_A+n_B-\tau+1} \longrightarrow \Re \in C^1$, $\tau \leq n_B$. In the sequel it is assumed that the feedforward neural network with one hidden layer and linear output [1] is used as the function g in (5). Output of the model can be expressed as

$$y(k) = w_0^2 + \sum_{i=1}^{K} w_i^2 v_i(k) = w_0^2 + \sum_{i=1}^{K} w_i^2 \varphi(z_i(k)) \tag{6}$$

where $z_i(k)$ and $v_i(k)$ are the sum of inputs and the output of the i-th hidden node, respectively, $\varphi : \Re \longrightarrow \Re$ is the nonlinear transfer function, K is the

number of hidden nodes. Recalling the input arguments of the general nonlinear model (5) one has

$$z_i(k) = w^1_{i,0} + \sum_{j=1}^{I_u} w^1_{i,j} u(k-\tau+1-j) + \sum_{j=1}^{n_A} w^1_{i,I_u+j} y(k-j) \tag{7}$$

The weights of the network are denoted by $w^1_{i,j}$, $i = 1, \ldots, K$, $j = 0, \ldots, n_A + n_B - \tau + 1$, and w^2_i, $i = 0, \ldots, K$, for the first and the second layer, respectively. The number of the network's input nodes depending on input signal u is $I_u = n_B - \tau + 1$. Total number of weights is $(n_A + n_B - \tau + 2)K + K + 1$.

Considering the prediction over the horizon N, the quantities $z_i(k+p|k)$ and consequently $\hat{y}(k+p|k)$ will depend on future values of control signal (i.e. decision variables of the control algorithm), values of control signal applied to the plant at previous sampling instants, future output predictions and measured values of the plant output signal. From the equation (7) one has

$$\begin{aligned} z_i(k+p|k) = w^1_{i,0} &+ \sum_{j=1}^{I_{uf}(p)} w^1_{i,j} u(k-\tau+1-j+p|k) + \\ &+ \sum_{j=I_{uf}(p)+1}^{I_u} w^1_{i,j} u(k-\tau+1-j+p) + \\ &+ \sum_{j=1}^{I_{yp}(p)} w^1_{i,I_u+j} \hat{y}(k-j+p|k) + \sum_{j=I_{yp}(p)+1}^{n_A} w^1_{i,I_u+j} y(k-j+p) \end{aligned} \tag{8}$$

where $I_{uf}(p) = \max(\min(p-\tau+1, I_u), 0)$ is the number of the network's input nodes depending on future control signals and $I_{yp}(p) = \min(p-1, n_A)$ is the number of the network's input nodes depending on output predictions.

Defining a linearisation point as the vector composed of past input and output signal values

$$\bar{x}(k) = [\bar{u}(k-\tau) \ \ldots \ \bar{u}(k-n_B) \ \bar{y}(k-1) \ \ldots \ \bar{y}(k-n_A)]^T \tag{9}$$

the linearised model has the form

$$\begin{aligned} y(k) = g(\bar{x}(k)) &+ \sum_{l=1}^{n_B} b_l(\bar{x}(k))(u(k-l) - \bar{u}(k-l)) \\ &- \sum_{l=1}^{n_A} a_l(\bar{x}(k))(y(k-l) - \bar{y}(k-l)) \end{aligned} \tag{10}$$

Taking into account the structure of the neural model and corresponding equations (6) and (7), the coefficients of the linearised model are calculated from

$$a_l(\bar{x}(k)) = -\frac{\partial g(\bar{x}(k))}{\partial y(k-l)} = -\sum_{i=1}^{K} w^2_i \frac{d\varphi(z_i(\bar{x}(k)))}{dz_i(\bar{x}(k))} w^1_{i,I_u+l} \quad l = 1, \ldots, n_A \tag{11}$$

and

$$b_l(\bar{x}(k)) = \begin{cases} 0 & l = 1, \ldots, \tau - 1 \\ \dfrac{\partial g(\bar{x}(k))}{\partial u(k-l)} = \displaystyle\sum_{i=1}^{K} w_i^2 \frac{d\varphi(z_i(\bar{x}(k)))}{dz_i(\bar{x}(k))} w_{i,l-\tau+1}^1 & l = \tau, \ldots, n_B \end{cases} \tag{12}$$

Let $a_l(k) = a_l(\bar{x}(k))$, $b_l(k) = b_l(\bar{x}(k))$. Redefining the variables $y(k) := y(k) - g(\bar{x}(k))$, $y(k-i) := y(k-i) - \bar{y}(k-i)$, $l = 1, \ldots, n_A$, $u(k-i) := u(k-i) - \bar{u}(k-i)$, $l = 1, \ldots, n_B$ the linear approximation of the model (5), obtained at sampling instant k, can be expressed as

$$\boldsymbol{A}(k, z^{-1}) y(k) = \boldsymbol{B}(k, z^{-1}) u(k) \tag{13}$$

where

$$\begin{aligned} \boldsymbol{A}(k, z^{-1}) &= 1 + a_1(k) z^{-1} + \ldots + a_{n_A}(k) z^{-n_A} \\ \boldsymbol{B}(k, z^{-1}) &= b_1(k) z^{-1} + \ldots + b_{n_B}(k) z^{-n_B} \end{aligned} \tag{14}$$

It can be noticed that the linearisation point given by (9) and hence the coefficient $a_l(k)$, $b_l(k)$ are not influenced by the most recent output value $y(k)$, which is available to be measured. It may be crucial in the case of fast processes. Therefore, it is recommended to use

$$\bar{x}(k) = [\bar{u}(k - \tau + 1) \ \ldots \ \bar{u}(k - n_B + 1) \ \bar{y} \ (k) \ \ldots \ \bar{y}(k - n_A + 1)]^T \tag{15}$$

If $\tau = 1$ for linearisation purposes one may set $\bar{u}(k) = u(k-1)$ or $\bar{u}(k) = u(k|k-1)$.

It is assumed that the superposition principle holds true, i.e. the predicted output trajectory $\hat{y}(k+p|k)$ over the prediction horizon can be expressed as the sum of a free trajectory $y^0(k+p|k)$, which depends only on the past (i.e. on the control signal values applied at previous sampling instants and measured values of the output signal) and a forced trajectory, which depends only on the future (i.e. on decision variables.) Defining the vectors

$$\boldsymbol{y}^{ref}(k) = \left[y^{ref}(k+1|k) \ldots y^{ref}(k+N|k)\right]^T \tag{16}$$

$$\hat{\boldsymbol{y}}(k) = [\hat{y}(k+1|k) \ldots \hat{y}(k+N|k)]^T \tag{17}$$

$$\boldsymbol{y}^0(k) = \left[y^0(k+1|k) \ldots y^0(k+N|k)\right]^T \tag{18}$$

the predicted output trajectory can be expressed by the equation [5], [15]

$$\hat{\boldsymbol{y}}(k) = \boldsymbol{y}^0(k) + \boldsymbol{G}(k) \Delta \boldsymbol{u}(k) \tag{19}$$

The matrix $\boldsymbol{G}(k)$ is calculated on-line from the nonlinear model taking into account the current state of the plant. It has the following structure

$$\boldsymbol{G}(k) = \begin{bmatrix} s_1(k) & 0 & \ldots & 0 \\ s_2(k) & s_1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_N(k) & s_{N-1}(k) & \ldots & s_{N-N_u+1}(k) \end{bmatrix} \tag{20}$$

where the step-response coefficients of the linearised model are determined from

$$s_j(k) = \sum_{i=1}^{\min(j,n_B)} b_i(k) - \sum_{i=1}^{\min(j-1,n_A)} a_i(k) s_{j-i}(k) \tag{21}$$

Using the superposition principle (19), the cost function (2) becomes a quadratic function of the decision variables

$$J(k) = \left\| \boldsymbol{y}^{ref}(k) - \boldsymbol{y}^0(k) - \boldsymbol{G}(k)\Delta\boldsymbol{u}(k) \right\|^2 + \lambda \left\| \Delta\boldsymbol{u}(k) \right\|^2 \tag{22}$$

The MPC-NPL algorithm can be summarised as follows:

1. Linearisation: obtain the matrix $\boldsymbol{G}(k)$.
2. Calculate the nonlinear free response $\boldsymbol{y}^0(k)$.
3. Solve the quadratic programming problem (4) with the cost function (22) to determine $\Delta\boldsymbol{u}(k)$.
4. Apply $u(k) = \Delta u(k|k) + u(k-1)$.
5. Set $k := k+1$, go to step 1.

The nonlinear free response $y^0(k+p|k)$, $p = 1, \ldots, N$, is calculated on-line recursively from the general prediction equation

$$\hat{y}(k+p|k) = y(k+p|k) + d(k) \tag{23}$$

where the quantities $y(k+p|k)$ are calculated from the nonlinear neural model. The above formulation uses the "DMC type" disturbance model, in which the unmeasured disturbance $d(k)$ is assumed to be constant over the prediction horizon. It is estimated from the equation

$$d(k) = y(k) - y(k|k-1) = y(k) - \left(w_0^2 + \sum_{i=1}^{K} w_i^2 v_i(k) \right) \tag{24}$$

where $y(k)$ is a measured value while the quantity $y(k|k-1)$ is calculated from the model (6). From (23) the nonlinear free response is given by

$$y^0(k+p|k) = w_0^2 + \sum_{i=1}^{K} w_i^2 \varphi(z_i^0(k+p|k)) + d(k) \tag{25}$$

The quantities $z_i^0(k+p|k)$ are determined from (8) assuming no changes in control signals from sampling instant k onwards and replacing predicted output signals from $k+1$ by corresponding values of the free response

$$\begin{aligned} u(k+p|k) &:= u(k-1) \quad p \geq 0 \\ \hat{y}(k+p|k) &:= y^0(k+p|k) \quad p \geq 1 \end{aligned} \tag{26}$$

hence

$$z_i^0(k+p|k) = w_{i,0}^1 + \sum_{j=1}^{I_{uf}(p)} w_1(i,j)u(k-1) + \\ + \sum_{j=I_{uf}(p)+1}^{I_u} w_{i,j}^1 u(k-\tau+1-j+p) + \\ + \sum_{j=1}^{I_{yp}(p)} w_{i,I_u+j}^1 y^0(k-j+p|k) + \sum_{j=I_{yp}(p)+1}^{n_A} w_{i,I_u+j}^1 y(k-j+p). \tag{27}$$

The extension of the presented MPC-NPL algorithm with neural networks to systems with many inputs and many outputs (MIMO) is discussed in [5]. The algorithm can be also combined with the stabilising dual-mode approach [4], [5] developed by H. Michalska and D. Q. Mayne [9].

4 Simulation Results

The plant under consideration is a high purity, high pressure (1.93 MPa) ethylene-ethane distillation column shown in Fig. 1, [5]. The feed stream consists of ethylene (approx. 80%), ethane (approx. 20%) and traces of hydrogen, methane and propylene. The product of the distillation is ethylene which can contain up to 1000 ppm (parts per million) of ethane. The objective is to develop a supervisory controller which would be able to increase relatively fast the impurity level when the composition changes in the feed stream are insignificant. Reducing the purity of the product, of course taking into account the technological limit, results in decreasing energy consumption. Production scale is very big, nominal value of the product stream flow rate is 43 tons/h. The column has 121 trays, the feed stream is delivered to the tray number 37.

The basic control layer comprises 3 fast single-loop PID controllers (denoted as LC and TC.) They are used to stabilise the levels in reflux and bottom product tanks and the temperature on the tray number 13. The supervisory control loop has one manipulated variable r, which is the reflux ratio $r = \frac{R}{P}$, where R and P are reflux and product stream flow rates, respectively, and one controlled variable z, which represents the impurity of the product. The reflux is delivered to the column by the top tray and the product is taken from the tray number 110. Sample time of the MPC algorithms is 40 min. (slow composition analyser.)

Four models of the plant were used. The first one was used as the real process during the simulations, it was based on technological considerations [5]. An identification procedure was carried out, as a result two linear models for different operating points and a neural one were obtained. For the empirical models $n_A = 1$, $\tau = n_B = 3$. The horizons were set to $N = 10$, $N_u = 3$, the weighting coefficient λ to 2. It is assumed that at sampling instant $k = 1$ the set-point value was changed from 100 ppm to 350 ppm, 600 ppm and 850 ppm. The following constraints were imposed on the reflux ratio: $r^{min} = 4.051$, $r^{max} = 4.4571$.

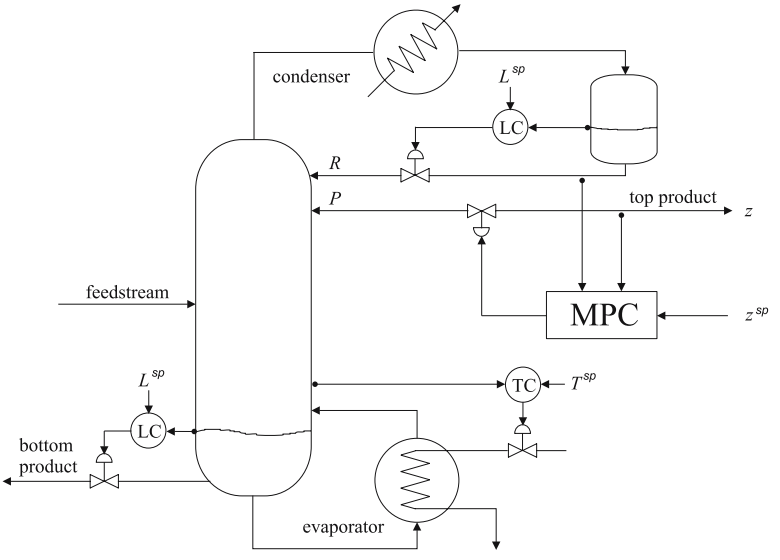


Fig. 1. High-purity ethylene-ethane distillation column control system structure

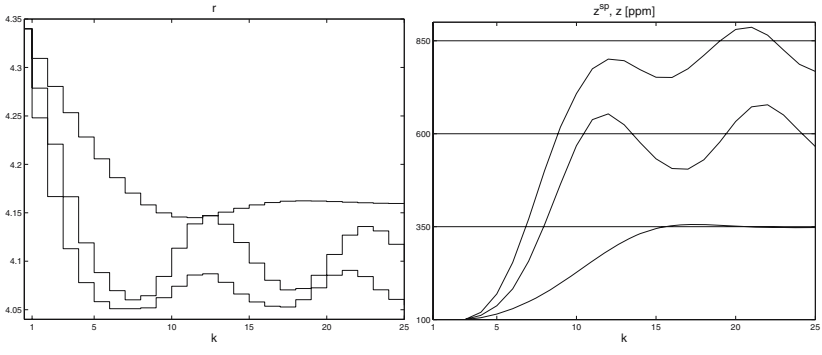


Fig. 2. Simulation results of the ethylene-ethane distillation column with MPC algorithm with linear model valid for "low" impurity level

At first, the MPC algorithms based on two linear models were developed. The first one is valid for "low" impurity level and the resulting control algorithm works well in this region but exhibits unacceptable oscillatory behaviour for medium and big set-point changes as it is shown in Fig. 2. The second linear model captures the process properties for "high" impurity level and the closed-loop response is fast enough for the biggest set-point change but is very slow for smaller ones as it is shown in Fig. 3. Simulation results of the MPC-NPL and MPC with Nonlinear Optimisation (MPC-NO) algorithms with the same neural network model are depicted in Fig. 4. The closed-loop performance obtained in

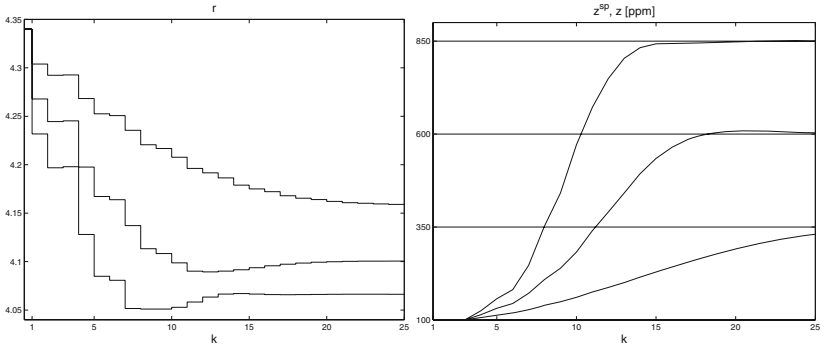


Fig. 3. Simulation results of the ethylene-ethane distillation column with MPC algorithm with linear model valid for "high" impurity level

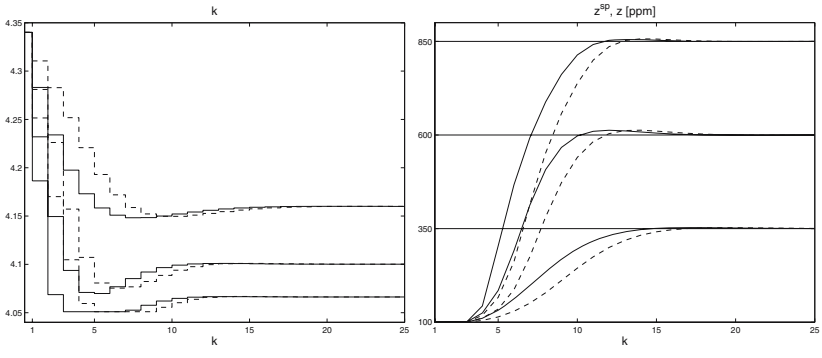


Fig. 4. Simulation results of the ethylene-ethane distillation column with MPC-NPL (dashed line) and MPC-NO (solid line) algorithms with neural network model

the suboptimal MPC-NPL algorithm is close to that obtained in computationally prohibitive MPC-NO approach.

5 Summary

The paper describes the MPC-NPL algorithm with feedforward neural network models and discusses its application to a high-purity ethylene-ethane distillation process. The presented algorithm is able to control effectively highly nonlinear, multivariable processes with constraints.

The emphasis is put on controller's reliability, computational efficiency and closed-loop accuracy. The MPC-NPL algorithm uses on-line only a quadratic programming procedure, the necessity of full nonlinear optimisation is avoided. Moreover, the algorithm, although being suboptimal, in practice gives performance comparable to that obtained in MPC schemes with nonlinear optimisation.

Feedforward neural networks are used as process models. Having excellent approximation abilities, in comparison with popular fuzzy models they do not

suffer from "the curse of dimensionality", which is troublesome in multivariable cases. Furthermore, unlike fundamental models, feedforward neural models have simple, regular structure. Hence, they can be easily incorporated into the MPC-NPL algorithm and efficiently used on-line.

Acknowledgement

This work was partly supported by Polish national budget funds 2005-2007 for science as a research project.

References

1. Haykin, S.: Neural networks – a comprehensive foundation. Prentice Hall. (1999)
2. Henson, M. A.: Nonlinear model predictive control: current status and future directions. Computers and Chemical Engineering. **23** (1998) 187–202
3. Liu, G. P., Kadirkamanathan, V., Billings, S. A.: Predictive control for nonlinear systems using neural networks. International Journal of Control. **71** (1998) 1119–1132
4. Ławryńczuk, M., Tatjewski, P.: A stable dual-mode type nonlinear predictive control algorithm based on on-line linearisation and quadratic programming. Proceedings of the 10^{th} International Conference on Methods and Models in Automation and Robotics. Miedzyzdroje, Poland. (2004) 503–510
5. Ławryńczuk, M.: Nonlinear model predictive control algorithms with neural models of processes (in Polish). PhD thesis. Warsaw University of Technology (2003)
6. Ławryńczuk, M., Tatjewski, P.: A computationally efficient nonlinear predictive control algorithm based on neural models. Proceedings of the 8^{th} International Conference on Methods and Models in Automation and Robotics. Szczecin, Poland. (2002) 781–786
7. Maciejowski, J. M.: Predictive control with constraints. Prentice Hall. (2002)
8. Mahfouf, M., Linkens, D. A.: Non-linear generalized predictive control (NLGPC) applied to muscle relaxant anaesthesia. International Journal of Control. **71** (1998) 239–257
9. Michalska, H., Mayne, D. Q.: Robust receding horizon control of constrained nonlinear systems. IEEE Transactions on Automatic Control. **38** (1993) 1623–1633
10. Morari, M., Lee, J. H.: Model predictive control: past, present and future. Computers and Chemical Engineering. **23** (1999) 667–682
11. Parisini, T., Sanguineti, M., Zoppoli, R.: Nonlinear stabilization by receding-horizon neural regulators. International Journal of Control. **70** (1998) 341–362.
12. Piche, S., Sayyar-Rodsari, B., Johnson, D., Gerules, M.: Nonlinear model predictive control using neural networks. IEEE Control Systems Magazine. **20** (2000) 56–62
13. Qin, S. J., Badgwell, T. A.: A survey of industrial model predictive control technology. Control Engineering Practice. **11** (2003) 733–764
14. Tatjewski, P., Ławryńczuk, M.: Soft computing in model-based predictive control. International Journal of Applied Mathematics and Computer Science. **16** (2006) 101–120.
15. Tatjewski P.: Advanced Control of Industrial Processes. Structures and Algorithms (in Polish). EXIT Academic Publishing House. Warsaw. (2002)

Creativity of Neural Networks

Urszula Markowska-Kaczmar and Katarzyna Czeczot

Wroclaw University of Technology, Poland
urszula.markowska-kaczmar@pwr.wroc.pl

Abstract. In the paper the ability of neural networks in creativity is tested. The creation of new words was chosen as an example task of creativity. Three different approaches based on the neural networks were designed and implemented to perform experiments. From all concerned solutions the best results was produced by the recurrent neural network.

1 Introduction

Human beings always have tended to attribute their features to whole environment. After invention of computers people found another thing to dream of. Books and movies have been created, that described machines becoming more like humans. The truth is that computers can substitute people in many activities that require lots of calculations and can be expressed in an algorithmic way. But the common opinion is that they lack few important things, like flexibility or creativity, and over all intelligence. The purpose of this work is to show that skills of artificial neural networks can change this opinion.

Intelligence is hard to define and even more hard to measure. The taxonomy distinguishes different types of intelligence among of them is cognitive, and creative one [1]. In this work we decided to focus on the last one. The reason was that computers are quite good in mimicking cognitive intelligence. They are able to solve complex problems if they are shown how to do it. But they can not come up with their own solution and even a small change in the problem blocks the ability to obtain the solution with the previous algorithm.

There were several attempts to force neural networks to become creative and innovative, mostly in speech, art and games domains. Chen, in [2], applied a simple recurrent network to create new melodies. He assumed that music is a kind of language, with its own grammar and rules. According to the opinion of the author, results were quite good, although whole melodies were not always satisfactory. In [3] author tried to train a neural networks to predict past forms of English verbs. He used multilayered perceptron taught by supervised growing neural gas method. After 918 epochs of the network training 100% irregular and 99,8% regular verbs were recognized.

The most interesting example is Creativity Machine, developed by S. Thaler. It is neural network creativity paradigm. He applied it to different problems and forced neural networks to design: drinks, toothbrush, melodies and ultra hard materials. His solution is described in details in [4].

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 86–93, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 The Problem Formulation

In this paper as an example of a creativity skill the problem of new words creation was chosen. After a neural network training with a representative set of English words its capability to create new words was tested. This skill can be demonstrated by the neural network if it is able to generate words already existing in the vocabulary, but not included in the training set. Other way to estimate the network progress in this area is to test possibility of pronouncing created words. but this criterion is very subjective.

The ability of neural network to solve this problem can be seen as the capability to discover a kind of rules existing in the language represented by the words included in the dictionary. These rules can be discovered by applying one of the rule extraction method. However, this ability can be very useful in case when we look for some new names, for example the name of a brand-new line of cars or other products. Three methods based on the neural netw ork approaches were designed, implemented and tested to solve this problem. They are described further.

3 Coding of Letters

The first problem to solve was how to represent words for neural networks. At the beginning, each letter in a given word was coded as its ASCII code normalized to the range [-1, 1]. The results were not satisfying. In the next approach, each letter in the alphabet was encoded on twenty six bits. For each letter only one of them was always set to 1 (one matching the position of the letter in the alphabet). The rest of them was set to -1. In the last attempt, the number of bits for one letter was equal 28, where the last two bits were used to encode whether the letter is the vowel or not. The rest of bits has the same meaning as in the previous case.

4 The Examined Neural Networks

The basic network used in the experiments was an *autoassociative memory*, which was realised as the feedforward neural network that was trained to reproduce its inputs. The second one was *Creation by Refinement* [5] and the last one – recurrent neural network [6].

4.1 Autoassociative Memory

The architecture of the multilayered perceptron with one hidden layer was chosen in this case. To train the neural network backpropagation algorithm was used. The main idea was to train a set of English words, and next to perturb its weights according to the following equation with the assumed probability of change:

$$w_{j,i} = w_{j,i} \pm \alpha, \tag{1}$$

where $w_{j,i}$ is the i-th weight of j-th neuron; α defines the intensity of perturbation. In the effect the neural network should react to the known patterns in a nondeterministic way.

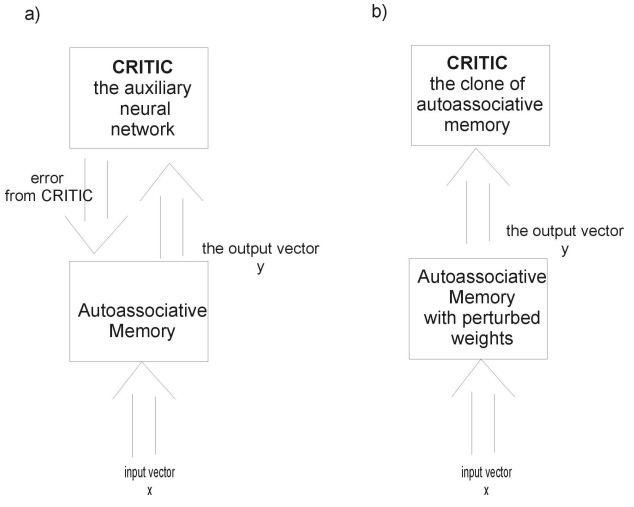


Fig. 1. The methods of retraining of autoassociative memory: a) the scheme of the algorithm *Models and Critics*; b) the clone of autoassociative memory

The problem that occurred was the representation of words of a different length. Neural networks usually have the fixed number of input neurons while language includes words with a various length. To solve this problem three solutions were considered: the first one was that the neurons in the hidden layer exceeding beyond the length of word were not considered. In the second solution the special mark was used to represent an empty places in the assumed number of letters representing a word. In the third solution an ensemble of neural networks was considered. In the ensemble each neural network was responsible for one of the assumed length of a word.

To simplify the problem two dictionaries were created, the first one containing only four-letter words and the other one containing words with a various number of letters. In the creation phase the neural network was stimulated by the pattern in different way. The following three stimulations were considered: all inputs were stimulated in the same way (input vector consists of bits set to 1), the random stimulation, which does not represent a correct word or a stimulation defined by a user.

After a stimulation the output of neural network was produced. In case when the produced output was not satisfying (the sequence of vowels or the sequence of letters have arrived which was difficult to pronounce) the network was retrained. It could be achieved by using human critic who pointed at incorrect letters in the generated sequence or by applying – *Models and Critics* method (Fig. 1a) [7] or the *clone* neural network Fig. 1b).

In case of *Models and Critics* there is only one neural network input vector, which is taken from the associative memory. The *critic* neural network is trained to recognize whether a given string is a word or not. The idea is shown in Fig. 1a. The input vector processed by the associative memory is sent to the *critic*. It generates the evaluation of this input. The errors from the hidden layer of the *critic* neural net-

work are sent to the outputs of associative memory and they are treated as the errors for this layer like in the standard backpropagation method. The last considered approach in the retraining process was the *clone* neural network. It relies on the making a copy of trained autoassociative memory before its perturbation. When as an input to the *clone* network the answer of associative memory is sent the *clone* network has a tendency to repair the obtained sequence to the pattern, which is most similar to the one it was trained.

4.2 Creation by Refinement

The next method of new words formation is based on the neural network creation paradigm – *Creation by Refinement*. The method is described in [5] in details. As the *critic* again the multilayered perceptron trained with backpropagation

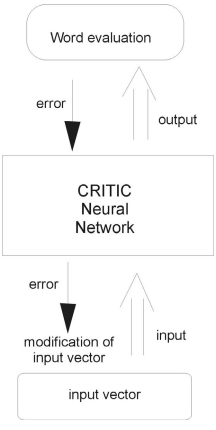


Fig. 2. The scheme of creation phase in the method *Creation by refinement*

algorithm was used. The general scheme of the architecture of this paradigm is shown in Fig. 2. In the creation phase the propagation of pattern in the *critic* neural network as well as backpropagation of error is proceeded like in the standard backpropagation algorithm but without of change of weights. Instead, input vector is being modified and improved on the basis of the calculated and back propagated error. It is done by establishing the position with the biggest error for a given letter and setting this position in the input vector to 1. The stopping criteria of the method is defined as suitably small error of the output of neural network. Finally, decoding of an input pattern is required. During creation phase the neural network is stimulated by random patterns.

4.3 Simple Recurrent Network

The third adapted neural approach is the recurrent neural network [6], called also Elman network (Fig. 3). The neural network takes a single letter as an

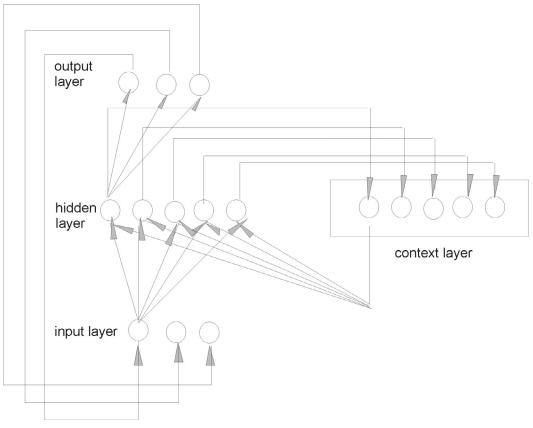


Fig. 3. Simple recurrent neural network

input and produces the next letter of a given word. Words are separated by the additional mark. To encode it three methods were considered: a sequence of 1, a sequence of -1 and the third one relies on adding of twenty seventh letter. The last case requires encoding letters on 27 bits.

The end of learning phase takes place when the neural network properly recognises the sequence of letters. During the creation phase one assumed that a word is generated till the additional mark or tenth letter is generated. Because this method would generate always the same word, a weights perturbation was introduced.

4.4 Evaluation of Created Word

In order to replace human in the evaluation of a created word two mechanisms were considered – the table of probabilities and the *critic* neural network. The table of probabilities contains probability of an appearance of the three-letter sequence in a given language. To evaluate a word, its groups are checked and their values are assumed. *Critic* network is the same multilayered neural network as the network used in *Creation by Refinement* method. Its role here is to evaluate quality of the created word, not to create them. Simply, the creation phase is skipped.

5 Experimental Studies

The aim of the described bellow experiments was an evaluation of the efficiency of the proposed methods in new words creation. In all experiments the network was trained using one of the following training set. The first one composes of 256 of words with the length of 4 letters, generated on the basis of 1000 most popular English words (they were used for all types of considered networks). The second set contains 290 negative patterns (sequences of letters that do not

represent words), generated in the random way (for *critic*-network). The third set includes 214 positive examples of words with a various length (for associative memory).

First, the optimal values of parameters for each method were searched. At the beginning we investigated the influence of the probability and an intensity of the perturbation. For the autoassociative memory the less is the perturbation the more correct strings it generates. When the perturbation coefficient is very high the network is set up very quickly. In case when the perturbation coefficient is too small the network produces mostly trained words. The optimal value was assumed as 0.3. The probability of perturbation set to 40 % gave the best results. Interesting observation was made during the experiments with recurrent neural network. Studying the probability and the intensity of perturbation one can notice that it was very sensible to the probability of perturbation and less to the intensity. With the growth of probability of perturbation from 20 % to 30% sudden disappearance of ability to create of correct words was observed (from 70 % to 30%). The change of the intensity of perturbation does not cause similar phenomenon.

The influence of the number of neurons in the hidden layer was evaluated, as well. The produced words were evaluated by human in this case. The ability to pronounce was the main criterion of evaluation but the number of neighbouring vowels or repeated neighbouring letters were also taken into account. But, it has to be underline that (Table 1) the comparison of the human evaluation and that made on the basis of *clone* network or on the table of probability sometimes differs.

Table 1. The examples of word evaluation made by critic neural networks, human and table of probability in associative memory approach (p- correct, n-incorrect)

word	critic	human	table
cois	p	p	p
roes	p	p	p
coss	p	p	p
cols	p	p	p
cors	p	p	p
coms	p	p	p
comb	p	p	p
rpat	n	p	p
rpab	n	p	n
gnls	p	n	n
onls	p	n	p
rilr	p	p	n
ovar	n	p	n
ovao	n	p	n
ovis	n	p	n
ovis	n	p	n
roeq	n	p	n
ooeq	n	n	n
ooif	n	n	n

For autoassociative memory the best results were obtained with the number of neurons set to 60. The best results were observed with the stimulation of the network by the words from the training set. For the *Creation by Refinement* it was difficult to find the dependence between the number of neurons and the creation of correct words. The results depend on the ability to generalisation of the neural network. For the recurrent neural network the more neurons are in the hidden layer, the more clearly the ability to generate correct words falls down. One can observe that with the growth of the number of hidden neurons the generated words are longer and longer. Independently of the number of hidden neuron the networks have their favourite sequences, that means the strings that are generated more frequently comparing to other strings.

Table 2. The example of words generated by using recurrent neural network, the sign ! stands for additional mark

correct new words							
teen!	she!	one!	her!	mist!	how!	shit!	hood!
too!	sea!	see!	tea!	fist!	seen!	bare!	hey!

In case of autoassociative memory additional experiments were performed testing the results of its retraining using *Models and Critics* method and the *clone* neural network. The first one prevents the network from converging in the local minima and it gives more diverse sequences. The results of both methods were not satisfying. After retraining the network learns by rote instead of recognition of incorrect word. Experimenting with the recurrent neural network, one can noticed that despite it was trained on the four-letter words only, it was able to generate longer sequences and new four-letter words (ie. words that were not contained in the training set). In the Table 2 the examples of three- and four-letter words discovered by the network that exist in English are presented. They were generated with different runs and various settings of parameters. But the fact that the network was able to find them can be seen as a success.

6 Conclusions and Future Plans

On the basis of the performed experiments we can conclude that in the ability of neural network to create new words the best results gave the recurrent neural network. Some ideas still need experimental studies. It applies to another letter encoding and the setting of the perturbation parameters depending on the performance of the network. However creation of new words was chosen as an example of creation, relatively simply the solutions can be used to realise any problem of creation, for example creation of images or an equipment design what will be the aim of the further studies.

References

1. Howard, G.: Frames of Mind. Basic Books (1983)
2. Chen, C.C.J.: Creating melodies with evolving recurrent neural networks. in. In: Proceedings of the 2001 International Joint Conference on Neural Networks. (2001)
3. Westermann, G.: A constructivist neural network learns the past tense of english. In: Proceedings of the GALA '97 Conference on Language Acquisition. (1997) 393–398
4. Thaler, S.: Autonomous ultrahard materials discovery via spreadsheet-implemented neural network cascades. JOM **49** (1997)
5. Levis, J.P.: Shape and texture generation by neural network creation paradigm. (Computer Graphics Laboratory) www.idiom.com/ zilla/Work/texcbrfinal.
6. Elman, J.L.: Learning and development in neural networks: The importance of starting small. Cognition **48** (1993) 71–99
7. Hertz, J., Krogh, A., Palmer, R.: Introduction to the theory of neural computation. Addison Wesley Publishing Company (1991)

Speed Up of the SAMANN Neural Network Retraining

Viktor Medvedev and Gintautas Dzemyda

Institute of Mathematics and Informatics,
Akademijos 4, LT-08663 Vilnius, Lithuania
{viktor.m, dzemyda}@ktl.mii.lt

Abstract. Sammon's mapping is a well-known procedure for mapping data from a higher-dimensional space onto a lower-dimensional one. The original algorithm has a disadvantage. It lacks generalization, which means that new points cannot be added to the obtained map without recalculating it. SAMANN neural network, that realizes Sammon's algorithm, provides a generalization capability of projecting new data. Speed up of the SAMANN network retraining when the new data points appear has been analyzed in this paper. Two strategies for retraining the neural network that realizes the multidimensional data visualization have been proposed and then the analysis has been made.

1 Introduction

Searching for better and suitable data projection methods has always been an integral objective of pattern recognition and data analysis. Such a method will enable us to observe and detect underlying data distributions, patterns, and structures. Feature extraction is the process of mapping the original features into fewer features, which preserve the main information of the data structure. Such visualizations are useful especially in exploratory analyses: they provide overviews of the similarity relationships in high-dimensional datasets that would be hard to acquire without the visualization.

The problem of data projection is defined as follows: given a set of high-dimensional data points, project them to a low-dimensional space so that the result configuration would perform better than the original data in further processing such as clustering, classification, indexing and searching [2], [4]. The visual inspection of the data can provide a deeper insight into the data, since clustering tendencies or a low intrinsic dimensionality in the data may become apparent from the projection. In general, this projection problem can be formulated as mapping a set of n vectors from an d-dimensional space onto an m-dimensional space, with $m < d$.

There exist lots of methods that can be used for reducing the dimensionality of data by projecting high-dimensional datasets as points on a low-dimensional, usually 2D, display. For example multidimensional scaling [11] and its partial case – Sammon's mapping [8] – are such technique, which, according to some predefined error criterion, try to map the original data space into a lower-dimensional space,

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 94–103, 2006.
© Springer-Verlag Berlin Heidelberg 2006

hereby preserving as much as possible the local structure of the original space. The methods differ in what properties of the dataset they try to preserve. The simplest methods, such as the principal component analysis (PCA) [1], are based on linear projection. A more complex set of traditional methods, that are based on multidimensional scaling (MDS) [11], try to preserve the pairwise distances of the data samples as well as possible. That is, the pairwise distances after projection approximate the original distances. Another method, Sammon mapping [8], emphasizes the preservation of local distances relative to the larger ones.

With the development of neural networks, new possibilities for non-linear mapping were created. Among them, Self-Organizing Maps are probably the most well-known [12]. The self-organizing map (SOM), proposed by Kohonen [12], is a class of neural networks that are trained in an unsupervised manner using competitive learning. It is a well-known method for mapping a high-dimensional space onto a low-dimensional one. Usually a mapping onto a two-dimensional grid of neurons is used. The advantages of the SOM are its unsupervised training and that it combines clustering and projection operations. Using the SOM-based approach, we can draw a table with cells corresponding to the neurons. The cells corresponding to the so-called neurons-winners are filled with the order numbers of the analyzed data vectors. Some cells may remain empty. One can make a decision visually on the distribution of the vectors in the n-dimensional space in accordance with their distribution among the cells of the table. However, the table does not answer the question, how much the vectors of the neighboring cells are close in the n-dimensional space. A natural idea comes to apply the distance-preserving projection method to additional mapping of vectors-winners in the SOM. Sammon's mapping may be used for such purposes. In [13] a consequent combination of the SOM and Sammon's mapping examined and grounded experimentally. More sophisticated combination is suggested in [14]. The comparative analysis of the graphical result presentation in the SOM software is presented in [15].

Mao and Jain [6] have suggested a neural network implementation of Sammon's mapping. A specific backpropagation-like learning rule has been developed to allow a normal feedforward artificial neural network to learn Sammon's mapping in an unsupervised way, called SAMANN. In Mao and Jain's implementation the network is able to project new vectors after training – a property Sammon's mapping does not have. A drawback of using SAMANN is that it is rather difficult to train and it is extremely slow.

In this paper, we proposed two strategies for retraining the neural network that realizes multidimensional data visualization.

2 A Neural Network for Sammon's Projection

Sammon's nonlinear mapping is an iterative procedure to project high-dimensional data into low-dimensional configurations. The algorithm was originally proposed by Sammon to assist analysis of structure within volumes of data. It tries to keep the same interpattern distances between points in the low-dimensional

space. Suppose that we have n data points (vectors), $X_i = (x_{i1}, x_{i2} \ldots, x_{id})$, $i = 1, \ldots, n$, in a d-space and, respectively, we define n points, $Y_i = (y_{i1}, y_{i2} \ldots, y_{im})$, $i = 1, \ldots, n$, in a m-space ($m < d$). Without loss of generality, only projection onto a two-dimensional space are studied ($m = 2$). The pending problem is to visualize these d-dimensional vectors X_i, $i = 1, ..., d$ onto the plane R^2. Let d^*_{ij} denotes the Euclidean distance between X_i and X_j in the input space and is defined as $d^*_{ij} = \sqrt{\sum_{k=1}^{d}(x_{ik} - x_{jk})^2}$. d_{ij} denotes the Euclidean distance between the corresponding points Y_i andY_j in the projected space and is defined as $d_{ij} = \sqrt{\sum_{k=1}^{m}(y_{ik} - y_{jk})^2}$. The projection (mapping) error measure E is as follows:

$$E = \frac{1}{\sum_{i,j=1, i<j}^{n} d^*_{ij}} \sum_{\substack{i,j=1 \\ i<j}}^{n} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}}. \tag{1}$$

Sammon's algorithm involves a large amount of computations. Since, $n(n-1)/2$ distances have to be computed for every step within an iteration, the algorithm soon becomes impractical for a large number of vectors. Sammon's algorithm has no generalization capability. In order to project new data, one has to run the program again on pooled data (old data and new data) [3].

Since Sammon's algorithm was primarily designed for data analysis and visualization, one of its major drawbacks is that it does not yield a map or algorithm that might allow one to generalize the transformation to unseen points. The SAMANN for Sammon's nonlinear projection was a neural network training paradigm proposed in [6] to circumvent this problem. The SAMANN is an improvisation on the backpropagation algorithm used for training multi-layer perceptrons. It takes as input, vectors in the d-dimensional space and has in its output layer m-nodes. After training the network, one is able to use it to generalize on previously unseen data.

SAMANN network for two-dimensional projection is given in Fig. 1. It is a feedforward neural network where the number of input units is set to be the feature space dimension d, and the number of output units is specified as the

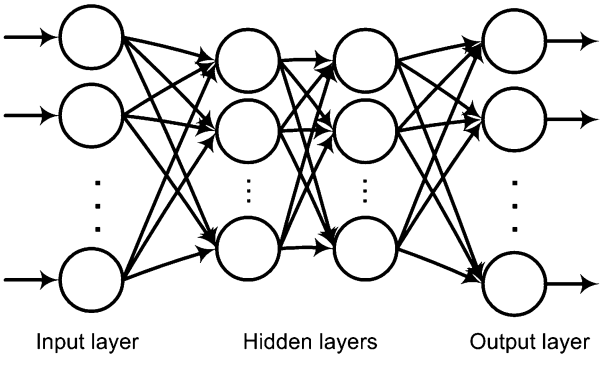


Fig. 1. SAMANN network for two-dimensional projection

extracted feature space dimension m. Mao and Jain [6] have derived a weight updating rule for the multilayer perceptron neural network that minimizes Sammon's stress, based on the gradient descent method. The general updating rule for all the hidden layers, $l = 1, \ldots, L-1$ and for the output layer ($l = L$) is:

$$\Delta\omega_{jk}^{(l)} = -\eta \frac{\partial E_{\mu\nu}}{\partial \omega_{jk}^{(l)}} = -\eta\big(\Delta_{jk}^{(l)}(\mu) y_j^{l-1}(\mu) - \Delta_{jk}^{(l)}(\nu) y_j^{l-1}(\nu)\big), \tag{2}$$

where ω_{jk} is the weight between the unit j in the layer $l-1$ and the unit k in the layer l, η is the learning rate, $y_j^{(l)}$ is the output of the jth unit in the layer l, and μ and ν are two vectors. The $\Delta_{jk}^{(l)}$ are the errors accumulated in each layer and backpropagated to a preceding layer, similarly to the standard backpropagation. A momentum term can be added to the updating rule to speed up the learning process. A momentum component will help to damp the fluctuations around the optimality by encouraging the adjustments to stay in the same direction. The sigmoid activation function whose range is $(0.0, 1.0)$ is used for each unit. However, in the neural network implementation of Sammon's mapping the errors in the output layer are functions of the interpattern distances. The network takes a pair of input vectors at each time in the training. The outputs of each neuron are stored for both points. The distance between the neural network output vectors can be calculated and an error measure can be defined in terms of this distance and the distance between the points in the input space. From this error measure a weight update rule can be derived. Since no output examples are necessary, this is an unsupervised algorithm.

In the SAMANN, all the inter-point distances have to be normalized before being input to the network. This will result clamping of any new data points whose distances to previous data points are large the initial normalizing scale.

The SAMANN Unsupervised Backpropagation Algorithm [6] is as follows:

1. Initialize the weights randomly in the SAMANN network.
2. Select a pair of vectors randomly, present them to the network one at a time, and evaluate the network in a feedforward fashion.
3. Update the weights in the backpropagation fashion starting from the output layer.
4. Repeat steps 2-3 a number of times.
5. Present all the vectors and evaluate the outputs of the network; compute Sammon's stress; if the value of Sammon's stress is below a prespecified threshold or the number of iterations (from steps 2-5) exceeds the prespecified maximum number, then stop; otherwise, go to step 2.

One iteration in our research means showing all pairs of vectors to the neural network once.

The rate, at which artificial neural networks learns, depends upon several controllable factors. Obviously, a slower rate means that a lot more time is spent in accomplishing the learning to produce an adequately trained system. At the faster learning rates, however, the network may not be able to make the

fine discriminations possible with a system that learns more slowly. When the learning rate is very small, the weight adjustments tend to be very small. Thus, if η is small when the algorithm is initialized, the network will probably take an unacceptably long time to converge.

3 Strategies for Retraining of the SAMANN Network

After training the SAMANN network, a set of weights of the neural network are fixed. A new vector shown to the network is mapped into the plane very fast and quite exactly without any additional calculations. However, while working with large data amounts there may appear a lot of new vectors, which entails retraining of the SAMANN network after some time. It has been established that training of the SAMANN neural network requires much calculations, therefore we strive to obtain new weights and a precise data projection as soon as possible.

Let us name the set of vectors that have been used to train the network by the primary set, and the set of the new vectors, that have not been used for training yet, by the new set.

Two strategies for retraining the neural network that visualizes multidimensional data have been proposed and investigated. The first strategy uses all possible pairs of data vectors (both from primary and new datasets) for retraining. The second strategy uses restricted number of pairs of the vectors (vector from the primary set – vector from the new set).

The strategies of the neural network retraining data are as follows:

1. The SAMANN network is trained by N_1 initial vectors, a set of weights ω_1 is obtained, then the projection error $E(N_1)$ is calculated and vector projections are localized on the plane. After the emergence of N_2 new vectors, the neural network is retrained with all the $N_1 + N_2$ vectors, and after each iteration the projection error $E(N_1 + N_2)$ is calculated and the computing time is measured. The new set of SAMANN network weights ω_2 is found.
2. The SAMANN network is trained by N_1 initial vectors, a set of weights ω_1 is obtained, and the projection error $E(N_1)$ is calculated. Since in order to renew the weights ω, a pair of vectors μ and ν is simultaneously provided for the neural network, the neural network is retrained with $2 * N_2$ vectors at each iteration: at each step of training one vector is taken from the primary dataset and the other from the new one. After each iteration the projection error $E(N_1 + N_2)$ is calculated and the computing time is measured. The new set of network weights ω_2 is found.

Tree datasets have been used in the experiments:

1. Iris Dataset (Fisher‘s iris dataset) [9]. A real dataset with 150 random samples of flowers from iris species setosa, versicolor, and virginica. From each species there are 50 observations of sepal length, sepal width, petal length, and petal width in cm. The iris flowers are described by 4 attributes.

2. 300 randomly generated vectors $X_i = (x_{i1}, ..., x_{id}) \in R^d$ (three spherical clusters with 100 vectors each, $d = 5$).
3. Austra dataset (Australian Credit Approval) [10]. The dataset concerns credit card applications. The dataset consists of 690 14-dimensional patterns from two classes.

These three datasets were divided into two parts: the primary dataset and the set of new vectors (new dataset). The first part is used for primary training of the SAMANN network, while the new part together with the primary dataset – for retraining the network.

When projecting data, it is of great importance to achieve good results in a short time interval. In the case of the SAMANN network, it has been observed that the mapping error depends on different parameters. The latest investigations have revealed that in order to achieve good results, one needs to select the learning rate η properly. The experimental investigation [5] shows that the optimal value of the learning rate is in the interval [5, 30]. By selecting such values of the learning rate, the significant economy of the computing time is possible. Smaller values of the learning rate within the interval (0.0,1.0) guarantee a more stable convergence to the minimum of the mapping error.

In the analysis of strategies for the network retraining, a particular case of the SAMANN network was considered: a feedforward artificial neural network with one hidden layer and two outputs ($m = 2$). In each case, the same number ($n_2 = 20$) of neurons of the hidden layer was taken and the set of initial weights was fixed in advance. To visualize the initial dataset, the following parameters were employed: the number of iterations $M = 10000$, the training parameter $\eta = 10$; to visualize the set of new vectors: the training parameter was $\eta = 1$, and the number of iterations depended on the strategy chosen. In the case of sufficiently large d (e.g., $d > 10$), some fluctuations in the dependence of the projection error on the learning time appears (see Austra dataset analysis in Fig. 4). To decrease the fluctuations, $\eta = 0.7$ was selected in case of Austra dataset for visualizing the new points.

In the Iris dataset, 50 vectors were used for retraining. In the randomly generated set, 90 vectors were used for retraining: 30 vectors for the different clusters. In the Austra dataset, 230 vectors were used for retraining.

When calculating, the time of algorithm performance was measured. Figs. 2, 3 and 4 demonstrate the results of calculation. Only the results of retraining the SAMANN network with the new vectors (points) are indicated in the figures.

The first strategy yields good results, however retraining of the network is slow. The best visualization results are obtained by taking points for network retraining from the primary dataset and the new dataset. This is a basis of the second strategy. The second strategy enables us to attain good visualization results in a very short time as well as to get smaller visualization errors and to improve the accuracy of projection as compared to the first strategy (Fig. 3 illustrates this fact best in the experiment with the dataset of randomly generated vectors). The second strategy makes it possible to reduce the duration of calculation because there are considerably less new vectors than in the primary

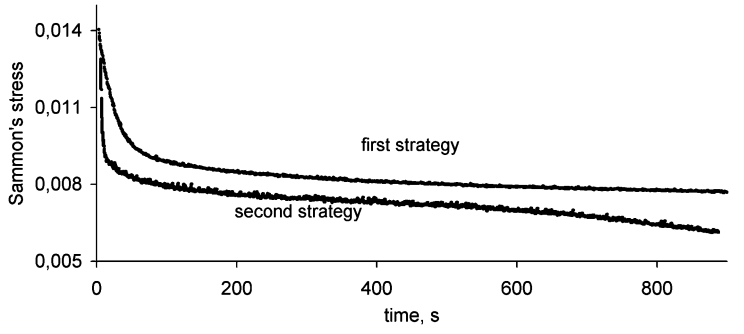


Fig. 2. Dependence of the projection error on the computing time for the Iris dataset

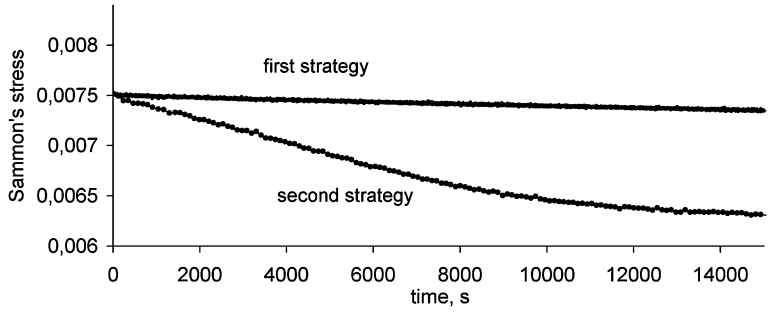


Fig. 3. Dependence of the projection error on the computing time for randomly generated vectors

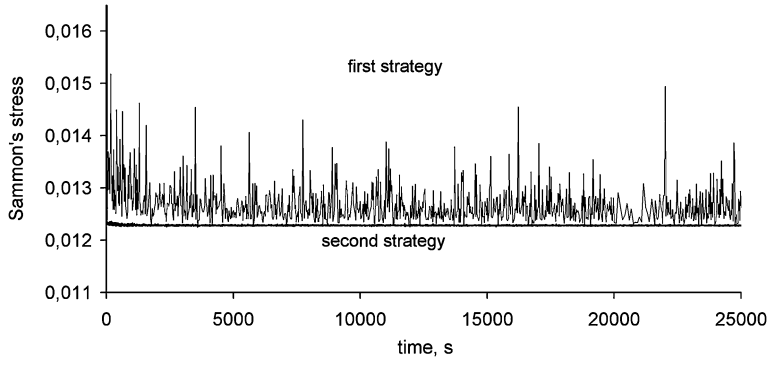


Fig. 4. Dependence of the projection error on the computing time for the Austra dataset

set. Figs. 5, 6 and 7 illustrate mapping results (primary and new datasets) of the Iris dataset, randomly generated vectors and Austra dataset.

The experiments indicate that the second strategy performs better than the first one (see Figs. 2, 3 and 4). Moreover, intense fluctuations of the projection

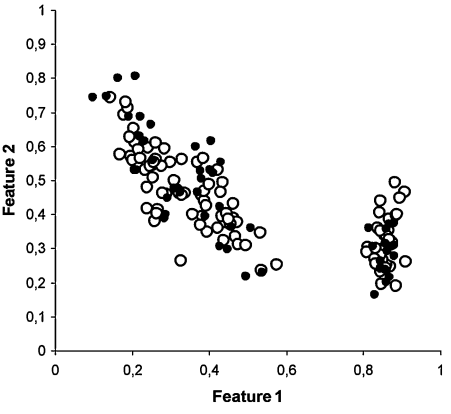


Fig. 5. Mapping results of the Iris dataset

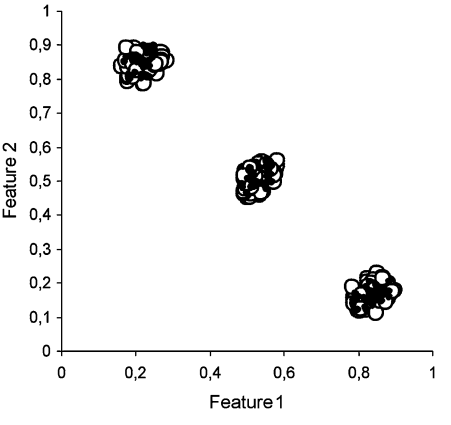


Fig. 6. Mapping results of the randomly generated vectors

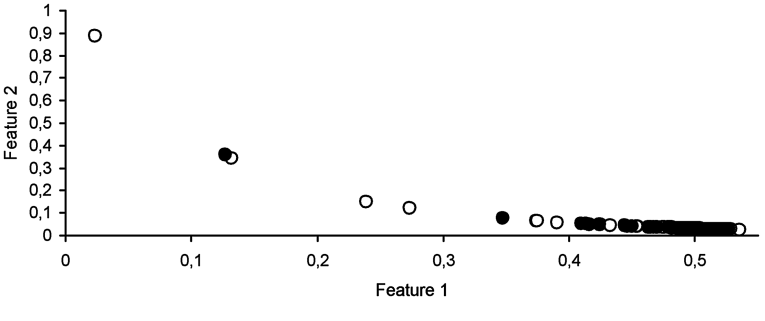


Fig. 7. Mapping results of Austra dataset

error are possible when the first strategy is applied (see Fig. 4). When visualizing the Austra dataset, we observe the small mapping errors from the very beginning of the training. These errors do not decrease significantly in future iterations. This cannot be concluded analyzing results of Figs. 2 and 3. Here we observe the fast stabilization of the projection error in case of the first strategy, while the mapping error by the second strategy decreases. This is an advantage of the second strategy.

4 Conclusions

Speed up of the SAMANN network retraining when the new data points appear has been analyzed in this paper. It is important that the retraining of the neural network is efficient and the training algorithm is faster convergent, therefore effort was put to obtain a new set of weights in a shorter time. Two strategies for retraining the neural network that visualizes multidimensional data have been proposed and investigated. The first strategy uses all possible pairs of data vectors (both from primary and new datasets) for retraining. The second strategy uses restricted number of pairs of the vectors (vector from the primary set – vector from the new set). The experiments both on artificial and real data have shown that it is expedient to take one vector from the primary dataset and the other from the new one at every step of training. This strategy yields smaller projection errors faster.

The experiments lead to the idea of possibility to minimize the SAMANN neural network training time via dividing the training process into two subprocesses: (1) training of the network by a part of the data vectors; (2) retraining of the network by the remaining part of the dataset. In this case, the training set will consist of some subsets. The smaller number of pairs of vectors will be used when training the network by vectors of the subsets. This allows to obtain the similar visualization quality much faster.

References

1. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, Vol. 24 (1993) 417–441, 498–520
2. Jain, A.K. and Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall (1988)
3. Jain, A.K. and Mao, J.: Artificial neural network for nonlinear projection of multivariate data. Proc. IEEE International Joint Conference Neural Network, Vol. 3 (1992) 335–340
4. Jain, A.K., Duin, R., and Mao, J.: Statistical pattern recognition: A review. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 22, No. 1 (2000) 4–37
5. Medvedev, V., Dzemyda, G.: Optimization of the local search in the training for SAMANN neural network. To appear in Journal of Global Optimization, Springer
6. Mao, J. and Jain, A.K.: Artificial neural networks for feature extraction and multivariate data projection, IEEE Trans. Neural Networks, Vol. 6 (1995) 296–317
7. de Ridder, D., Duin, R.P.W.: Sammon's mapping using neural networks: A comparison. Pattern Recognition Letters, Vol. 18 (1997) 1307–1316

8. Sammon, J.J.: A nonlinear mapping for data structure analysis. IEEE Trans. Computer, C-18(5) (1969) 401–409
9. Fisher, R. A.: The use of multiple measurements in taxonomic problem. Annual Eugenics, Vol. 7, Part II (1936) 179–188
10. Australian Credit Approval http://www.niaad.liacc.up.pt/old/statlog/datasets/australian/australian.doc.html
11. Torgerson, W.S.: Multidimensional scaling, I: theory and methods. Psychometrica, Vol. 17 (1952) 401–419
12. Kohonen, T.: Self-Organizing Maps, 3rd Ed. Springer Series in Information Sciences, Vol. 30. Springer (2001)
13. Dzemyda, G.: Visualization of a set of parameters characterized by their correlation matrix. Computational Statistics & Data Analysis, Vol. 36 (1) (2001) 15–30
14. Dzemyda, G., Kurasova, O.: Heuristic approach for minimizing the projection error in the integrated mapping. European Journal of Operational Research, Vol. 171 (2006) 859–878
15. Dzemyda, G., Kurasova, O.: Comparative analysis of the graphical result presentation in the SOM software. Informatica, Vol. 13(3) (2002) 275–286

Application of Neural Networks in Chain Curve Modelling

Andrzej Piegat[1,2], Izabela Rejer[1], and Marek Mikolajczyk[1]

[1] University of Szczecin, 64/66 Mickiewicza Street, 71-101 Poland
[2] Technical University of Szczecin, 49 Zolnierska Street, 71-210 Poland
{andrzej.piegat, izabela.rejer, marek.mikolajczyk}@uoo.univ.szczecin.pl

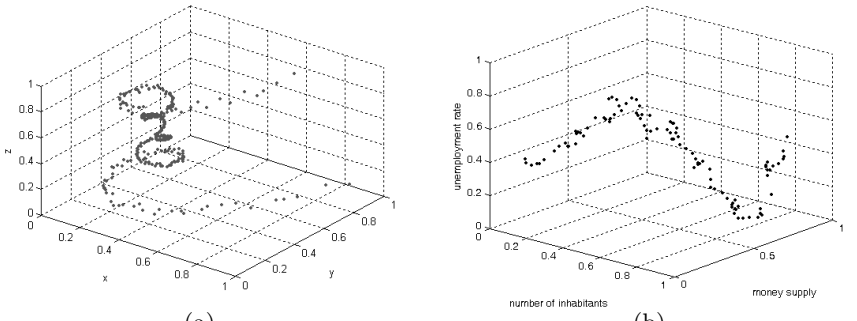
Abstract. A modelling process of an unknown multi-dimensional system is mostly performed with methods which describe the system by a multi-dimensional surface (e.g. neural networks (NNs)). Some systems, however, does not have a surface nature. On the contrary - their behavior resembles multi-dimensional chains. Obviously, as it was proven in numerous applications, always better results can be obtained when the modelling method corresponds to the system nature. Therefore, when a data distribution of an unknown system has a chain characteristic, the system should be also modelled with a chain, not a surface, method. The aim of this article is to present the alternative approach to the modelling process, in which the multi-dimensional model of an unknown system is built on the basis of a set of two-dimensional NNs instead of one multi-dimensional NN. The proposed approach results in a chain multi-dimensional model of an analyzed system.

1 Introduction

The most popular approach to the modelling process of an unknown real system is to describe its behavior by a surface mapping its input variables into output variables. This approach often results in a correct and satisfactory input-output mapping, however, there are some situations when it should be avoided. The most obvious one is when a system of a very large number of input variables is regarded. This is due to the fact that the data distribution in a multi-dimensional input space is often not appropriate to build a surface model. As Gershenfeld [2] proved, almost all data contained in a data set describing a high-dimensional space come from the border of the analyzing space, instead of its center. That means that most analysis of a high-dimensional space are dominated by "border effects" [3].

This tendency can be observed for example when the Monte-Carlo method is applied for generating points in a multi-dimensional region. It can be noticed that a ratio of points situated inside the region to all points tends to one when the region dimension increases. An important fact is that this effect appears relatively quick (for example for 20 dimensions about 90% of data is located on the region border [3]).

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 104–112, 2006.
© Springer-Verlag Berlin Heidelberg 2006



(a) (b)

Fig. 1. a) System which should not be modelled by a surface. b) Economic system of a chain characteristic.

The multi-dimensional space is not the only one in which surface approximation can sometimes result in not satisfactory input-output mapping. Even in 3D space examples of systems which cannot be properly modelled by a surface model can be found (Fig. 1a).

The system shown in Fig.1a is an example of an artificially generated system. However, not only artificial systems can have a chain characteristic. Systems of this kind are also very common in real life. The best example are economic time series systems which are mostly of a chain nature - even when the time variable is not explicitly given. Figure 1b presents an exemplary economic time series system of an unemployment rate in Poland in years 1992-1999 [6].

In situations in which it is not reasonable to approximate an unknown system by a surface model another approaches should be considered. The proposition of the authors of this article is to replace a multi-dimensional surface model, created for instance with a multi-dimensional NN, with a multi-dimensional parametric curve built on the basis of a set of two-dimensional NNs.

2 Parametric Curve Modelling Method in a Multi-dimensional Space

The main idea of the parametric curve modelling method is to build a set of two-dimensional NNs, where each NN describes the behavior of a single variable (input or output) in regard to the known parameter t. These two-dimensional NNs are then assembled together in order to create a multi-dimensional model describing the input-output mapping in the whole space.

The idea of the parametric curve modelling method will be discussed in details via a 3D system described by the following parametric equation:

$$\begin{cases} x = \frac{sin(t)}{t} \\ y = \frac{cos(t)}{t} \\ z = t \end{cases} \tag{1}$$

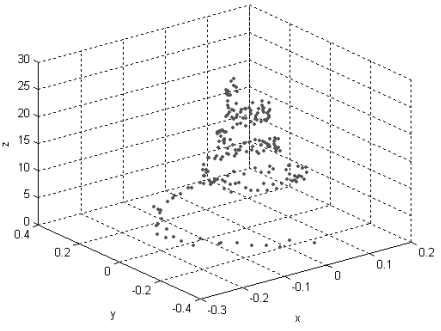


Fig. 2. Interfered data set built with eq. 1

Before starting the parametric curve modelling process the data set of one thousand data points was created. Data was generated from eq. (1) for parameter $t = [\Pi, 8\Pi]$. In order to make the problem more realistic the data was interfered by adding random noise from interval $\langle -10\%, 10\% \rangle$. The interfered data set is shown in Fig. 2. To prepare the data for a NNs training all variables (input and output) were normalized to the interval $\langle 0.1, 0.9 \rangle$, according to eq. (2) [4]:

$$V_n = 0.1 + 0.8 * \frac{V - V_{min}}{V_{max} - V_{min}} \tag{2}$$

Next, a set of two-dimensional NN models was created. The parameters of NNs used in the survey were as follows [4] [5]:

- flow of signals: one-way,
- architecture of connections between layers: all to all,
- hidden layers: one with suitable sigmoid neurons,
- output layer: 1 linear neuron,
- training method: backpropagation algorithm with momentum and changing learning rates,

Table 1. Parameters of NNs described by eq. (3), (4) and (5)

	net. 1 (eq.3)			net. 2 (eq.4)			net. 3 (eq.5)		
i	iw	b	lw	iw	b	lw	iw	b	lw
1	-56.01	55.99	-0.72	55.99	-56.01	3.21	8.51	-6.60	1.94
2	-55.96	49.85	-0.69	57.31	-48.21	0.66	-6.02	0.76	-3.93
3	55.81	-43.68	-0.97	-57.50	41.55	0.82			
4	-55.88	37.42	-1.09	57.38	-35.18	0.96			
5	-56.09	30.84	1.42	-57.45	28.31	1.21			
6	55.86	-24.70	1.71	57.21	-22.01	1.48			
7	-56.52	17.94	2.49	-57.11	14.41	2.81			
8	-54.45	11.07	-4.19	-56.57	9.97	-2.18			
9	56.71	-5.52	-5.53	-55.72	6.77	-3.13			
10	56.01	0.07	2.27	55.99	-0.07	1.92			

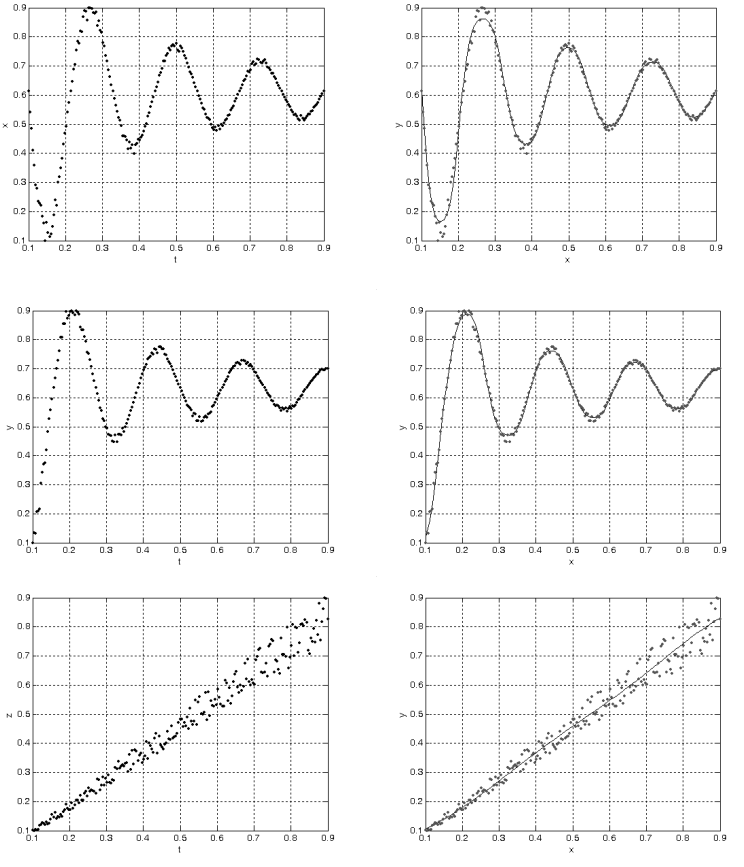


Fig. 3. Two-dimensional models created with NNs

- training aim: to minimize mean absolute error,
- training time: 10000 epoch,
- testing method: visual control.

Models built with NNs of above parameters are shown in Fig. 3 and described by eq. (3), (4) and (5). The parameters (weights and biases) of all three networks are gathered together in Table 1.

$$x = 3.95 + \sum_{i=1}^{10} lw_i * \frac{1}{1 + e^{-(iw_i + b_i)}} \tag{3}$$

$$y = -4.17 + \sum_{i=1}^{10} lw_i * \frac{1}{1 + e^{-(iw_i + b_i)}} \tag{4}$$

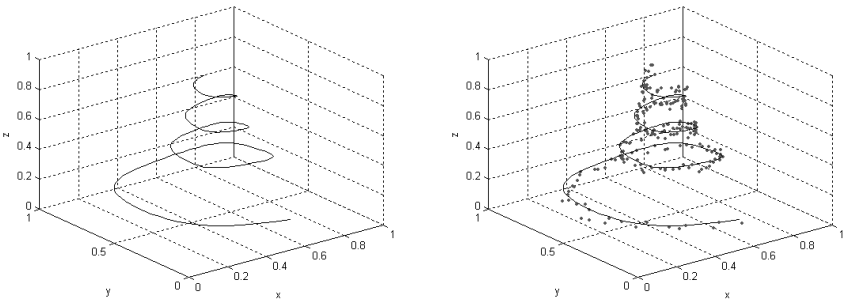


Fig. 4. a) Parametric curve model of the analyzed system. b) Model and data points.

$$z = 0.09 + \sum_{i=1}^{2} lw_i * \frac{1}{1 + e^{-(iw_i + b_i)}} \tag{5}$$

where: iw - input weights vector, lw - output weights vector, b - biases vector

By assembling together eq. (3), (4) and (4) a three-dimensional model of the analyzed system was created (Fig. 4). The approximated accuracy of this model (4.9%) was calculated using MAE (mean absolute error) measure (eq.6) [1], which was equal to 4.9%.

$$error = \frac{\sum_{k=1}^{n} |z_i^* - z_i|}{n} \tag{6}$$

where: z_i^* - empirical values, z_i - theoretical values, n - number of data points. In order to calculate the theoretical values, the 2D models were equally sampled (10000 points) in regard to t dimension. Next, the points were projected onto the x-y space and for each empirical value the closest point (the theoretical value) was found.

3 Surface Model Performance vs Parametric Curve Model Performance

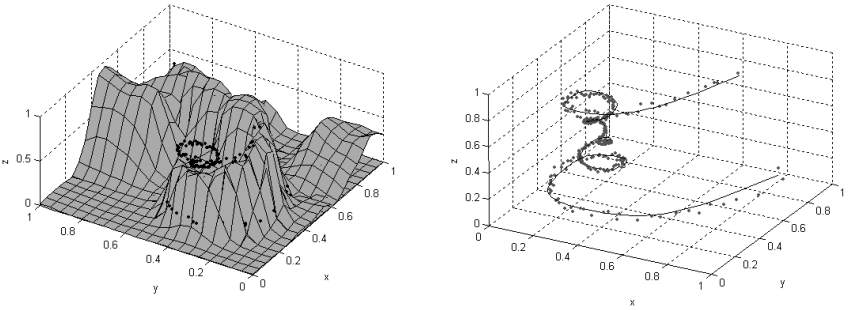
As it was mentioned in the introduction there is a group of systems which cannot be successfully approximated with surface methods. In order to prove this statement two 3D models of the system presented in the introduction (Fig. 1a) were constructed.

The first model was created with use of a neural network. The parameters of the network were mostly the same as described in section 2. The differences were only in the number of hidden neurons (which was set to 10) and the number of training epochs (which was set to 10000). The model error calculated with eq. 6 was equal to 14.57%. The model surface is shown in Fig. 5a.

In order to decrease the model error, a lot of experiments (during which the network parameters were changed) were carried out. Their results are shown in

Table 2. Parameters of the surface models built for the system presented in Fig. 1a

No. of experiment	No. of neurons	No. of epochs	MAE (%)
1	10	10000	14.57
2	20	10000	12.11
3	30	10000	12.09
4	40	10000	11.81
5	10	50000	12.66
6	20	50000	10.53
7	30	50000	10.60
8	40	50000	9.88
9	10	100000	11.40
10	20	100000	11.19
11	30	100000	9.73
12	40	100000	8.55



(a) (b)

Fig. 5. Two models of the system presented in Fig. 1a: a) the surface model, b) the parametric curve model

Table 2. After completing the experiments, it occurred that neither increasing the number of hidden neurons or lengthen the learning time caused a significant decrease in the model error. Even in case of the model of 40 hidden units (trained by 100000 epoches) MAE was still not satisfactory (8.55%). The only effect of adding new hidden neurons was a potential overparametrization of the model.

The second model was built with the method described in the previous section. In contrast to the surface method, the application of the parametric curve modelling method resulted in the model of a very high precision (MAE=2.3%). The model shape is shown in Fig. 5b.

4 Other Advantages of Parametric Curve Models

The previous section described a system which was modelled much more precisely with the parametric curve modelling method than with the surface method. The better performance of the chain model stemmed from the fact that the system

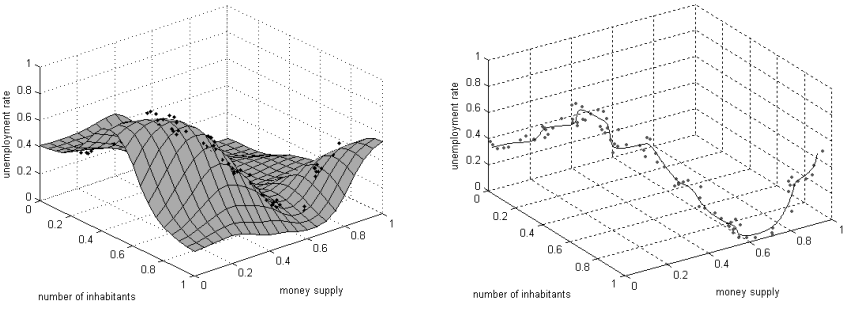


Fig. 6. Two models of the unemployment rate system: a) the surface model, b) the parametric curve model

was described by data of a chain distribution. This relation, however, is not always observed. Sometimes even a system of a strict chain distribution can be described with a higher precision by the surface model than by the chain model. This is due to the fact that the surface model spreads its surface over data fitting its very precisely in input dimensions and balancing only values describing the output dimension. Figure 6 presents two models of a system of a strict chain characteristic which was discussed in the introduction (Fig. 1b). The first model was constructed with a neural network (Fig. 6a) and the second one with the parametric curve modelling method (Fig. 6b). As it can be observed, the first model is a bit more precisely fitted to the data (MAE = 2.18%) than the second one (MAE = 2.28%).

The system illustrated in Fig. 1b presents a chain system which can be described with a little higher rate of precision by the surface model than by the chain model. However, even in such case building the parametric curve model of a system with a chain data distribution is more advantageous. First of all, the process of constructing the parametric curve model allows the researcher to control the overfitting phenomena which is one of the main drawbacks of a multi-dimensional neural network training. Theoretically, there are some methods which can be used in order to control this phenomena (e.g. division of data set into two or more subsets or the cross validation method [4]), however, none of them guarantees that the resulting multi-dimensional model is not an overfitted one. The only method which can give the absolute certainty is a visual control. Obviously this method can be used only when two or three-dimensional problems are considered. In other cases there is no possibility of creating figures simultaneously showing a data distribution and a model shape in the whole analyzed space. Since the proposed method of the parametric curve modelling is based on creating two-dimensional models, it enables the graphic control of the overfitting phenomena regardless of the number of input dimensions.

A two-dimensional analysis of the input-output dependencies has one more essential feature - it requires far less data points for building a non-overfitted model of the multi-dimensional system than a multi-dimensional analysis. This is due to

Table 3. The increase in the number of parameters of the chain and surface model in regard to the increase in the number of system input dimensions

No. of input dimensions	Name of input variable added to model	Neural m. (no. of par.)	Chain m. (no. of par.)	Difference in no. of par.
1	number of inhabitants	16	9	7
2	money supply	21	12	9
3	dollar's rate of exchange	26	17	9
4	rediscount rate	31	22	9
5	number of workers	36	26	10

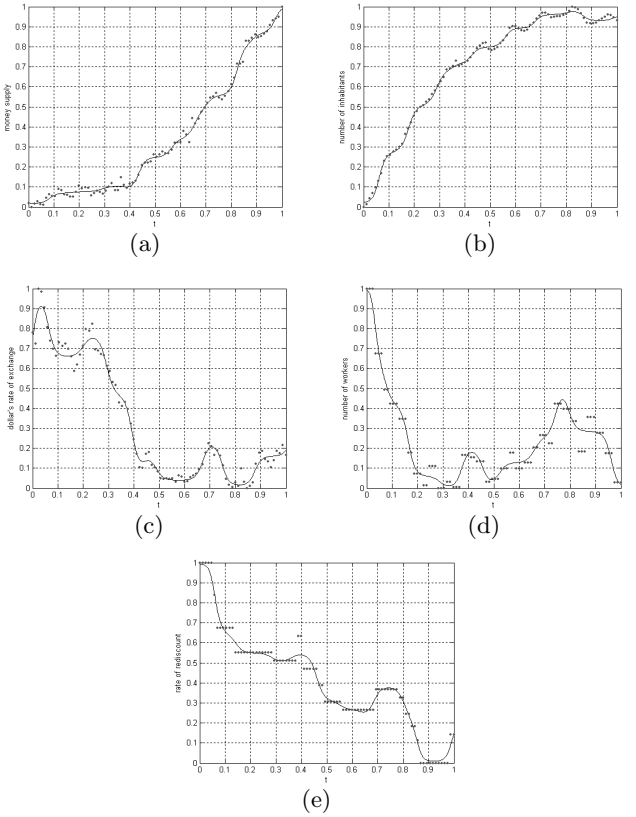


Fig. 7. Parametric curves for input variables of the system of an unemployment rate in Poland in years 1992-1999: a) money supply, b) number of inhabitants, c) dollar's rate of exchange, d) number of workers, e) rediscount rate

the fact that a parametric curve model, constructed by joining two-dimensional models, requires far less parameters than corresponding multidimensional surface model. Table 3 presents the increase in the number of parameters of the chain and surface model in regard to the increase in the number of system input dimensions.

The survey, which results are gathered in Table 3, were also based on the system of unemployment rate in Poland, however, this time five following input variables were used: *money supply, number of inhabitants, dollar's rate of exchange, number of workers* and *rediscount rate*. As it can be observed in Fig. 7 all input variables applied in the survey were of a chain characteristic which means that the data distribution in the whole system was also a chain one.

5 Conclusion

The aim of this article was to present the modelling method based on a multi-dimensional parametric curve built on the basis of two-dimensional NNs. The intention of the authors of the article was also to show that this method, applied in systems of a specific - so called "chain" - nature, can result in a more precise approximation than this obtained with multi-dimensional NNs (or other surface modelling methods).

Naturally, it has to be underlined that the application of the modelling method based on the multi-dimensional parametric curve has one serious limitation - it can be used only when the data sequence is known. Therefore, so far the parametric curve modelling method was successfully applied by the authors of this article only in time series systems where the t parameter could be interpreted as time. The survey aimed at finding a way of applying the parametric curve modelling method in non-time series systems is now carried out by the authors of this articles. The results will be presented soon.

References

1. Aczel, A., "Complete Business Statistics", Richard D. Irwin, Inc., Sydney, Austalia, 1993.
2. Gershenfeld, N., "The Nature of Mathematical Modeling", Cambridge University Press, Cambridge, United Kingdom, 1999.
3. Klesk P., "The method of setting suitable extrapolation capabilities for neurofuzzy models of multidimensional systems", PhD Thesis, Technical University of Szczecin, 2005.
4. Masters T., "Practical Neural Networks Recipes in C++", Academic Press Inc, 1993.
5. Osowski S., "Neural networks for information processing", The publishing house of the Technical University of Warsaw, Warsaw, 2000.
6. Rejer I., "A method of modeling a multi-dimensional system via artificial intelligence methods on the example of an unemployment in Poland", The publishing house of the Szczecin University, Szczecin, 2003.

RBF Nets in Faults Localization

Ewaryst Rafajłowicz

Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50 370 Wrocław, Poland
raf@immt.pwr.wroc.pl

Abstract. The task of faults localization is discussed in a model-free setting. As a tool for its solution we consider a multiclass pattern recognition problem with a metric in the label space. Then, this problem is approximately solved, providing hints on selecting appropriate RBF nets. It was shown that the approximate solution is the exact one in several important cases. Finally, we propose the algorithm for learning the proposed RBF net. The results of its testing are briefly reported.

1 Introduction

Problems of detection and localization of faults have been the subject of intensive research in recent years (see monograph [9] for a wide perspective and extensive bibliographies). Roughly speaking, two main approaches to these problems can be distinguished: a model based and a model free approach, the latter being frequently motivated by hypothesis testing or by the theory of pattern recognition. Also the approach considered here relies on this theory. Our aim is to discuss a multi-class pattern recognition problem, which differs from the classical, statistical pattern recognition problem in that a metric (or more generally, a topology) is defined in the space of outputs (labels). In other words, in apposite to the classical setting, in which labels of the classes are arbitrary and unordered, we consider the family of problems with class labels that have neighbors, which are closer or further in a specified metric. There are many possible applications where a metric on labels is defined in a quite natural way:

1) Consider a production process of resistors, which are classified to 1%, 5%, 10% and 20% classes of their accuracy. It is reasonable to attach a larger loss when a resistor from a 5% class is recognized as that of a 20% class, than if it is recognized as coming from a 10% class of accuracy. In fact, the above example is typical for every quality control procedure that classifies products to more than two classes, which can be ordered according to the increasing quality of production.
2) Consider a fault localization problem on a sufficiently fine grid. For simplicity assume that the grid is two dimensional with the nodes numbered as (i, j), say. Suppose that a fault is located at (i^0, j^0) node. It is reasonable to attach a smaller loss if the fault is recognized to be at the position $(i^0 + 1, j^0 + 1)$ than if it is recognized to be at node $(i^0 + 10, j^0 - 13)$, say.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 113–122, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Problems similar to 2) arise in the localization of targets as in military or medical applications (the target is, e.g., a group of cancer cells).

The above examples lead to considering pattern recognition problems with loss functions, which measure a kind of distance between labels of classes. The Euclidean metric is of special importance, since we are able to find simple decision rules, as is shown in the rest of the paper. There are two more important reasons for considering quadratic loss functions instead of zero-one loss, which has been mainly considered in the literature so far (see [4], [2] for extensive bibliographies).

A) It is well known that most of the existing pattern recognition methods can change a decision, if the learning sequence is enlarged by even one example. This feature is, to some extent, unavoidable, but in multiple class classification problems with unordered class labels such a change can have far reaching consequences. In opposite, if we are able to introduce a metric on labels, then this type of changing decisions will result in selecting a decision, which is close to the previous one. Thus, a kind of robustness is built-in into the process of training.
B) For many years a widespread opinion on multiple class recognition problems was that they are not worth of special attention, since a multiple class problem easily reduces to a sequences of dichotomies. On theoretical ground this is, of course, the truth, but in practice, multiple class problems are known to cause troubles (see, [1], [5], [7]).

It should be mentioned that different metrics were suggested in the literature (see [12], [13] and the bibliographies cited therein), but they were introduced in the pattern space, while here, we consider metrics in the space of labels.

2 Problem Statement

Consider firstly the statistical pattern recognition problem in the standard setting (see, e.g., [2] or [4]). Let $X \in R^d$ be a random vector, representing a pattern, which is a member of one of I classes, labelled as $1, 2, \ldots, I$. Pair (X, i) is a random vector representing a pattern and its correct classification, which is unknown for a new pattern X to be classified, but we usually have also a learning sequence $(X^{(k)}, i^{(k)})$, $k = 1, 2, \ldots, n$ of observed patterns $X_k \in R^d$ and their correct classifications $i^{(k)} \in \{1, 2, \ldots, I\}$. Denote by $0 \leq q(i) \leq 1$, a priori probability that X comes from i-th class, $i = 1, 2, \ldots, I$, $\sum_{i=1}^{I} q(i) = 1$.

For simplicity of the exposition assume the existence of probability densities $f(x|i)$, which describes the conditional p.d.f. of X, provided that it was drawn from i-th class. The next ingredient of the problem setting is a loss function, $L(i, j)$ say, which attaches loss $L(i, j)$ if a pattern from i-th class is classified to j-th class. The aim is to find (or to approximate from a learning sequence) a decision function $\Psi(X)$, which specifies a label of the class for X and such that it minimizes the expected loss given by:

$$R(\Psi) \;=\; E_X \left[\sum_{i=1}^{I} L(i, \Psi(X))\, P(i|X) \right] \;, \tag{1}$$

where E_X denotes the expectation w.r.t. X, while $P(i|X)$ is the a posteriori probability that observed pattern X comes from i-th class. In other words, $P(i|X = x)$ is the conditional probability that label i is correct classification of a given pattern $X = x$. According to the Bayes rule, $P(i|X = x)$ is given by

$$P(i|X = x) \;=\; \frac{f(x|i)\,q(i)}{f(x)}, \quad i = 1, 2, \ldots, I, \qquad f(x) \stackrel{def}{=} \sum_{l=1}^{I} f(x|l)\,q(l)\;. \quad (2)$$

Our aim is to minimize the risk $R(\Psi)$, provided that the minimizer $\Psi^*(x)$, say, is a measurable function. It is well known, that in order to minimize $R(\Psi)$ it suffices to minimize the conditional risk

$$r(\psi, x) \stackrel{def}{=} \sum_{i=1}^{I} L(i, \psi)\,P(i|X = x) \quad (3)$$

with respect to ψ, which is a real variable taking values in the range of $\Psi(x)$, while x is treated as a parameter. According to the above statement, the optimal decision rule $\Psi^*(x)$ is obtained as $\Psi^*(x) \;=\; \arg\min_{\psi}\; r(\psi, x)$ for all $x \in R^d$ in the range of X and it is called the Bayes classifier.

3 Approximate and Exact Decision Rules for Quadratic Loss Function

Now, we are at the position to put our idea into the above general framework. Firstly, we slightly generalize the notion of class labels admitting i and j to be vectors. For simplicity of exposition we work with two-dimensional labels, i.e., $i = (i_a, i_b)$, where $i_a \in \{1, 2, \ldots, I_a\}$, $i_b \in \{1, 2, \ldots, I_b\}$, say, and the total number of classes is $I = I_a \cdot I_b$. Analogously, our decisions are also labelled in two dimensions $j = (j_a, j_b)$, where $j_a \in \{1, 2, \ldots, I_a\}$, $j_b \in \{1, 2, \ldots, I_b\}$. Thus, $\Psi(x)$ is of the form (j_a, j_b).

In opposite to the classical stream of research, in which it is customary to select $L(i, j)$ as $0 - 1$ loss, we choose the quadratic loss function

$$L(i, j) \;=\; (i_a - j_a)^2 + (i_b - j_b)^2.$$

Now, the problem of finding the Bayes classifier $\Psi^*(x) = (j_a^*(x), j_b^*(x))$ reads as follows

$$(j_a^*(x), j_b^*(x)) = \arg\min_{j_a, j_b} \sum_{i_a=1}^{I_a} \sum_{i_b=1}^{I_b} \left[(i_a - j_a)^2 + (i_b - j_b)^2\right] P((i_a, i_b)|X = x)\;, \quad (4)$$

where the minimization in (4) is carried out for $j_a \in \{1, 2, \ldots, I_a\}$ and $j_b \in \{1, 2, \ldots, I_b\}$.

From the optimization theory point of view, (4) is the discrete optimization problem, which can be quite (or even prohibitively) laborious when I_a and I_b are

large, e.g., each of the order 1000, as we can meet when (i_a, i_b) are coordinates of pixels in an image. For this reason we later relax the problem and treat j_a and j_b in (4) as real variables. Thus, we can now use the standard way of finding the minimizers of (4) and then to round them to the nearest integers. These steps yield

$$j_a^* \;=\; \mathrm{Round}_a \left[\sum_{i_a=1}^{I_a} \sum_{i_b=1}^{I_b} i_a \, P((i_a, i_b)|X = x) \right] \tag{5}$$

$$j_b^* \;=\; \mathrm{Round}_b \left[\sum_{i_a=1}^{I_a} \sum_{i_b=1}^{I_b} i_b \, P((i_a, i_b)|X = x) \right] \;, \tag{6}$$

where $\mathrm{Round}[.]_a$ and $\mathrm{Round}[.]_b$ denote usual rounding to the nearest integer with the exception that $\mathrm{Round}_l[t] = 1$ for $t \le 1$, $l = a, b$ and $\mathrm{Round}_a[t] = I_a$ for $t \ge I_a$, $\mathrm{Round}_b[t] = I_b$ for $t \ge I_b$. In deriving (5) and (6) we have used the fact that for every x

$$\sum_{i_a=1}^{I_a} \sum_{i_b=1}^{I_b} P((i_a, i_b)|X = x) = 1 \;.$$

These formulas can further be simplified if labels i_a and i_b are conditionally independent given $X = x$.

Formally, this approach provides only an approximate solution, but when $P((i_a, i_b)|X = x)$ are smooth functions of i_a and i_b, then one can expect that the risk is not much larger than for the optimal solution. This hope is supported by the following results.

Theorem 1. *Decision rules (5), (6) are exact minimizers of the conditional risk (4) in the following cases.*

1. *One can identify the class labels with a set* $\{1, 2, \ldots, I\}$.
2. *The classes are separable, i.e., there exist subsets* $\mathcal{R}_m \subset R^d$, $m = 1, 2 \ldots, M$, *which are mutually disjoint, covering all* R^d *space and such that each label is attached to one and only one* $\mathcal{R}_m$.
3. *The class labels form the grid*[1] $\{1, 2, \ldots, I_a\} \times \{1, 2, \ldots, I_b\}$ *and their components are conditionally independent, given* X, *i.e.,*

$$P((i_a,\, i_b)|X) \;=\; P(i_a|X)\, P(i_b)|X) \;. \tag{7}$$

Before sketching the proof, a few remarks are in order.

- The above theorem provides the list of sufficient conditions for decision rules (5), (6) to be exact minimizers of (4). If none of these conditions is fulfilled, it does not necessarily mean that (5), (6) are not exact minimizers of (4). In other words, this list can be extended.
- In case 1 we do not impose any additional restrictions on probability distributions of patterns.

[1] For simplicity we write the grid of labels as two-dimensional, but the results can easily be generalized.

- The geometry of labels is not restricted in case 2.
- If the probability distributions of patterns possess densities $f(x|i)$, $i = 1, 2, \ldots d$, then in case 2 the requirement that $\mathcal{R}_m \subset R^d$, $m = 1, 2 \ldots, M$, are mutually disjoint can be weakened to the condition that they are either disjoint or just touching, i.e., they can have common boundary points or surfaces in R^{d-1}. The result still holds, since then the probability that pattern X is drawn from the common boundary surface equals zero.
- One may attempt to prove the Bayes risk consistency of the empirical counterparts of (5), (6) using the results from [6] as guidelines.

Assertion 1 follows from the fact that for univariate and ordered labels the approximate minimizer (5) is closer to that integer for which

$$\sum_{i=1}^{I} (i-j)^2 P(i|X=x) \tag{8}$$

is smaller, since this function is quadratic in j.

To prove assertion 2 note that for separable classes if $x \in \mathcal{R}_m$, then $P(i|X = x) = 1$ only for $i = m$ and $P(i|X = x) = 0$ for all other classes. Thus, the loss function equals to $(m-j)^2$ and it has the same same minimizer as arising from the rounding of the a posteriori mean.

Under condition (7) the quadratic loss in (4) reduces to the sum of expressions which have the form (8). Thus, assertion 3 follows by applying assertion 1 several times.

4 RBF Net for Faults Localization

Formulas (5) and (6) form the basis for constructing an RBF net for determining the positions of faults. The classical way of converting decision rules (5) and (6) into empirical decision rules, which are based on the learning sequence instead of formulas for $P(i|X = x)$, is to estimate these functions from $(X^{(k)}, i^{(k)})$, $k = 1, 2, \ldots, n$ with the aid of (2). This way provides the so called plug-in decision rules (see [4]) and it requires a long learning sequence when $f(x|i)$, $i = 1, 2, \ldots, I$ are estimated by nonparametric methods.

If I_a and I_b are large we cannot follow the plug-in method, since it would require estimating hundreds of densities. It seems that a reasonable alternative is to estimate each $f(x|i)$ by an element of parametrized family of densities such as Gaussian or elliptically contoured densities and to estimate their parameters from the learning sequence. In some cases one can have good reason to select such a family of densities. Otherwise, we are forced to select such a family arbitrarily, e.g., from the class of radial basis functions (see [16] for the results on their approximation abilities and [11], [15], [8], [10], [14] for recent results in related directions). Below we follow the former approach, but we insist on selecting positions of RBF centers as if they were mean vectors of densities corresponding to these RBF's.

Denote by $K(t) \geq 0$, $t \in R$ a kernel of RBF's, which is such that

$$\int_{-\infty}^{\infty} K(t) = 1 \ , \quad \int_{-\infty}^{\infty} t\, K(t) = 0 \ ,$$

i.e., K fulfils the same conditions as a density function. Instead of $f(x|i)$ we shall use $K_h(||x - c(i)||)$, where $K_h(t) \stackrel{def}{=} h^{-1}K(t/h)$, $h > 0$ a smoothing parameter and $||.||$ is a norm in R^d. $c(i) \in R^d$, $i = 1, 2, \ldots, I$ are centers of RBF's. In more details, for two-dimensional class labels $i = (i_a, i_b)$ we shall write $c(i_a, i_b)$. Also h may depend on (i_a, i_b), but later it is not displayed. Now, $f(x|(i_a, i_b))$ is roughly estimated as $K_h(||x - c(i_a, i_b)||)$. Inserting these expressions to (2) and (5), (6) we obtain the following plug-in decision rules

$$\widehat{j}_a \ = \ \mathrm{Round}_a \left[\frac{1}{\hat{f}(x)} \sum_{i_a=1}^{I_a} \sum_{i_b=1}^{I_b} i_a\, \hat{q}(i_a, i_b)\, K_h(||x - c(i_a, i_b)||) \right] \ , \tag{9}$$

$$\widehat{j}_b \ = \ \mathrm{Round}_b \left[\frac{1}{\hat{f}(x)} \sum_{i_a=1}^{I_a} \sum_{i_b=1}^{I_b} i_b\, \hat{q}(i_a, i_b)\, K_h(||x - c(i_a, i_b)||) \right] \ , \tag{10}$$

where $\hat{q}(i_a, i_b)$ is an estimate of a priori probability that a pattern comes from class (i_a, i_b), while

$$\hat{f}(x) \stackrel{def}{=} \sum_{i_a=1}^{I_a} \sum_{i_b=1}^{I_b} \hat{q}(i_a, i_b)\, K_h(||x - c(i_a, i_b)||) \ .$$

We have also added hats over j_a and j_b in order to indicate that these are decisions, which are based on the learning samples and they can be different than j_a^* and j_b^*. Examining the nominators and the denominators in (9) and (10), one can easily notice that their structure is exactly the same as the classical RBF nets. Thus, for d-dimensional label space we need $2\,d + 1$ RBF nets to be tuned. There is, however, a difference in tuning between usual RBF's and our case, since parameters $\hat{q}(i_a, i_b)$ and $c(i_a, i_b)$ appear in all these nets and can not be tuned independently. Fortunately, these parameters have their own statistical interpretation, which makes their selection easier.

Namely, we propose to select them as follows. Having the learning sequence $(X^{(k)}, (i_a^{(k)}, i_b^{(k)})$, $k = 1, 2, \ldots, n$ at our disposal, we put

$$\hat{q}(i_a, i_b) = n(i_a, i_b)/n \ , \quad i_a = 1, 2, \ldots, I_a \ , \quad i_b = 1, 2, \ldots, I_b \ , \tag{11}$$

where $n(i_a, i_b)$ denotes the number of observations with the label (i_a, i_b) in the learning sequence. Clearly, $\sum_{i_a=1}^{I_a} \sum_{i_b=1}^{I_b} n(i_a, i_b) = n$. Analogously,

$$c(i_a, i_b) \ = \ n^{-1}(i_a, i_b) \sum_{k=1}^{n} X^{(k)}\, \chi(k, (i_a, i_b)) \ , \tag{12}$$

where $\chi(k,(i_a,i_b))=1$ if in the learning sequence observation $X^{(k)}$ has the label (i_a,i_b) and $\chi(k,(i_a,i_b))=0$, otherwise. In other words, $c(i_a,i_b)$ is the mean vector of those observations, which were assigned to class (i_a,i_b).

One can express formulas (11) and (12) in the recursive form. These formulas can either be used directly for tuning the above nets, as in the example below, or one can treat them as good starting point for a direct search of centers and weights. Smoothing parameter(s) h (or $h(i_a,i_b)$) can be selected in the standard way, i.e., by minimizing an estimate of the expected loss, with the same care in estimating the loss as when RBF's are used for approximating surfaces, e.g., splitting the learning sequence and using cross-validation.

5 Detection of Hot Spots – Example

As is known, the presence of unexpected heat sources in electronic devices indicates faults (or potential faults). Difficulties in their localization stem from the fact that the temperature can usually be measured in locations, which are far from possible positions of faults. Additionally, a simple linear model of heat transfer (as the one used here for simulating training data) is not adequate. In such cases it is reasonable to train the net in locating possible faults, basing on indirect measurements.

In our simulation experiment observations were generated using the solution of the linear heat equation, considered in the steady state. Four temperature

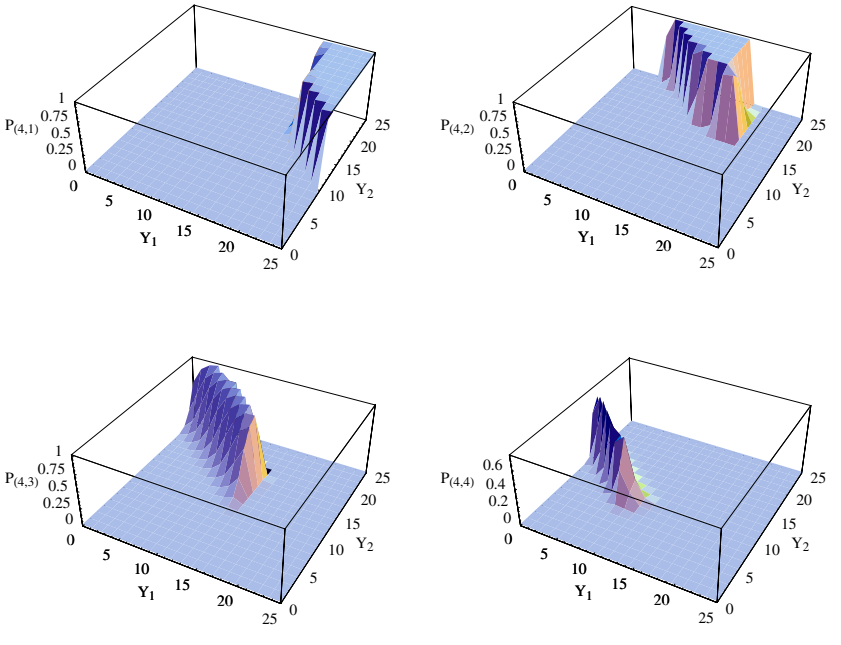


Fig. 1. A posteriori probabilities of detecting the second component of fault labels vs observations Y_1, Y_2 (see example)

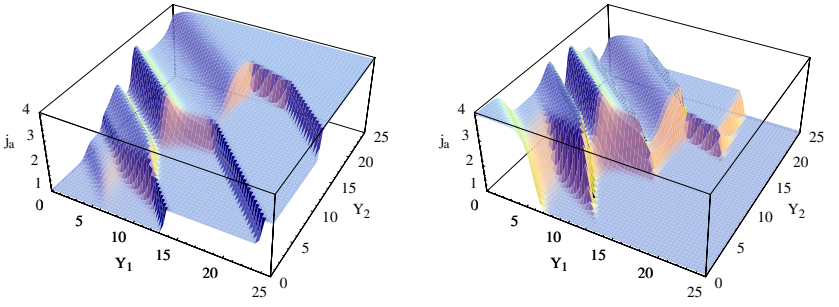


Fig. 2. The decisions on fault locations j_a (*left panel*) and j_b (*right panel*) vs observations Y_1, Y_2 (see example)

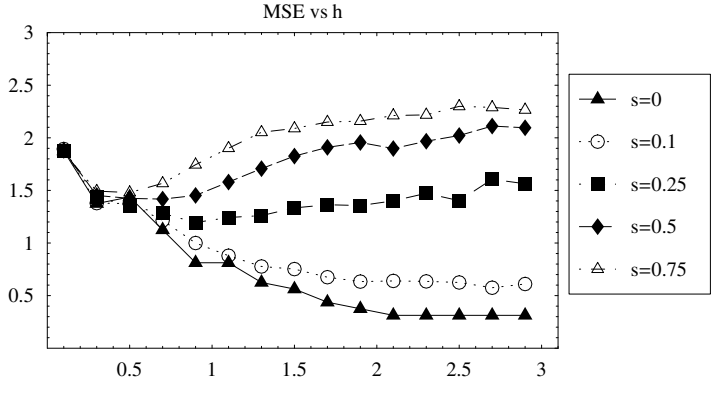


Fig. 3. Dependence of the empirical loss on smoothing parameter h (see example)

sensors were located in positions, which were far from possible positions of heating sources. Gaussian noises were added to the measurements. A parasite heat source could occur at four different locations with four different intensities. We have implemented decision rules (9), (10), together with formulas (11), (12) for learning.

Two temperature sensors were used at the first stage of our simulations in order to visualise decision surfaces. Assuming that the temperature is measured with Gaussian errors[2] and selecting kernel K also as the normal distribution, we can additionally visualize a posteriori probabilities as a function of the observations Y_1 and Y_2, say. These probabilities are shown in Fig. 1 for four different locations of the parasite heat source, keeping the source intensity constant. As one can notice from these figures, the network can clearly distinguish between positions of the hot point. Note, however, that this is an idealized case with known errors distribution with small variances (see below for more realistic simulations). The corresponding decision surfaces are shown in Fig. 2. Rounding the

[2] We underline that knowledge of measurement errors distribution is not neceessary for functioning of the proposed neural net. The normality assumption was made only for the purposes of simulations.

vertical coordinates provides decisions on the coordinates and the intensities of the parasite heat source.

In order to demonstrate dependence of the empirical loss on h we took advantage of simulations when we know the locations of faults. This time, four temperature sensors were used. For simplicity the same parameter h was applied at each RBF center. The dependence, obtained by averaging over 100 noise realizations, is shown in Fig. 3 for various dispersion of measurement errors $s \geq 0$. With the exception of small s, the existence of a flat area of loss minimizing h is clearly visible. In practice, one may hope to locate such areas using cross-validation techniques. The level of averaged loss is also satisfactory, since for reasonable chosen h it is less than or equal to 2, which means that the most frequently committed decision errors were not larger than one grid step in each direction.

6 Concluding Remarks

The algorithms of training a RBF type neural network to the task of fault location are proposed and – to some extent – investigated. They are based on the solution of the pattern recognition problem with a metric in the label space. It seems that this problem has a wider potential utility than described here and it deserve further attention.

Acknowledgement. This work was partly supported by the grant of the Polish Council for Scientific Research under a grant ranging from 2002 to 2005 and by the Polish Foundation for Science.

References

1. Allwein, A., Schapire, R., Singer, Y.: Reducing multiclass to binary: A unifying approach for margin classifiers. J. Machine Learning Research, **1** (2000) 113-141
2. Bishop, C.: Neural Networks for Pattern Recognition. Oxford Univ. Press, (1995)
3. Devroye, L., Györfi, L.: Nonparametric Density Estimation. The L_1 View. Wiley, New York (1985)
4. Devroye, L., Györfi, L., Lugosi, G.: Probabilistic Theory of Pattern Recognition. Springer-Verlag, New York, (1996)
5. Dietterich, T., Bakiri, G.: Solving Multiclass Learning Problems via Error-Correcting Output Codes. J. Artificial Intelligence Research **2** (1995) 263-86
6. Greblicki, W., Pawlak, M.: Necessary and Sufficient Conditions for Bayes Risk Consistency of Recursive Kernel Classification Rule. IEEE Trans. Information Theory, **33** (1987) 408–412
7. Hastie T., Tibshirani, R.: Classification by Pairwise Coupling. The Annals of Statistics **26** (1998) 451-471
8. Karayiannis, N.B., Randolph-Gips, M. M.: On the Construction and Training of Reformulated Radial Basis Function Neural Networks, IEEE Trans. on Neural Networks **14** (2003) 835–846.

9. Korbicz J., Kocielny J.M., Kowalczuk Z., Cholewa W. (eds.): Fault Diagnosis. Models, Artificial Intelligence, Applications. Springer-Verlag, Berlin Heidelberg New York (2004)
10. Krzyżak A., Skubalska-Rafajłowicz E.: Combining Space-Filling Curves and Radial Basis Function Networks. Artificial Intelligence and Soft Computing. Leszek Rutkowski et all (eds.): Lecture Notes in Artificial Intelligence, Vol. 3070. Springer-Verlag, Berlin Heidelberg New York (2004) 229–234
11. Pawlak, M., Siu, D.: Classification with Noisy Features. Advances in Pattern Recognition **1451** (1999) 845–852
12. Skubalska-Rafajłowicz E.: Pattern Recognition Algorithms Based on Space-Filling Curves and Orthogonal Expansions. IEEE Trans. Information Theory **47** (2001) 1915–1927
13. Skubalska-Rafajłowicz, E., Krzyżak, A.: Fast k-NN Classification Rule Using Metric on Space-Filling Curves. Proceedings of the 13th International Conference on Pattern Recognition, Vienna (1996) Vol. 2 121–125
14. Skubalska-Rafajłowicz, E.: Data Compression for Pattern Recognition Based on Space-Filling Curve Pseudo-Inverse Mapping. Nonlinear Analysis, Theory, Methods and Applications **47** 315–326
15. Skubalska-Rafajłowicz, E.: RBF Neural Network for Probability Density Function Estimation and Detecting Changes in Multivariate Processes, Artificial intelligence and soft computing - ICAISC 2006. 8th Int. Conf. Proceedings (in print).
16. Xu L., Krzyżak A., Yuille A.: On Radial Basis Function Nets and Kernel Regression: Statistical Consistency, Convergence Rates and Receptive Field Size. Neural Networks **4** (1994) 609–628

A Hypertube as a Possible Interpolation Region of a Neural Model

Izabela Rejer and Marek Mikolajczyk

University of Szczecin, 64/66 Mickiewicza Street, 71-101 Poland
izabela.rejer@uoo.univ.szczecin.pl,
marek.mikolajczyk@uoo.univ.szczecin.pl

Abstract. The aim of this article is to present a method which can be applied to determine interpolation region of a multidimensional neural model. The method is based on the parametric curve modelling. The idea of it is to surround the parametric curve model with the hypertube covering most of the data points used in a neural model training. The practical application of the method will be shown via a system of an unemployment rate in Poland in years 1992-1999.

1 Introduction

A characteristic feature of a neural model is that it not only covers the region of data points used in the training process but also spreads out over the whole domain of an analyzed system. The implication of this fact is that while the neural model shows the correct rules of the system behavior in the region of fit, its surface beyond this region is mostly a random one. Therefore, it can be stated that the application of the neural model beyond the region of training data (an extrapolation case) is unjustified. According to Niederlinski "there seems to be no engineering justification whatever for extrapolating any model, be it polynomial or be it neural, beyond the region of fit used in the identification experiment. On the contrary, there are plenty of counterexamples showing that systems described by models established for some region of fit may break down when driven beyond this region" [Ni1].

Since the neural model application is justified only in case of data points located in the range of training data (an interpolation case), a very important issue is to determine properly the interpolation region of a neural model. While determining this region for a two-dimensional model is a relatively easy task (it can be done on the basis of analyzing two-dimensional graphs), it can be very hard in a multi-dimensional case. The aim of this article is to present a method which can be used for determining the interpolation region of a multi-dimensional neural model in a specific class of systems - chain systems.

The distinct feature of a chain system is that it can be described not only by a surface model but also by a parametric curve. This feature is very useful when the interpolation region of a neural model is under consideration because the chain model is located in the very center of the training samples. That means it indicates the location of data points in the whole system domain simultaneously.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 123–132, 2006.
© Springer-Verlag Berlin Heidelberg 2006

The application of the method proposed in this article will be illustrated via a real economic system of an unemployment rate in Poland in years 1992-2001. The output variable of this system is *unemployment rate* and the input variables are: *money supply* and *number of inhabitants*. The data for the survey was provided by the Polish Statistic Department.

First section of the article shortly discusses some approaches for establishing the interpolation region of a multi-dimensional model, second section introduces the problem of parametric curve modelling, third - explains the basis of the proposed method and the last one - presents the application of this method in the system of an unemployment rate in Poland.

2 Interpolation Region of a Multi-dimensional Model

Sometimes it is assumed that interpolation takes place only when a model surface is fitted to the data in interpolation nodes which are situated exactly in the data points [Fl1]. Such approach, however, is not very useful when real systems described by huge sets of noisy data are under consideration. This is due to the fact that situating the interpolation nodes in data points often results in an overfitted models which reflect the noise existed in the training data instead of the general rule of the analyzed system.

The term interpolation can be also addressed in a broader sense - as the term opposite to extrapolation. In this context interpolation is a process of determining the model answers for data points located inside the region covered by the training data. In case of analyzing multi-dimensional systems the biggest problem is establishing the boundary beyond which the interpolation is replaced by the extrapolation. The most popular approach to deal with this problem is to determine a hypercube covering all data points of the analyzed system. The hypercube edges are established on the basis of the minimal and maximal values of each variable. Figure 1a presents an example of a hypercube in a three-dimensional space.

The approach based on the hypercube is very easy to implement, however, as it can be noticed in Fig. 1, it generates the interpolation region only partially covered by data points. A more strict approach to establish the border of the interpolation region is to build a convex hull spread over all data points in a multi-dimensional space. The convex hull of a set of data points can be defined as the smallest closed convex region that contains all data from this set (Fig. 2b). There are a lot methods which can be applied in the process of building the convex hull e.g. greedy algorithm [Kl1], Beneath-Beyond algorithm [Kr1], NFECH method [Kl2]. This kind of methods generate a much more narrow interpolation region then the method which is based on building a hypercube, however, in case of systems of a chain data distribution this region is still too large. This is due to the fact that the data distribution in chain systems is very often a non-convex one, which means that these systems should not be described by the convex hull (Fig. 3).

The article presents a method of determining the interpolation region of a neural model which can be applied in systems of a chain data distribution. Since

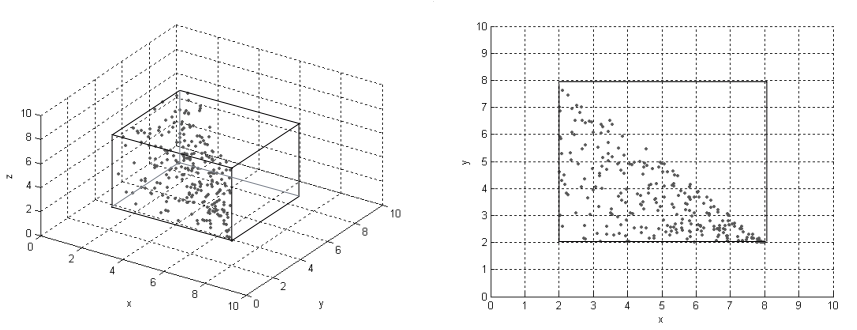


Fig. 1. Hypercube in a three-dimensional space (a); Hypercube in a two-dimensional space (b)

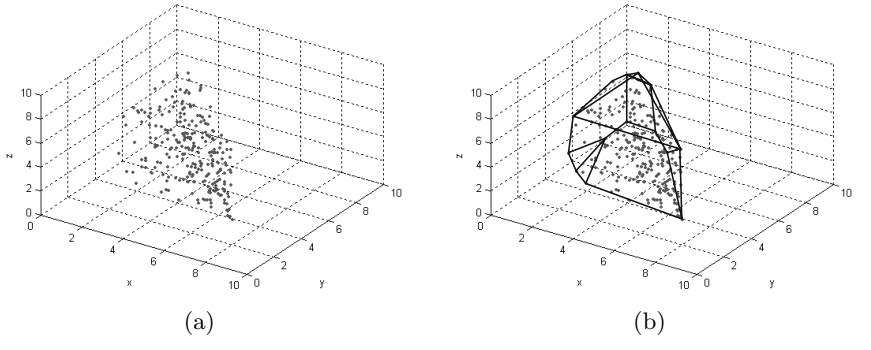


Fig. 2. Convex hull in a three-dimensional space

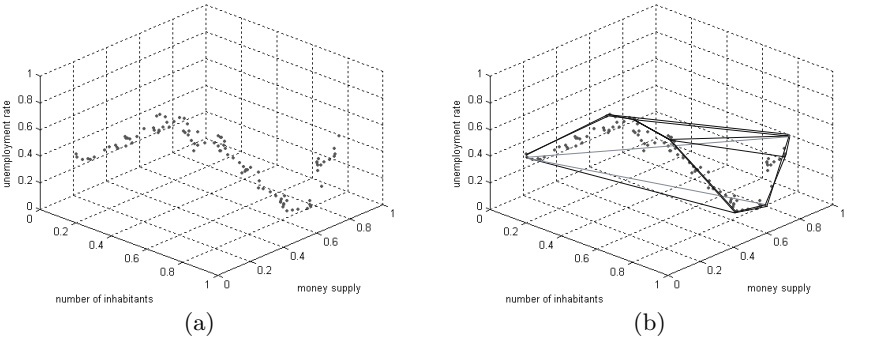


Fig. 3. System of a chain data distribution (a); Convex hull built over the data from fig. 3a (b)

the proposed approach is well suited to the system characteristic, it generates a much narrower (and of course also much more credible) interpolation region of a neural model than approaches mentioned above.

3 Parametric Curve Modelling Method

The main feature of a multi-dimensional chain system is that its decomposition into two-dimensional time series gives a set of tight chain systems. Therefore, in order to verify whether a system is of a chain profile, the reverse analysis should be performed. This analysis should be based on the time characteristics of all system variables. Tight chain dependencies, visible on the two-dimensional time series graphs, will indicate chain characteristic of the analyzed multi-dimensional system.

The chain system can be described not only by a surface model but also by a parametric curve model. The main idea of the parametric curve modelling method is to build a set of two-dimensional models, where each model describes the behavior of one variable (input or output) in regard to the known parameter t. These two-dimensional models can be created with many different mathematic techniques e.g. non-linear neural networks, polynomial regression, splines, etc. Two-dimensional models, built with one of the mentioned techniques, are then assembled together in order to create a multi-dimensional model describing the input-output mapping in the whole space (eq. 1).

$$\begin{cases} x_1 = f_1(t) \\ x_2 = f_2(t) \\ \quad ... \\ x_n = f_n(t) \end{cases} \tag{1}$$

In order to illustrate the process of parametric curve modelling, the chain model of the system mentioned in introduction was created. The neural networks of following parameters were used to build two-dimensional time series models of all system variables [Ma1] [DB1]:

- flow of signals: one-way,
- architecture of connections between layers: all to all,
- hidden layers: 1 hidden layer with suitable number of sigmoid neurons (5 for variable *number of inhabitants*, 3 for variable *money supply*, 4 for variable *unemployment rate*),
- output layer: 1 linear neuron,
- training method: backpropagation algorithm with momentum and changing learning rates,
- training aim: minimize mean absolute error (MAE),
- training time: 20000 epoch,
- testing method: visual control.

Models built with neural networks described above are shown in Fig. 4(a, b, c). By assembling together equations of all three models the chain curve model

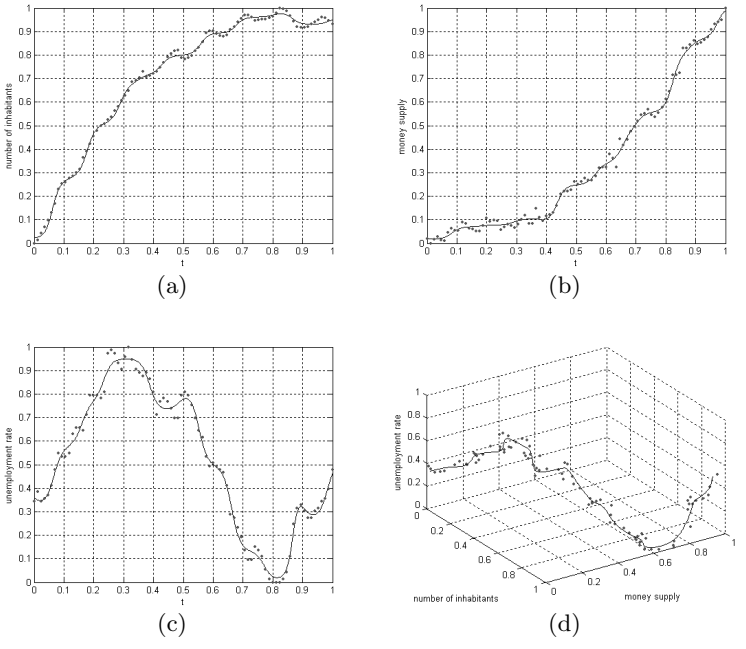


Fig. 4. Two-dimensional time series models (a,b,c); The parametric curve model (d)

of the analyzed system was created (Fig. 4d). The approximated accuracy of this model was calculated using MAE (eq. 2) [Ac1] and was equal to 2.28%.

$$error = \frac{\sum_{k=1}^{n} |z_i^* - z_i|}{n} \tag{2}$$

where: z_i^* - empirical values, z_i - theoretical values, n - number of data points.

4 Interpolation Region of a Neural Model

One the characteristic feature of the interpolation region of a surface model is that it indicates where it is possible to make credible prognosis for new data. Hence, the problem of establishing the interpolation region of a surface model can be replaced by the problem of determining the most credible prognostic region.

It can be said that a prediction made on the basis of a surface model is credible when two following criteria are fulfilled:

1. a new data point is situated in the range of data points used in the training process,
2. a part of the surface model used for predicting an output value of a new point is also situated in the range of data points used in the training process.

The chain model, introduced shortly in previous section, is a tool which allows to deal with both criteria. Its main feature is that it is situated in the center of data distribution - which means that it shows the center of training data.

Obviously, the knowledge of the center of data points is not sufficient to establish the interpolation region of a neural model. The second point which has to be addressed is a width of this region in a multi-dimensional space. Assuming that the width of the interpolation region will be the same in each direction, it can be calculated as a radius of a hypertube surrounding the chain model in a multi-dimensional space (in case of two-dimensional space it will be a band). In order to establish the radius of this hypertube, the distribution of the distances between each training data point and the chain model, calculated in a multi-dimensional space, can be considered.

A well known statistic measure which informs about the amount of data located in the selected region is a quantile. Among all quantiles, the most commonly used are quartiles, which divide a data set into four parts of equal amount of data points. In case of analyzing the absolute distances between data points and the chain model quartile first indicates the region surrounding the chain model covered by 25% of data points, quartile second (median) - 50% and quartile third - 75%.

At first look, the distance between the most remote data point and the model seems to be appropriate for establishing the radius of the hypertube determining the interpolation region of a surface model. However, taking into account the outliers problem, often meet in real systems [Ma1], it seems that it is more reasonable to establish the radius shorter than the greatest distance. In case of making prognoses in real systems the value of the quartile third could be regarded as the most appropriate one. On the other hand, however, when the mathematic system of a normal data distribution is considered, choosing the value of the third quartile as the radius of the hypertube would result in a false conclusions about the unreliability of the prognoses made for data points located behind the quartile third, while in reality a model of a system of a normal data distribution gives reliable prognoses in the whole range of the training data. Therefore, in order to build a hypertube appropriate for both kind of systems (real and mathematic ones), the hypertube radius cannot be equal to the value of the quartile third but has to be a little greater than that.

Regarding the well-known fact that in a system of a normal data distribution quartiles divide the data space into four parts [Ac1], the width of one of these parts can be used as the value which expands the hypertube radius. Since, these parts are only approximately equal, the added value cannot be calculated on the basis of any width but it ought to be equal to the minimal one (this is because the value of the hypertube radius cannot go beyond the most remote data point). Hence, the radius of the hypertube indicating the interpolation region of the neural model is proposed to be calculated as follows (eq. 3):

$$R_h = Q3 + min(Q1; Q2 - Q1; Q3 - Q2) \tag{3}$$

where: $Q1, Q2, Q3$ - value of quartile first, second and third, R_h - radius of the hypertube.

A hypertube of a radius calculated according to equation 3 indicates a reliable interpolation region of a neural model for both - mathematic systems and systems of an unknown data distribution. This is the implication of the following facts:

1. The hypertube created on the basis of systems of a normal data distribution covers approximately the whole range of training data.
2. The hypertube created on the basis of systems of an unknown data distribution eliminates the influence of the outliers problem on the width of the interpolation region.

Theoretically, when a neural and a chain model are built correctly, then the chain model is situated exactly on the surface of the neural model. However, it has to be underlined that this case is an ideal one. In most real situations there is no possibility to adapt a multi-dimensional neural network to such degree that it would be able to overlap the whole chain model - in such case both models are located in a distance from each other. Of course, a larger distance between both models indicates a smaller interpolation region of the neural model. In the most extreme case, when the whole neural model is located beyond the hypertube built around the chain model, it will not have any reliable interpolation region. This situation, however, should not be taken into account because it indicates that the neural network was not trained properly or the analyzed system does not have the surface characteristic (and cannot be described by a surface model). Theoretically, this situation could be also caused by an incorrectly trained chain model. However, in practice it is rather improbable because the learning process of the two-dimensional networks, forming this model, can be controlled visually.

Summing up the ideas introduced in this section, it can be said that the most reliable interpolation region of a neural model is given by the intersection of a neural model and a hypertube created around the chain model. Therefore, in order to verify whether a chosen point from the neural model is located in the interpolation region of this model, the distance between this point and the chain model should be calculated. The distance smaller than the radius of the hybertube means that the analyzed point is located in the interpolation region of the neural model.

5 Practical Application

According to prof. A. Piegat economic systems have mostly chain characteristic. Therefore in order to present a practical application of the method described in the previous section, an economic system (described in introduction) was chosen. The process of determining the interpolation region of a neural model of this system was carried out in three steps, in which:

1. a neural model was created,
2. a chain model was created,
3. an interpolation region of the neural model was established.

In the first step of the survey a neural model of the analyzed system was created. The parameters of the neural networks used in the process of building a neural model were as follows:

- flow of signals: one-way,
- architecture of connections between layers: all to all,
- hidden layers: 1 hidden layer with 8 sigmoid neurons,
- output layer: 1 linear neuron,
- training method: backpropagation algorithm with momentum and changing learning rates,
- training aim: minimize mean absolute error (eq. 2),
- training time: 50000 epoch,
- testing method: 16-cross-fold validation.

The MAE of the model was equal to 2.35%. The model surface is shown in Fig. 5.

(a) (b)

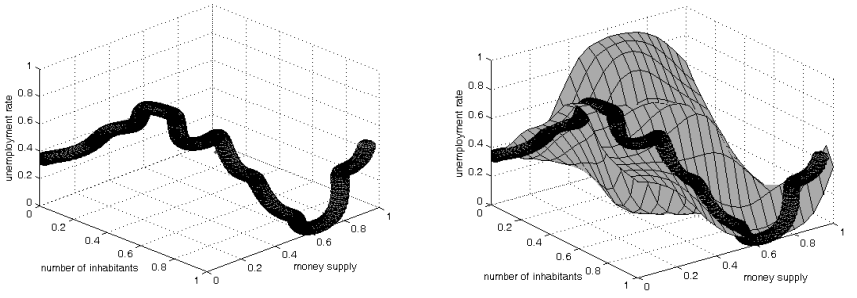
Fig. 5. Neural model of the system of an unemployment rate in Poland in years 1992-1999 (a); Neural model and data distribution

As it can be observed in Fig. 5b, the model surface is reasonable only in the region covered by data points and is randomly shaped beyond this region. This indicates that this model cannot be used for estimating the value of the output variable in the whole space but only in the closest neighborhood of the known data points.

In order to establish the interpolation region of the neural model, a chain model of the analyzed system had to be created. The process of building the chain model of the system of an unemployment rate was discussed in details in section three - the shape of the final model is shown in Fig. 4d.

The chain model was then used as a center of a multi-dimensional hypertube containing most of the data points of the analyzed system. The radius of the hypertube, calculated on the basis of eq. 3, was equal to 0.0387. The hypertube is shown in Fig. 6a and the most credible interpolation region of the neural model - in Fig. 6b (where the overlapping of both, the hypertube and the surface model

is visible). As it can be observed in Fig. 6 the proposed method of establishing the interpolation region of the multi-dimensional model resulted in a much more narrow interpolation region than the methods based on creating the convex hull, which example was shown in section two (Fig. 3b)



(a) (b)

Fig. 6. Hypertube (a); Hypertube and surface model (b)

6 Conclusion

The application of any surface model (also a neural model) outside its interpolation region is unjustified because it can result in random model responses. Therefore, it is very important to establish a reliable interpolation region - especially when multi-dimensional models are considered. To deal with this task some methods mentioned in section two can be applied. However, in a special class of systems (system of a chain data distribution) standard methods can generate too wide interpolation region (Fig. 3b).

The aim of this article was to present an approach which can be used for establishing the most probable interpolation region of neural models built for unknown multi-dimensional systems of a chain characteristic. The proposed approach utilizes the knowledge coming from the chain models constructed for such systems and is based on the analysis of hypertubes surrounding these models. The application of the method in the systems of a chain data distribution results in a more reliable interpolation region than the application of standard methods (Fig. 6).

It has to be underlined that at this moment of the survey the application of the approach described in the article can be used only in systems in which the data sequence is known (e.g. in time series systems).

References

[Ac1] Aczel A.D., "Complete Business Statistics", Richard D. Irwin, Inc., Sydney, 1993.

[DB1] Demuth H., Beale M., "Neural Network Toolbox User's Guide", The Math Works Inc., Natick MA USA, 2000.

[Fl1] Flannery B.P., Press W.H, Teukolsky S.A.,Vetterling W. T., "Numerical Recipes in C : The Art of Scientific Computing", Cambridge University Press, 1992.

[Kl1] Klesk P., "Algorithm for automatic definition of validated and nonvalidated region in multidimensional space", 10th International Conference on Advanced Computer Systems, Miedzyzdroje, 2003.

[Kl2] Klesk P., "The method of setting suitable extrapolation capabilities for neuro-fuzzy models of multidimensional systems", PhD Thesis, Technical University of Szczecin, 2005.

[Kr1] Krivsky S., Lang B., "Verified computation of HigherDimensional convex hulls and the solution of linear systems", Electronic Journal on Mathematics of Computation, 2003.

[Ma1] Masters T., "Practical Neural Networks Recipes in C++", Academic Press,1993.

[Ni1] Niederlinski A., "Polynomial and Neural Input-Output Models for Control - a Comparison", MMAR'97, Poland, 1997.

[Re1] Rejer I., "A method of modeling a multi-dimensional system via artificial intelligence methods on the example of an unemployment in Poland", The publishing house of the Szczecin University, Szczecin, 2003.

RBF Neural Network for Probability Density Function Estimation and Detecting Changes in Multivariate Processes

Ewa Skubalska-Rafajłowicz

Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50 370 Wrocław, Poland
ewa.rafajlowicz@pwr.wroc.pl

Abstract. We propose a new radial basis function (RBF) neural network for probability density function estimation. This network is used for detecting changes in multivariate processes. The performance of the proposed model is tested in terms of the average run lengths (ARL), i.e., the average time delays of the change detection. The network allows the processing of large streams of data, memorizing only a small part of them. The advantage of the proposed approach is in the short and reliable net training phase.

1 Introduction

Statistical control charts are designed in order to detect abnormalities (out-of control states) in the process under consideration. The most common abnormalities are mean shifts, variance changes and trends.

Suppose $X_1, X_2, \ldots$ are independent random vectors observed sequentially and X_1 to X_{q-1} have a distribution function with probability density f_0 while $X_q, X_{q+1}, \ldots$ have a distribution function with probability density $f_1 \neq f_0$.

q is unknown and some action should be taken after undesirable change in the process. One has to decide, on the basis of given observations

$$X_t = (x_{t1}, \ldots, x_{td}) \ ,$$

whether X_t is r.v. with pdf f_0 , i.e., process is "in-control" or if X_t is another r.v. - process is "out-of-control", i.e., changes in the process occurred. We assume, that probability densities f_0 and f_1 exist but are unknown.

There is extensive literature on statistical methods for statistical process control (SPC) and control charts, see [14] and the bibliography cited therein.

Classical control charts require prior assumptions about the probability density distribution of the observed process variables. Typically it is assumed that monitored data follow univariate or multivariate Gaussian (or sometimes other known) distribution. For multivariate statistical process control with individual observations, the Hotelling T^2 control chart or charts (based on Mahalanobis distance) are usually recommended.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 133–141, 2006.
© Springer-Verlag Berlin Heidelberg 2006

A neural network based approach to statistical process control and out-of control state detection allows in-control data density distribution to be non-Gaussian. Most of the neural network models designed for detecting changes in (mostly univariate) statistical process work in pattern recognition settings, i.e. on the assumption that also abnormal observation (out-of control states) are available and their class-membership (in-control and out-of control labels) are known [8], [3], [7], [6], [10], [9], [4], [5], [2], [12].

A neural network-based approach used when only in-control data is available has been considered in only a few papers (see [18] and [22]).

In the former paper the author proposes a vector quantization neural network with Kohonen's type learning algorithm to define the acceptance region. The multi-variate data is transformed onto unit interval using quasi-inverse of a space-filling curve [19], [20]. The method uses only one current vector observation to decide about the state of the process and for normal (Gaussian) data it is comparable to the Hotelling T^2 control chart [18], [13]. Zorriassatine et al.[22] uses a novelty detection method [1] for bivariate time series.

When constructing a control chart it is desirable to have a long average run length (ARL) in the in-control state, since this means a low level rate of false alarms. On the other hand, a short out-of control ARL is desired, which guarantees that any unacceptable changes will be identified as soon as possible.

Here we propose a new, easy to learn, radial basis function (RBF) neural network model for detecting changes in a multivariate process. The detection is based on one vector observation as in the classical T^2 control chart. Furthermore, we assume that the a'priori probabilities of in-control and out of control process states are not given.

In this paper the possibilities of detecting changes in the process mean vector (mean shifts) are investigated in terms of in-control and out-of-control ARL's.

2 RBF Neural Network Model for Detection of Changes

The radial basis function networks have been extensively applied to pattern recognition, function approximation or regression function estimation.

A basic radial-basis function (RBF) network consists of three layers having entirely different roles: an input layer, a hidden layer, which applies a nonlinear transformation from the input space to the hidden space and a linear output layer. Hence,

$$f_N(x) = \sum_{i=1}^{N} w_i G(||x - c_i||) \ , \tag{1}$$

where $x \in R^d$, $c_i \in R^d$, are tunable vectors, w_i are tunable weights, and N is a number of neurons.

Usually $||x||$ is the Euclidean norm, however also generalized weighted norm $||x||_{Q_i}$, defined by the quadratic form $||x||^2_{Q_i} = x^T Q_i^T Q_i x$ can be used, where Q_i are (usually tunable) $d \times d$ matrices.

The most popular are Gausssian RBF nets:

$$G(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \text{ for some } \sigma > 0 \text{ and } r \in \mathcal{R} \ .$$

There are three groups of parameters in the RBF networks which may be learnable or arbitrarily chosen: the weights w_i, the centers c_i and some parameters of radial basis functions, for example σ or Q_i matrices.

RBF networks can be related to Parzen window estimators of a probability density [16] or to Nadaraya-Watson regression estimators [21], [1], [15]. Similarities between the RBF network structure and kernel regression estimators lead to RBF networks with the centers chosen to be a subset of the training input vectors and associated weights which directly correspond to Y_i's [21]. Other approaches related to Nadaraya-Watson regression estimators were proposed in [17] and [11]

Usually, parameters of the network (1) are obtained from an n-sample observation data set (learning sequence) $L_n = ((X_1, Y_1), \ldots, (X_n, Y_n)$.

As regards our problem, labels $Y_i,\ i = 1, \ldots, n$ are set as "in-control" and do not carry any information. Thus, we need a net which will self-organize and generalize information about distribution of the in-control states.

In this context, we choose RBF neural networks related to Parzen kernel estimators. Bishop [1] discusses a number of heuristics for learning RBF parameters in such a way, that the basis functions approximate the distribution of the input data.

The Parzen window estimator [16], [1] with Gaussian kernel functions takes the form:

$$\frac{1}{n(2\pi\sigma^2)^{d/2}} \sum_{i=1}^{n} \exp\left(-\frac{||X - X_i||^2}{2\sigma^2}\right) \ , \tag{2}$$

where d is the dimensionality of the input data.

Let N be a number of centers. Assuming that the centers should be distributed according the same probability distribution as the learning data, the centers are simply a subset of the training input vectors. One can take, for example, N first elements from the leaning sequence $(X_1, \ldots, X_n)$.

Note that if X_i is close to a center C, then

$$G(||X - X_i||) - G(||X - C||) \approx 0$$

So, we can replace each X_j in the sum (2) by its nearest neighbor among a set of centers $\{C_1, C_2, \ldots, C_N\}$ breaking ties at random. Note, that the same C_i can be the nearest neighbor for several X_j's and that each C_i has at least one point from the learning sequence (namely itself) as a neighbor, since every center is taken from the learning set.

Let n_j stands for the number of points closest to the center C_j, i.e.,

$$n_j = \text{card}[\{X_i : ||X_i - C_j|| < ||X_i - C_k||\}] \ .$$

Thus, we obtain the approximate version of (2):

$$y(X) = \frac{1}{n(2\pi\sigma^2)^{d/2}} \sum_{j=1}^{N} n_j \exp\left(-\frac{||X - C_j||^2}{2\sigma^2}\right) \ . \tag{3}$$

Observe, that this kind of the probability density approximation appears in [17], where it is used as a common denominator in the RBF neural network mimicking the Nadaraya-Watson regression estimators.

The decision about in-control or out of control state of the current vector observation X is made according to the estimate of the probability density $y(X)$.

If $y(X) < \lambda$, where λ is chosen acceptance level, then classify X as abnormal (out of control) state of the process. Otherwise, accept X as an in-control state. The set $y(X) \geq \lambda$ forms the confidence region.

2.1 Algorithm for Tuning RBF Net

Step 1. Choose centers C_j, $j = \overline{1, N}$ at random from the learning sequence $\{X_1, X_2, \ldots, X_n\}$.
Step 2. Set $n_j = 0$, $j = \overline{1, N}$.
Step 3. For $i = \overline{1, n}$ perform the following steps.
1. Find $j^* = arg\ \min_{1 \leq j \leq N} ||X_i - C_j||$.
2. Update the corresponding weight:$n_{j^*} = n_{j^*} + 1$.

Step 4. Form the net

$$y(X) = \frac{1}{n(2\pi h\sigma^2)^{d/2}} \sum_{j=1}^{N} n_j \exp\left(-\frac{||X - C_j||^2}{2\sigma^2}|\right) .$$

Step 5. Choose the acceptance level (threshold) λ. If $y(X)$ is greater than λ accept vector observation as in-control, otherwise alarm, since an out-of-control state is detected.

This algorithm should be accompanied by a method of selecting the bandwidth $\sigma > 0$ and threshold parameter λ. One can choose any known method, e.g., the cross-validation for selecting $h\sigma$. Having selected centers and using a formula (3) one can considerably reduce the computational burden needed for selecting σ in a data-driven way.

Furthermore, reducing the number of kernels (to the number of centers) leads to the formula less sensitive to the σ choice. The threshold level λ governs the false alarm probability α. The average run length to the false alarm (the in-control ARL) equals to $1/\alpha$[14], but the distribution function of in-control states is usually not (fully) known. Thus, as in classical control charts, the value of the threshold, which guarantees desired in-control ARL should be chosen experimentally.

3 Experimental Results

In the following sections we present the results of applying the RBF control chart to a series of simulated data sets. We have tested the proposed method using three different data sets:

A. A 2-D normal distribution with $(0, 0)$ mean and covariance matrix I
B. A mixture of two equiprobable 2-D normal distributions with vector means: $(0, 0)$ and $(2, 0)$ and the same I.

C. A mixture of three equiprobable 2-D normal distributions with vector means: $(0,0)$, $(2,0)$ and $(0,0-2)$ and the same I (see Figure 1).

We compared the performance of the proposed RBF control chart with the results given by the classical T^2 chart based on the Hotelling statistic (also known as the squared Mahalanobis distance) [14], [13]:

$$T^2(x) = (x-m)^T \Sigma^{-1}(x-m) \ ,$$

where x is a given observation vector, m is a mean vector and Σ the covariance matrix. Both of them, m and Σ, are estimated from the learning data. If $T^2(x) > t$, where t is an experimentally chosen value, it is assumed that the observation x is rejected as out-of-control data. Thus, the region of acceptance of T^2 control chart form an ellipsoid with the center m and the other parameters defined by the covariance matrix Σ.

In two cases (A and B) the neural network model was tuned using 10^5 in-control learning samples. We examined the changes in the process caused by the following mean shifts $||\Delta m|| = 0.5,\ 1, 2$, and 3. The number of centers was equal to 100.

The in-control ARL's were obtained from 10^6 examples and out-of-control ARL's were estimated from 10^5 repetitions. The results were averages over four different shift directions. The comparisons for examples A and B are given in Table 1.

Table 1. Comparison of RBF neural network chart with T^2 chart, $d = 2$

	Example A		Example B	
$\|\|\Delta m\|\|$	T^2	RBF net	T^2	RBF net
	$t = 10.6$	$\lambda = 0.00049$	$t = 9.8$	$\lambda = 0.0003$
0.0	200.0	199.0	200.4	198.2
0.5	116.0	117.6	124.0	118.2
1.0	42.0	44.5	53.3	48.1
2.0	6.9	7.7	10.1	8.7
3.0	2.2	2.3	3.2	2.9

The second column of Table 1 contains analytically obtained ARL's for T^2 Hotelling chart applied to multivariate normal data (see for example [13]). The RBF net based control chart attains almost the same ARL times, however the knowledge about probability density distribution is not used in the process of designing the RBF net chart. The value of parameter λ was chosen experimentally on test data in such a way as to obtain ARL0=200.

The third column consists of empirically determined ARL's for T^2 Hotelling chart applied to multivariate non-normal data (example B). This time, the RBF

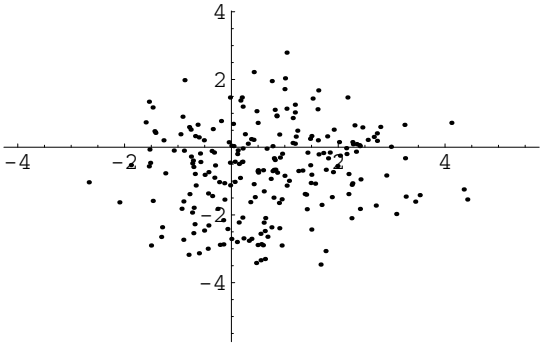


Fig. 1. Learning data - example C

net chart leads to shorter average detection times (smaller out-of-control ARL's), than the T^2 chart.

Experiments on the third example (the mixture of three normals – C) were performed on 300, 000 learning samples (see Figure 1). In this case the changes in the process were introduced by the mean shifts $||\Delta m||/\ (det\Sigma_C)^{1/4} = 0.5,\ 1, 2, 3$, where Σ_C is the empirical covariance matrix obtained for the distribution C (estimated on the basis of all 300000 learning samples). The results for every shift's length were averages over eight different shift directions. The same empirical covariance matrix Σ_C was used in T^2 calculations. The number of centers of the RBF network was equal to 200. The value of parameter λ was chosen experimentally on test data in such a way as to obtain ARL0=200. The comparisons for the example C are given in Table 2.

Table 2. Comparison of RBF neural network chart with T^2 chart for mixture of three normals $d = 2$

$\|\|\Delta m\|\|/det\Sigma_C^{1/4}$	Example C	
	T^2	RBF net
	$t = 41.5$	$\lambda = 0.00029$
0.0	201.6	200.0
0.5	137.1	97.6
1.0	85.8	29.3
2.0	21.8	4.64
3.0	3.76	1.82

The RBF control chart proposed here gave this time evidently better results than that obtained with T^2 control chart. The ARL times estimated for T^2 chart are even worse than relative ARL's computed for examples A and B, since the

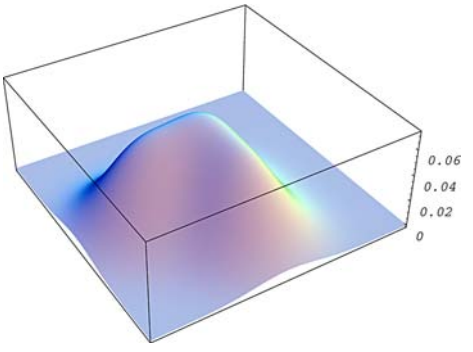


Fig. 2. Probability density function of data from example C

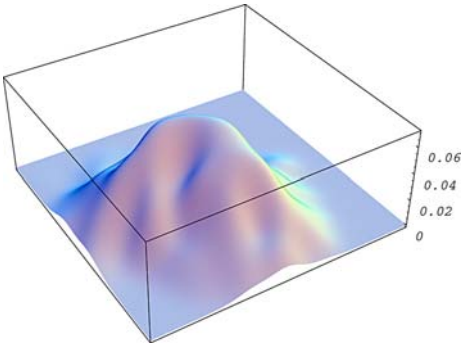


Fig. 3. Probability density function of data from example C estimated by RBF net

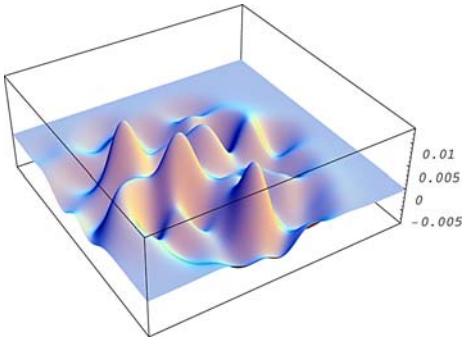


Fig. 4. Error of probability density estimation for example C

mixture of three normals distributions is not similar to any two-dimensional normal distribution. Figure 2 shows the true density function of data from example C. Figure 3 presents the probability density function of data under considerations estimated using the RBF neural network (obtained by formula 3). Error of this probability density estimation is given in Figure 4. The acceptance region obtained during experiments with in-control data is illustrated in Figure 5 as a

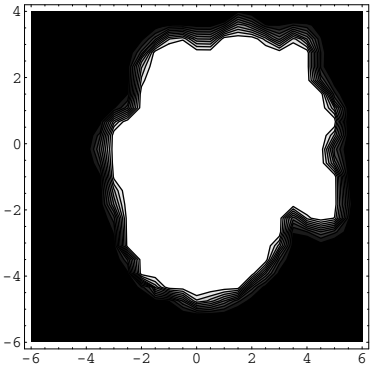


Fig. 5. Acceptance region obtained using a RBF net with $\lambda = 0.00029$

white shape. This is a set of points where the value of parameter λ is smaller than the probability density value estimated in those points by RBF network.

4 Concluding Remarks

The crucial problem faced in this paper is in designing the simple and robust nonparametric probability density function estimator for time independent multivariate processes. A new version of a RBF neural network allows the processing of large streams of data, memorizing only a small part of them. The network was successfully applied to the detection of changes in multivariate processes. The advantage of the proposed approach is in the short and reliable net training phase.

Acknowledgments

This paper was partially supported by a Polish Government Grant 2003–2005 and by the Polish Foundation for Science.

References

1. Bishop C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
2. Chang S.I., Aw C.A.: A neural fuzzy control chart for detecting and classifying process mean shifts. International Journal of Production Research **34** (1996) 2265–2278
3. Chen L., Wang T.: Artificial neural networks to classify mean shifts from multivariate χ^2 charts signals. Computers and Industrial Engineering **47**(2004) 195–205
4. Cheng C.-S.: A neural network approach for the analysis of control charts patterns. International Journal of Production Reseach **235**(1997) 667–697

5. Cheng C.-S.: A multi-layer neural network model for detecting changes in the process mean. Computers and Industrial Engineering **28** (1995) 51–61
6. Cheng C.-S., Cheng S.: A neural network-based procedure for the monitoring of exponential mean. Computers and Industrial Engineering **40**(2001) 309–321
7. Chinnam R.B.: Support vector machines for recognizing shifts in correlated and other manufacturing processes. International Journal of Production Research **40**(2002) 4449–4466
8. Guh R.: A hybrid learning based model for on-line detection and analysis of control chart patterns. Computers and Industrial Engineering **49**(2005) 35–62
9. Guh R., Hsieh Y.: A neural network based model for abnormal pattern recognition of control charts. Computers and Industrial Engineering **36**(1999) 97–108
10. Ho E.S., Chang S.I.: An integrated neural network approach for simultaneously monitoring of process mean and variance shifts - a comparative study. International Journal of Production Research **37**(1999) 1881–1901
11. Krzyżak A., Skubalska-Rafajłowicz E.: Combining Space-filling Curves and Radial Basis Function Networks.In: Artificial Intelligence and Soft Computing ICAISC 2004. Lecture Notes in Artificial Intelligence Vol.3070. Springer-Verlag, Berlin Heidelberg New York (2004) 229–234
12. Low C., Hsu C., Yu F.:Analysis of variations in a multi-variate process using neural networks.Int. J Adv Manuf Technology **22** (2003) 911–921
13. Lowry C.A., Woodall W.H.: A Multivariate Exponentially Weighted Moving Average Control Chart. Technometrics **34** (1992)46–53
14. Montgomery D.C.:Introduction to statistical quality control.John Wiley and Sons, Inc.(2001)
15. Nadaraya A.: On the integral mean square error of some nonparametric estimates for the density function. Theory of Probability and its Applications **19** (1974) 133–141
16. Parzen E.:On estimation of a probability density function and mode.Annals of Mathematical Statistics **33** (1962) 1065–1076
17. Rafajłowicz E., Skubalska-Rafajłowicz E.: RBF nets based on equidistributed points. in Proc. of 9th IEEE International Conf. Methods and Models in Automation and Robotics MMAR 2003 **2** (2003) 921–926
18. Skubalska-Rafajłowicz E.: On using space-filling curves and vector quantization for constructing multidimensional control charts. Fifth Conf. "Neural Network and Soft Computing" Zakopane June 6-10, 2000 (2000) 162–167
19. Skubalska-Rafajłowicz E. : Pattern recognition algorithm based on space-filling curves and orthogonal expansion. IEEE Trans. on Information Theory **47** (2001) 1915–1927
20. Skubalska-Rafajłowicz E.: Space-filling Curves in Multivariate Decision Problems. (in Polish) Wrocław University of Technology Press, Wrocław (2001)
21. Xu L., Krzyżak A., Yuille A.: On Radial Basis Function Nets and Kernel Regression: Statistical Consistency, Convergence Rates, and Receptive Field Size. Neural Networks **7** (1994) 609–628
22. Zorriassatine F., Tannock J.D.T., O'Brien C.: Using novelty detection to identify abnormalities caused by mean shifts in bivariate processes.Computers and Industrial Engineering **44** (2003) 385–408

Fast Orthogonal Neural Networks

Bartłomiej Stasiak and Mykhaylo Yatsymirskyy

Institute of Computer Science, Technical University of Łódź
ul. Wólczańska 215, 93-005 Łódź, Poland
basta@ics.p.lodz.pl, jacym@ics.p.lodz.pl

Abstract. The paper presents a novel approach to the construction and learning of linear neural networks based on fast orthogonal transforms. The orthogonality of basic operations associated with the algorithm of a given transform is used in order to substantially reduce the number of adapted weights of the network. Two new types of neurons corresponding to orthogonal basic operations are introduced and formulas for architecture-independent error backpropagation and weights adaptation are presented.

1 Introduction

Linear neural networks represent linear transforms of input signals. One layer of linear neurons is capable of learning an arbitrary linear transform [1,2], which involves determining of $O\left(N^2\right)$ weights, where N is the dimension of the transformed space.

For special types of linear transforms, including discrete Fourier transform (DFT), discrete cosine transform (DCT), discrete sine transform (DST) and discrete Hartley transform (DHT), a symmetry-based factorization of their matrices leads to reduction in computational complexity [3,4,5]. Following the factorization scheme in neural network architecture it is possible to obtain a fast multilayer linear network with sparsely connected layers, containing $O\left(N \log\left(N\right)\right)$ weights [6,7]. One of the substantial advantages of such an approach is the possibility of efficient hardware implementations, based on the existing DSP architectures [8].

In this paper we consider a new approach to constructing and teaching neural networks of this type, based on the orthogonality of basic operations in the underlying algorithms. A notion of a basic operation orthogonal neuron (BOON) is introduced and two types of BOONs are presented.

The main contribution of the paper is a method of BOON-based network teaching in an architecture-independent way, applicable to a wide class of known orthogonal transforms. It is also shown that the application of BOONs leads to a two-fold or a four-fold reduction in the number of weights and to an increase in the stability of the learning process. The fast cosine transform, type II and its variant with tangent multipliers [9] have been chosen to demonstrate the network construction, but other known fast orthogonal transforms (e.g. [10]) may be also easily realized.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 142–149, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Fast Two-Stage Orthogonal Transforms Algorithms

In homogeneous two-stage algorithms of fast orthogonal transforms, best suited for hardware implementation, two main types of basic operations are typically used: trivial operations (addition/subtraction) and non-trivial ones involving multiplications by twiddle factors. In the next part of the paper two variants of the fast cosine transform, type II will be presented as examples of a homogeneous two-stage algorithm construction.

2.1 Fast Cosine Transform, Type II

Let $x(n)$ be an N-point real sequence, where $n = 0, 1, ..., N-1; N = 2^m; m \in \mathbb{N}$. The discrete cosine transform, type II of $x(n)$ is defined as [11]:

$$L_N^{II}(k) = \mathrm{DCT}_N^{II}\{x(n)\} = \sum_{n=0}^{N-1} x(n) C_{4N}^{(2n+1)k} \quad , \tag{1}$$

where $n, k = 0, 1, ..., N-1; C_K^r = \cos(2\pi r/K)$.

The basic computational procedure of the fast cosine transform, type II (FCT2) may be given as [9]:

$$\begin{aligned} L_N^{II}(0) &= L_1(0), L_N^{II}(N/2) = \sqrt{2}/2 \cdot L_2(0) \quad , \\ L_N^{II}(k) &= C_{4N}^k L_1(k) + S_{4N}^k L_2(N/2-k) \quad , \\ L_N^{II}(N-k) &= -S_{4N}^k L_1(k) + C_{4N}^k L_2(N/2-k) \quad , \\ k &= 1, 2, ..., N/2-1 \quad , \end{aligned} \tag{2}$$

where $L_1(k) = \mathrm{DCT}_{N/2}^{II}\{a(n)\}, L_2(k) = \mathrm{DCT}_{N/2}^{II}\{b(n)\}$ and the sequences $a(n)$ and $b(n)$ are formed from the input sequence $x(n)$ as follows:

$$\begin{aligned} a(n) &= x(2n) + x(2n+1) \quad , \\ b(n) &= (-1)^n (x(2n) - x(2n+1)) \quad , \\ n &= 0, 1, ..., N/2-1 \quad . \end{aligned} \tag{3}$$

Formulas (2) and (3) may be applied recursively, leading to a homogeneous FCT2 algorithm, concisely described in the form of a directed graph (Fig. 1, 2).

2.2 FCT2 Algorithm with Tangent Multipliers (mFCT2)

Multiplying and dividing the formulas (2) by C_{4N}^k enables an additional optimization [9] by means of cumulating the coefficients C_{4N}^k as a single multiplication performed as the last step of the transform (Fig. 3, 4).

3 Orthogonal Neural Networks

3.1 Orthogonal Basic Operations

Based on the diagram in Fig. 1 a neural network architecture where each non-trivial basic operation is replaced by two neurons [6,7] may be built. Both neurons corresponding to a single basic operation have two inputs, i.e. the number

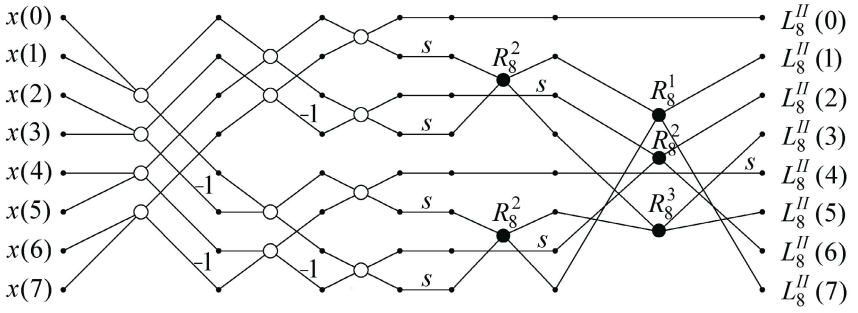


Fig. 1. Directed graph of the FCT2 algorithm for $N = 8$, $s = \sqrt{2}/2$

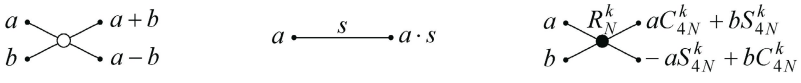


Fig. 2. Basic operations of the FCT2 algorithm

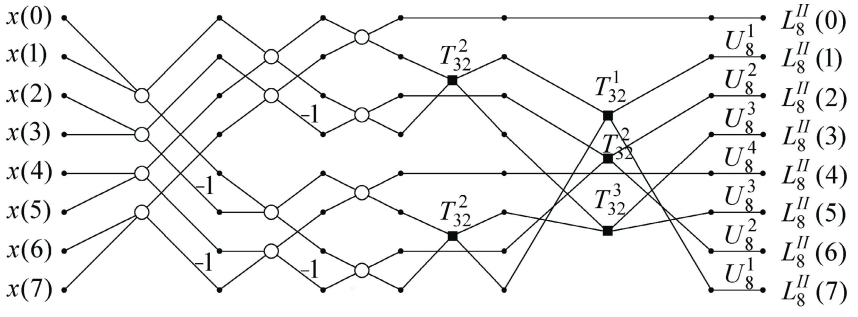


Fig. 3. Directed graph of the mFCT2 algorithm for $N = 8$

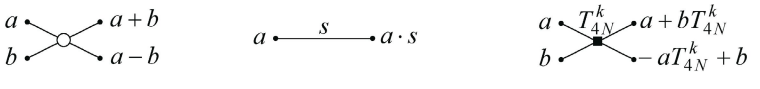


Fig. 4. Basic operations of the mFCT2 algorithm

of weights to adapt equals 4. A basic operation may be therefore seen as a 2-by-2 matrix multiplication

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = P_4 \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ , where } P_4 = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \text{ .} \tag{4}$$

However, considering the matrix representation of the third operation in Fig. 2 we notice that only two weights are actually needed

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = P_2 \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ , where } P_2 = \begin{bmatrix} u & w \\ -w & u \end{bmatrix} \text{ .} \tag{5}$$

The significant difference lies in the orthogonality of matrix P_2. In fact, P_2 satisfies even more restrictive condition: not only are its rows/columns orthogonal, but they are also of equal norm (the orthogonality condition itself would imply adapting three independent weights).

Taking into account the explicit target values of P_2 elements, $u = C_{4N}^k$; $w = S_{4N}^k$, it is also possible to express the weights of a basic operation as functions of one parameter α, e.g. $u = \cos(\alpha)\,; w = \sin(\alpha)$. This is equivalent to defining the rows/columns of P_2 as orthonormal vectors. Such an approach would, however, result in the necessity of trigonometric functions computations in the course of the learning process, which is undesirable.

The solution to the last inconvenience may be obtained by implementing the neural network on the basis of the mFCT2 algorithm. This implies considering

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = P_1 \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ , where } P_1 = \begin{bmatrix} 1 & t \\ -t & 1 \end{bmatrix} \text{ ,} \tag{6}$$

according to the third operation presented in Fig. 4.

3.2 Teaching Methods

The main practical issue resulting from the application of matrices P_2 or P_1 affects the methods of neuronal weights adaptation. The classical definition of a neuron should be modified here to reflect the relationship between the outputs of the basic operation. We would either talk about two associated neurons, orthogonal to each other, or simply about a basic operation orthogonal neuron (BOON) with two outputs. It is also worth noting that the matrix P_4 does not require any special treatment, as its rows may be seen as representations of classical independent neurons with two inputs. As gradient backpropagation methods seem the best choice for teaching the considered types of neural networks [7], a proper algorithm suited for the special forms of matrices P_2 and P_1 is necessary.

Considering a simple case of two connected layers shown in Fig. 5 and assuming that the basic operation matrix has a form defined by (5) we can explicitly express the outputs of the network as functions of its inputs

$$\begin{array}{l} y_1 = \quad u_1^{(2)} v_1 + w_1^{(2)} v_2 \\ y_2 = -w_1^{(2)} v_1 + u_1^{(2)} v_2 \\ y_3 = \quad u_2^{(2)} v_3 + w_2^{(2)} v_4 \\ y_4 = -w_2^{(2)} v_3 + u_2^{(2)} v_4 \end{array} \text{ , where } \begin{array}{l} v_1 = \quad u_1^{(1)} x_1 + w_1^{(1)} x_3 \\ v_2 = \quad u_2^{(1)} x_2 + w_2^{(1)} x_4 \\ v_3 = -w_1^{(1)} x_1 + u_1^{(1)} x_3 \\ v_4 = -w_2^{(1)} x_2 + u_2^{(1)} x_4 \end{array} \tag{7}$$

and where the expressions $u_k^{(l)}$ and $w_k^{(l)}$ refer to the k-th operation of the l-th layer. Our goal is to minimize the error function given as:

$$E = \frac{1}{2} \sum_{i=1}^{N} (y_i - d_i)^2 \quad , \tag{8}$$

where $N = 4$ and d_i represents an expected value of the i-th output.

Substituting (7) into (8) and computing derivatives for the weights in the second and in the first layer we arrive at formulas defining the components of the gradient vector and the error vector for a single basic operation

$$\begin{bmatrix} \frac{\partial E}{\partial u} \\ \frac{\partial E}{\partial w} \end{bmatrix} = \begin{bmatrix} v_1 & v_2 \\ v_2 & -v_1 \end{bmatrix} \cdot \begin{bmatrix} e_1^{(n)} \\ e_2^{(n)} \end{bmatrix} \quad , \tag{9}$$

$$\begin{bmatrix} e_1^{(n-1)} \\ e_2^{(n-1)} \end{bmatrix} = P_2^{\mathrm{T}} \cdot \begin{bmatrix} e_1^{(n)} \\ e_2^{(n)} \end{bmatrix} \quad , \tag{10}$$

where P_2^{T} denotes the transpose of P_2.

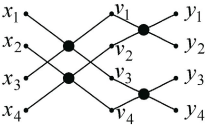


Fig. 5. Two layers of an orthogonal network, each containing 2 basic operation neurons

The parameters v_1 and v_2 represent the inputs of the basic operation, the vector $\left[e_1^{(n)},\ e_2^{(n)} \right]^{\mathrm{T}}$ refers to the error values propagated back from the next layer and the vector $\left[e_1^{(n-1)},\ e_2^{(n-1)} \right]^{\mathrm{T}}$ defines the error values to be propagated back from the current layer to the previous one. As the matrix P_1 (6) is a special case of the matrix P_2 (5) for $u = 1$, the corresponding formulas, suitable for teaching an mFCT2-based network, can be derived from (9), (10) as follows:

$$\frac{\partial E}{\partial t} = \left[v_2,\ -v_1 \right] \cdot \begin{bmatrix} e_1^{(n)} \\ e_2^{(n)} \end{bmatrix} \quad , \tag{11}$$

$$\begin{bmatrix} e_1^{(n-1)} \\ e_2^{(n-1)} \end{bmatrix} = P_1^{\mathrm{T}} \cdot \begin{bmatrix} e_1^{(n)} \\ e_2^{(n)} \end{bmatrix} \quad . \tag{12}$$

The formulas (9) - (12) have a general meaning, i.e. they are applicable to a basic operation irrespective of its location in the network architecture. Moreover, no specific architecture is imposed as the information about the indexes of the interconnected basic operations' inputs/outputs is sufficient. Given the components of the gradient vector, any known gradient method may be successfully applied to minimize the error function of the network.

The numbers of multiplications (μ) and additions (α) for the matrices are:

$$\begin{array}{lll} \mu(P_4) = 8 & \mu(P_2) = 8 & \mu(P_1) = 4 \\ \alpha(P_4) = 2 & \alpha(P_2) = 4 & \alpha(P_1) = 3 \end{array} . \tag{13}$$

These values concern gradient computation and error backpropagation only. As one of the most crucial parameters influencing the efficiency of gradient minimization algorithms is the number of the adapted weights, its two-fold (P_2) and four-fold (P_1) reduction will actually play the most important role in a global computational complexity improvement.

It should also be noted that learning of the inverse transform may be easily realized by changing all the matrices to their (properly scaled) transpositions.

3.3 Experimental Validation

The presented methods of teaching the BOONs were implemented within a framework developed for testing neural networks with arbitrary connections.

Three groups of tests were performed to compare the capabilities of a non-orthogonal FCT2-based network (type P_4 BOONs) and of two orthogonal networks: FCT2 and mFCT2-based (type P_2 BOONs and type P_1 BOONs). Several datasets, varied by the length and the number of input vectors, were used for all groups. The teaching was repeated ten times for each dataset, from a random starting point. The averaged results are presented in Tables 1, 2, 3, respectively.

The first two columns contain the size (N) and the number (P) of random input vectors and std is the standard deviation. Target vectors for all the datasets were computed according to the formula (1).

The conjugate gradient method was applied as an algorithm of error function minimization [1] and the teaching was stopped when the error was lower than 1e-9. The tests were performed on a computer with Intel Celeron M, 1.40 GHz processor.

Table 1. Results of FCT2-based non-orthogonal network training

N	P	Mean epochs	Epochs std	Mean time [s]	Time std	Weights
8	4	49	6.063	0.1329	0.0619442	20
16	8	119	23.7445	1.2891	0.245518	68
32	16	125	37.5847	8.2673	2.41184	196
64	32	172	81.5784	78.6579	37.2662	516

Table 2. Results of FCT2-based orthogonal network training

N	P	Mean epochs	Epochs std	Mean time [s]	Time std	Weights
8	4	25	1.96214	0.0782	0.0474443	10
16	8	46	3.74299	0.525	0.0564659	34
32	16	58	1.84662	3.9814	0.126828	98
64	32	71	2.5865	33.3281	1.16706	258

Table 3. Results of mFCT2-based orthogonal network training

N	P	Mean epochs	Epochs std	Mean time [s]	Time std	Weights
8	4	14	1.22066	0.0625	0.0394899	5
16	8	25	0.916515	0.297	0.0270222	17
32	16	24	0.538516	1.7124	0.04304	49
64	32	22	0	10.9764	0.1246	129

The teaching of the orthogonal networks proved to be a stable process in terms of the standard deviation of its length. The most interesting observation is the almost constant number of epochs in the case of mFCT2-based network. A closer examination revealed that the final state of the network was always similar for a given dataset and equal to the state obtainable by computing the weights values directly.

The relatively high mean time values for higher N result from the generality and flexibility of the framework which involves operations on large connection matrices. The comparison between the tables, however, shows a definite superiority of the orthogonal networks, which is particularly clear in the case of type P_1 BOONs.

4 Conclusion

A new method of constructing and teaching neural networks based on fast orthogonal transforms was presented. Respecting the orthogonality of the basic operations allowed to reduce the number of the adapted weights in comparison to the non-orthogonal network, increasing the efficiency and stability of the learning process. Owing to the generality of the presented solutions, the proposed BOONs may be used in the construction of neural networks realizing a wide class of known orthogonal transforms.

References

1. Osowski, S.: Neural networks for information processing. (in Polish) OWPW, Warsaw (2000)
2. Rutkowski, L.: Methods and techniques of artificial intelligence. (in Polish) Polish Scientific Publishers PWN (2005)
3. Wang, Z.: Fast algorithms for the discrete W transform and for the discrete Fourier transform. IEEE Trans. on Acoustics, Speech, and Signal Processing **32** (1984) 803-816
4. Yatsymirskii, M.N.: Fast algorithms for the discrete cosine transformation. Comput. Maths Math. Phys **33** (1993) 267-270
5. Egner, S., Püschel, M.: Automatic generation of fast discrete signal transforms. IEEE Trans. on Signal Processing **49** (2001) 1992-2002
6. Jacymirski, M., Szczepaniak, P.S.: Neural realization of fast linear filters. In: Proc. of the 4th EURASIP - IEEE Region 8 International Symposium on Video/Image Processing and Multimedia Communications. (2002) 153-157

7. Szczepaniak, P.S.: Intelligent computations, fast transforms and classifiers. (in Polish) EXIT Academic Publishing House, Warsaw (2004)
8. Rabiner, L.R., Gold, B.: Theory and application of digital signal processing. Prentice-Hall (1975)
9. Jacymirski, M.: Fast homogeneous algorithms of cosine transforms, type II and III with tangent multipliers. (in Polish) Automatics **7** AGH University of Science and Technology Press, Cracow (2003) 727-741
10. Yatsymirskyy, M.M.: Shifted in the time and frequency domains cosine and sine transforms fast algorithms with homogeneous structure. (in Russian) Izvestiya Vysshikh Uchebnykh Zavedenii, Radioelektronika **43**, Kiev (2000) 66-75
11. Ahmed, N., Rao, K.R.: Orthogonal transforms for digital signal processing. Springer-Verlag, New York (1975)

AI Methods in Solving Systems of Interval Linear Equations

Nguyen Hoang Viet and Michal Kleiber

Institute of Fundamental Technological Research, Polish Academy of Sciences,
Swietokrzyska 21, 00-049 Warsaw, Poland
{viet, mkleiber}@ippt.gov.pl

Abstract. The problem of solving systems of interval linear equations with use of AI based approaches is studied in this paper. First, this problem is viewed in terms of an optimization task. A cost function with interval variables is defined. Next, for a given system of equations, instead of the exact algebraic solution its approximation is determined by minimizing the cost function. This is done by use of two different approaches: the NN based approach and the GA based one. A number of numerical evaluations are provided in order to verify the proposed techniques. The results are compared, discussed and some final conclusions are drawn.

1 Introduction

Solving real world problems, one usually has to deal with various kinds of uncertainties. These uncertainties may come from many sources, just to mention the inexactness of the data, the imperfection of the model, the presence of judgment of experts expressed in terms of linguistic values, the varying environment etc.. Moreover uncertainties can also have either the nature of randomness or impreciseness [3,13]. Randomness is related to events that may or may not take place in the future, whereas impreciseness is related to values, concepts etc. which can not be precisely defined.

It is inevitable to deal with uncertainties in a proper way in order to achieve a reliable solution for the given problem. Uncertainties can be mathematically modeled by random variables, random sets, fuzzy sets, fuzzy random variables etc. depending on their source and nature. When one only has to deal with impreciseness of the form: X *can take real values between* v_L *and* v_R, interval computation can be used [1].

Systems of linear equations play an important role in many theoretical as well as practical, real world problems, just to mention structure mechanics, solid mechanics, heat transfer analysis, etc.. The original problem can be transformed into that of solving a system of linear equations in various ways. When imprecise information is present in the underlying problem, that impreciseness must be as well considered in the mentioned system of equations, which is now no longer *crisp*.

A system of linear equations Ax = b is called a System of Interval Linear Equations (denoted hereby as SILE) if the left side matrix A is an interval

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 150–159, 2006.
© Springer-Verlag Berlin Heidelberg 2006

matrix and the right side vector b is an interval vector. Solving SILEs has been the topic of many researches for decades. Different definitions for the solution of a SILE have been considered and various techniques have been successfully developed [2,5,6,9,10,11]. The algebraic solutions [6] of systems of interval linear equations are considered in this paper. It is well known that finding the exact algebraic solution of a given SILE is an NP-hard problem. To find the exact algebraic solution of a SILE of size n, one has to solve 2^n systems of (*crisp*) linear equations of size n. In many cases, such a solution may not exist at all. The aim of this paper is to provide some AI based techniques to approximate the *exact* algebraic solution, if any exists, or to find an *acceptable* one in the opposite case.

In the next section, some basic notions regarding interval computation together with the formulation of the SILE problem are provided. The SILE is then viewed in terms of an optimization problem. Next, the NN based and the GA based techniques to approximate the algebraic solution of SILEs are discussed in sections 3 and 4 respectively. Numerical results, comparison between the two proposed techniques can be found in section 5. Some final conclusions are drawn in section 6.

2 Systems of Interval Linear Equations

2.1 Basic Notions

A real interval number $\overline{a}$ can be considered as a pair $\left[a^{\mathrm{L}},\, a^{\mathrm{R}}\right]$, where a^{L}, a^{R} ($a^{\mathrm{L|R}} \in \mathbb{R}$, $a^{\mathrm{L}} \leq a^{\mathrm{R}}$) are called the left and right endpoints of $\overline{a}$. Precisely a real interval number $\overline{a}$ is a subset of $\mathbb{R}$:

$$\overline{a} \;=\; \left[a^{\mathrm{L}},\, a^{\mathrm{R}}\right] \;\equiv\; \left\{x \in \mathbb{R} : a^{\mathrm{L}} \leq x \leq a^{\mathrm{R}}\right\}$$

For sake of simplicity, real interval numbers are referred to hereby as interval numbers. The set of all interval numbers is denoted by $\mathbb{IR}$. Let $d : \mathbb{IR} \times \mathbb{IR} \rightarrow \mathbb{R}^+$ be defined as follows:

$$d\left(\overline{a}, \overline{b}\right) = \sqrt{\left(b_L - a_L\right)^2 + \left(b_R - a_R\right)^2} \tag{1}$$

then $\{\mathbb{IR}, d\}$ is a metric space.

Arithmetic operations in $\mathbb{R}$ can be extended to those in $\mathbb{IR}$. Namely, a binary operation $*$ (which can be either addition, subtraction, multiplication or division) between two real interval numbers $\overline{a}$ and $\overline{b}$ can be defined as:

$$\overline{a} * \overline{b} \;\equiv\; \left\{c = a * b : a \in \overline{a},\; b \in \overline{b}\right\} \tag{2}$$

where in the case of division, $\overline{b}$ is assumed not to contain zero. It can be easily observed that the right hand set in (2) is a real interval number as well. Arithmetic operations in $\mathbb{IR}$ can be expressed in terms of endpoint values, e.g.:

$$\begin{aligned}
\left[a^{\mathrm{L}},\, a^{\mathrm{R}}\right] \,+\, \left[b^{\mathrm{L}},\, b^{\mathrm{R}}\right] &= \left[a^{\mathrm{L}} + b^{\mathrm{L}},\, a^{\mathrm{R}} + b^{\mathrm{R}}\right] \\
\left[a^{\mathrm{L}},\, a^{\mathrm{R}}\right] \,-\, \left[b^{\mathrm{L}},\, b^{\mathrm{R}}\right] &= \left[a^{\mathrm{L}} - b^{\mathrm{R}},\, a^{\mathrm{R}} - b^{\mathrm{L}}\right] \\
\left[a^{\mathrm{L}},\, a^{\mathrm{R}}\right] \,\times\, \left[b^{\mathrm{L}},\, b^{\mathrm{R}}\right] &= \left[\min_{\alpha,\beta\in\{\mathrm{L,R}\}} \left\{a^{\alpha}b^{\beta}\right\}, \max_{\alpha,\beta\in\{\mathrm{L,R}\}} \left\{a^{\alpha}b^{\beta}\right\}\right] \\
\left[a^{\mathrm{L}},\, a^{\mathrm{R}}\right] \,/\, \left[b^{\mathrm{L}},\, b^{\mathrm{R}}\right] &= \left[\min_{\alpha,\beta\in\{\mathrm{L,R}\}} \left\{a^{\alpha}/b^{\beta}\right\}, \max_{\alpha,\beta\in\{\mathrm{L,R}\}} \left\{a^{\alpha}/b^{\beta}\right\}\right]
\end{aligned} \tag{3}$$

Denote by $\mathbb{R}_m^n$ the set of all $n \times m$ real matrices. Let $\mathrm{A} = [a_{ij}]_m^n \in \mathbb{R}_m^n$ and $\mathrm{B} = [b_{ij}]_m^n \in \mathbb{R}_m^n$. Let $\preceq \equiv \left\{(\mathrm{A},\mathrm{B}) : \forall i = \overline{1,n}, \forall j = \overline{1,m}, a_{ij} \leq b_{ij}\right\}$ be a relation in $\mathbb{R}_m^n$. Now let A^{L} and A^{R} be two real matrices so that $\mathrm{A}^{\mathrm{L}} \preceq \mathrm{A}^{\mathrm{R}}$, then the set:

$$\overline{\mathrm{A}} \equiv \left\{\mathrm{A} \in \mathbb{R}_m^n : \mathrm{A}^{\mathrm{L}} \preceq \mathrm{A} \preceq \mathrm{A}^{\mathrm{R}}\right\}$$

is called an real interval matrix (or shortly interval matrix). The set of all interval matrices over $\mathbb{R}_m^n$ is denoted in this paper by $\mathbb{IR}_m^n$. It can be observed from the above definition that in fact, interval matrices are matrices of interval numbers. Hence each interval matrix $\overline{\mathrm{A}} \in \mathbb{IR}_m^n$ can be represented as $\overline{\mathrm{A}} = [\overline{a}_{ij}]_m^n$ where $\forall i = \overline{1,n}, \forall j = \overline{1,m}, \overline{a}_{ij} \in \mathbb{IR}$.

Interval matrices over $\mathbb{R}_1^n$ are called interval vectors. The set of all such vectors are denoted by $\mathbb{IR}^n$. Let $\overline{\mathrm{x}} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)^{\mathrm{T}}, \overline{\mathrm{y}} = (\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_n)^{\mathrm{T}} \in \mathbb{IR}^n$. A metric function ρ in $\mathbb{IR}^n$ can be defined as:

$$\rho\left(\overline{\mathrm{x}}, \overline{\mathrm{y}}\right) = \sqrt{\sum_{i=1}^{n} d^2\left(\overline{x}_i, \overline{y}_i\right)} \tag{4}$$

Let $\overline{\mathrm{A}} = [\overline{a}_{ij}]_n^m \in \mathbb{IR}_n^m$ and $\overline{\mathrm{x}} = [\overline{x}_j]^n \in \mathbb{IR}^n$. The product of $\overline{\mathrm{A}}$ and $\overline{\mathrm{x}}$ is an interval vector in $\mathbb{IR}^m$ which is defined as follows:

$$\overline{\mathrm{A}}\overline{\mathrm{x}} = [\overline{a}_{ij}]_n^m [\overline{x}_j]^n \equiv \left[\sum_{k=1}^{n} \overline{a}_{ik} \times \overline{x}_k\right]^m \tag{5}$$

Using the endpoint notations, this product can be rewritten as:

$$\overline{\mathrm{A}}\overline{\mathrm{x}} = \left[\left[\sum_{k=1}^{n} \min_{\alpha,\beta\in\{\mathrm{L,R}\}} a_{ik}^{\alpha} x_k^{\beta}, \sum_{k=1}^{n} \max_{\alpha,\beta\in\{\mathrm{L,R}\}} a_{ik}^{\alpha} x_k^{\beta}\right]\right]^m. \tag{6}$$

2.2 Algebraic Solutions of Systems of Interval Linear Equations

Let us now consider the system of interval linear equations:

$$\overline{\mathrm{A}}x \,=\, \overline{\mathrm{b}}, \tag{7}$$

where $\overline{\mathrm{A}} \in \mathbb{IR}_n^n$ and $\overline{\mathrm{b}} \in \mathbb{IR}^n$. The algebraic solution of (7) is an interval vector $\overline{\mathrm{x}}^* \in \mathbb{IR}^n$, for which the product $\overline{\mathrm{A}}\overline{\mathrm{x}}^*$, according to (5) is equal to $\overline{\mathrm{b}}$ [6]. Finding $\overline{\mathrm{x}}^*$ is known as a complex problem due to (3). Moreover, in many cases

the algebraic solution may not exist at all. A more detailed analysis of algebraic solutions for (7) can be found in [6].

The aim of this work is to find an interval approximation of the exact solution $\overline{\mathrm{x}}^*$, if such exists. In the opposite case, the approximating interval vector may give an idea of the solution for some solvable SILE, whose left-side matrix and/or right-side vector are slightly different from $\overline{\mathrm{A}}$ and $\overline{\mathrm{b}}$, respectively. This will be considered in the context of an optimization task in the next section.

2.3 Solving SILE as an Optimization Problem

The SILE problem (7) can be viewed in terms of an optimization (minimization) problem with the following cost function F:

$$\begin{aligned} \mathrm{F}\left(\overline{\mathrm{x}}\right) &= \frac{1}{2}\rho^2\left(\overline{\mathrm{A}}\overline{\mathrm{x}}, \overline{\mathrm{b}}\right) \\ &= \frac{1}{2}\sum_{i=1}^{n}\left(\left(\left[\overline{\mathrm{A}}\overline{\mathrm{x}}\right]_i^{\mathrm{L}} - b_i^{\mathrm{L}}\right)^2 + \left(\left[\overline{\mathrm{A}}\overline{\mathrm{x}}\right]_i^{\mathrm{R}} - b_i^{\mathrm{R}}\right)^2\right) \end{aligned} \tag{8}$$

where as it was stated in (6):

$$\left[\overline{\mathrm{A}}\overline{\mathrm{x}}\right]_i^{\mathrm{L}} = \sum_{k=1}^{n} \min_{\alpha,\beta\in\{\mathrm{L},\mathrm{R}\}} a_{ik}^{\alpha} x_k^{\beta}$$

$$\left[\overline{\mathrm{A}}\overline{\mathrm{x}}\right]_i^{\mathrm{R}} = \sum_{k=1}^{n} \max_{\alpha,\beta\in\{\mathrm{L},\mathrm{R}\}} a_{ik}^{\alpha} x_k^{\beta}$$

If there exists an exact algebraic solution $\overline{\mathrm{x}}^*$ for (7) then such solution corresponds to the global minimum of the cost function $\mathrm{F}\left(\overline{\mathrm{x}}\right)$, i.e. $\mathrm{F}\left(\overline{\mathrm{x}}^*\right) = 0$. Hence one can approximate $\overline{\mathrm{x}}^*$ by minimizing F.

Due to the presence of min and max functions, F is not everywhere differentiable, and as a consequence gradient based optimization techniques cannot be directly applied. In this paper two different AI based approaches are proposed. In the first one, a neural network is involved to form a modified, differentiable cost function, which can be in turn minimized with a gradient based technique. In the second one, genetic algorithms are used. This will be the topics of the next two sections.

3 The NN-Based Technique

The approach to be presented here is based on the following observations: Firstly, if the multiplication of interval numbers defined as in (3) can be replaced by a differential function m_1 of the endpoints (i.e. $m_1 : \mathbb{R}^4 \to \mathbb{R}^2$), then the product of interval matrix and interval vector in (6) can be computed as well by another differential function $m_2 : \mathbb{R}^{2n(m+1)} \to \mathbb{R}^{2m}$. As the consequence, the cost function (8) can also be approximated by a differential function. Secondly, any sigmoid multilayer feedforward neural network with p input units and q output neurons can be interpreted as a differential function:

$$\mathcal{N} : \mathbb{R}^p \to \mathbb{R}^q$$

The main idea of this approach is to approximate the product of real interval numbers by use of a simple feedforward neural network and then to use that network to create a new differential cost function. Next this modified cost function is minimized by a gradient based technique. The approximating network $\mathcal{N}$ contains $p = 4$ input neurons, $q = 2$ output neurons and one hidden layer with K hidden neurons. All hidden and output neurons are sigmoid with the tanh activation function. The architecture of the network is presented in Fig. 1.

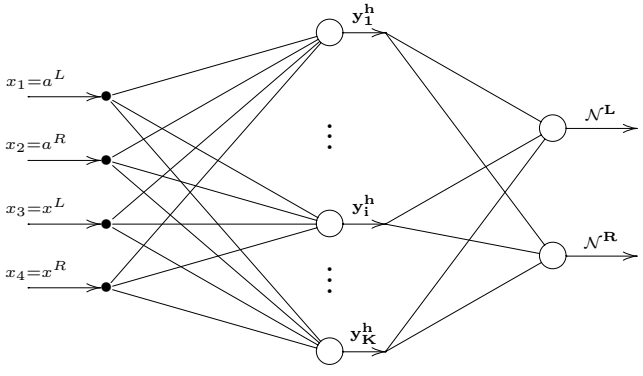


Fig. 1. Neural network for approximating the product of interval numbers

To compute the product of $\overline{a}$ and $\overline{x}$ their endpoint values (i.e. a^{L}, a^{R}, x^{L} and x^{R}) are taken as inputes to the network. The network produces two output values representing the left and the right endpoints of the interval product. This approximating network is hereby referred to as a function $\mathcal{N} = \left[\mathcal{N}^{\mathrm{L}}, \mathcal{N}^{\mathrm{R}}\right] : \mathbb{R}^4 \to \mathbb{R}^2$. By replacing the product of interval numbers with $\mathcal{N}$, a modified cost function $f : \mathbb{R}^{2n} \to \mathbb{R}^+$ is to be minimized:

$$\mathrm{f}\left(\mathrm{x}\right) = \frac{1}{2}\sum_{i=1}^{n}\left[\left(\sum_{k=1}^{n}\mathcal{N}^{\mathrm{L}}\left(a_{ik}^{\mathrm{L|R}}, x_k^{\mathrm{L|R}}\right) - b_i^{\mathrm{L}}\right)^2 + \left(\sum_{k=1}^{n}\mathcal{N}^{\mathrm{R}}\left(a_{ik}^{\mathrm{L|R}}, x_k^{\mathrm{L|R}}\right) - b_i^{\mathrm{R}}\right)^2\right] \tag{9}$$

where $\mathrm{x} = \left[x_1^{\mathrm{L}}, x_1^{\mathrm{R}}, x_2^{\mathrm{L}}, x_2^{\mathrm{R}}, \ldots, x_n^{\mathrm{L}}, x_n^{\mathrm{R}}\right]^{\mathrm{T}} \in \mathbb{R}^{2n}$ is the vector of the endpoints of the unknown interval vector $\overline{\mathrm{x}}$ in (8).

This modified cost function f can be now minimized using any of the well known gradient based methods. These methods require computations of the partial derivatives of f with respect to each of its variables $x_k^{\mathrm{L|R}}$, $k = \overline{1,n}$, i.e. $\partial f / \partial x_k^{\mathrm{L|R}}$. In consequence, one has to compute the partial derivatives

$\partial \mathcal{N}^{\mathrm{L|R}}/\partial x_k^{\mathrm{L|R}}$. As it was shown in our previous work [12], the computation of $\partial \mathcal{N}^{\mathrm{L|R}}/\partial x_k^{\mathrm{L|R}}$ can be facilitated by mean of a *dual* hybrid network $\mathcal{N}^d$, which can be built directly from $\mathcal{N}$.

4 The GA-Based Technique

The cost function introduced in (8) can be as well minimized with use of genetic algorithms ([4,7]), a particular class of evolutionary algorithms. GA based techniques are global search methods which do not require the gradient of the objective function to be computed. The only information used in the search process is the values of an appropriate fitness function. There are various versions of genetic algorithms. In this paper, the steady state strategy is applied. It will be described in more details in the following section.

4.1 The Steady State Algorithm

As it is common in GA approaches, the search process starts with a population of randomly generated individuals (genomes). At each generation it creates a temporary set of new individuals by performing selection, crossover and mutation on individuals from the current population. These offsprings are then added to this population and their fitness values are computed. At this point, the worst individuals are removed in order to return the population to its original size. A new individual, if not proven to be good enough may not make any contribution to the next population.

Denote by $\mathrm{P}(t)$ the population at step t, by S the number of individuals in the population. Let M be the number of new individuals additionally created at each iteration ($M < S$). Let T be the maximal number of iterations. The steady state strategy can be described in the following pseudo code:

```
BEGIN PROCEDURE SteadyStateGA
P(0) = {};
FOR i = 1 TO S DO
    Randomly generate an individual g(i);
    ComputeFitness(g(i)); //compute the fitness value for g(i)
    P(0) = P(0) + {g(i)};
END FOR;
FOR t = 1 TO T DO
    P(t) = P(t-1);
    FOR j = 1 TO M/2 DO
        f = DoSelection(P(t-1));  //perform selection on P(t-1)
        m = DoSelection(P(t-1));
        {b, s} = CrossOver(f,m);  //perform crossing over
        IF (b should be mutated) THEN
            Mutate(b);
        END IF;
        IF (s should be mutated) THEN
```

```
            Mutate(s);
        END IF;
        ComputeFitness(b); ComputeFitness(s);
        P(t) = P(t) + {b, s};
    END FOR;
    Remove M worst individual from P(t);
END FOR;
Return the best individual as the solution;
END PROCEDURE;
```

4.2 Genetic Operators

In order to be able to apply the above algorithm in solving systems of interval linear equations, one has to specify what the genomes are and to define the appropriate genetic operators. In this work, each genome is simply an interval vector in $\mathbb{IR}^n$, i.e. $\overline{\mathrm{x}} = \left(\left[x_i^{\mathrm{L}},\, x_i^{\mathrm{R}}\right]_{i=1}^{n}\right)^{\mathrm{T}}$ which represents a potential solution for (7).

Selection. Various selection schemes: roulette wheel selection, tournament selection, deterministic selection and stochastic remainder sampling selection can be applied. Selection is made based on the fitness values of genomes. Here the fitness values of genomes are computed from their associating cost function values (as defined in (8)) by applying a linear scaling scheme.

Crossover. Two mating methods - the single point crossover and the arithmetic crossover - are used during evolution. The aim of crossing over is to allow genetic material to be exchanged between individuals. Let $\overline{\mathrm{x}} = \left(\left[x_i^{\mathrm{L}},\, x_i^{\mathrm{R}}\right]_{i=1}^{n}\right)^{\mathrm{T}}$ and $\overline{\mathrm{y}} = \left(\left[y_i^{\mathrm{L}},\, y_i^{\mathrm{R}}\right]_{i=1}^{n}\right)^{\mathrm{T}}$ be two genomes that are selected for mating. In the single point crossover approach, a random crossing point $k = \overline{1,n}$ is chosen and two offsprings:

$$\overline{\mathrm{b}} = \left(\left[x_i^{\mathrm{L}},\, x_i^{\mathrm{R}}\right]_{i=1}^{k}, \left[y_j^{\mathrm{L}},\, y_j^{\mathrm{R}}\right]_{j=k+1}^{n}\right)^{\mathrm{T}}$$

$$\overline{\mathrm{s}} = \left(\left[y_i^{\mathrm{L}},\, y_i^{\mathrm{R}}\right]_{i=1}^{k}, \left[x_j^{\mathrm{L}},\, x_j^{\mathrm{R}}\right]_{j=k+1}^{n}\right)^{\mathrm{T}}$$

are generated. In the second approach, a random real number $p \in (0,1)$ is chosen and the new genomes $\overline{\mathrm{b}} = \left(\left[b_i^{\mathrm{L}},\, b_i^{\mathrm{R}}\right]_{i=1}^{n}\right)^{\mathrm{T}}$ and $\overline{\mathrm{s}} = \left(\left[s_i^{\mathrm{L}},\, s_i^{\mathrm{R}}\right]_{i=1}^{n}\right)^{\mathrm{T}}$ are produced, where for $\alpha \in \{\mathrm{L}, \mathrm{R}\}$:

$$b_i^{\alpha} = p.x_i^{\alpha} + (1-p)\,.y_i^{\alpha}$$
$$s_i^{\alpha} = (1-p)\,.x_i^{\alpha} + p.y_i^{\alpha}$$

Mutation. The aim of mutation is to maintain the diversity of genomes from generation to generation, which in turn prevents the search process from premature convergence in a local minimum. Let $\Delta(t,r)$ ($t \in \mathbb{N}$ and $r \in \mathbb{R}^+$ are the parameters) be a random number generator with the following properties:

1. $\Delta(t,r) \in [0,r]$, i.e. $\Delta(t,r)$ generates random numbers between 0 and r,
2. $\forall \epsilon > 0, \forall t_1, t_2 \in \mathbb{R}^+ : t_1 < t_2, \text{Prob}\{\Delta(t_2,r) < \epsilon\} > \text{Prob}\{\Delta(t_1,r) < \epsilon\}$. In other words, the probability of $\Delta(t,r)$ to produce a value close to 0 increases when t increases.

Let $[\mathrm{L}_i, \mathrm{R}_i] \subset \mathbb{R}$ be the domain of the i-th component of the genomes in the population. Let $\overline{\mathrm{x}} = \left(\left[x_i^{\mathrm{L}}, x_i^{\mathrm{R}}\right]_{i=1}^{n}\right)^{\mathrm{T}}$ be an individual selected for mutation at step t of the evolution process. For this given vector, a component $\left[x_j^{\mathrm{L}}, x_j^{\mathrm{R}}\right]$ is randomly chosen. This j-th component of $\overline{\mathrm{x}}$ is then replaced by $\left[\min\left\{\left(x_k^{\mathrm{L}}\right)', \left(x_k^{\mathrm{R}}\right)'\right\}, \max\left\{\left(x_k^{\mathrm{L}}\right)', \left(x_k^{\mathrm{R}}\right)'\right\}\right]$, where

$$\left(x_k^{\mathrm{L}}\right)' = x_k^{\mathrm{L}} + \Delta\left(t, \mathrm{R}_k - x_k^{\mathrm{L}}\right) \text{ or } \left(x_k^{\mathrm{L}}\right)' = x_k^{\mathrm{L}} - \Delta\left(t, x_k^{\mathrm{L}} - \mathrm{L}_k\right)$$
$$\left(x_k^{\mathrm{R}}\right)' = x_k^{\mathrm{R}} + \Delta\left(t, \mathrm{R}_k - x_k^{\mathrm{R}}\right) \text{ or } \left(x_k^{\mathrm{R}}\right)' = x_k^{\mathrm{R}} - \Delta\left(t, x_k^{\mathrm{R}} - \mathrm{L}_k\right)$$

In this paper, the following random number generator $\Delta(t,r)$ was applied:

$$\Delta(t,r) = r\left(1 - \omega^{1-t/T}\right)$$

Here T is the maximal number of populations, ω is a uniform random variable in $[0,1]$. Due to the properties of $\Delta(t,r)$, this mutation scheme allows the GA to search for the solution in the whole input space at the beginning phase, whereas as evolution goes on, this process becomes local.

5 Numerical Evaluation

To verify the NN based approach, 50 neural networks for approximating the product of interval numbers with 5 hidden neurons were trained. The training set is composed of 500 samples, whereas the validating set is composed of 300 samples. Both training and validating sample sets were randomly generated. After the training process all networks were tested against a set of 2000 randomly generated testing samples. The network with the best performance (in terms of the mean absolute square error) was chosen as the approximating network $\mathcal{N}$ to be used in the modified cost function (9). Next this modified cost function was minimized with use of the Scaled Conjugate Gradient Algorithm [8].

For the GA based approach, the size of the population S, the maximal number of iterations T, the probability of crossing over p_{C} as well as the probability of mutation p_{M} were chosen depending on the size of the problem. For instance, in the case when $n = 20$, these settings were: S = 200, T = 5000, $p_{\mathrm{C}} = 0.85$ and $p_{\mathrm{M}} = 0.1$.

The proposed techniques were applied to approximate the algebraic solution of various systems of interval linear equations (7) with various values of n. For each value of n, a set of 20 different configurations of $\overline{\mathrm{A}}$ and the exact algebraic solution $\overline{\mathrm{x}}^*$ were first randomly generated (each component of $\overline{\mathrm{A}}$ as well as of $\overline{\mathrm{x}}^*$ is an interval number in $[-1,1]$). The interval vectors $\overline{\mathrm{b}}$ were then computed according to: $\overline{\mathrm{b}} = \overline{\mathrm{A}}\overline{\mathrm{x}}^*$.

For each configuration of $\{\overline{\mathrm{A}}, \overline{\mathrm{b}}\}$, the simulation was repeated 10 times (i.e. with 10 different randomly chosen starting points in the case of the NN based approach and 10 initial populations in the case of the GA based technique). The solution with the smallest value of the (modified) cost function was chosen as the final solution for the system. In Table 1 the experimental results[1] for $n = 20$, in terms of the Mean Absolute Square Error (E) are shown:

Table 1. The min., max. and avg. MASE for $n = 20$

	NN based method	GA based method
Min.E	1.88e-2	1.99e-1
Max.E	1.53e-1	3.53e-1
Avg.E	6.41e-2	2.74e-1

Various numerical experiments showed that in the case of the GA based approach, the arithmetic crossover operator is slightly better than the single point one. If one compares the NN based and the GA based techniques, the first approach outperforms the later one taking into consideration the MASE. Moreover the number of cost function evaluations in the NN based approach is also smaller than that in the GA based approach.

6 Conclusions

As it was described in the paper, neural networks and genetic algorithms can be applied to approximate the algebraic solution of systems of interval linear equations. The original problem of solving SILE was transformed into a problem of minimization of a cost function. Due to the specific properties of interval arithmetics, this cost function was minimized either by using genetic algorithms or by employing a simple neural network for interval number multiplication.

It can be observed that in the case when there is no exact solution for a given SILE, a *possible solution* can still be found. The proposed approaches can also be used to solve the system (7) when $\overline{\mathrm{A}} \in \mathbb{IR}^{n\times m}$,$\overline{\mathrm{b}} \in \mathbb{IR}^{n}$ and $n \neq m$. With the use of neural networks and genetic algorithms, it is possible to parallelize the computation process that takes place in each neural unit, or during genome evaluation in the GA based case. This is an essential issue in solving complex, large scale problems.

It may be interesting to combine both GA based and NN based approaches into one single method. GA may be used at the beginning phase for global search, whereas the NN based technique may be used in the final, local search phase to improve the solution obtained by GA.

[1] In the case of GA based approach, the results presented here were obtained with the roulette wheel selection and arithmetic crossover involved.

References

1. Alefeld, G., Herzberger, J.: Introduction to Interval Computation. Academic Press, London, 1983.
2. Alefeld, G., Kreinovich, V., Mayer, G.: On the Solution Sets of Particular Classes of Linear Interval Systems. Journal of Computaional and Applied Mathematics, **152**, 2003, 1–15.
3. Bertoluzza, C., Gil, M.A., Ralescu, D.A. (eds.): Statistical Modeling, Analysis and Management of Fuzzy Data. Physical-Verlag, New York, 2002.
4. Goldberg, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning. Kluwer Academic Publishers, Boston, MA, 1989.
5. Hansen, E.: Bounding the solution of interval linear equations. SIAM Journal of Numerical Analysis, **29**(5), 1992, 1493–1503.
6. Markov, S.: An Iterative Method for Algebraic Solution to Interval Equations. Applied Numerical Mathematics, **30**, 1999, 225–239.
7. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, Berlin, 1996.
8. Möller, M.F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks, **6**, 1993, 525-533.
9. Neumaier, A.: Interval Methods for Systems of Linear Equations. Cambridge University Press, 1990.
10. Ning, S., Kearfott, R.B.: A comparison of some methods for solving linear interval equations. SIAM Journal of Numerical Analysis, **34**(4), 1997, 1289–1305.
11. Shary, S.P.: Algebraic Approach in the Outer Problem for Interval Linear Equations. Reliable Computing, **3**, 1997, 103–135.
12. Viet, N.H., Kleiber, M.: Neural Network in Solving Systems of Interval Linear Equations. Foundation of Computing and Decision Sciences, **30**(3), 2005, 263–277.
13. Zimmerman, H.J.: Fuzzy Set Theory and its Applications. Kluwer Academic Press, Boston, 1991.

A Fast and Numerically Robust Neural Network Training Algorithm

Youmin Zhang

Department of Computer Science and Engineering
Aalborg University Esbjerg
Niels Bohrs Vej 8, 6700 Esbjerg, Denmark
ymzhang@cs.aaue.dk

Abstract. A fast and numerically robust algorithm for training feedforward neural networks (FNNs) using a recursive prediction error (RPE) method, along with an adaptively-adjustable time-varying forgetting factor technique, is presented first. Then a U-D factorization-based RPE (UD-RPE) algorithm is proposed to further improve the training rate and accuracy of the FNNs. In comparison with the backpropagation (BP) and existing RPE based training algorithms, the proposed UD-RPE can provide substantially better results in fewer iterations with fewer hidden nodes and improve significantly convergence rate and numerical stability. In addition, it is less sensitive to start-up parameters, such as initial weights and covariance matrix. It has also good model prediction ability and need less training time. The effectiveness of the proposed algorithm is demonstrated via two nonlinear systems modeling and identification examples.

1 Introduction

The classical method for training a multilayer FNN is the steepest descent backpropagation (BP) algorithm. Although successfully used in many cases, the BP algorithm suffers from a number of shortcomings. One such shortcoming is the slow convergence rate. Furthermore, the learning rate and the momentum term have to be tuned in a heuristic manner to obtain quick convergence. An improper choice of these parameters may incur problems of stability, slow convergence, or lead to only local minimal solutions. These facts motivated the research for faster learning algorithms. Several modifications have been proposed in the literature, in an attempt to speed up the convergence. The earlier research falls roughly into two categories. The first involves the development of heuristic techniques, which arose out of a study of the distinctive performance of the standard BP algorithm. These heuristic techniques include such ideas as varying learning rate parameter, using momentum and rescaling variables [5]. It has been shown that the learning rate can be improved by setting a large learning rate parameter. Unfortunately, this also increases the possibility for the algorithm to be trapped in a local minimum, to oscillate around the global minimum, or to numerically overflow according to author's experience. Other research has focused on numerical optimization techniques other than the gradient descent

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 160–169, 2006.
© Springer-Verlag Berlin Heidelberg 2006

method to accelerate convergence. Meanwhile, the learning algorithm based optimal estimation has also received considerable attention. Chen and Billings [2,3] suggested the use of the recursive prediction error (RPE) method for the nonlinear system modeling and identification based on the multiple layer perceptron (MLP) network. These improve the convergence rate considerably. But the RPE method increases computational cost and may suffer from numerical instability [6]. Some other estimation-based training algorithms, such as extended Kalman filter (EKF) [8,9] and recursive least squares method [4], have also been proposed. These algorithms improve considerably the convergence of the BP algorithm and exhibit good performance. The storage and computational cost have been more or less reduced. However, the numerical stability is not guaranteed, which may degrade convergence. The numerical stability, convergence rate, learning accuracy and the computational requirement of these algorithms can be improved further by using the numerically robust U-D factorization of covariance matrices [6,9] combined with a time-varying forgetting factor.

In view of the above facts and to obtain better convergence and accuracy of the training algorithm, an adaptively-adjustable time-varying forgetting factor technique, combined with the RPE algorithm and its two modified implementations, for training FNNs is presented in this paper first. Then, a much faster and more robust training algorithm is proposed by using forgetting factor technique together with a U-D factorization-based RPE (UD-RPE) algorithm.

The paper is organized as follows. In Section 2, the RPE and two modified RPE algorithms, along with the developed adaptively-adjustable time-varying forgetting factor technique, are presented. The UD-RPE is proposed in Section 3 for nonlinear systems with multiple as well as single output, In Section 4, simulation results for comparison of the three algorithms for nonlinear systems modeling and identification are given, and finally Section 5 presents the main conclusions.

2 RPE Algorithms with Adaptively-Adjustable Time-Varying Forgetting Factor

2.1 RPE Algorithm and Its Variants

The recursive prediction error (RPE) algorithm, referred to as the **RPE(C)** algorithm, is given as follows [2,3]:

$$\begin{bmatrix} \hat{\mathbf{y}}(t) \\ \Psi(t) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}}[\mathbf{x}(t); \hat{\mathbf{\Theta}}(t-1)] \\ G[\mathbf{x}(t); \hat{\mathbf{\Theta}}(t-1)] \end{bmatrix} \tag{1}$$

$$\varepsilon(t) = \mathbf{y}(t) - \hat{\mathbf{y}}(t) \tag{2}$$

$$R(t) = R(t-1) + \gamma(t)[\Psi(t)\Lambda^{-1}\Psi^T(t) + \delta I - R(t-1)] \tag{3}$$

$$\hat{\mathbf{\Theta}}(t) = \hat{\mathbf{\Theta}}(t-1) + \gamma(t)R^{-1}(t)\Psi(t)\Lambda^{-1}\varepsilon(t) \tag{4}$$

where $\hat{\mathbf{\Theta}}(t)$ denotes the estimate of $\mathbf{\Theta}$ at t and $\gamma(t)$ is the gain at t.

In practice, the above algorithm is not implemented in a straightforward way with matrix inverses R^{-1} and Λ^{-1}. Based on the matrix inversion lemma, an equivalent form of (3)-(4) can be obtained [3,6] (with $\delta = 0$) as following:

$$P(t) = P(t-1) - P(t-1)\Psi(t)[\Psi^T(t)P(t-1)\Psi(t) + \lambda(t)\Lambda]^{-1}\Psi^T(t)P(t-1)\}/\lambda(t) \quad (5)$$

$$\hat{\mathbf{\Theta}}(t) = \hat{\mathbf{\Theta}}(t-1) + P(t)\Psi(t)\Lambda^{-1}\varepsilon(t) \quad (6)$$

where

$$P(t) = \gamma(t)R^{-1}(t) \quad \text{and} \quad \lambda(t) = \gamma(t-1)(1-\gamma(t))/\gamma(t) \quad (7)$$

The simplest choice for Λ is I but Λ can also be chosen to be time-varying as follows [3]:

$$\Lambda(t) = \Lambda(t-1) + \gamma(t)[\varepsilon(t)\varepsilon^T(t) - \Lambda(t-1)] \quad (8)$$

Note that the factor $P(t-1)\Psi(t)$ appears three times in the computation of the $P(t)$. By utilizing this feature and taking advantage of symmetry and positive definiteness of the covariance matrix $P(t-1)$, the following modified RPE algorithm, referred to as the **RPE(M)** algorithm, can be obtained:

$$\begin{aligned}
\varepsilon(\mathbf{t}) &= \mathbf{y}(t) - \hat{\mathbf{y}}(t) \\
L(t) &= P(t-1)\Psi(t) \\
S(t) &= \Psi^T(t)P(t-1)\Psi(t) + \lambda(t)\Lambda \\
P(t) &= [P(t-1) - L(t)S^{-1}(t)L^T(t)]/\lambda(t) \\
\hat{\mathbf{\Theta}}(t) &= \hat{\mathbf{\Theta}}(t-1) + P(t)\Psi(t)\Lambda^{-1}\varepsilon(t)
\end{aligned} \quad (9)$$

It can be shown that this modified RPE implementation is computationally efficient, has slightly higher numerical stability, due to the use of symmetry and positive definiteness of $P(t-1)$, and a faster rate of convergence. This will be shown later via simulation.

Moreover, using the expression for $P(t)$ in (7), the gain $L(t)$ can be defined and written in a computationally more convenient form

$$\begin{aligned}
L(t) &= \gamma(t)R^{-1}(t)\Psi(t)\Lambda^{-1} \\
&= P(t)\Psi(t)\Lambda^{-1} \\
&= P(t-1)\Psi(t)[\Psi^T(t)P(t-1)\Psi(t) + \lambda(t)\Lambda]^{-1}
\end{aligned} \quad (10)$$

Then (2)-(6) can also be written in the following modified form, referred to as the **RPE(L)** algorithm

$$\begin{aligned}
\varepsilon(t) &= \mathbf{y}(t) - \hat{\mathbf{y}}(t) \\
S(t) &= \Psi^T(t)P(t-1)\Psi(t) + \lambda(t)\Lambda \\
L(t) &= P(t-1)\Psi(t)S^{-1}(t) \\
P(t) &= [P(t-1) - L(t)S(t)L^T(t)]/\lambda(t) \\
\hat{\mathbf{\Theta}}(t) &= \hat{\mathbf{\Theta}}(t-1) + L(t)\varepsilon(t)
\end{aligned} \quad (11)$$

In the above algorithms, $\lambda(t)$ is a time-varying forgetting factor and can be chosen as follows [6]

$$\lambda(t) = \lambda_0\lambda(t-1) + (1-\lambda_0) \quad (12)$$

where the rate λ_0 and the initial value $\lambda(0)$ are design parameters. Typical values are $\lambda_0 = 0.99$ and $\lambda(0) = 0.95$. If $\lambda(t) \leq 1$, then $P(t)$ will keep larger than the case when $\lambda(t) = 1$ and will not tend to zero, and the RPE algorithm will always be altered to tracking the changes in the system.

RPE(C), RPE(M) and RPE(L) are algebraically equivalent. Their differences lie in the computation of $P(t)$ from $P(t-1)$. RPE(M) and RPE(L) are computationally less demanding. Their difference is the definition of $L(t)$. The computational requirements and numerical stability of RPE(M) and RPE(L) are improved due to the use of symmetry and positive definiteness of the $P(t)$.

2.2 The Adaptive Adjustment of Time-Varying Forgetting Factor

The introduction of time-varying forgetting factor in identification algorithms makes it possible for a recursive identification to track the time-varying dynamic systems [6]. This feature can be exploited for the training of FNNs discussed in this paper. When the training accuracy reaches a specified accuracy, corresponding to the steady-state of the identification, smaller weights for current input and output should be given and, therefore, the forgetting factor should adaptively change to 1 so that a fine adjustment of the weights of the FNN can be acquired. On the other hand, if the specified accuracy is not reached in the subsequent training iterations, the time-varying factor should be in effect so as to speed up the convergence rate. In this way, both a fast convergence and a high learning accuracy can be obtained simultaneously. This technique is important for improving simultaneously the convergence rate and learning accuracy of the RPE-based learning algorithms. The adaptive adjustment process is implemented by a logic based on the detection of a "large" training error.

The root mean square errors (RMSE) between the estimated and the actual output (prediction error) of the FNN can be used as a measure of the accuracy of the training algorithm, which is defined as

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{j=1}^{N_{iter}} \sum_{i=1}^{N} \varepsilon_j^2(i)} \tag{13}$$

where N_{iter} denotes the number of iterations for FNN while N denotes the number of training samples. When the RMSE is larger than some specified training accuracy σ (e.g., $\sigma = 10^{-3}$ or $\sigma = 10^{-4}$), $\lambda(t)$ should be smaller but close to 1, as computed by (12); otherwise, set $\lambda(t) = 1$. That is

$$\lambda(t) = \begin{cases} \lambda_0 \lambda(t-1) + (1-\lambda_0) & \text{if RMSE} \geq \sigma \\ 1 & \text{if RMSE} < \sigma \end{cases} \tag{14}$$

The threshold σ should be chosen to have a balanced training accuracy and smoothness of the training process.

This "adaptively-adjustable time-varying forgetting factor" technique can smooth out weight changes and improve significantly the convergence rate and training accuracy. However, the faster and numerically robust learning algorithm

can be obtained by the combination of the above forgetting factor technique with the U-D factorization of the covariance matrix to be described in next section.

3 U-D Factorization-Based RPE Training Algorithm

In the above RPE algorithms, $P(t)$ plays an important role. Its computational load and numerical stability are a major consideration of the algorithm implementation, especially for the case of real-time application of the algorithm. Due to the finite word length (in particular, the truncation errors) of a computer, the algorithm may lead to the loss of symmetry and positive definiteness of $P(t)$, or even divergence, even though theoretically $P(t)$ is always positive definite. Several covariance matrix factorization methods, such as square root method (Cholesky decomposition), U-D factorization [1], and singular value decomposition (SVD) have been developed to improve the computation of $P(t)$. The U-D factorization method is the most popular one [1,6] due to its computational efficiency. In this method, the covariance $P(t)$ is decomposed into $P(t) = U(t)D(t)U^T(t)$, where $U(t)$ and $D(t)$ are a unit-upper triangular matrix and a diagonal matrix, respectively. Then, $U(t)$ and $D(t)$, instead of $P(t)$, are updated at every time step. This decomposition guarantees the positive-definiteness and symmetry of $P(t)$, and thus high estimation accuracy and robustness can be attained [1,6,9]. In view of the superiority of the U-D factorization, a U-D factorization-based RPE algorithm is proposed to improve the performance of the training algorithm for FNNs.

3.1 Multiple Input Single Output (MISO) Case

Suppose $P(t-1)$ has been U-D decomposed at time $t-1$:

$$P(t-1) = U(t-1)D(t-1)U^T(t-1)$$

Eq. (5) then becomes

$$\begin{aligned} P(t) &= \frac{1}{\lambda(t)}\{U(t-1)D(t-1)U^T(t-1) \\ &\quad - \frac{U(t-1)D(t-1)U^T(t-1)\Psi(t)\Psi^T(t)U(t-1)D(t-1)U^T(t-1)}{\Psi^T(t)U(t-1)D(t-1)U^T(t-1)\Psi(t)+\lambda(t)\Lambda}\} \quad (15) \\ &= \frac{1}{\lambda(t)}U(t-1)\left[D(t-1) - \frac{D(t-1)\mathbf{e}\ \mathbf{e}^T D(t-1)}{\mathbf{e}^T D(t-1)\mathbf{e}+\lambda(t)\Lambda}\right]U^T(t-1) \quad (16) \\ &= \frac{1}{\lambda(t)}U(t-1)\left[D(t-1) - \frac{\mathbf{g}\ \mathbf{g}^T}{\beta}\right]U^T(t-1) \quad (17) \end{aligned}$$

where

$$\mathbf{e} = U^T(t-1)\Psi(t) \quad (18)$$

$$\mathbf{g} = D(t-1)\mathbf{e} \quad (19)$$

$$\beta = \mathbf{e}^T D(t-1)\mathbf{e}+\lambda(t)\Lambda = \mathbf{e}^T\mathbf{g}+\lambda(t) \quad (20)$$

and $\Psi(t)$, $\mathbf{e}$ and $\mathbf{g}$ are n_θ-vectors for an FNN with single output; Λ is equal to 1 and β is a scalar.

If $\bar{U}(t)$ and $\bar{D}(t)$ are U-D factors for the matrix $[D(t-1) - \frac{\mathbf{g}\,\mathbf{g}^T}{\beta}]$, that is

$$[D(t-1) - \frac{\mathbf{g}\,\mathbf{g}^T}{\beta}] = \bar{U}(t)\bar{D}(t)\bar{U}^T(t) \tag{21}$$

then the U-D factors for $P(t) = U(t)D(t)U^T(t)$ can be given as:

$$U(t) = U(t-1)\bar{U}(t) \tag{22}$$

$$D(t) = \bar{D}(t)/\lambda(t) \tag{23}$$

where a detailed deviation of $\bar{U}(t)$ and $\bar{D}(t)$ can be found in [1,6].

Let e_i and $g_i, i = 1, ..., n_\theta$, be the elements of $\mathbf{e}$ and $\mathbf{g}$, respectively. Further, let $U_{ij}(t)$ and $D_{ij}(t)$ represent the (i,j)th element of $U(t)$ and $D(t)$, respectively. Then, the implementation of U-D factorization form of the RPE(C) algorithm can be given as follows:

Step 1. Compute $\mathbf{e} = [e_1, ..., e_{n_\theta}]^T := U^T(t-1)\Psi(t)$, $\mathbf{g} = [g_1, ..., g_{n_\theta}]^T := D(t-1)\mathbf{e}$, $\alpha_0 := \lambda(t)$

Step 2. For $j = 1, ..., n_\theta$, compute

$$\begin{cases} \alpha_j := \alpha_{j-1} + e_j g_j \\ D_{jj}(t) := D_{jj}(t-1)\alpha_{j-1}/\alpha_j\lambda(t) \\ \nu_j := g_j \\ \mu_j := -e_j/\alpha_{j-1} \end{cases} \tag{24}$$

For $i = 1, ..., j-1$, compute

$$\begin{cases} U_{ij}(t) := U_{ij}(t-1) + \nu_i\mu_j \\ \nu_i := \nu_i + U_{ij}(t-1)\nu_j \end{cases} \tag{25}$$

Step 3.

$$L(t) = [\nu_1, ..., \nu_{n_\theta}]^T/\alpha_{n_\theta} \tag{26}$$

Step 4.

$$\hat{\mathbf{\Theta}}(t) = \hat{\mathbf{\Theta}}(t-1) + L(t)\varepsilon(t) \tag{27}$$

where the scalar α_{n_θ} obtained after the n_θ-th cycle of Step 2 is the "innovation variance", i.e., $\alpha_{n_\theta} = \Psi^T(t)P(t-1)\Psi(t) + \lambda(t)$.

3.2 Multiple Input Multiple Output (MIMO) Case

For MIMO systems, one approach is to transform the problem into a sequence of scalar problems [1,6]. Based on this idea, a UD-RPE for multiple (vector) output can be obtained by processing the m-dimensional output with m sequential applications of the above single output UD-RPE algorithm. Due to the space limitation, the detailed algorithm is omitted here.

4 Application to System Modeling and Identification

In this section, simulation results for two nonlinear system identification problems are presented. Performance comparison of the proposed UD-RPE, RPEs and the BP algorithm is carried out.

Example 1. Consider the following nonlinear system:

$$y(t) = 2.0\frac{x(t)}{1+x^2(t)} \tag{28}$$

where $y(t) \in [-1, 1]$ when $x(t) \in [-10, 10]$. The UD-RPE, RPEs and BP algorithms were employed to train a neural network with an architecture of 1-10-1 consisting of a total of 30 adjustable weights to approximate (28).

Training and prediction accuracy. Table 1 shows the mean and standard deviation (s.d.) of training and prediction output errors of the three algorithms with or without the forgetting factor. Fig. 1 illustrates the RMSE and output error curves of different algorithms. It is clear that the training and prediction accuracy as well as the convergence of the UD-RPE algorithm are much better than those of the RPEs and BP algorithms. The RPEs algorithms also have much better accuracy and convergence than BP. In the BP algorithm, the learning rate and momentum constant are selected as $\eta_w = 0.2$ and $\rho_w = 0.2$. There are some oscillations in Fig. 1 for RPE(C) and RPE(L) due to the problem of numerical stability and the sensitivity to the time-varying forgetting factor. The RPE(M) gives slightly smoother and faster convergence than RPE(L) and RPE(C).

Table 1. Comparison of training and prediction accuracy of three algorithms

		BP		RPE			UD-RPE	
		$\eta_w = 0.2$	$\lambda(t) = 1$	RPE(M)	RPE(L)	RPE(C)	$\lambda(t) = 1$	$\lambda(t)$
learning	mean	-8.2208e-1	-6.0577e-6	9.6370e-6	-1.3220e-6	-1.2576e-6	3.0063e-6	4.4357e-7
	s.d.	7.5517e-1	3.7639e-3	3.1881e-5	6.4208e-5	4.2196e-5	2.1238e-3	5.9914e-6
prediction	mean	6.0716e-1	2.7231e–3	-4.8791e-4	-3.1823e-4	2.0132e-3	8.8712e-3	-1.7920e-4
	s.d.	7.6116e-1	6.5024e-3	1.1455e-3	7.6485e-4	5.0064e-3	1.9154e-2	4.9411e-4

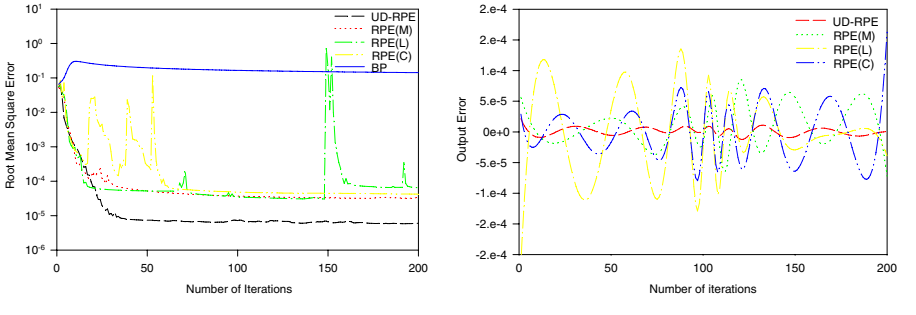


Fig. 1. Comparison of RMSE and output errors of three algorithms

The influence of number of hidden nodes. Fig. 2 illustrates the RMSE curves from the three algorithms with different numbers of hidden nodes. It is clear that better learning accuracy and convergence rate were obtained by the UD-RPE with just 7 nodes than RPE(M) with 10 nodes, and the UD-RPE with only 2 neurons has much better results than the BP with 10 neurons. In addition, the RPE(M) has much better results than BP. This demonstrates the superiority of the UD-RPE algorithm. It also means that, to obtain a specific modeling accuracy, fewer hidden neurons and less computation time are needed by the UD-RI

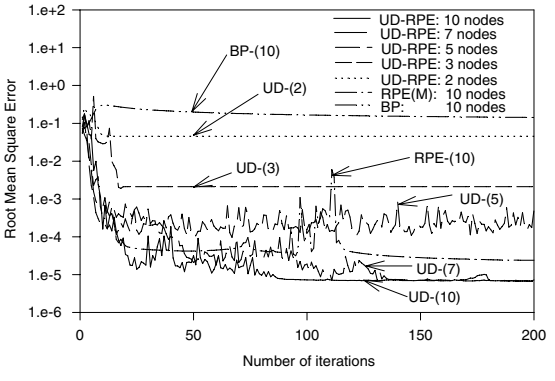


Fig. 2. Performance comparison of different hidden nodes

The effectiveness of "variable" forgetting factor. As discussed previously, forgetting factor is an important design parameter which assures the fast convergence and high training accuracy of the proposed training algorithms. Fig. 3 shows the RMSE curves for different specification of training accuracy using different "variable" forgetting factors. It can be observed that the RPE(M) is sensitive to the values of the forgetting factor. As shown in Fig. 3, the RMSEs are quite different for different specifications $10^{-3}, 10^{-4}$ and 10^{-5}. A large oscillation exists in RPE(M) when the specified accuracy in RMSE is smaller than 10^{-5}. On the other hand, the learning accuracy will be limited to a large bound if the RMSE is chosen as 10^{-3}, although the convergence is smooth. In contrast, the UD-RPE is not sensitive to forgetting factor, and a much higher learning accuracy and smooth convergence can be acquired even when the accuracy is specified as smaller than 10^{-6}.

Convergence and computation complexity. As a comparison of the convergence rate and computational requirement for different algorithms to reach the same identification accuracy, Table 2 lists the required iteration numbers of the three algorithms for the RMSE of $10^{-2}, 10^{-3}, 10^{-4}$ and 10^{-5}, respectively. In Table 2, "-" denotes that the corresponding algorithm has not reached the expected accuracy within 200,000 iterations for BP and 2000 iterations for RPEs and UD-RPE, respectively. Table 3 lists the computation time per iteration of the three algorithms for the cases with different hidden nodes.

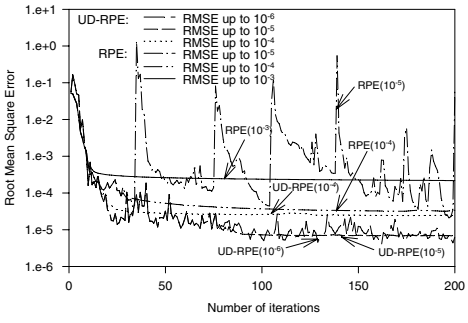


Fig. 3. Performance comparison of different forgetting factors and specified accuracy

Table 2. Iteration number/computation time (second) for specified accuracy

RMSE	BP	RPE				UD-RPE	
		$\lambda(t)=1$	RPE(M)	RPE(L)	RPE(C)	$\lambda(t)=1$	$\lambda(t)$
10^{-2}	-	16/35	6/12	7/12	14/279	17/17	6/7
10^{-3}	-	-	10/21	14/24	24/489	-	10/11
10^{-4}	-	-	27/58	53/91	-	-	17/18
10^{-5}	-	-	-	-	-	-	77/80

Table 3. Computation time per iteration with different hidden node numbers

n_h	BP	RPE(M)	RPE(L)	RPE(C)	UD-RPE
10	0.035	2.13	1.71	19.49	1.035
7	0.03	1.06	0.85	7.08	0.52
5	0.02	0.54	0.44	2.65	0.27
3	0.015	0.21	0.16	0.67	0.11
2	0.01	0.09	0.08	0.23	0.05

It is found that the UD-RPE gives the best convergence. Although the computation time per iteration for UD-RPE and RPEs are larger than BP, the computation time required by UD-RPE for convergence to a specified accuracy is much smaller than RPEs and BP.

Example 2. Consider the MIMO system given in [7]:

$$\begin{bmatrix} y_1(t+1) \\ y_2(t+1) \end{bmatrix} = \begin{bmatrix} \frac{y_1(t)}{1+y_2^2(t)} \\ \frac{y_1(t)y_2(t)}{1+y_2^2(t)} \end{bmatrix} + \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \tag{29}$$

An FNN with an architecture of 4-20-2 was used to model and identify the above MIMO system. Table 4 shows the mean and s.d. of training and prediction output errors of the three algorithms. It is obvious that the proposed UD-RPE has the fastest convergence and highest accuracy. RPE(M) has faster convergence than BP.

Table 4. Modeling and identification results comparison with three algorithms

Results		y_1		y_2	
Algorithms		Mean	s.d.	Mean	s.d.
Learning	UD-RPE	$-4.5249969e^{-6}$	$9.4609054e^{-4}$	$-6.3272516e^{-6}$	$1.3137960e^{-3}$
	RPE(DP)	$1.3641758e^{-4}$	$1.7913972e^{-2}$	$1.2526319e^{-4}$	$2.1014970e^{-2}$
	BP	$2.0058365e^{-2}$	$1.3416174e^{-1}$	$1.3929651e^{-3}$	$6.7967173e^{-1}$
Generalization	UD-RPE	$-4.3691909e^{-3}$	$1.7857307e^{-1}$	$-2.9752850e^{-2}$	$1.9580073e^{-1}$
	RPE(DP)	$-4.3006508e^{-2}$	$2.8429252e^{-1}$	$1.2770477e^{-2}$	$3.1416870e^{-1}$
	BP	$-1.9512380e^{-1}$	$8.7109413e^{-1}$	$1.2051304e^{-1}$	$1.0580563e^{0}$

5 Conclusions

Combined with an adaptively-adjustable time-varying forgetting factor technique, two fast learning algorithms using the recursive prediction error (RPE) method and the U-D factorization-based RPE for training FNNs have been proposed. It has been demonstrated that the UD-RPE algorithm with the adaptive forgetting factor greatly improves the convergence rate, numerical stability, and provides much more accurate results in fewer iterations with fewer hidden nodes. In addition, it is less sensitive to the choice of initial weights, initial covariance matrix. It has also been illustrated that the proposed algorithm can be used successfully for modeling and identification of highly nonlinear systems. The algorithm proposed here is not restricted to networks of a specific topology. It can be used effectively for other types of neural networks.

References

1. G. J. Bierman. *Factorization Methods for Discrete Sequential Estimation*. Academic Press, New York, 1977.
2. S. Chen and S. A. Billings. Neural networks for nonlinear dynamical system modeling and identification. *Int. J. Control*, 56(2):319–346, 1992.
3. S. Chen, S. A. Billings, and P. M. Grant. Non-linear system identification using neural networks. *Int. J. Control*, 51(6):1191–1214, 1990.
4. S. Chen, C. F. N. Cowan, and P. M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. Neural Networks*, 2(2):302–309, 1991.
5. S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, Upper Saddle River, NJ, 1999.
6. L. Ljung and T. Soderstrom. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, MA, 1983.
7. K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks*, 1(1):4–27, Mar. 1990.
8. S. Singhal and L. Wu. Training feed-forward networks with the extended Kalman algorithm. In *Proc. Int. Conf. on ASSP*, pages 1187–1190, 1989.
9. Y. M. Zhang and X. R. Li. A fast U-D factorization-based learning algorithm with applications to nonlinear system modeling and identification. *IEEE Trans. Neural Networks*, 10(4):930–938, July 1999.

On Interpretation of Non-atomic Values and Induction of Decision Rules in Fuzzy Relational Databases

Rafal A. Angryk

Department of Computer Science
Montana State University
Bozeman, MT 59717-3880, USA
angryk@cs.montana.edu

Abstract. In this paper, we propose two new ways to interpret uncertain information reflected by non-atomic descriptors. We focus our research on data stored in a proximity-based fuzzy relational database as the database provides convenient mechanisms for recording and interpretation of uncertain information. In proximity-based fuzzy databases the lack of certainty about obtained information can be reflected via insertion of non-atomic attribute values. In addition, the database extends classical equivalence relations with fuzzy proximity relations, which provide users with interesting analytical capabilities. In this paper we concentrate on both of these properties when proposing new approaches to interpretation of non-atomic values for decision making purposes.

1 Introduction

The aptitude to make correct decisions is one of crucial human skills. According to the surveys published by portal KDnuggets [1], extraction of decision rules was the most frequently used data mining technique for the last five years. In real life, imperfect information occurs frequently (e.g. the lack of suitable precision of measuring instruments, subjective judgments of human beings) in multiple areas (e.g. customer surveys, genotype characteristics, police-reports). An ability to generate decision rules from such data has crucial importance for many real-world applications.

Currently available standard decision tree algorithms (e.g. ID3, C4.5, J48 [10-13, 16]) do not allow processing of non-atomic variables, as they focus only on statistically comparable entities. In this work we concentrate our attention on the similarity-based interpretation of non-atomic descriptors and extraction of decision trees from such data stored in a fuzzy relational database.

In the next section, we provide an overview of the proximity based fuzzy relational database model, which we used in our research, and briefly discuss entropy-based decision tree algorithms. In the sec. 3, we present two ways to interpret non-atomic categorical values and discuss a method letting us utilize the fuzzy proximity relation to support precise decision tree induction. Finally, in sec. 4, we briefly summarize conclusions coming from our investigation.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 170–181, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Related Work

The similarity-based fuzzy model of a relational database, proposed originally by Buckles and Petry [2-3], is an extension of the ordinary relational database [4]. The fuzzy model (based on the max-min composition of a fuzzy similarity relation, proposed by Zadeh in [5]), was further extended by Shenoi and Melton [6-7] using the concept of proximity relations (derived from work published by Tamura et al. [8]). An interesting discussion of fuzzy relational database properties was recently presented by Kumar De et al. in [9].

There are two fundamental properties of fuzzy relational databases: (1) acceptance of non-atomic domain values when characterizing attributes of recorded entities, and (2) generation of multi-level equivalence classes based on the domain-specific and expert-specified fuzzy relations applied in the place of traditional equivalence relations.

As mentioned above, each attribute value of the fuzzy database record is allowed to be a subset of the whole base set of attribute values describing a particular domain. Formally, if we denote a set of acceptable attribute values as D_j, and we let d_{ij} symbolize a j^{th} attribute value of the i^{th} tuple, instead of $d_{ij} \in D_j$ (required by Codd's 1NF [4]), the more general case $d_{ij} \subseteq D_j$ is allowed in fuzzy databases. That is, any member of the power set of accepted domain values can be used as an attribute value except the null set. This lets us represent inexactness arising from the original source of information. In the case when a particular entity's attribute cannot be clearly characterized by an atomic descriptor, such uncertainty can be reflected by insertion of multiple attribute values.

The second feature characterizing proximity-based fuzzy databases is substitution of the ordinary equivalence relation, defining the notion of redundancy in the ordinary database, with an explicitly declared proximity relation (denoted by P_α) of which both the identity and fuzzy similarity relations are actually special cases. Since the original definition of fuzzy proximity relations was only reflexive and symmetric, which is not sufficient to effectively replace the classical equivalence relation, the transitivity of proximity relation was added [6-7]. It was achieved by adding to the original definition of the fuzzy proximity relation a transitivity property, generated by sequences of similarities, proposed by Tamura et al. [8], and known as similarity chains. Such extended relation is often referred as α-proximity relation (symbol: P_α^+).

Each of the attributes in the fuzzy database has its own proximity table, which includes the degrees of proximity between all values occurring for the particular attribute. An example of such table for the domain COUNTRY is presented in Tab. 1.

The proximity table can be transformed using Tamura's similarity chains [8] to represent an α-proximity relation, which has properties identical to the fuzzy similarity relation specified by Zadeh [5]. Results of this transformation are in Tab. 2.

Now the disjoint classes of attribute values, considered to be equivalent at a specific α-level, can be extracted from the Table 2 (they are marked by shadings). Such separation of the equivalence classes arises mainly due to the sequential

Table 1. Proximity Table for domain COUNTRY [15]

	Canada	*USA*	*Mexico*	*Colomb.*	*Venezue.*	*Australia*	*N. Zlnd.*
Canada	1.0	0.8	0.5	0.1	0.1	0.0	0.0
USA	0.8	1.0	0.8	0.3	0.2	0.0	0.0
Mexico	0.5	0.8	1.0	0.4	0.2	0.0	0.0
Colomb.	0.1	0.3	0.4	1.0	0.8	0.0	0.0
Venezuela	0.1	0.2	0.2	0.8	1.0	0.0	0.0
Australia	0.0	0.0	0.0	0.0	0.0	1.0	0.8
N. Zlnd.	0.0	0.0	0.0	0.0	0.0	0.8	1.0

similarity proposed by Tamura et al. in [8]. For instance, despite the fact that the proximity degree, presented in Tab. 1, between the concepts *Canada* and *Venezuela* is 0.1, the α-proximity is 0.4. Using the sequence of the original proximity degrees from Tab. 1: $Canada P_\alpha Mexico$=0.5 $\wedge$ $Mexico P_\alpha Colombia$=0.4 $\wedge$ $Colombia P_\alpha Venezuela$=0.8, we can obtain $Canada P^+_\alpha Venezuela$=0.4, as presented in Tab. 2.

Table 2. α-proximity Table for domain COUNTRY [15]

	Canada	*USA*	*Mexico*	*Colomb.*	*Venezue.*	*Australia*	*N. Zlnd.*
Canada	1.0	0.8	0.8	0.4	0.4	0.0	0.0
USA	0.8	1.0	0.8	0.4	0.4	0.0	0.0
Mexico	0.8	0.8	1.0	0.4	0.4	0.0	0.0
Colomb.	0.4	0.4	0.4	1.0	0.8	0.0	0.0
Venezuela	0.4	0.4	0.4	0.8	1.0	0.0	0.0
Australia	0.0	0.0	0.0	0.0	0.0	1.0	0.8
N. Zlnd.	0.0	0.0	0.0	0.0	0.0	0.8	1.0

From the propagation of shadings in Tab. 2, we can observe that the equivalence classes marked in the table have a nested character [5, 15]. This property allows for the extraction of a concept hierarchy (identical to that presented in [5]), representing disjoint α-equivalence classes on multiple levels of α (e.g. Fig. 1 and 2). For instance, a space of equivalence classes at 0.8-level for the domain COUNTRY consists of three subsets (i.e. $\left\| P^+_{0.8,COUNTRY} \right\|$=3): {{*Canada, USA, Mexico*}, {*Colombia, Venezuela*}, {*Australia, N. Zealand*}}.

2.1 ID3-Type Algorithms

In 1986 Quinlan [10] presented his first ID3 algorithm, which became the most common entropy-based decision rules learning technique [11]. Due to its popularity, many improvements have been introduced, generating the whole family of popular ID3-type algorithms [10-13, 16].

A decision tree is n-ary tree structure, which can be used to represent a set of decision rules letting a decision maker assign currently analyzed cases to earlier discovered classes. In the tree every leaf node represents a class (several nodes

can represent the same class), and each internal node represents an attribute of an analyzed object (i.e. data-item) and has one child for each of the possible attribute's values. Every path in that tree reflects a sequence of decisions (i.e. nodes), which need to be taken to assign a new object to a particular class of objects, named in one of the leaves.

ID3-type algorithms use an information gain measure, as a heuristics for selecting the attribute that will best separate the samples' individual classes. This attribute becomes a "test attribute", used to point the user into a direction leading to a leaf, representing a particular class of training samples. A separate branch is created for each known value of the "test attribute", and the samples are partitioned accordingly.

The major issue in the ID3 approach is to choose the best attribute for a split at each node (i.e. to find the best ordering of attributes in every decision rule). Since there may be many decision trees that correctly classify the training set, a smaller one is typically preferred. In his work, Quinlan used entropy measure [14] to calculate the value of information gain (i.e. entropy reduction) for every attribute remaining in the current algorithm's iteration (i.e. the current node of the decision tree).

The expected information needed to classify a given sample is caclulated as follows:

$$I(s_1, s_2, ..., s_m) = -\sum_{k=1}^{m} \frac{s_k}{s} \cdot \log(\frac{s_k}{s}) \tag{1}$$

where s denotes the cardinality of the training set S with m classes and s_k samples of k^{th} class ($k = 1, 2, \ldots, m$).

If an attribute D have v different domain values $\{d_1, d_2, \ldots, d_v\}$, it can be used to partition training set S into v subsets: $\{S_1, S_2, \ldots, S_v\}$, where subset S_j contains all those samples from S that have the same domain value d_j of D. If we let s_{ij} to represent the number of samples of class C_i in a subset S_j, the entropy, based on the partitioning of s samples into subsets by the analyzed attribute D, is given by:

$$E(D) = \sum_{j=1}^{v} \frac{s_{1j} + s_{2j} + ... + s_{mj}}{s} I(s_{1j}, s_{2j}, ..., s_{mj}) \tag{2}$$

The smaller the entropy value $E(D)$, the greater is the purity of the subset partitions. The encoding information that would be gained by branching the set of training samples using values of atrribute D is:

$$Gain(D) = I(s_1, s_2, ..., s_m) - E(D) \tag{3}$$

The attribute with the highest information gain for the given set of training samples is put as a new separation node of the generated decision tree.

3 Interpretations of Tuples with Non-atomic Values

As we mentioned in section 2.1, fuzzy databases let the user reflect uncertainty about inserted information via the insertion of multiple descriptors in every

column of the data table. At the same time, all of the decision tree induction algorithms, mentioned in section 2.2, were originally created for processing only atomic attribute values.

This leads to the challenge of developing a mechanism allowing consistent interpretation of the non-atomic values in the fuzzy databases. In other words, we want to map (i.e. defuzzify) non-atomic values from fuzzy relation into a set of atomic descriptors, that are compatible with 1NF definition and can be analyzed using regular data mining techniques.

We need to keep in mind that interpretation of values in fuzzy database depends on the character of the α-proximity relation or, in other words, on the shape of the partition tree extracted from the relation. The hierarchical and nested distribution of equivalence classes in the partition tree provides data miners with an interesting knowledge allowing for consistent and accurate interpretations of fuzzy database relations. Different attribute values can be considered as identical or totally different, depending on the preferred level of the partition tree. It gives us the ability to analyze the same data on multiple levels of granularity. Moreover, assuming that proximity relation reflects true dependencies among data, the analysis of entered values may give us a hint on granularity level (i.e. α) the persons, who were entering values, were able or willing to register.

In this section, we are proposing two major approaches to the interpretation of fuzzy tuples: (1) *lossless approach*, preserving all original entries stored in the fuzzy relation, and (2) *lossy approach*, which transfers the original data entries into one common α-level. We will now discuss briefly our interpretation of these concepts.

(1) *Lossless approach* is the most universal, and at the same time the most extreme case, as it separates every combination of values (i.e. any subset of values entered for a particular attribute of a specific entity) and maps it to a unique single (and therefore atomic) descriptor. Since the mapping has one-to-one character, we called it *lossless*, as both the number of entries, and its similarity level can be backtracked if necessary.

This approach could be interpreted as a tranformation of the multidimensional representation of data (where the dimensionality is defined by (1) the number of similarity levels (α's), (2) the number of equivalence classes distinguished in the partition tree, and (3) by the number of non-atomic values actually entered to the original data set) into a one dimensional space that allows us to transfer non-atomic, originally inserted values into atomic ones. For instance, $\{Canada\}$ would be interpreted as $\{Canada\}$, whereas entry $\{Canada,\ USA\}$ as some type of unique combination (e.g. union) of these two values, i.e. $\{CanadaUSA\}$. Such process can generate n unique combinations of descriptors for a single domain, such that: $n \leq 2^{\|P^+_{1.0}\|}$. We can expect $n = \|P^+_{1.0}\|$for an attribute that contains only precise data, and $n = \sum_{\alpha=1.0}^{\alpha=0.0} \|P^+_\alpha\|$for an imprecise attribute with α-proximity table that reflects perfectly all dependencies occurring in real-life data. Since a database relation is a subset of cross product of all domains' values, we

may ended up with $2^{\left\|P^+_{1.0,D_1}\right\|} \times 2^{\left\|P^+_{1.0,D_2}\right\|} \times ... \times 2^{\left\|P^+_{1.0,D_k}\right\|}$ unique descriptors for table containing k attributes.

This approach to fuzzy data interpretation generates a new data table that is in 1NF, and has a size identical with the original fuzzy data set. The 1NF is guaranteed, as all (atomic and non-atomic) variables are artificially mapped to atomic descriptors (e.g. atomic values are copied, whereas the non-atomic variables are interpreted as some kind of intermediate, but now atomic, values). Since all occurring descriptors are mapped to unique values, preservation of the original character of the data, as well as its original size, is assured.

Despite some advantages of the *lossless* interpretations (accuracy, preservation of original data character, etc.), there are many good reasons to perform *lossy* interpretation of the data. As we showed above, the *lossless* interpretation may cause exponential growth of new "atomic" descriptors, whereas mainitaing the original size of the data table. This can have a significant influence on results generated by entropy-based classification, as it couses considerable expansion of granularity of our search space, leading to decrease in support for individual decision rules. Moreover, this type of interpretation can be very confusing for a customer, who may have troubles interpreting the artificially created one-to-one mapping.

One very good reason to perform *lossy* interpretation is typically a client's preference. Quite often a customer (e.g. decision-maker) is not interested in a detailed representation of the warehouse data, as the knowledge he/she hopes to discover from the data set is located at the more abstract level. The transformation of data to a certain granularity level may permit the reduction of the size of the original data repository (leading in consequence to faster data processing), or simply provide a more human-friendly data representation.

(2) *Lossy approach* to interpretation of fuzzy tuples is driven by a need for the transformation of all originally entered values, representing different equivalence classes in the partition tree, to a single, common level, where atomic descriptors/interpretations can be assigned to them. Such transformation is going to generate the $\left\|P^+_\ell\right\|$ of unique descriptors for a single domain, such that $\left\|P^+_{0.0}\right\| \leq \left\|P^+_\ell\right\| \leq \left\|P^+_{1.0}\right\|$, where ℓ is the preferred abstraction level, and $\left\|P^+_\ell\right\|$ denotes the number of unique descriptors (i.e. equivalent classes) at that level.

The most common case of *lossy* interpretation would be the transformation of a fuzzy relation into a table, where all entries are atomic and reflected at the lowest level of the partition tree (ℓ=1.0). After such transformation we can interpret information almost in the same way we would interpret a regular (in 1NF) relational database table. We will call this approach a *full data-specialization* (which, could be also interpreted as a *complete data-defuzzification*), as it requires transformation of all higher-level (i.e. multi-valued) descriptors into the classes distinguished at the very bottom of the partition tree (where α=1.0).

This requires development of a technique allowing appropriate splitting of a vote, carried by every fuzzy tuple, into multiple parts, reflecting distribution of all originally entered non-atomic descriptors and their similarities. We propose one possible approach in the next section of this paper.

To better explain our motivation for applying *lossy* interpretation, we will use a trivial example. For instance, if in the column COUNTRY_OF_ORIGIN we have two entries {*Canada, USA*} we could interpret such a record as two halves of a record, one having an atomic value {*Canada*}, and another saying precisely {*USA*}, each with half of the vote carried by the original record (usually having a value of unity) to reflect the uncertainty represented in the original fuzzy record.

When generating *lossy* interpretations of fuzzy tuples, we need to remember to add the attribute COUNT to the output relation. The attribute is necessary to preserve a consistent representation of the original data after the *lossy* transformation, as this transformation may cause merging identical fractions of originally separate records (e.g. half of the vote with now atomic value {*USA*} can be merged with other record that contains a descriptor belonging to the same equvalence class, as long as values in all other attributes are the same). In such case, an accurate count of votes is essential to guarantee proper data mining results. It protects every original tuple from being counted more or less than once, when the *lossy* data interpretation is performed. By representing each record of the original data table as a single vote, when having it either split (i.e. fractioned) into multiple equivalence subclasses (when transfering a tuple to match equvalence classes at the lower level of partition tree), or generalized (i.e. summarized) to a more abstract equivalence class(es) (when going up in the partition tree), we are guaranteed that original proportions among the data (i.e. distribution of originally inserted values) remain accurate in the new data set.

After performing *full data-specialization* (i.e. *complete data-defuzzification* – a specialization reaching bottom of partition tree) we are capable of reducing a size of the original relation by merging all fractions of tuples that look identical after the transformation. Unfortunately, an exact representation of uncertainty distribution (i.e. assignment of non-atomic values to specific tuples) cannot be maintained anymore. That is the major reason we called this approach a *lossy* one.

Obviously, with the *lossy* interpretation, we are not obligated to always choose only the terms at the lowest level of the partition tree as our data representation. We can choose any (e.g. customer-preferred) level of data granularity (i.e. α-level) and transfer the whole dataset to the level of our preference.

3.1 Similarity-Driven Vote Distribution Method

Due to tree-like character of partition hierarchies, *data-generalization* of fuzzy tuples has a typically straightforward character. We can replace the originally inserted values with broader concepts (i.e. equivalence classes occurring at the higher level of similarity tree), where the original values belong to and then treat a record with originally lower-level descriptors, as a full member of higher-level class. If the proximity table accurately reflects real-life dependencies among the data, the vote of the generalized tuple may never need to be split, as the equivalence classes in the partition hierarchy have a nested character.

The problem arises however when we have to deal with multiple attribute values that need to be *data-specialized*. E.g. in what COUNTRY (α=1.0) should we expect to find a drugs dealer who, as a not-confirmed report says, was

recently seen in {*Canada, Colombia, Venezuela*}? The most trivial solution would be to split the vote equally among all inserted descriptors: {*Canada|0.(3), Colombia|0.(3), Venezuela|0.(3)*}. This approach however does not take into consideration real life dependencies, which are reflected not only in the number of inserted descriptors, but also in their similarity.

In our investigation we used a simple heuristic [15] letting us to replace the even distribution of a vote with a nonlinear spread, dependent both on the similarity of inserted values and on their quantity. Using the partition tree built from the α-proximity table (grey structure in Fig. 1), we can extract from the set of the originally inserted values those concepts which are more similar to each other than to the remaining descriptors. We call them *subsets of resemblances* (e.g. {*Colombia, Venezuela*} from the above example). Then we use them as a basis for calculating a distribution of a vote's fractions. An important aspect of this approach is extraction of the *subsets of resemblances* at the lowest possible level of their common occurrence, since the nested character of α-proximity relation guarantees that above this α-level they are going to co-occur regularly.

Our algorithm is straightforward. Given (1) a set of attribute values inserted as a description of particular entity, and (2) a hierarchical structure reflecting Zadeh's partition tree [5] for the particular attribute; we want to extract a table, which includes (a) the list of all subsets of resemblances from the given set of descriptors, and (b) the highest level of α-proximity of their common occurrence. We then use the list to fairly distribute fractions of the original record.

Our algorithm uses preorder recursive traversal for searching the partition tree. If any subset of the given set of descriptors occurs at the particular node of the concept hierarchy we store the values that were recognized as similar, and the adequate value of α. An example of such a search for subsets of resemblances in a tuple with the values {*Canada, Colombia, Venezuela*} is depicted in Fig. 1. Numbers on the links in the tree represent the order in which the particular subsets of similarities were extracted.

After extracting the *subsets of resemblances*, we apply a summarization of α values as a measure reflecting both the frequency of occurrence of the particular attribute values in the *subsets of resemblances*, as well as the abstraction level of these occurrences. Since the country *Canada* was reported only twice, we assigned it a grade 1.4(i.e. 1.0 + 0.4). For *Colombia* we get: *Colombia|(1.0 + 0.8 + 0.4) = Colombia|2.2*, and for the last value: *Venezuela|(1.0 + 0.8 + 0.4) = Venezuela|2.2*.

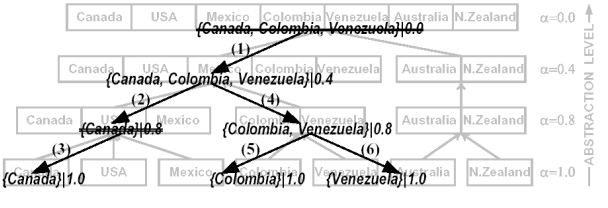


Fig. 1. Subsets of Resemblances extracted from the Partition tree

At the very end we normalize grades assigned to each of the entered values: *Canada |(1.4/5.8) = Canada |0.24, Colombia |(2.2/5.8) = Colombia | 0.38, Venezuela |(2.2/5.8) = Venezuela |0.38.* This leads to the new distribution of the vote's fractions, which more accurately reflects real life dependencies than a linear weighting approach.

3.2 Example of Decision Tree Induction Via Lossless and Lossy Interpretations

In this section, we will analyze a short example of the entropy-based classification of fuzzy tuples (presented in Tab. 3). Our example is a modified version of a non-fuzzy problem presented in a popular textbook on AI [17].

Let us assume we gathered data from different clients regarding their decision about waiting (or not) for a table at a restaurant. Some of the clients did not exactly remember the situations they were reporting on, so we used fuzzy representation to reflect such lack of certainty. The data has four attributes, i.e.: *Waiting Time* (referred also as *WT*), *Day of Week* (i.e. *DoW*), *Food Type* (i.e. *FT*), and *Did Wait?*. The last attribute has a non-fuzzy character that answers

Table 3. Fuzzy Relation for decision about waiting problem

ID	*Wait Time (WT)*	*Day Of Week*	*Food Type (FT)*	*Did Wait?*
1	30-60	Thr	Greek	No
2	10-30	Fri	Italian	Yes
3	30-60, Above 60	Thr	Burger, Barbeque	No
4	30-60	Fri, Sat	Greek	Yes
5	Above 60	Tue	Barbeque, Burger	No
6	10-30	Tue	Burger, Barbeque	Yes
7	30-60	Mon	Italian	No
8	30-60	Fri, Sun	Burger	Yes
9	0-10	Fri	Italian, Barbeque, Greek	Yes
10	30-60	Sun	Chinese	No
11	30-60	Mon, Tue	Barbeque	No
12	10-30	Mon	Greek	No
13	30-60	Fri	French, Italian	Yes
14	30-60	Fri, Sat	Barbeque	Yes
15	10-30	Sun	Barbeque, Burger	Yes
16	30-60	Sun	French, Greek, Italian	No
17	30-60	Wed, Thr	French	No
18	10-30	Thr	French	Yes
19	0-10	Fri, Sat, Sun	Barbeque	Yes
20	30-60, Above 60	Sun, Mon, Tue	Sushi	No
21	10-30	Wed	Chinese	No
22	30-60	Fri	Chinese, Sushi	No
23	10-30	Sat	Sushi, Chinese	Yes
24	30-60	Sun	Barbeque	Yes
25	0-10	Wed	Chinese, Sushi	Yes

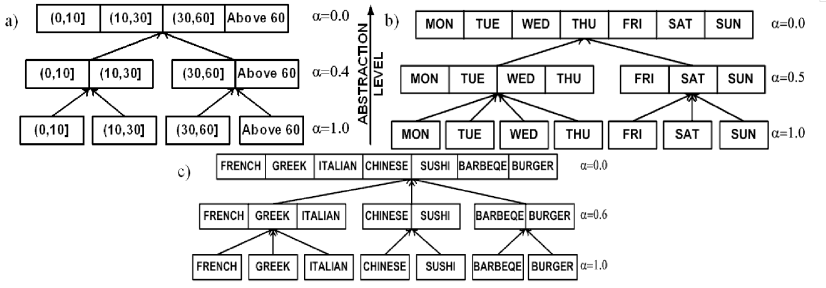


Fig. 2. Partition trees for the analyzed example

the question whether a particular client decided to wait or not for a table, when the circumstances described in the remaining three attributes were taking place. As expected for fuzzy relational databases, the similarity relations are specified for all of the attributes. To make our example more understandable, we present the similarity relations using Zadeh's partition trees [5]. Figure 2a presents a similarity tree for the domain *WT*. Figure 2b presents partition tree for *DoW*, and figure 2c is for *Food Type*. As explained in the previous sections, we can transfer table 3 to multiple sets, containing only atomic values, using different approaches to fuzzy data interpretation. In table 4 we present some results of the Similarity-driven Vote Distribution algorithm when *lossy* interpretation to 1.0-level values is requested. Obviously, tuples identified as 9b and 19a can be merged, as long as the value in COUNT attribute is appropriately modified (COUNT for the resulting tuple: *0.24+0.333=0.573*).

When *fully defuzzifying* our original relation (i.e. performing *lossy* interpretation where ℓ=1.0 for all of the attributes), we initially generated 76 fractions of tuples (with the cumulative vote remaining at its original level of 25), and managed to merge them into 51 unique records in 1NF (with the cumulative vote unchanged). Of course, table 3 could be generalized further, even to only 2 tuples (where 12 votes would be stored in COUNT for class *No*, and 13 votes for answer *Yes*, and ℓ=0.0 for all of the remaining attributes). Such a transformation could take a place if we would decide to generalize all attributes (except the *Did Wait?*) to the highest level of granularity. This observation brings up a seemingly obvious conclusion that level of entropy carried by a given data set may change with the data generalization or specialization. Choice of the transformation has also some influence on support of the generated classification rules. As we can observe from table 5, we may need 780 different data points to fully cover the space generated by *lossless* interpretation of data in table 3. Whereas 84 unique combinations of attribute values would be enough to cover the whole space of possibilites generated by the second case of *lossy* interpretation.

3.3 Conclusions and Future Work

In this paper we discussed new ways in which the similarity and proximity relations, implemented in the fuzzy databases, can be successfully applied to

Table 4. Part of Fuzzy Relation from Table 3 after defuzzification to the 1.0-level

ID	*Wait Time (WT)*	*Day Of Week (DoW)*	*Food Type (FT)*	*COUNT*	*Did Wait?*
9a	0-10	Fri	Italian	0.38	Yes
9b	0-10	Fri	Barbeque	0.24	Yes
9c	0-10	Fri	Greek	0.38	Yes
...	...	...	...	...	...
19a	0-10	Fri	Barbeque	0.333	Yes
19b	0-10	Sat	Barbeque	0.333	Yes
19c	0-10	Sun	Barbeque	0.333	Yes

Table 5. Characterization of three different interpretations of Table 3. First 3 rows represent *lossless* interpretation, whereas the remaining 6 characterize two different *lossy* transformations.

Attribute (i.e. D)	ℓ	$\left\|P^{+}_{\ell,D}\right\|$	$E(D)$	*Gain(D)*	*Characterization of the generated dataset*
Wait Time (WT)	all	5	0.71220	0.28683	25 unique tuples
Day Of Week	all	13	0.56000	0.43884	Σ COUNT = 25
Food Type (FT)	all	12	0.67019	0.32865	$\|\|WT\|\| \times \|\|DoW\|\| \times \|\|FT\|\|$=780
Wait Time (WT)	1.0	4	0.73592	0.26333	51 unique tuples
Day Of Week	1.0	7	0.67976	0.31949	Σ COUNT = 25
Food Type (FT)	1.0	7	0.93254	0.06671	$\|\|WT\|\| \times \|\|DoW\|\| \times \|\|FT\|\|$=196
Wait Time (WT)	1.0	4	0.73592	0.26332	34 unique tuples
Day Of Week	1.0	7	0.67972	0.31952	Σ COUNT = 25
Food Type (FT)	0.6	3	0.94392	0.05533	$\|\|WT\|\| \times \|\|DoW\|\| \times \|\|FT\|\|$=84

imprecise data interpretation and to decision rules induction. We showed how the fuzzy relational databases, due to their interesting properties allowing for multi-level data representations, could be successfully utilized to reduce space size while maintaining its original entropy. Finally, we pointed out important consequences of both methods.

References

1. KDnuggets, *http://www.kdnuggets.com/polls/*
2. B.P. Buckles and F.E. Petry, "A fuzzy representation of data for relational databases", Fuzzy Sets and Systems, 7(3), 1982, pp. 213-226.
3. F.E. Petry, Fuzzy Databases: Principles and Applications, Kluwer Academic Publishers, Boston, MA, 1996.
4. F.E. Codd, "A relational model of data for large share data banks", Communications of the ACM, 13(6), 1970, pp. 377-387.
5. L.A. Zadeh, "Similarity relations and fuzzy orderings", Information Sciences, 3(2), 1970, pp. 177-200.
6. S. Shenoi and A. Melton, "Proximity Relations in the Fuzzy Relational Database Model", International Journal of Fuzzy Sets and Systems, 31(3), 1989, pp. 285-296.

7. S. Shenoi, A. Melton, L. T. Fan, "Functional Dependencies and Normal Forms in the Fuzzy Relational Database Model", Information Sciences, 60(1-2), 1992, p. 1-28.
8. S. Tamura, S. Higuchi, and K. Tanaka, "Pattern Classification Based on Fuzzy Relations", IEEE Transactions on Systems, Man, and Cybernetics, SMC-1(1), 1971, pp. 61-66.
9. S. Kumar De, R. Biswas, A.R. Roy, "On extended fuzzy relational database model with proximity relations", Fuzzy Sets and Systems 117, 2001, pp. 195-201
10. Quinlan J, "Induction of decision trees", Machine Learning, 1(1), 1986, p. 81-106.
11. J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, New York, NY, 2000.
12. Quinlan J, "Simplifying decision trees", International Journal of Man-Machine Studies, 27, 1987, pp. 221-234.
13. Quinlan J, "C4.5: Programs for Machine Learning", Morgan Kauffman, 1993.
14. Shannon C.E., "A Mathematical Theory of Communication", Bell System Technical Journal, 27, 1948, pp. 379-423, and 623-656.
15. R. Angryk, F. Petry, "Discovery of Abstract Knowledge from Non-Atomic Attribute Values in Fuzzy Relational Databases," in: Modern Information Processing, From Theory to Applications, Elsevier, 2006, pp. 171-182.
16. I. H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, June 2005.
17. S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 2002.

A Genetic-Programming-Based Approach for the Learning of Compact Fuzzy Rule-Based Classification Systems*

F.J. Berlanga[1], M.J. del Jesus[1], M.J. Gacto[2], and F. Herrera[2]

[1] University of Jaén, Dept. of Computer Science, E-23071 Jaén, Spain
{berlanga, mjjesus}@ujaen.es
[2] University of Granada, Dept. of Computer Science and Artificial Intelligence, E-18071 Granada, Spain
mjgacto@ugr.es, herrera@decsai.ugr.es

Abstract. In the design of an interpretable fuzzy rule-based classification system (FRBCS) the precision as much as the simplicity of the extracted knowledge must be considered as objectives. In any inductive learning algorithm, when we deal with problems with a large number of features, the exponential growth of the fuzzy rule search space makes the learning process more difficult. Moreover it leads to an FRBCS with a rule base with a high cardinality. In this paper, we propose a genetic-programming-based method for the learning of an FRBCS, where disjunctive normal form (DNF) rules compete and cooperate among themselves in order to obtain an understandable and compact set of fuzzy rules, which presents a good classification performance with high dimensionality problems. This proposal uses a token competition mechanism to maintain the diversity of the population. The good results obtained with several classification problems support our proposal.

1 Introduction

The Fuzzy Rule-Based Systems have been successfully applied to various fields such as control, modelling and classification. Traditionally, the main goal in the design of this kind of fuzzy systems has been the maximization of the precision, although the interpretability has also been taken into account in some recent studies [1]. This objective is more difficult to reach when the number of features for the problem increase due the exponential growth of the fuzzy rule search space. This growth makes the learning process more difficult and, in most cases, it leads to an FRBCS with a rule base with a high cardinality (with respect to the number of rules and the number of features included in each rule).

An analysis of the specialized literature indicates that exist two principal ways to tackle the problem of the high dimensionality:

1. Compacting and reducing the previously learned rule set in a postprocessing stage ([10], [13]), and

* Supported by the Spanish Ministry of Science and Technology under the Projects TIN-2005-08386-C05-03 and TIN-2005-08386-C05-01.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 182–191, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2. Carrying out a feature selection process, that determines the most relevant variables before or during the inductive learning process of the FRBCS ([7], [16], [17]).

In this paper, we tackle the learning of FRBCSs with high interpretability by means of a genetic-programming (GP) based approach. The genetic programming [12] is a kind of evolutionary algorithm that uses variable-length trees to represent the different individuals in the population, instead of fixed-sized vectors (with binary, integer or real codification) as the Genetics Algorithms (GAs) do.

The FRBCSs learning has already been done previously in the specialized literature by means the GP. An initial paper in this topic is Fuzzy GP, developed by Geyer-Schulz [6], which combines a simple GA that operates on a context-free language with a context-free fuzzy rule language. Sánchez et al. propose an FRBCS learning process in [18] and [5] by combining GP operators with simulated annealing and GA respectively to establish the membership functions. Mendes et al. develop in [14] a co-evolutionary algorithm which includes a GP based algorithm for FRBCS learning and an evolutionary algorithm to evolve membership functions. Tsakonas et al. propose in [19] a GP-based algorithm with a Pittsburgh representation for the learning of FRBCSs.

In our proposal, the definition of a context-free grammar that allows the learning of DNF fuzzy rules, together with the use of a competition mechanism between rules which deletes irrelevant rules during the learning process, allow us to obtain of compact FRBCSs (with few rules and conditions per rule) with a high-generalization capability.

The paper is organized as follows. Our proposal is explained in Section 2. Section 3 presents the experimental study and the analysis carried out. Finally, the conclusions obtained are presented in Section 4.

2 The Genetic-Programming-Based Proposal

The first feature of our proposal is the kind of fuzzy rule used. Our method learns DNF fuzzy rules, which have the following form:

$$If\ X_1\ is\ \hat{A}_1\ and\ \ldots\ and\ X_n\ is\ \hat{A}_n\ then\ Y\ is\ C\ with\ CD$$

where each input variable X_i takes as a value a set of linguistic terms or labels $\hat{A}_i = \{A_{i1}\ or\ \ldots\ or\ A_{iL_i}\}$ joined by a disjunctive operator, whilst the output variable (Y) has one of the class values. The definition of the fuzzy sets that specify the meaning of each linguistic term or label, is done by using expert knowledge, or in its absence, by using triangular fuzzy sets divided in a uniform way.

This rule also includes a certainty degree ($CD \in [0, 1]$), which represents the confidence of the classification in the class represented by the consequent. In our proposal, this certainty degree is obtained as the quotient Sj / S, where Sj is the sum of the matching degrees for the training examples belonging to class represented by the consequent which are covered by the antecedent of the rule, and S the sum of the matching degrees for all the training examples which are

covered by the antecedent of the rule, independently of the class they belong to. Is important to point out that, in our proposal, this kind of rule is generated according to the production rules of a context-free grammar. The definition of this grammar is completely explained in subsection 2.1.

One of the most important aspects in any evolutionary approach is its coding scheme. The different evolutionary methods follow two approaches in order to encode rules within a population of individuals [4]:

- The "Chromosome = Set of rules", also called the *Pittsburgh* approach, in which each individual represents a rule set.
- The "Chromosome = Rule" approach, in which each individual codifies a single rule, and the whole rule set is provided by combining several individuals in the population.

In turn, within the "Chromosome = Rule" approach, there are three generic proposals:

- The *Michigan* approach, in which each individual codifies a single rule. This kind of system is usually called a learning classifier system. It is rule-based, message-passing system that employs reinforcement learning and the GA to learn rules that guide its performance in a given environment [11].
- The *IRL (Iterative Rule Learning)* approach, in which each chromosome represents a rule, but the solution is the best individual obtained and the global solution is formed by the best individuals obtained when the algorithm is run multiple times. SLAVE [8] is a proposal that follows this approach.
- The *cooperative-competitive* approach, in which the complete population or a subset of it codifies the rule base. LOGENPRO [21] is an example with this kind of representation.

Our method follows the *cooperative-competitive* approach. However, this kind of representation makes necessary to introduce a mechanism to maintain the diversity of the population. In this proposal we use a token competition mechanism to promote the diversity in order to avoid that all the individuals in the population converge into the same area of the search space. This mechanism is described in subsection 2.1, together with the remaining components of the evolutionary learning process.

Finally, is important to point out that our proposal is made up by two different stages:

- The first stage consists in an evolutionary learning process that uses GP to obtain a compact fuzzy rule base with a good classification performance.
- The second one, consists in a postprocessing stage that eliminates redundant rules from the rule base in order to increase the interpretability.

In the following two subsections we describe these two stages.

2.1 Evolutionary Learning Process

The GP process starts with an initial population of rules that is randomly generated according to the grammar production rules.

In each iteration of the evolutionary process, parents are selected to generate offspring by the ranking selection scheme. All the individuals in the new population generate one descendant by means one of the genetic operators. Individuals in the population and offspring obtained by the application of the genetic operators, are joined to form a new population. The size of the resulting population is double the original. The individuals of this new population are ordered by their fitness score and the token competition is carried out. The token competition modifies the fitness of the individuals in order to maintain the diversity of the population. Once token competition is finished, the individuals in the population are ordered again by their modified fitness score, and the population size is set to its original one. This evolutionary process is repeated until a certain number of calls to the fitness function is reached.

Once the evolutionary process is concluded, the best evolved population is returned as the final rule set. The best population obtained in the evolutionary process is selected according to a certain measure of global fitness score.

In the following, the most important components of our method are described.

1. Grammar Definition: In our method, is necessary to define a grammar that allows the learning of DNF fuzzy rules and the absence of some input features. In Table 1, an example of the grammar for a classification problem with two features (X_1, X_2), three linguistic labels per feature (Low, Medium, High) and three classes (C1, C2, C3) is shown.

Table 1. Grammar example

$Start \longrightarrow [If],\ antec,\ [then],\ conseq,\ [.].$
$antec \longrightarrow descriptor1,\ [and],\ descriptor2.$
$descriptor1 \longrightarrow [any].$
$descriptor1 \longrightarrow [X_1\ is]\ label.$
$descriptor2 \longrightarrow [any].$
$descriptor2 \longrightarrow [X_2\ is]\ label.$
$label \longrightarrow \{member(?a,\ [L,\ M,\ H,\ L\ or\ M,\ L\ or\ H,$ $M\ or\ H,\ L\ or\ M\ or\ H])\},\ [?a].$
$conseq \longrightarrow [Class\ is]\ descriptorClass.$
$descriptorClass \longrightarrow \{member(?a,\ [C1,\ C2,\ C3])\},\ [?a].$

2. Genetic Operators: Offspring are generated by one of the next three genetic operators (these operators are selected in a probabilistic way):

1. Crossover: Produces one child from two parents. A part in the first parent is randomly selected and replaced by another part, randomly selected, in the second one, but under the constraint that the offspring produced must be valid according to the grammar.
2. Mutation: A part of the rule is selected and replaced by a randomly generated new part. Since the offspring have to be valid according to the grammar, a selected part can only mutate to another part with a compatible structure.

3. Dropping Condition: Due the probabilistic nature of GP, redundant constraints may be generated in the rule. Thus, it is necessary to generalize the rules, to represent the actual knowledge in a more concise form. Dropping condition selects randomly one descriptor in the antecedent part and then turns it into "any". The attribute in the descriptor is no longer considered in the rule, hence, the rule can be generalized.

3. Fitness Function: Our method uses a fitness function based on the estimation of two measures:

1. *Confidence*, which measures the accuracy of an individual, that is, the confidence of the consequent to be true if the antecedent is verified

$$confidence = \frac{tp}{(tp + fp)} * \frac{tn}{(fn + tn)} . \tag{1}$$

2. *Support*, which measures the coverage of the knowledge represented in the individual

$$support = \frac{tp}{(tp + fn)} * \frac{tn}{(fp + tn)} . \tag{2}$$

where tp and fp are the sums of the matching degrees for true and false positives, and tn and fn are the number of true and false negatives, respectively.

Both measures are combined to make up the fitness function in the following way

$$raw_fitness = \begin{cases} support, & \text{if } support < min_support \\ support * confidence, & \text{otherwise} \end{cases} . \tag{3}$$

If the support of the rule is below a user-defined minimum threshold, the confidence value should not be considered to avoid the waste of effort to evolve those individuals with a high confidence but low support.

4. Maintaining the Diversity of the Population: Token Competition [21] has been used as mechanism to maintain the diversity in the population in our approach. It assumes that each example in the training set can provide a resource called a token, for which all chromosomes in the population will compete to capture. If an individual (i.e. a rule) can match the example, it sets a flag to indicate that the token is seized. Other weaker individuals then cannot get the token.

The priority of receiving tokens is determined by the strength of the individuals. The individuals with a high fitness score will seize as many tokens as they can. The other ones will have their strength decreased because they cannot compete with the stronger ones. The fitness score of each individual is penalized based on the tokens it can seize. The penalized fitness is defined as:

$$Penalized_fitness = raw_fitness * \frac{count}{ideal} . \tag{4}$$

where *raw_fitness* is the fitness score obtained from the evaluation function, *count* is the number of tokens that the individual actually seized and *ideal* is the total

number of tokens that it can seize, which is equal to the number of examples that the individual matches.

As a result of token competition, there exist individuals that cannot seize any token. These individuals are considered as irrelevant, and they can be eliminated from the population due to all of their examples are covered by other stronger individuals.

In order to increase the diversity into the population, new rules are also generated to cover those examples whose tokens have not been seized by any rule (if those examples exist).

5. Selecting the Best Population: At the end of each iteration in the evolutionary process, a process that keeps the best evolved population is carried out. This process checks if the current population is better than the others that have been evolved. One population A is considered better than other B if the global fitness score of A is greater than the global fitness score of B. The global fitness score is calculated adding four different measures

$$Global_fitness = Percent \ + \ Nvar \ + Ncond \ + \ Nrules \, . \tag{5}$$

where *Percent* indicates the correct percentage on training, *Nvar* the average number of variables per individual (rule) in the population, *Ncond* the average number of labels per individual (rule) in the population and *Nrules* the number of rules in the population. These four measures are defined in the following way

$$Percent = W_1 \ * \ \%Tra \, . \tag{6}$$

$$Nvar = W_2 \ * \ (1.0 \ - \ \#V) \, . \tag{7}$$

$$Ncond = W_3 \ * \ (1.0 \ - \ \#C) \, . \tag{8}$$

$$Nrules = W_4 \ * \ (1.0 \ - \ \#R) \, . \tag{9}$$

where $\%Tra$ is the normalized value of the correct percentage on training, $\#V$ is the normalized value of the number of variables per rule, $\#C$ is the normalized value of the number of labels per rule, $\#R$ is the normalized value of the number of rules and W_i are some weights that allows give more importance to any of the four measures (in our experiments we have used the same value for all the weights, $W_i = 1$).

2.2 Rule Base Simplification

Once the evolutionary process has finished, a postprocessing stage is carried out for eliminating redundant rules. During the rule base learning process it may happen that the algorithm learns two rules, one included in the other. For example, in the following two rules

$$R1: \ If \ X_1 \ is \ Low \ then \ Class \ is \ C1 \ with \ \alpha_1$$

$$R2: \ If \ X_1 \ is \ Low \ or \ Medium \ then \ Class \ is \ C1 \ with \ \alpha_2$$

the second rule includes the first one, hence, it does not make sense to keep both of them in the rule set. In this case, it must be deleted the first rule because the examples that it covers are also covered by the second rule. Sometimes, it is also necessary to recalculate the certainty degree of the remaining rule. This process aims to increase the interpretability of the previously learned FRBCS, by deleting redundant rules.

3 Experimental Study

In order to analyse the behaviour of the proposed method, we use Pima, Wisconsin, and Wine databases from the UCI repository of machine learning Databases (*http://www.ics.uci.edu/mlearn/MLRepository.html*). Our method (from now on called FRBCS_GP) has been compared to other fuzzy rule learning techniques:

1. An extension of Wang & Mendel algorithm [20] for classification problems proposed by Chi et al. [2], that generates a fuzzy rule for each example in the training set and does not carry out any feature selection process.
2. A process for deriving fuzzy rules for high-dimensionality classification problems developed by Ravi et al. [16]. This approach uses a reduced set of features extracted from the original ones by the principal component analysis. After that, a fuzzy rule learning process is carried out following the method proposed in [9] which divides the pattern space in several fuzzy subspaces, learning a rule for each one. Finally, a modified threshold accepting algorithm [17] is used to build a compact rule subset with a high classification accuracy, from rule set obtained in the previous stage.
3. SLAVE, a GA-based method for the learning of DNF fuzzy rules proposed by Gonzalez et al. [8]. In [7], this method is extended by the inclusion of a feature selection process. This extension will be called 2SLAVE from now.
4. A GP-based FRBCS learning process designed by Tsakonas et al. [19] which uses a Pittsburgh approach to represent the solutions.
5. C4.5, a classification algorithm proposed by Quinlan [15] that constructs a decision tree, which can be later transformed into a crisp rule set.

The parameters of our algorithm are the following: It stops after 5000 evaluations, initial population size is 20, crossover probability is 0.5, mutation probability is 0.4, dropping condition probability is 0.1, and the minimum threshold for the support used in fitness function is 0.01. We have used 5 linguistic labels per variable in all the experiments. For each different database, we have used 10-fold cross-validation (except for the Sonar database, where only two partitions for training and test, with a distribution of 50%, have been used).

The results are showed in Table 2, where #R indicates the average rule number, #Var the average antecedent variables per rule, #Cond the average antecedent conditions number per rule and the %Test the correct percentage with test examples. The subscripts in %Test are related to the fuzzy reasoning method (FRM) [3] used, so 1 corresponds to the classical FRM (max-min) and 2 with the normalised sum respectively (except in the C4.5 algorithm in where FRM is

Table 2. Databases results

Pima

Method	**#R**	**#Var**	**#Cond**	**%Test$_1$**	**%Test$_2$**
WM	472.6	8	8	70.18	70.70
Ravi	354.9	5	5	70.06	70.06
2SLAVE	3.43	4.26	11.28	65.47	65.37
Tsakonas	17.4	1.41	1.74	52.8	51.6
C4.5	47.2	4.47	5.6	71.46	-
FRBCS_GP	10.27	1.14	2.29	71.37	72.73

Sonar

Method	**#R**	**#Var**	**#Cond**	**%Test$_1$**	**%Test$_2$**
WM	104	60	60	43.27	43.27
Ravi	277	6	6	73.08	75
2SLAVE	7.67	22.39	41.43	68.33	68.33
Tsakonas	18.33	1.65	1.73	49	50.67
C4.5	10	3.1	3.1	74	-
FRBCS_GP	11.33	6.5	14.62	75.33	75.33

Wisconsin

Method	**#R**	**#Var**	**#Cond**	**%Test$_1$**	**%Test$_2$**
WM	296.5	9	9	66.34	66.19
Ravi	344.3	5	5	93.85	93.85
2SLAVE	5.73	6.02	16.09	88.53	91.93
Tsakonas	22.7	1.15	1.19	72.2	56.9
C4.5	38.4	4.46	5.11	94.29	-
FRBCS_GP	11.6	1.67	3.81	93.87	94.5

Wine

Method	**#R**	**#Var**	**#Cond**	**%Test$_1$**	**%Test$_2$**
WM	159.4	13	13	78.56	79.74
Ravi	231.8	5	5	92.22	92.22
2SLAVE	5.67	6.9	16.86	89.87	89.87
Tsakonas	19.53	1.38	1.49	35.63	39.13
C4.5	7.1	2.15	2.28	90.51	-
FRBCS_GP	8.53	1.69	3.15	91.77	92.43

not used, so it has been decided to place the results in the first of the last two columns).

Analysing the results, we can point out the following considerations:

– Our method learns rule sets with a low number of variables and labels per rule. It also learns rule bases with a small number of rules. Therefore the resulting FRBCSs have a high interpretability level. Our results are comparable with the ones obtained by 2SLAVE.

– Analysing the performance of our approach, it presents a good performance in test for all the problems, obtaining the best results for all the different databases with the normalised sum FRM.

4 Conclusions

In this work, we have proposed a genetic-programming-based method to obtain FRBCSs with a high interpretability. The definition of a context-free grammar that allows the learning of DNF fuzzy rules and the absence of some input features, allows the obtaining of rules with fewer antecedent conditions. On the other hand, the use of token competition mechanism to increase the diversity into the population makes the rules compete among themselves giving out a smaller number of rules with a high-generalization capability.

The effectiveness of the method has been demonstrated over several classification problems and the results are promising. Therefore, we consider this approach to be an interesting alternative for the learning of interpretable FRBCSs.

As future work we will incorporate a proper multiobjective approach within the learning process.

References

1. Casillas J., Cordón O., Herrera F., Magdalena L. (Eds.): Interpretability Issues in Fuzzy Modeling. Springer-Verlag. Series Studies in Fuzziness and Soft Computing, Vol. 128 (2003)
2. Chi Z., Wu J., Yan H.: Handwritten numeral recognition using self-organizing maps and fuzzy rules. Pattern Recognition 28:1 (1995) 59–66
3. Cordón O., del Jesus M.J., Herrera F.: A Proposal on Reasoning Methods in Fuzzy Rule-Based Classification Systems. International Journal of Approximate Reasoning 20 (1999) 21–45
4. Cordón O., Herrera F., Hoffmann F., Magdalena L.: Genetic Fuzzy Systems. Evolutionary tuning and learning of fuzzy knowledge bases. World Scientific (2001)
5. García S., González F., Sánchez L.: Evolving fuzzy based classifiers with GA-P: A grammatical approach. Lecture Notes in Computer Science, Vol. 1598. Genetic Programming: Second European Workshop, EuroGP'99 (1999) 203–210
6. Geyer-Schulz A.: Fuzzy rule-based expert systems and genetic machine learning. Heidel-berg: Physica-Verlag (1995)
7. González A., Pérez R.: Selection of relevant features in a fuzzy genetic learning algorithm. IEEE Transactions on Systems, Man and Cybernetics Part B 31:3 (2001) 417–425
8. González A. Pérez R.: SLAVE: A genetic learning system based on an iterative approach. IEEE Transactions on Fuzzy Systems 27 (1999) 176–191
9. Ishibuchi H., Nozaki K., Tanaka H.: Distributed representation of fuzzy rules and its application to pattern classification. Fuzzy Sets and Systems 52 (1992) 21–32
10. Ishibuchi H., Nozaki K., Yamamoto N., Tanaka N.: Selecting fuzzy if-then rules for classification problems using genetic algorithms. IEEE Trans. Fuzzy Systems 3:3 (1995) 260–270

11. Kovacs T.: Strength or Accuracy: Credit Assignment in Learning Classifier Systems. Springer-Verlag (2004).
12. Koza J.R.: Genetic programming on the programming of computers by means of natural selection. Cambridge MA, USA: The MIT Press (1992)
13. Krone A., Krause P., Slawinski T.: A new rule reduction method for finding interpretable and small rule bases in high dimensional search spaces. Proc. of the 9th IEEE International Conference on Fuzzy Systems vol. 2 (2000) 694–699
14. Mendes R.R.F., Voznika F. de B., Freitas A.A., Nievola J.C.: Discovering Fuzzy Classification Rules with Genetic Programming and Co-evolution. Principles of Data Mining and Knowledge Discovery: 5th European Conference (PKDD'01). Springer-Verlag. Lecture Notes in Computer Science, Vol. 2168 (2001) 314
15. Quinlan J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
16. Ravi V., Reddy P.J., Zimmermann H.J.: Pattern classification with principal component analysis and fuzzy rule bases. European Journal of Operational Research 126:3 (2000) 526–533
17. Ravi V., Zimmermann H.J.: Fuzzy rule based classification with FeatureSelector and modified threshold accepting. European Journal of Operational Research 123:1 (2000) 16–28
18. Sánchez L., Couso I., Corrales J.A.: Combining GP operators with SA search to evolve fuzzy rule based classifiers. Information Sciences 136:1–4 (2001) 175–191
19. Tsakonas A., Dounias G., Jantzen J., Axer H., Bjerregaard B., von Keyserlingk D.G.: Evolving rule-based systems in two medical domains using genetic programming. Artificial Intelligence in Medicine 32:3 (2004) 195–216
20. Wang L.X., Mendel J.M.: Generating fuzzy rules by learning from examples. IEEE Transactions on Systems, Man, and Cybernetics 22:6 (1992) 1414–1427
21. Wong M.L., Leung K.S.: Data Mining using grammar based genetic programming and applications. Kluwer Academics Publishers (2000)

Performance Evaluation of Fuzzy-Neural HTTP Request Distribution for Web Clusters

Leszek Borzemski[1] and Krzysztof Zatwarnicki[2]

[1] Institute of Information Science and Engineering,
Wroclaw University of Technology, Wroclaw, Poland
leszek.borzemski@pwr.wroc.pl
[2] Department of Electrical Engineering and Automatic Control,
Technical University of Opole, Opole, Poland
KZatwarnicki@po.opole.pl

Abstract. In this paper we present the performance evaluation of our fuzzy-neural HTTP request distribution algorithm called FNRD, which assigns each incoming request to the server in the Web cluster with the quickest expected response time. The fuzzy mechanism is used to estimate the expected response times. A neural-based feedback loop is used for real-time tuning of response time estimates. To evaluate the system, we have developed a detailed simulation and workload model using CSIM19 package. Our simulations show that FNRD can be more effective than its competitors.

1 Introduction

Users perceive good Internet performance as characterized by low latency and high throughput. When browsing the Web, users are concerned with the performance of downloading entire pages. Various solutions impact the performance of downloading individual Web objects as well as the base page. Among them there are Web site architectures. A current trend is to organize a number of servers in a cluster with front-end components, called dispatchers or Web switches, which distribute the incoming requests among the Web servers. Such Web site system architectures are scalable and can better overcome the peak demands for Web services. The content-aware Web switches with adaptive policies, effectively combining client and server information, may provide better performance in Web services support, especially when we are interested in optimizing the quality of Web service according to the idea of 'Quality of Web-based Services' (QoWS).

To meet these requirements we propose the use of artificial intelligence methods, namely *fuzzy sets* and *neural networks* in the development of a content-aware Web switch that may take advantage of the fuzzy-neural approach in the problem solving. Fuzzy and neural networks-based problem formulation work to solve issues for different types of computer systems, for example in computer networks [9], distributed computer systems [11], and WWW servers [10,12]. However, not in the area of Web switch design, except our research [4,5].

In [4] we developed the fuzzy-neural adaptive HTTP request distribution algorithm called FNRD (Fuzzy-Neural Request Distribution). Our previous works

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 192–201, 2006.
© Springer-Verlag Berlin Heidelberg 2006

evaluated FNRD via trace-driven simulation [5] (using real-life trace data from the 1998 World Cup Soccer Web site [1]) and via benchmarking experiments [4] (using our Web switch prototype, real Web servers and own benchmarking system). This paper presents new results of a more general study on FNRD performance. Our research is based on a well-accepted modeling approach [7] and CSIM19 tool [14] for Web systems simulations.

FNRD minimizes the request response time and because of that it meets the demand that the Web switch should distribute requests taking into account the user perspective. It dispatches each incoming request to the server with the quickest expected response time which is estimated at the moment for that individual request based on the current knowledge on the server load state. A fuzzy estimation mechanism with client and server awareness is used. Due to the neural network-based mechanism which is employed for real-time tuning of request response estimates, our Web switch learns how the changes in the Web cluster configuration, server loading, and the workload characteristics affect the response times, and tunes its decisions according to these changes. Special attention has been devoted to the development of a new combined system- and component-level model for a Web cluster with back-end databases equipped with our Web switch, as well as the application of a new Web workload model.

The rest of this paper is organized as follows. In the next section we review related work and introduce the problem of HTTP distribution. Section 3 presents the fuzzy adaptive request distribution mechanism. Section 4 shows the performance evaluation experiments and discusses the results. Section 5 concludes the paper and proposes future work.

2 Related Work and Background

HyperText Transfer Protocol (HTTP) is the method used to transfer information on the World Wide Web. It is an Internet protocol providing a way to receive HTML pages from WWW servers. Although a user's browser issues one request at a time for a Web page, it usually causes multiple client-to-server interactions because retrieving one Web page requires, on the average, $1+n$ accesses on the server (1 access to retrieve the HTML file and n accesses to retrieve its n embedded objects (resources)). All these interactions are referred to us as HTTP requests.

In the paper we deal with Web-server clusters consisting of multiple server nodes, built on a local area and equipped with a mechanism to distribute client requests among the nodes. Web servers in a cluster work collectively as a single Web resource in the network. The distribution algorithm is embedded in a Web switch providing the required level of performance and availability of a Web service. Taking into account the OSI protocol stack, we may think about *Layer-4* and *Layer-7* Web switches. Layer-4 switch is *content-blind* as it knows only information about TCP/IP connections. Layer-7 switch is *content-aware* because, in addition to layer 4 information, it makes use of the layer-7 data (that is HTTP content).

The main classes of request distribution algorithms are *static*, those that do not consider any information about the current state of the system at the time of decision-making, and *dynamic*, those that use system state information to prepare the distribution decisions. Both policies have been widely studied, e.g. see valuable literature in [8]. Further taxonomy may include *adaptive* algorithms for systems where some parameters of the distribution algorithm are changing due to fluctuating system and workload conditions. Adaptive algorithms are not yet deeply studied for the Web and need further research [8]. Finally, the request distribution algorithms can be client information aware [7]. Then the Web switch takes into account information about the user that can be extracted from the HTTP request. Alternatively, the policy can be server information aware when server state information is utilized in decision-making [7].

Two basic performance strategies are mainly considered in Web switch operation: *load balancing* and *load sharing*. Typical load metrics are either the number of new and active TCP connections for each server or server CPU utilization, or queue length. Load balancing policies are aimed at equalizing the load among all servers in a cluster, while load sharing algorithms attempt to smooth out transient peak overload periods on some cluster nodes. Layer-4 switches distribute requests to even the load. The Layer-7 Web switch-based clustered architectures are aimed at sharing more than balancing the cluster workload. Both performance strategies focus on the most advantageous Web site resource utilization. However, they represent different perspectives. Load balancing comes from the site administrator's viewpoint, whereas load sharing would take into account the end-user's point of view. Such user-oriented distribution approach is a new direction in Web switch design.

Paper [8] presents the extensive taxonomy of HTTP requests, distribution methods and algorithms, including local and geographical, static and dynamic, content-blind and content-aware, client-aware and server-aware approaches, and algorithms. The authors concluded that the Web switch architectures need further research in the direction of the development of new content-aware adaptive policies combining effectively client and server information.

We have responded to the challenge issued in the literature and developed a novel HTTP request fuzzy-neural distribution algorithm, FNRD [4,5], which is a client-and-server-aware, dynamic and adaptive dispatching policy. The dispatching decisions are dynamic, as they are based on current knowledge about client request and server state. We think that our solution would be valuable for the quality of Web service aware developments and might be well-suited for user perceived end-to-end performance, e.g. in situations when a Web server support for differentiated services is required [7]. To the best of our knowledge, no other works in the area use a learning approach. Also fuzzy logic has not been applied yet. Fuzzy approach may be especially beneficial when a small amount of uncertain and imprecise information about a system under consideration is provided whereas neural networks have learning and adaptive capabilities. These system modeling approaches have been very widely used in studying several decision-making systems, also designed for the need of computer networks. For example,

[10,12] propose a fuzzy controller for Apache Web server and [9] presents a QoS-provisioning neural-fuzzy connection admission controller for multimedia high-speed networks.

In this paper, for comparison reasons, we use two widely known request distribution policies: *Round Robin* (RR) [7] and *Locality Aware Request Distribution* (LARD) [2,15]. RR is a content-blind baseline policy that allocates arriving requests based on a round robin discipline. It is often used in evaluation studies as 'a point of reference'. LARD is known to be the best content-aware dynamic policy that is aimed to distribute files among servers and increase the RAM cache hit rate. LARD works as follows. When a Web switch receives a request it extracts the URL and if this URL is requested the first time then the Web switch selects the server with the lowest load (taking into account the number of connections) and forwards the request to the selected server. If the URL is known for the Web switch, it forwards it to the previously used server for that URL. If that server is overloaded then a request is forwarded to a lightly-loaded server. Because a URL is always forwarded to the same server, very likely the server will have the requested object in its RAM.

3 Fuzzy-Neural HTTP Request Distribution

We deal with an algorithm for a Layer-7 Web switch and use two-way architecture of cluster-based Web system where all HTTP requests arrive through the Web switch and the obtained resources are sent back through the Web switch. The goal of the Web switch is to optimize the request response time that is the time between the opening and closing of the TCP session that is established between the Web switch and a target Web server to get a resource. This goal can be realized by the redirection of the request to the server that is expected to provide the fastest response time. We assume that the resources are fully replicated on all servers and each server can handle each request. Fig. 1a presents the major components of our fuzzy-neural Web switch: *Executor*, *MIN* and *Model of Server* modules. Each server has its own corresponding Model of Server (Fig. 1b).

In our model we evaluate the server (each server consists of the Web server and a database server) load by the combination of two independently observed load indexes: the Web server load a (the number of TCP connections active on a Web server) and the back-end database server load b (the number of TCP connections active on a database server). The values of these measures are discrete. The server load is described by the tuple (a, b). They are represented in the load table (Fig. 1b) as the tuples (Aan, Abm), where Aan and Abm are fuzzy sets. Based on the semantics of both loads we define the same eleven membership functions (Fig. 1c). The shape and distribution of the fuzzy sets underlying control variables correspond to the way in which control is expressed in the model. Increasing the number of fuzzy sets generally increases the precision of the estimation but at a price of more number of rules and more computations. The model is capable of balancing between the high precision of the estimation of the

request response time desired for the loads lower than (and around) the system operating point, and the computation time needed to calculate the estimator. The operating point is the number of requests that are handled by the system simultaneously in typical situations. It was determined on the basis of empirical data as eight requests. Consequently, we use eight singleton membership functions for the first eight values of inputs and triangular and trapezoid membership functions distributed for higher loads. The requests x_i, i=1,..., are serviced independently in the order of their arrival. When x_i request arrives, each Model of Server (Fig. 1b) calculates its own estimator of the response time t_i^{*s}, $s = 1, .., S$ assuming that this request is to be serviced by the given server, and making use of current server load information $(a, b)^s, s = 1, .., S$ and the knowledge about the request x_i (type and size of the object).

First, the request x_i is classified to the k-th class using the classification function $\Phi(x_i)$. K classes are defined. After that, the *estimation mechanism* determines the expected response time t_i^{*s} on the basis of the set Δ_{ikj} of elements in the k-th row of the load table for the s-th server, where Δ_{ikj}=$\{t_{ikj}\}$, for given i, k and for *fired j's* , $j \in J^*, J^* \subset J$, where J=$\{1, \ldots, 121\}$ is the set of indexes of rules defined in the *rule base* with 121 *if-then* rules. J^* is the set of indexes of rules in the rule base that are fired by a current crisp value of server load $(a, b)^s$ (i.e. such that $\mu_{Rj}(a, b)^s > 0$).

The rule base is as follows: $R1$: IF (a=$Aa1$) AND (b=$Ab1$) THEN (t=$T1$), ... , Rj: IF (a=Aam) AND (b=Abn) THEN (t=Tj),..., $R121$: IF (a=$Aa11$) AND (b=$Ab11$) THEN (t=$T121$), where a, b - server loads (number of active connections), t - response time, $Aa1, \ldots, Aa11$, $Ab1, \ldots, Ab11$ - servers load fuzzy sets, $T1, \ldots, T121$- sets of output t.

Next, the MIN module determines the target server as the one for which the estimated response time is minimal. Our estimation mechanism is based on a fuzzy-neural model and follows Mamdani's model [13]. Figure 1d presents the proposed estimation mechanism with adaptability. The fuzzification block and the rule base are constructed like typical fuzzy models, whereas in the defuzification block we propose to use the artificial neuron. The input in the model is the load tuple (a, b). For both loads we use eleven fuzzy sets that cover all possible load ranges. We have K^*S estimation mechanisms because we need to have an individual estimation mechanism for each class of objects in each Model of Server module. The system continuously adapts to the changes in the workload and Web cluster characteristics. Information about server loads is collected independently of request handling. When the request is completed by the target server, then the Executor measures the response time, and uses this measured value as the observed value of response time $\widetilde{t}_i^{\,s}$ for request x_i in the *learning algorithm* to recalculate the set $\Delta_{(i+1)kj}$ of new values of the response times, where $\Delta_{(i+1)kj}$ =$\{t_{(i+1)kj}\}$, for given i, k and for *fired j's* , $j \in J^*$. Set $\Delta_{(i+1)kj}$ refines the old values stored in the k-th row of the load table of the s-th server. To teach the network we use the traditional *error back propagation* algorithm.

The new weights $t_{(i+1)kj}$ are calculated in accordance with the learning rule for the ADALINE neuron, each time after completing the request, according to the

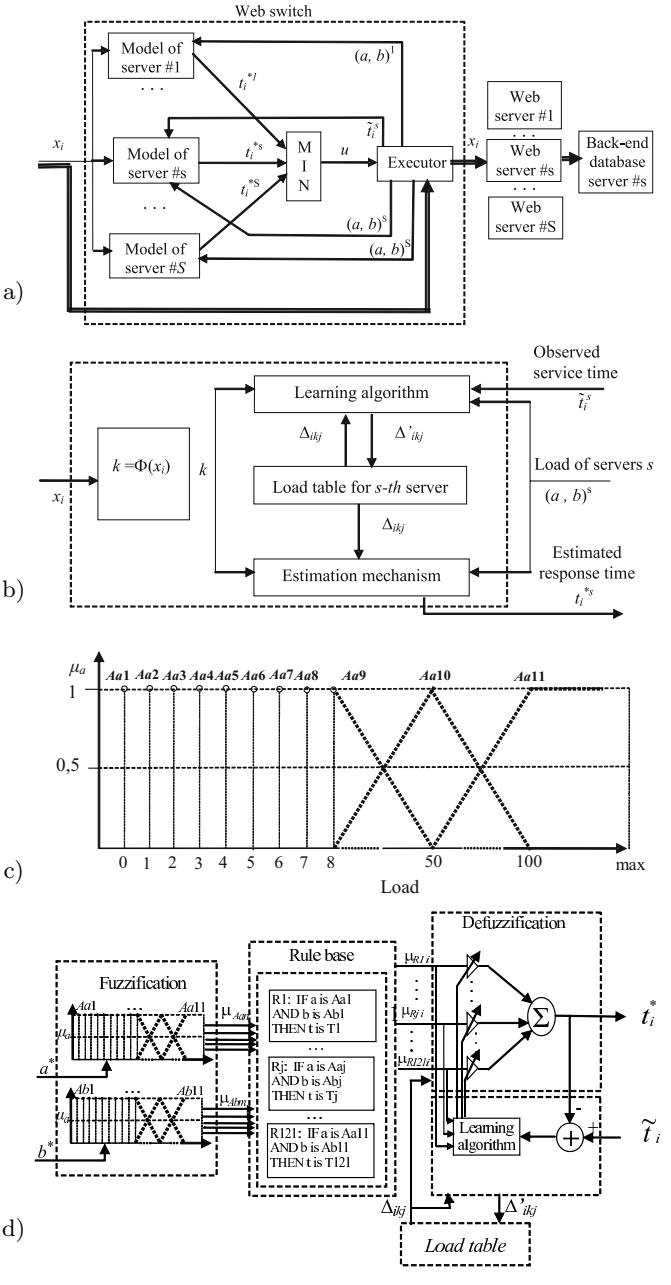


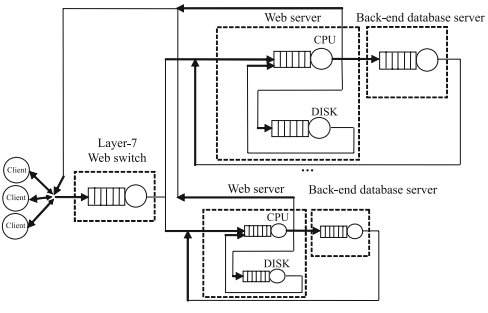
Fig. 1. Fuzzy-Neural Web switch: (a) Architecture; (b) Model of Server module; (c) Membership functions for inputs; (d) Estimation mechanism

formula $t_{(i+1)kj} = t_{ikj} + \eta\mu_{Rj}(\tilde{t}_i - t_i^*)$, where t_i^* is the estimated request response time, $\tilde{t}_i$ is the observed request response time, and η=0.13 is the learning rate index.

4 Simulation and Results

The model of a cluster-based Web system used in our simulation is shown in Fig. 2a. We assumed that both the Web switch and local area network are fast enough and do not introduce significant delay that might influence results. The main delay in request servicing is assumed to be introduced by Web servers and database servers. Our CSIM19 based simulator runs for the similar values of such system parameters as CPU speed, amount of memory, number of disks and other as assumed in [2,7,15].

a)



b)

Static requests

Category	Distribution	Paremeters
Requests per session	Inverse Gaussian	μ=3.86, λ=9.46
User think time	Pareto	α=1.4, k=1
Objects per request	Pareto	α=1.33, k=2
HTML object size	Lognormal Pareto	μ=7.630, σ=1.001 α=1, k=10240
Embedded object size	Lognormal	μ=8.215, σ=1.46

Dynamic requests

Type	Mean service time [msec]	Frequency
Low intensive	5	0.85
Medium intensive	10	0.14
High intensive	20	0.01

Fig. 2. (a) A simulation model; (b) Workload model parameters

The processing costs are calculated for Pentium II 300 MHz PC with FreeBSD 2.2.5 and Apache 1.3.3 Web server. We use real-life parameters to setup the Web cluster components. Connection establishment and teardown costs are set at 278 μs of CPU time and 527 μs, respectively, while transmit processing incurs 24 μs per 512 bytes. Disc costs are the following: reading a file from disk has a latency of 28 ms, the disk transfer time is 410 μs per 4 KByte. Additionally, for files larger than 44 KBytes, an additional 14 ms is charged for every 44 KBytes of file length in excess of 44 KBytes. The Least Recently Used (LRU) cache replacement policy is used, however files with a size of more than 500 KB are never cached. The total memory size used for caching is 85 MB.

To evaluate the system we have developed new detailed simulation (Fig. 2a) and workload (Fig. 2b) models. CSIM19 package [14] was used in the simulation as it is a well-accepted simulation tool used also for building Web systems models. We evaluate the system performance using the workload model incorporating the most recent research on Web load which is heavy-tailed and self-similar [3, 7]. We consider two classes of requests: *static* and *dynamic*. Static requests are serviced directly by the Web server, whereas in the dynamic requests the objects are dynamically generated by the back-end servers. Figure 2b shows the probability

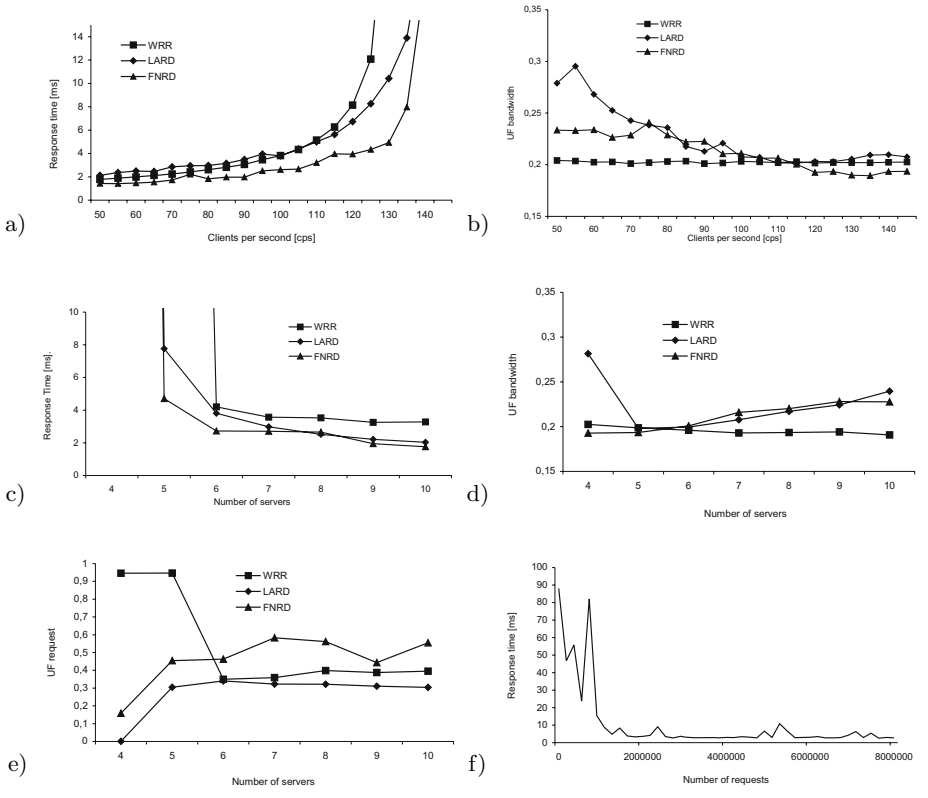


Fig. 3. Simulation results: (a) response time v. load; (b) *UF request* v. load; (c) response time v. S; (d) *UF bandwidth* v. S; (e) *UF request* v. S; (f) convergence characteristics

distributions and parameters we use in the workload model. The size of dynamic requests is simulated based on the same size distribution as the static ones. They are additionally classified into three classes, namely *high*, *medium* and *low intensive*, according to the workload size incurred while database processing. The service time on the database server for a dynamic request is modeled according to a hyper-exponential distribution with the parameters as shown in Fig. 2b. In the simulation we assumed the basic workload composition consisting of 80% of static requests and 20% of dynamic requests. We simulated the browsing of a Web site with the total size of 200 MB size. The first simulation we performed with 4 servers (each server is the system consisting of the Web and database server) in the cluster and measured the average response times versus the number of Web clients serviced by the cluster per second. As the load balance measure we propose to use the *Load Balance Metric* (LBM), which is the weighted average of the instantaneous peak-to-mean ratios [6]. Let $load_{i,j}$ and $peak_load_j$ denote the load of server i (of S servers) at the $j-th$ sampling point (of M such points) and the highest load on any server at the $j-th$ sampling point, respectively. Then the *peak-to-mean ratio* and the *LBM* are defined as follows:

$$peak-to-mean\,ratio = \frac{peak_load_j}{\left(\sum_{i=1}^{S} load_{i,j}\right)/S} \text{ and } LBM = \frac{\sum_{j=1}^{M} peak_load_j}{\left(\sum_{j=1}^{M}\sum_{i=1}^{S} load_{i,j}\right)/S}\ .$$

LBM can range from 1 to at most S. Small values of the LBM indicate better load balancing performance, i.e. smaller peak-to-mean load ratios, than larger values. The optimal LBM value is 1. We can interpret the LBM metric in the following way. On a two-server system, an LBM of 1.0 indicates perfect load balance. A load imbalance where one server holds 60% of load and the other server has 40% of load yields an LBM of 1.2 (60/((60+40)/2)=1.2. We can also define the *unbalancefactor* as the percentage variation of the LBM value with respect to the optimal LBM value, $UF = ((LBM-1)/(S-1))100$, where $0 \leq UF \leq 100$.

The unbalance factor UF is the performance measure that is independent of the number of servers S and defines the variation of the load with the respect to the even load balancing. UF also gives a good measure to compare the performance of different distribution algorithms. As the load can be expressed by either the total number of requests serviced by the Web cluster or by the total bandwidth of data transmitted back to the clients, we can consider either UF *request* or UF *bandwidth* unbalance factor, respectively. It is shown (Fig. 3a) that FNRD outperforms LARD and RR for the whole range of the load size. Figure 3b presents the UF *request* v. load characteristics for 4 servers. As in the previous experiment the FNRD outperforms other policies.

The next experiments were performed to study how the number of servers affects the performance of the Web cluster for a given load size. We generated the load of 140 clients/sec. Fig. 3c demonstrates that when considering the response time the FNRD algorithm is the best choice for all server configurations. However, the results on Figures 3d - 3e show that the fuzzy-neural FNRD policy is not a best load balancing policy for all configurations. Only for a small number of servers (S=4, 5 or 6) it outperforms LARD and RR in load balancing. Nevertheless, one should remember that FNRD was developed having in mind the response time optimization, not load balancing. Fig. 3f illustrates the convergence characteristic of FNRD when estimating the response time for a cluster with 4 servers under 130 clients/sec workload. The experiment showed that the average estimated response time converges starting at about 2000000 requests.

5 Conclusions and Future Work

A fuzzy-neural HTTP distribution policy called FNRD for Web cluster was evaluated through the simulation using CSIM19 package. We showed that a fuzzy-neural approach is useful in the design of the content-aware Web switch. FNRD optimizes request response time and outperforms substantially the most popular content-blind RR policy as well as the state-of-the-art content-aware LARD policy. In a load balancing challenge it is a winner for smaller clusters.

There is still the challenge in the design of fuzzy-neural Web switch. Below we would like to suggest some future research directions: (1) Performance evaluation of FNRD algorithm in the overloaded conditions; (2); Analysis of FNRD algorithm in the context of supporting differentiated service; and (3) Design of a new fuzzy-neural Web switch for distributing HTTP requests at the global (i.e. Internet) level.

References

1. Arlit M., Jin T.: A workload characterization study of the 1998 World Cup Web site. IEEE Network, 14(3), (2000) 30-37
2. Aron M., Druschel P., Zwaenepoel W.: Efficient support for P-HTTP in cluster-based Web servers. Proc. Usenix Ann. Techn. Conf., Monterey, CA. (1999)
3. Barford P., Crovella M.E.: A performance evaluation of Hyper Text Transfer Protocols. Proc. ACM SIGMETRICS '99, Atlanta, (1999) 188-197
4. Borzemski L., Zatwarnicki K.: A fuzzy adaptive request distribution algorithm for cluster-based Web systems. Proc. of 11^{th} Conf. on Parallel, Distributed and Network-based Processing, IEEE CS Press Los Alamitos (2003) 119-126
5. Borzemski L., Zatwarnicki K.: Using Adaptive Fuzzy-Neural Control to Minimize Response Time in Cluster-Based Web Systems. Advances in Web Intelligence. LNAI, Vol. 3528. Springer-Verlag Berlin (2005) 63-68
6. Bunt R., Eager D., Oster G., Wiliamson C.: Achieving load balance and effective caching in clustered web servers. Proc. 4^{th} Int'l Web Caching Workshop (1999)
7. Cardellini V., Casalicchio E., Colajanni M., Mambelli M.: Web switch support for differentiated services. ACM Perf. Eval. Rev., Vol. 29, No. 2 (2001) 14-19
8. Cardellini V., Casalicchio E., Colajanni M., Yu P.S.: The state of the art in locally distributed Web-server systems. ACM Comp. Surv. Vol. 34, No. 2 (2002) 263-311
9. Cheng R.G., Chang C.J.: A QoS-provisioning neural fuzzy connection admission controller for multimedia networks. IEEE Trans. on Networking, vol. 7, no. 1, Feb. (1999) 111-121
10. Diao Y., Hellerstein J.L., Parekh S.: Optimizing quality of service using fuzzy control, Proc. of Distributed Systems Operations and Management, (2002)
11. Kwok Y.-K., Cheung L.-S.: A new fuzzy-decision based load balancing system for distributed object computing. J. Parallel Distribut. Comput. 64 (2004) 238-253
12. Liu X., Sha L., Diao Y., Froehlich S., Hellerstein J.L., Parekh S.: Online response time optimization of Apache Web server. Int'l Workshop on Quality of Service (2003)
13. Mamdani E.H.: Application of fuzzy logic to approximate reasoning using linguistic synthesis. IEEE Trans. on Computers, vol. C-26, No.12, Dec. (1977) 1182-1191
14. Mesquite Software Inc. CSIM19 User's Guide. Austin, TX. http:www.mesquite.com
15. Pai V.S., Aront M., Banga G., Svendsen M., Druschel P., W. Zwaenpoel, Nahum E.: Locality-aware request distribution in cluster-based network servers. Proc. of 8th ACM Conf. on Arch. Support for Progr. Languages (1998)

Fuzzy Approach to Correlation Function

Bohdan S. Butkiewicz

Warsaw University of Technology, Warsaw, Poland
bb@ise.pw.edu.pl
http://www.ise.pw.edu.pl/~bb/index.html

Abstract. The paper presents second author's step toward building quite new theory of fuzzy signals. Before, in the first step author defined fuzzy Fourier transform. Here, an approach to description of fuzzy correlation functions is proposed. New definitions, called later fuzzy correlation and defuzzified correlation are proposed for fuzzy signals. The definitions are based on new concepts of the class L_f^2, L_{Tf}^2, and class M_f for fuzzy functions. Comparison with conventional approach and an example are shown.

1 Introduction

Concept of correlation plays an important part in the theory of many branches of science as electricity, optical or acoustical signal transmission and filtering, image processing, statistics, economy, etc. Conventional approach to signal theory uses determined functions for description of signals or random processes for uncertain signals. Here, author tries to use fuzzy description of uncertain signals. In previous works [1] [3] author introduced concept of fuzzy Fourier transform. It was the first step to build basis of spectral analysis of fuzzy functions. New definition of correlation of fuzzy functions is a second step to build basis of fuzzy signal theory. In this paper conventional definition of correlation function is enlarged to fuzzy functions. New definition is proposed in such a way to preserve general properties of correlation functions. Proofs of properties are not shown here. It can be found in [1], [2], [3].

2 Conventional Approach to Correlation

At the beginning conventional definitions of correlation functions are discussed below. Generally, correlation function is closely related to scalar product in appropriate Hilbert space [4]. Thus, the definition of correlation depends on the class of functions. Three classes of functions are most important: L^2, L_T^2, and M. In any case formal definition of correlation function is something different, but all the time based on scalar product. Therefore, for simplify notations in the paper the correlation functions will be denoting by the same symbols in different classes.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 202–211, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Class L^2. Function $x(t)$ (real or complex) belongs to the space $L^2(t_1, t_2)$ iif the integral below is finite

$$\| x \|^2_{L^2} = \int_{t_1}^{t_2} |x(t)|^2 dt \tag{1}$$

The square root of this integral plays a role of a norm in the space L^2. If the values of t_1, t_2 are infinite, then $x(t) \in L^2(-\infty, \infty)$. Physical interpretation of the $\| x \|^2_{L^2}$ is energy of signal $x(t)$. Thus, the class of such functions is sometimes called the class of *functions with finite energy.*

Scalar product in L^2 is defined by

$$(x, y)_{L^2} = \int_{t_1}^{t_2} x(t) y^*(t) dt \tag{2}$$

Definition (Correlation functions in L^2)
Let $x(t)$ and $y(t)$ belong to $L^2(t_1, t_2)$. Suppose that integrals

$$R_{xy}(\tau) = (x(t), y(t-\tau))_{L^2} = \int_{t_1}^{t_2} x(t) y^*(t-\tau) dt \tag{3}$$

$$R_{yx}(\tau) = (y(t), x(t-\tau))_{L^2} = \int_{t_1}^{t_2} y(t) x^*(t-\tau) dt \tag{4}$$

are finite. Symbol * denotes operation of conjugation when $x(t), y(t)$ are complex. Functions $R_{xy}(\tau)$ and $R_{yx}(\tau)$ are called *correlation functions* in L^2.

It can be shown that

$$R_{xy}(\tau) = \int_{t_1}^{t_2} x^*(t) y(t+\tau) dt \tag{5}$$

$$R_{yx}(\tau) = \int_{t_1}^{t_2} y^*(t) x(t+\tau) dt \tag{6}$$

Autocorrelation function is defined by $R_{xx}(\tau) = (x(t), x(t-\tau))_{L^2}$.

Class L^2_T. Periodic function $x(t)$ with period T belongs to the space L^2_T iif the integral

$$\| x \|^2_{L^2_T} = \frac{1}{T} \int_0^T |x(t)|^2 dt \tag{7}$$

is finite.

Physical interpretation of the integral is power of signal $x(t)$ in $[0, T]$. Square root of this integral plays a role of norm in the space L^2_T. Interpretation of the norm is effective value of $x(t)$. Thus, the class of such functions is called the class of *periodic functions with finite power.*

Scalar product in L^2_T is equal

$$(x, y)_{L^2_T} = \frac{1}{T} \int_0^T x(t) y^*(t) dt \tag{8}$$

Definition (Correlation function in L_T^2)
Let $x(t)$ and $y(t)$ belong to L_T^2. Suppose that integrals

$$R_{xy}(\tau) = (x(t), y(t-\tau))_{L_T^2} = \frac{1}{T}\int_0^T x(t)y^*(t-\tau)dt \tag{9}$$

$$R_{yx}(\tau) = (y(t), x(t-\tau))_{L_T^2} = \frac{1}{T}\int_0^T y(t)x^*(t-\tau)dt \tag{10}$$

are finite. Functions $R_{xy}(\tau)$ and $R_{yx}(\tau)$ are called *correlation functions* in L_T^2.

Similarly as before it can be shown that

$$R_{xy}(\tau) = \frac{1}{T}\int_0^T x^*(t)y(t+\tau)dt \tag{11}$$

$$R_{yx}(\tau) = \frac{1}{T}\int_0^T y^*(t)x(t+\tau)dt \tag{12}$$

Class *M*. Function $x(t)$ belongs to the space M (Marcinkiewicz space) iif exists the limit of integral

$$\| x \|_M^2 = \overline{\lim_{T\to\infty}}\frac{1}{T}\int_{-T/2}^{T/2} |x(t)|^2 dt \tag{13}$$

where $\overline{lim}$ denotes superior limit.

Physical interpretation of the integral is mean power of signal $x(t)$ in $(-\infty, \infty)$. Square root of the expression above is the norm in the space M. Interpretation of the norm is effective value of $x(t)$. Thus, the class of such functions is called the class of *functions with finite mean power*.

The integral

$$(x, y)_M = \overline{\lim_{T\to\infty}}\frac{1}{T}\int_{-T/2}^{T/2} x(t)y^*(t)dt \tag{14}$$

plays a role of pseudo-scalar product in M, because it not fulfils some axioms of scalar product.

Definition (Correlation in M)
Let $x(t)$ and $y(t)$ belong to M. Suppose that integrals

$$R_{xy}(\tau) = (x(t), y(t-\tau))_M = \overline{\lim_{T\to\infty}}\frac{1}{T}\int_{-T/2}^{T/2} x(t)y^*(t-\tau)dt \tag{15}$$

$$R_{yx}(\tau) = (y(t), x(t-\tau))_M = \overline{\lim_{T\to\infty}}\frac{1}{T}\int_{-T/2}^{T/2} y(t)x^*(t-\tau)dt \tag{16}$$

are finite. Function $R_{xy}(\tau)$ and $R_{yx}(\tau)$ are called *correlation functions* in M.

Class Γ. If $\xi(t)$ is second-order stationary random processes then it belongs to the space Γ iif expected square value

$$\| x \|_{\Gamma}^2 = \overline{\xi^2} = \int_{-\infty}^{\infty} x^2 f_{\xi}(x) dx \tag{17}$$

is finite, where $f_{\xi}(x)$ is probability density function of $\xi(t)$.

Definition (Correlation in Γ)
If stationary processes $\xi(t), \eta(t) \in \Gamma$ then correlation functions are defined as expected value

$$R_{\xi\eta}(\tau) = \overline{\xi(t)\eta(t-\tau)} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(t)y(t-\tau) f_{\xi\eta}(x,y;t,\tau) dxdy \tag{18}$$

$$R_{\eta\xi}(\tau) = \overline{\eta(t)\xi(t-\tau)} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} y(t)x(t-\tau) f_{\xi\eta}(x,y;t,\tau) dxdy \tag{19}$$

where $f_{\xi\eta}$ is two-dimensional probability density function.

As before both correlation functions are defined as scalar product, now in Γ

$$R_{xy}(\tau) = (x(t), y(t-\tau))_{\Gamma} \qquad R_{yx}(\tau) = (y(t), x(t-\tau))_{\Gamma} \tag{20}$$

3 Fuzzy Approach to Correlation

As it can be seen from further definitions, correlation function is the scalar product in appropriate functional space. Only class Γ describes uncertainty of a signal. Nevertheless, as it can be shown here, any of previously defined class can be enlarged for fuzzy functions. Thus, is necessary to define in reasonable way new spaces, the spaces of fuzzy functions.

Class L_f^2 and L_μ^2. Fuzzy function $x(t,\alpha)$ (real or complex), where α is a parameter being a fuzzy set, belongs to the space $L_f^2(t_1,t_2)$ iif for any $a \in supp(\alpha)$ the integral

$$\| x \|_{L_f^2}^2 = \int_{t_1}^{t_2} |x(t,a)|^2 dt \tag{21}$$

is finite.

It can be shown that square root of above integral fulfills for any $a \in supp(\alpha)$ all axioms required for a norm. This concept leads to *fuzzy norm*. If someone not agreed with fuzziness of the norm, he can use another definition of *crisp norm* of fuzzy function defined as below.
Fuzzy function $x(t,\alpha) \in L_\mu^2(t_1,t_2)$, where $t_1, t_2 \in (-\infty,\infty)$, iif exists an integral

$$\overline{|\, x(t,\alpha)\,|^2} \triangleq \| x \|_{L_\mu^2}^2 = \frac{\int_{t_1}^{t_2}\int_{supp(\alpha)} |x(t,a)|^2 \mu_\alpha(a) da\, dt}{\int_{supp(\alpha)} \mu_\alpha(a) da} \tag{22}$$

where overline denotes here mean value and denominator is the surface under $\mu_\alpha(a)$. The space can be called *space of fuzzy functions with finite energy.*

Scalar products of $x(t,\alpha),\ y(t,\beta)$ in L_f^2 is defined by

$$(x,y)_{L_f^2} = \int_{t_1}^{t_2} x(t,a)y^*(t,b)dt \tag{23}$$

where $a \in supp(\alpha)$, $b \in supp(\beta)$. The set of all values of $(x,y)_{L_f^2}$ will be called *scalar product* in L_f^2.

In the space L_μ^2 definition is based on the norm (22)

$$(x,y)_{L_\mu^2} = \frac{\int_{t_1}^{t_2}\int_{supp(\alpha)}\int_{supp(\beta)} x(t,a)y^*(t,b)\mu_\alpha(a)\mu_\beta(b)\,da\,db\,dt}{\int_{supp(\alpha)}\int_{supp(\beta)}\mu_\alpha(a)\mu_\beta(b)\,da\,db} \tag{24}$$

where $a \in supp(\alpha)$, $b \in supp(\beta)$. This scalar product is crisp.

In signal theory very important case is a situation when signal $x(t)$ is defined on the interval $(-\infty,\infty)$. Therefore, in the text below it is supposed $t \in (-\infty,\infty)$.

Definition (Fuzzy correlation functions in L_f^2)
Let $x(t,\alpha)$ and $y(t,\beta)$ belong to $L_f^2(-\infty,\infty)$. Suppose that integrals

$$R_{xy}(\tau,a,b) = (x(t,a),y(t-\tau,b))_{L_f^2} = \int_{-\infty}^{\infty} x(t,a)y^*(t-\tau,b)dt \tag{25}$$

$$R_{yx}(\tau,a,b) = (y(t,b),x(t-\tau,a))_{L_f^2} = \int_{-\infty}^{\infty} y(t,b)x^*(t-\tau,a)dt \tag{26}$$

are finite for any $a \in supp(\alpha),\ b \in supp(\beta)$. The sets $\mathcal{R}_{xy}(\tau,\alpha,\beta)$, $\mathcal{R}_{yx}(\tau,\alpha,\beta)$ of all functions $R_{xy}(\tau,a,b)$ and $R_{yx}(\tau,a,b)$ for $a \in supp(\alpha), b \in supp(\beta)$ will be called *fuzzy correlation functions* in L_f^2.

If $\mu_\alpha(a)$ and $\mu_\beta(b)$ are membership functions of sets α and β then membership function $\mu_\mathcal{R}(a,b)$ for $\mathcal{R}_{xy}(\tau,\alpha,\beta)$ can be found as product

$$\mu_\mathcal{R}(a,b) = \mu\alpha(a)\mu_\beta(b) \tag{27}$$

Generally, if α and β are in a fuzzy relation $\mathcal{R}(\alpha,\beta)$ with membership $\mu_\mathcal{R}(a,b)$ then fuzzy correlation function is described by

$$\mathcal{R}_{xy}(\tau,\alpha,\beta) = \int \mu_\mathcal{R}(a,b)/R_{xy}(\tau,a,b) \tag{28}$$

where Zadeh notation is used, i.e. the "integral" is considered as union [5].

It is possible to obtain defuzzified version of correlation using a defuzzification procedure, example area method

$$R_{xy}(\tau) = \frac{\int_{supp(\alpha)}\int_{supp(\beta)} R_{xy}(\tau,a,b)\,\mu_\mathcal{R}(a,b)\,da\,db}{\int_{supp(\alpha)}\int_{supp(\beta)}\mu_\mathcal{R}(a,b)\,da\,db} \tag{29}$$

The value $R_{xy}(\tau)$ will be called *defuzzified correlation function* in L_f^2.

Using formula (24) for scalar product $(x(t,\alpha), y(t,\beta))_{L^2_\mu}$ in L^2_μ definition of correlation in L^2_μ can take the following form

$$R_{xy}(\tau) = \frac{\int_{-\infty}^{\infty} \int_{supp(\alpha)} \int_{supp(\beta)} x(t,a) y^*(t-\tau,b)\, \mu_{\mathcal{R}}(a,b)\, da\, db\, dt}{\int_{supp(\alpha)} \int_{supp(\beta)} \mu_{\mathcal{R}}(a,b)\, da\, db} \tag{30}$$

where the product $\mu_\alpha(a)\mu_\beta(b)$ is replaced by more general form of relation $\mu_{\mathcal{R}}(a,b)$. Thus, this formula is similar to (29).

Autocorrelation function in L^2_μ is defined by

$$R_{xx}(\tau) = \frac{\int_{-\infty}^{\infty} \int_{supp(\alpha)} x(t,a) x^*(t-\tau,a)\, \mu_\alpha(a)\, da\, dt}{\int_{supp(\alpha)} \mu_\alpha(a)\, da} \tag{31}$$

It is known that autocorrelation in L^2 has property

$$R_{xx}(\tau) = R^*_{xx}(-\tau) \tag{32}$$

For real functions autocorrelation is even function of τ, i.e. $R_{xx}(\tau) = R_{xx}(-\tau)$. It is obvious that this property holds also for L^2_f and L^2_μ.

Class L^2_{Tf} and $L^2_{T\mu}$. Definitions for periodic signals can be enlarged similarly.

Definition (Fuzzy periodic signal)
Fuzzy function $x(t,\alpha)$ of crisp real argument t and fuzzy argument α will be called periodic if exist a real number T such that for all $t \in (-\infty, \infty)$ and all $a \in supp(\alpha)$

$$x(t+T,a) = x(t,a) \tag{33}$$

Fuzzy periodic function $x(t,\alpha)$ belongs to L^2_{Tf} iif for any $a \in supp(\alpha)$ the integral

$$\| x \|^2_{L^2_{Tf}} = \frac{1}{T} \int_0^T |x(t,a)|^2 dt \tag{34}$$

is finite.

Fuzzy periodic function $x(t,\alpha)$ belongs to $L^2_{T\mu}$ iif integral below is finite

$$\| x \|^2_{L^2_{T\mu}} = \frac{\frac{1}{T} \int_0^T \int_{supp(\alpha)} |x(t,a)|^2 \mu_\alpha(a)\, da\, dt}{\int_{supp(\alpha)} \mu_\alpha(a)\, da} \tag{35}$$

Similarly as before $\| x \|^2_{L^2_{Tf}}$ is fuzzy norm and $\| x \|^2_{L^2_{T\mu}}$ is crisp norm.

Definition (Fuzzy correlation functions in L^2_{Tf})
Let $x(t,\alpha)$ and $y(t,\beta)$ belong to L^2_{Tf}. Suppose that integrals

$$R_{xy}(\tau,a,b) = (x(t,a), y(t-\tau,b))_{L^2_{Tf}} = \frac{1}{T} \int_0^T x(t,a) y^*(t-\tau,b) dt \tag{36}$$

$$R_{yx}(\tau,a,b)=(y(t,b),x(t-\tau,a))_{L^2_{Tf}}=\frac{1}{T}\int_0^T y(t,b)x^*(t-\tau,a)dt \tag{37}$$

are finite where $a\in supp(\alpha),\ b\in supp(\beta)$.

The set $\mathcal{R}_{xy}(\tau,\alpha,\beta)$ of all functions $R_{xy}(\tau,a,b)$ for $a\in supp(\alpha),\ b\in supp(\beta)$ will be called *fuzzy correlation function* in L^2_{Tf}.

Membership function $\mu_{\mathcal{R}}(a,b)$ for $\mathcal{R}_{xy}(\tau,\alpha,\beta)$ can be found as product $\mu_{\mathcal{R}}(a,b)=\mu_\alpha(a)\mu_\beta(b)$ or more generally as membership of fuzzy relation $\mathcal{R}(\alpha,\beta)$.

Similarly for class $L^2_{T\mu}$ from formula (35) it follows that correlation function equals

$$R_{xy}(\tau)=\frac{\frac{1}{T}\int_0^T\int_{supp(\alpha)}\int_{supp(\beta)}x(t,a)y(t-\tau,b)\mu_{\mathcal{R}}(a,b)\,da\,db\,dt}{\int_{supp(\alpha)}\int_{supp(\beta)}\mu_{\mathcal{R}}(a,b)\,da\,db} \tag{38}$$

and it is defuzzified and normalized version of $\mathcal{R}_{xy}(\tau,\alpha,\beta)$.

Of course, all correlation functions (36) (37) (38) are periodic with period T.

Class M_f and M_μ. Fuzzy function $x(t,\alpha)$ belongs to the space M_f if for any $a\in supp(\alpha)$ exists the limit of integral

$$\|\ x\ \|^2_{M_f}=\overline{\lim_{T\to\infty}}\frac{1}{T}\int_{-T/2}^{T/2}|x(t,a)|^2dt \tag{39}$$

Fuzzy set $\|\ x\ \|_{M_f}$ will be called fuzzy norm in M_f.

Fuzzy function $x(t,\alpha)$ belongs to the space M_μ iif exists the limit of integral

$$\|\ x\ \|^2_{M_\mu}=\frac{\overline{\lim_{T\to\infty}}\frac{1}{T}\int_{-T/2}^{T/2}\int_{supp(\alpha)}|x(t,a)|^2\mu_\alpha(a)\,da\,dt}{\int_{supp(\alpha)}\mu_\alpha(a)\,da} \tag{40}$$

If $lim_{T\to\infty}$ exists in conventional sense then superior limit $\overline{\lim_{T\to\infty}}$ in formulas (39) (40) and formulas below can be replaced by ordinary limit.

In the class M scalar product not exists. Pseudo-scalar product is introduced only. Thus, correlation function in M_f is based on this pseudo-scalar product.

Definition (Fuzzy correlation function in M_f)
Let $x(t)$ and $y(t)$ belong to M_f. Suppose that integrals

$$R_{xy}(\tau,a,b)=\overline{\lim_{T\to\infty}}\frac{1}{T}\int_{-T/2}^{T/2}x(t,a)y^*(t-\tau,b)dt \tag{41}$$

$$R_{yx}(\tau,a,b)=\overline{\lim_{T\to\infty}}\frac{1}{T}\int_{-T/2}^{T/2}y(t,b)x^*(t-\tau,a)dt \tag{42}$$

are finite for any $a\in supp(\alpha)\ b\in supp(\beta)$. The set $\mathcal{R}_{xy}(\tau,\alpha,\beta)$ of all functions $R_{xy}(\tau,a,b)$ and appropriately set $\mathcal{R}_{yx}(\tau,\alpha,\beta)$ of all $R_{yx}(\tau,a,b)$ be called *fuzzy correlation functions* in M_f.

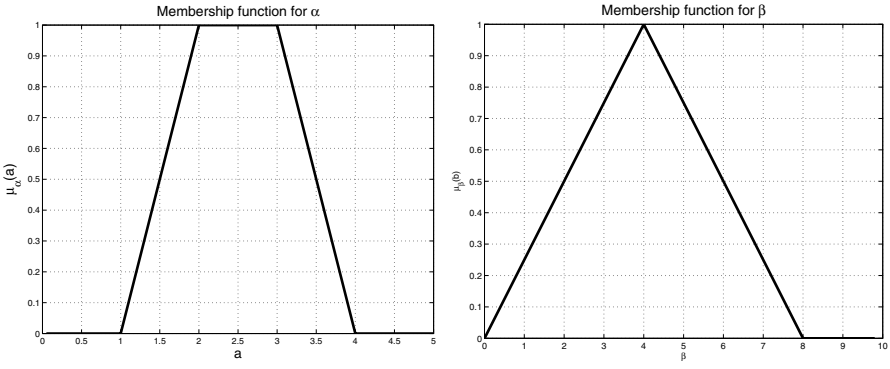


Fig. 1. Membership function for α (left) and for β (right)

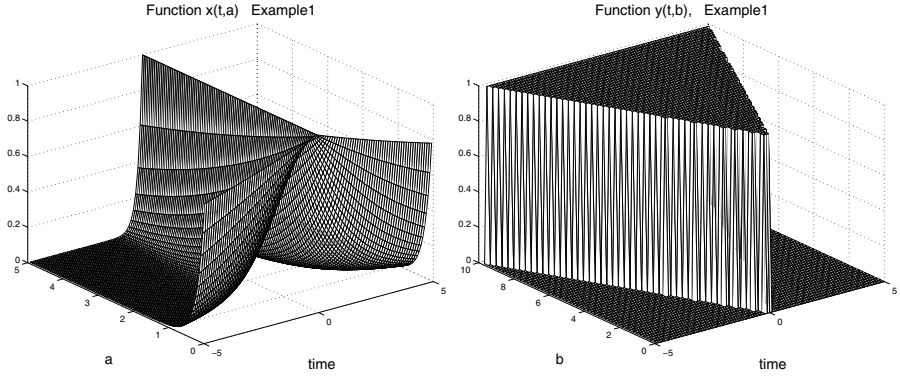


Fig. 2. Fuzzy functions $x(t, \alpha)$ (left) and $y(t, \beta)$ (right)

Membership function for $\mathcal{R}_{xy}(\tau, \alpha, \beta)$ can be found as product $\mu_{\mathcal{R}}(a, b) = \mu_\alpha(a)\mu_\beta(b)$ or more generally as membership of relation $\mathcal{R}(\alpha, \beta)$. The *defuzzified correlation function* in M_f can be found using for example area method.

Crisp correlation functions are possible to obtain also starting from appropriate pseudo-scalar product (14) and crisp norm (40). It obtains correlation function in M_μ

$$R_{xy}(\tau) = \frac{\overline{\lim_{T\to\infty}} \frac{1}{T} \int_{-T/2}^{T/2} \int_{supp(\alpha)} \int_{supp(\beta)} x(t,a) y^*(t-\tau, b) \mu_{\mathcal{R}}(a,b) da\, db\, dt}{\int_{supp(\alpha)} \int_{supp(\beta)} \mu_{\mathcal{R}}(a,b) da\, db} \tag{43}$$

Example. Fuzzy functions $x(t, \alpha) = exp(-\alpha \mid t \mid)$ and $y(t, \beta) = \prod(t/\beta)$ (rectangular shape) belong to L_f^2. Membership functions for fuzzy parameters have trapezoidal shape for α and triangular shape for β (Fig. 1). Fuzzy functions are shown in the Fig. 2.

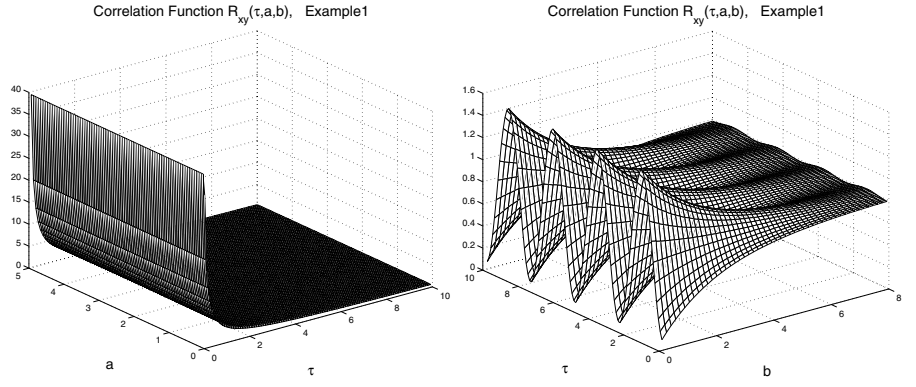


Fig. 3. Correlation function $\mathcal{R}_{xy}(\tau, \alpha, \beta)$ for constant $b = 4$ (left) and constant $a = 2.5$ (right)

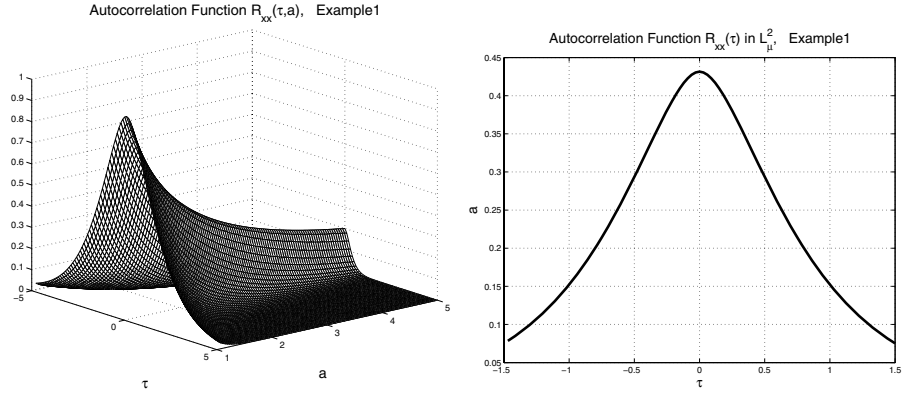


Fig. 4. Autocorrelation function $\mathcal{R}_{xx}(\tau, \alpha)$ in L_f^2 (left) and $R_{xx}(\tau)$ in L_μ^2 (right)

Fuzzy correlation function $R_{xy}(\tau, a, b)$ according to (25) is equal

$$R_{xy}(\tau, a, b) = \int_{-\infty}^{\infty} e^{-a|t|} \prod(\frac{t-\tau}{b}) dt = \int_{\tau-b/2}^{\tau+b/2} e^{-a|t|} dt = \frac{2}{a}[1 - e^{-b/2}\cos(a\tau)] \tag{44}$$

Function $R(\tau, a, b)$ depends on three arguments and presents a four dimensional surface. It can be presented as tree dimensional projections. It is shown in the Fig. 3 as projection for constant $a = 2.5$ and for constant $b = 4$. Autocorrelation function for $x(t, \alpha)$ in L_f^2 is found using symmetry property and is equal

$$R_{xx}(\tau, a) = \int_{-\infty}^{\infty} e^{-a|t|} e^{-a|t-\tau|} dt = \frac{1}{a} e^{-a|\tau|}[1 + a \mid \tau \mid] \tag{45}$$

Autocorrelation in L^2_μ is equal

$$R_{xx}(\tau) = \frac{\int_1^4 \frac{1}{a} e^{-a|\tau|}[1 + a \mid \tau \mid] \mu_\alpha(a)\, da}{\int_1^4 \mu_\alpha(a)\, da} \tag{46}$$

The formula for the result is complicated. Fig. 4 shows autocerrelation functions.

4 Remarks and Conclusions

In the paper the same notation was used for correlation function in different functional spaces in order to simplify notations and due to the same general definition - scalar product. Conventional definitions were generalized using this definition. Presented approach allows apply fuzzy description, which is more and more popular in different scientific areas, and introduce it in signal theory. Some proofs of shown properties were published in [3], and probably will appear in [2].

References

1. Butkiewicz, B. S.: Fuzzy Approach to Fourier Transform. Signal Processing Symposium, (2005), June 3-6, 1–6, Wilga, Poland.
2. Butkiewicz, B. S.: An Approach to Theory of Fuzzy Signals, Basic Definitions. Paper sent to IEEE Transactions on Fuzzy Systems.
3. Butkiewicz, B. S.: Fuzzy Approach to Fourier Transform. Proceedings of SPIE, vol. 6159, pp. 1045-1050, 2006.
4. Dieudonné, J.: Foundations of Modern Analysis, Academic Press, 1960.
5. L.A. Zadeh, *The Concept of a Linguistic Variable and its Application to Approximate Reasoning*, Part 1, Information Sciences, v. 8, pp. 199-249, 1975.

A Method for Designing Flexible Neuro-fuzzy Systems

Krzysztof Cpalka[1,2]

[1] Czestochowa University of Technology, Poland,
Department of Computer Engineering
[2] Academy of Humanities and Economics, Poland,
Department of Artificial Intelligence
cpalka@kik.pcz.czest.pl

Abstract. In the paper we develop a new method for designing and reduction of flexible neuro-fuzzy systems. The method allows to reduce number of discretization points in the defuzzifier, number of rules, number of inputs, and number of antecedents. The performance of our approach is illustrated on a typical benchmark.

1 Introduction

In the last decade various neuro-fuzzy systems have been developed (see e.g. [3], [5], [7], [12], [13]). They are characterized by natural language description and learning abilities. Typical applications include identification, pattern classification, prediction and control. Most of neuro-fuzzy systems are based on the Mamdani type reasoning described by a t-norm, e.g. product or min, applied to connect antecedents and consequences in the individual rules. Another approach is based on the logical method, e.g. an S-implication (see, e.g. [4], [6]) used to connect antecedents and consequences in the rule base. Flexible neuro-fuzzy systems have been developed in [1], [2], [8]-[10]. Such systems are characterized by various flexibility parameters incorporated into their construction. Moreover, they allow to combine the Mamdani type reasoning with the logical approach and to find a fuzzy reasoning (Mamdani or logical) in the process of learning. In this paper we continue to investigate flexible neuro-fuzzy systems and the goal is to develop a new method for their designing and complexity reduction.

In this paper we consider multi-input, single-output neuro-fuzzy system mapping $\mathbf{X} \rightarrow \mathbf{Y}$, where $\mathbf{X} \subset \mathbf{R}^n$ and $\mathbf{Y} \subset \mathbf{R}$. The fuzzy rule base of these systems consists of a collection of N fuzzy IF-THEN rules in the form

$$R^{(k)}: \quad \text{IF} \quad \mathbf{x} \quad \text{is} \quad \mathbf{A}^k \quad \text{THEN} \quad y \quad \text{is} \quad B^k, \tag{1}$$

where $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbf{X}$, $y \in \mathbf{Y}$, $A_1^k, A_2^k, \ldots, A_n^k$ are fuzzy sets characterized by membership functions $\mu_{A_i^k}(x_i)$, $\mathbf{A}^k = A_1^k \times A_2^k \times \ldots \times A_n^k$, and B^k are fuzzy sets characterized by membership functions $\mu_{B^k}(y)$, respectively, $k = 1, \ldots, N$.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 212–219, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Defuzzification in these systems is realised for example by COA (centre of area) method defined by the following formula

$$\bar{y} = \frac{\sum_{r=1}^{N} \bar{y}^r \cdot \mu_{B'}(\bar{y}^r)}{\sum_{r=1}^{N} \mu_{B'}(\bar{y}^r)} \tag{2}$$

where B' is the fuzzy set obtained from the linguistic model (1), using an appropriate fuzzy reasoning, and $\bar{y}^r$ denotes centres of the output membership functions $\mu_{B^r}(y)$, i.e. for $r = 1, \ldots, N$,

$$\mu_{B^r}(\bar{y}^r) = \max_{y \in \mathbf{Y}} \{\mu_{B^r}(y)\}. \tag{3}$$

2 New Flexible Neuro-fuzzy Systems

Neuro-fuzzy architectures developed so far in the literature are based on the formula (2) with the assumption that number of terms in both sums is equal to the number of rules N. In this paper we relax that assumption and replace formula (2) by

$$\bar{y} = \frac{\sum_{r=1}^{R} \bar{y}^r \cdot \mu_{B'}(\bar{y}^r)}{\sum_{r=1}^{R} \mu_{B'}(\bar{y}^r)}, \tag{4}$$

where $R \geq 1$. A great advantage of formula (4) over formula (2) is that an elimination of a single rule in (4) has no effect on number of discretization points.

For further investigations we choose flexible neuro-fuzzy systems of a logical type with an S-implication given by (for details see e.g. [8]-[10])

$$\bar{y} = \frac{\sum_{r=1}^{R} \bar{y}^r \cdot agr_r(\bar{\mathbf{x}}, \bar{y}^r)}{\sum_{r=1}^{R} agr_r(\bar{\mathbf{x}}, \bar{y}^r)}, \tag{5}$$

where

$$agr_r(\bar{\mathbf{x}}, \bar{y}^r) = \begin{pmatrix} (1-\alpha^{agr}) \operatorname{avg}(I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \ldots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r)) + \\ +\alpha^{agr} T^* \left\{ \begin{matrix} I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \ldots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r); \\ w_1^{agr}, \ldots, w_N^{agr} \end{matrix} \right\} \end{pmatrix}, \tag{6}$$

$$I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r) = \begin{pmatrix} \left(1-\alpha_k^I\right) \operatorname{avg}(1-\tau_k(\bar{\mathbf{x}}), \mu_{B^k}(\bar{y}^r)) + \\ +\alpha_k^I S\{1-\tau_k(\bar{\mathbf{x}}), \mu_{B^k}(\bar{y}^r)\} \end{pmatrix}, \tag{7}$$

and

$$\tau_k(\bar{\mathbf{x}}) = \begin{pmatrix} (1-\alpha_k^{\tau})\,\mathrm{avg}\left(\mu_{A_1^k}(\bar{x}_1),\ldots,\mu_{A_n^k}(\bar{x}_n)\right) + \\ +\alpha_k^{\tau} T^* \left\{ \begin{matrix} \mu_{A_1^k}(\bar{x}_1),\ldots,\mu_{A_n^k}(\bar{x}_n); \\ w_{1,k}^{\tau},\ldots,w_{n,k}^{\tau} \end{matrix} \right\} \end{pmatrix}. \tag{8}$$

In formulas (6) and (8) we apply the weighted t-norm [8] defined by

$$T^*\{a_1,\ldots,a_n;w_1,\ldots,w_n\} = \underset{i=1}{\overset{n}{T}} \{1-w_i(1-a_i)\} \tag{9}$$

to connect the antecedents in each rule, $k=1,\ldots,N$, and to aggregate the individual rules in the logical models, respectively. Observe that if $w_1=0$ then $T^*\{a_1,a_2;0,w_2\} = T\{1,1-w_2(1-a_2)\} = 1-w_2(1-a_2)$. Therefore, antecedent a_1 is discarded since its certainty is equal to 0.

We incorporate flexibility parameters [10] into construction of new neuro-fuzzy systems. These parameters have the following interpretation:

- weights in antecedents of the rules $w_{i,k}^{\tau}\in[0,1]$, $i=1,\ldots,n$, $k=1,\ldots,N$,
- weights in aggregation of the rules $w_k^{agr}\in[0,1]$, $k=1,\ldots,N$,
- soft strength of firing controlled by parameter α_k^{τ}, $k=1,\ldots,N$,
- soft implication controlled by parameter α_k^{I}, $k=1,\ldots,N$,
- soft aggregation of rules controlled by parameter α^{agr}.

The general architecture (see e.g. [8]) of the above system is depicted in Fig. 1. It is easily seen that system (4) contains $N(3n+5)+R+1$ parameters to be determined in the process of learning.

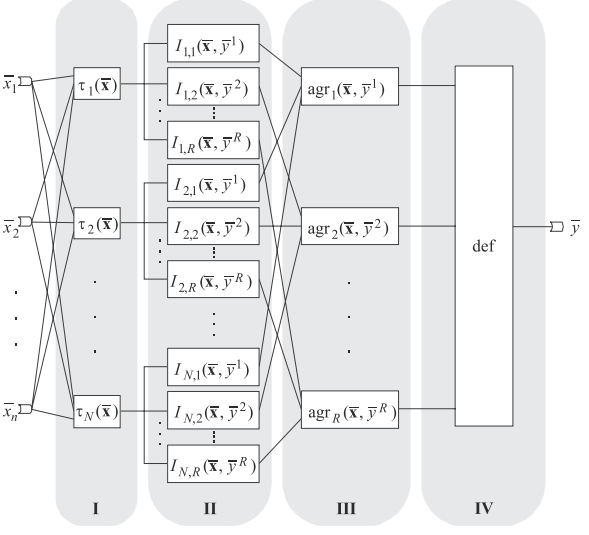


Fig. 1. The scheme of neuro-fuzzy system

3 Algorithm of Reduction of Neuro-fuzzy Systems

In this section we present an algorithm of reduction of neuro-fuzzy systems. The flowchart of the algorithm is depicted in Fig. 2.

It is assumed that the system under consideration works satisfactory after the learning process is finished. We apply the reduction procedure to that system in the following way:

- The initial system (structure and parameters) is saved before the reduction process starts.
- One parameter (discretization point in the defuzzifier, $r = 1, \ldots, R$, the whole rule, $k = 1, \ldots, N$, input, $i = 1, \ldots, n$, or antecedent, $i = 1, \ldots, n$, $k = 1, \ldots, N$) of the system is deleted.
- Learning by a single epoch is performed. Remaining parameters take over activity of the eliminated parameter.
- Performance of a reduced system is determined. If it is acceptable the reduced system is saved. Otherwise, the initial system is restored.

4 Simulation Results

The neuro-fuzzy system is simulated on Glass Identification problem [11]. The Glass Identification problem contains 214 instances and each instance is described by nine attributes (RI: refractive index, Na: sodium, Mg: magnesium, Al: aluminium, Si: silicon, K: potassium, Ca: calcium, Ba: barium, Fe: iron). All attributes are continuous. There are two classes: the window glass and the non-window glass. In our experiments, all sets are divided into a learning sequence (171 sets) and a testing sequence (43 sets). The study of the classification of the types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence if it is correctly identified.

The experimental results for the Glass Identification problem are depicted in tables 1, 2, 3, 4, 5 and figures 3, 4. In Table 1 we show the percentage of mistakes in the learning and testing sequences before and after reduction, e.g. for $N = 2$ and $R = 3$ we have 3.51%/2.34% for the learning sequence before and after reduction and 2.33%/2.33% for the testing sequence before and after reduction. In Table 2 we present number of inputs, number of rules, number of discretization points in the defuzzifier, number of antecedents and number of parameters before and after reduction. In Table 3 we show degree of learning time reduction [%] for a reduced system. In Table 4 we present reduced inputs, antecedents, rules and discretization points in the defuzzifier. In Table 5 we depict percentage of neuro-fuzzy systems having a particular input (attribute) after the reduction process and percentage of inputs (attributes) corresponding to a particular neuro-fuzzy system after the reduction process. In Fig. 3a we show degree of parameter number reduction [%], in Fig. 3b degree of learning time reduction [%], in Fig. 4a percentage of neuro-fuzzy systems having a particular input (attribute) after the reduction process, in Fig. 4b percentage of inputs (attributes) corresponding to a particular neuro-fuzzy system after the reduction process.

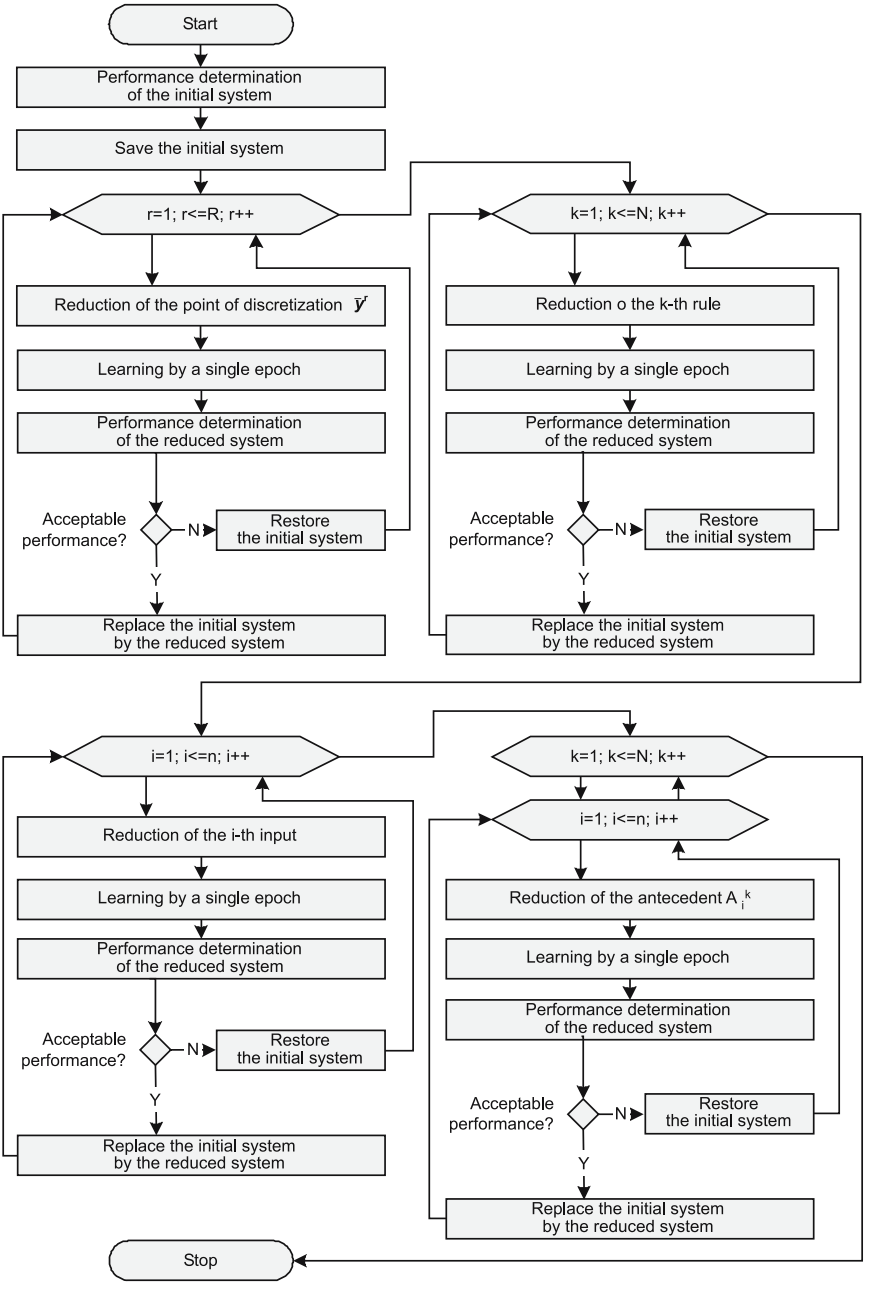


Fig. 2. The algorithm for reduction of flexible neuro-fuzzy systems

Table 1. Simulation results

Glass identification problem				
R	N			
	1	2	3	4
2	6.43%/5.85% 9.30%/9.30%	2.34%/2.34% 2.33%/2.33%	2.92%/2.92% 0.00%/0.00%	2.34%/2.34% 0.00%/0.00%
3	6.43%/5.85% 9.30%/9.30%	3.51%/2.34% 2.33%/2.33%	2.34%/2.34% 0.00%/0.00%	2.34%/2.34% 0.00%/0.00%
4	6.43%/6.43% 9.30%/9.30%	2.34%/2.34% 2.33%/2.33%	2.34%/2.34% 0.00%/0.00%	2.34%/2.34% 0.00%/0.00%

Table 2. Simulation results

Glass identification problem				
R	N			
	1	2	3	4
2	9/1/2/9/35 2/1/2/2/14	9/2/2/18/67 5/2/2/8/37	9/3/2/27/99 5/3/2/13/57	9/4/2/36/131 6/3/2/16/66
3	9/1/3/9/36 2/1/2/2/14	9/2/3/18/68 4/2/2/4/25	9/3/3/27/100 6/3/3/10/49	9/4/3/36/132 6/4/3/13/63
4	9/1/4/9/37 2/1/2/2/14	9/2/4/18/69 6/2/4/12/51	9/3/4/27/101 7/3/4/17/71	9/4/4/36/133 4/3/3/11/52

Table 3. Simulation results

Glass identification problem				
R	N			
	1	2	3	4
2	61%	58%	57%	57%
3	65%	70%	51%	48%
4	87%	50%	52%	67%

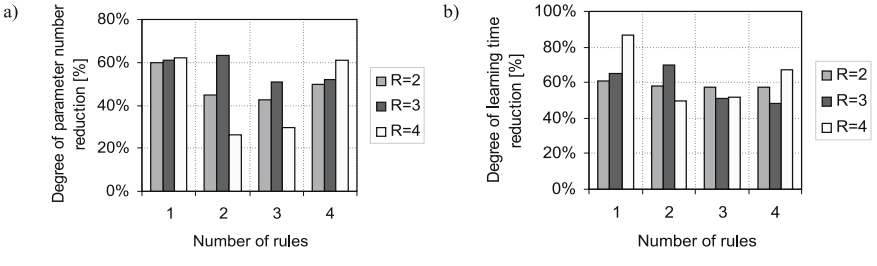


Fig. 3. Degree of a) parameter number reduction [%], b) learning time reduction [%]

Table 4. Simulation results

Glass identification problem				
R	N			
	1	2	3	4
2	$\bar{x}_1$, $\bar{x}_2$, $\bar{x}_4$, $\bar{x}_5$, $\bar{x}_6$, $\bar{x}_7$, $\bar{x}_9$	$\bar{x}_1$, $\bar{x}_2$, $\bar{x}_4$, $\bar{x}_5$, A_3^1, A_6^2	$\bar{x}_1$, $\bar{x}_5$, $\bar{x}_6$, $\bar{x}_7$, A_2^1, A_3^2	$\bar{x}_2$, $\bar{x}_6$, $\bar{x}_7$, A_1^1, A_5^2, $rule_4$
3	$\bar{x}_1$, $\bar{x}_2$, $\bar{x}_4$, $\bar{x}_5$, $\bar{x}_6 4$, $\bar{x}_7$, $\bar{x}_9$, $\bar{y}^1$	$\bar{x}_1$, $\bar{x}_2$, $\bar{x}_4$, $\bar{x}_5$, $\bar{x}_6$, A_3^1, A_9^1, A_7^2, A_8^2, $\bar{y}^1$	$\bar{x}_2$, $\bar{x}_5$, $\bar{x}_8$, A_1^1, A_4^1, A_9^1, A_3^2, A_9^2, A_3^3, A_4^3, A_6^3	$\bar{x}_2$, $\bar{x}_5$, $\bar{x}_7$, A_1^1, A_4^1, A_9^1, A_1^2, A_3^2, A_4^2, A_6^2, A_9^2, A_3^3, A_8^3, A_1^4
4	$\bar{x}_1$, $\bar{x}_2$, $\bar{x}_4$, $\bar{x}_5$, $\bar{x}_6$, $\bar{x}_7$, $\bar{x}_9$, $\bar{y}^1$, $\bar{y}^2$	$\bar{x}_1$, $\bar{x}_5$, $\bar{x}_7$	$\bar{x}_2$, $\bar{x}_4$, A_1^1, A_5^1, A_9^1, A_1^2	$\bar{x}_1$, $\bar{x}_2$, $\bar{x}_4$, $\bar{x}_5$, $\bar{x}_6$, A_7^1, $rule_4$

Table 5. Simulation results

Glass identification problem													
N	1	1	1	2	2	2	3	3	3	4	4	4	
R	2	3	4	2	3	4	2	3	4	2	3	4	
$\bar{x}_1$								v	v	v	v		33%
$\bar{x}_2$						v	v						17%
$\bar{x}_3$	v	v	v	v	v	v	v	v	v	v	v	v	100%
$\bar{x}_4$						v	v	v		v	v		42%
$\bar{x}_5$									v	v			17%
$\bar{x}_6$				v		v		v	v		v		42%
$\bar{x}_7$				v	v			v	v			v	42%
$\bar{x}_8$	v	v	v	v	v	v	v		v	v	v	v	92%
$\bar{x}_9$				v	v	v	v	v	v	v	v	v	75%
	22%	22%	22%	56%	44%	67%	56%	67%	78%	67%	67%	44%	

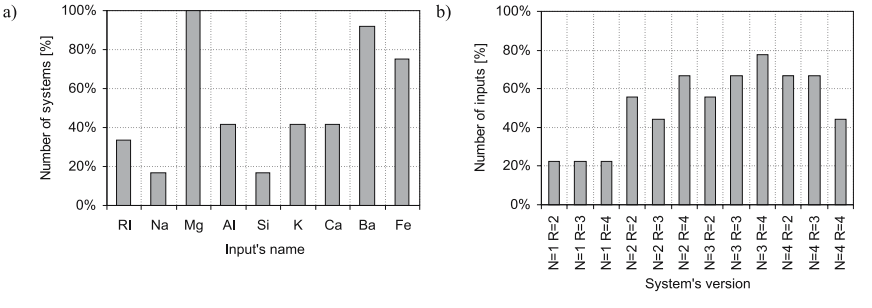


Fig. 4. Percentage of a) neuro-fuzzy systems having a particular input (attribute) after the reduction process, b) inputs (attributes) corresponding to a particular neuro-fuzzy system after the reduction process

5 Conclusions

In the paper we described a new method for designing and reduction of flexible neuro-fuzzy systems. From simulations it follows that the reduction process of neuro-fuzzy structures based on weighted triangular norms do not worsen the performance of these structures. The method allows to reduce number of discretization points in the defuzzifier, number of rules, number of inputs, and number of antecedents. It should be noted that our method allows to the decrease the learning time and to detect important features.

Acknowledgement

This work is supported by the Foundation for Polish Science (Professorial Grant 2005-2008), and the Polish State Committee for Scientific Research (Grant Nr T11C 04827).

References

1. K. Cpalka, L. Rutkowski, Neuro-Fuzzy Structures for Pattern Classification, WSEAS Trans. on Computers, Issue 7, vol.4, July 2005, pp. 697 688.
2. K. Cpalka, L. Rutkowski, Flexible Takagi-Sugeno Neuro-Fuzzy Structures for Nonlinear Approximation, WSEAS Trans. on Computers, Issue 9, vol.4, September 2005, pp. 1450-1458.
3. E. Czogala, J. Leski, Fuzzy and Neuro-Fuzzy Intelligent Systems, Physica Verlag, A Springer Verlag Company, Heidelberg, New York, 2000.
4. C. Fodor, On fuzzy implication, Fuzzy Sets and Systems, vol. 42, pp. 293 300, 1991.
5. M. Gorzalczany, Computational Intelligence Systems and Applications: Neuro-Fuzzy and Fuzzy Neural Synergisms. Springer Verlag 2002.
6. E. P. Klement, R. Mesiar, E. Pap, Triangular Norms, Kluwer Academic Publishers, Netherlands 2000.
7. D. Nauck and R. Kruse. Neuro-fuzzy systems for function approximation, Fuzzy Sets and Systems, vol. 101, pp. 261 271, 1999.
8. L. Rutkowski, Flexible Neuro-Fuzzy Systems. Kluwer Academic Publishers, 2004.
9. L. Rutkowski and K. Cpalka, Designing and learning of adjustable quasi triangular norms with applications to neuro-fuzzy systems, IEEE Trans. on Fuzzy Systems, vol. 13, Feb 2005, pp. 140 151.
10. L. Rutkowski, K. Cpalka, Flexible neuro-fuzzy systems, IEEE Transactions on Neural Networks, vol. 14, May 2003, pp. 554 574.
11. UCI respository of machine learning databases, Available online: http://ftp.ics.uci.edu/pub/machine learning databases/.
12. J. S. Wang and C. S. G. Lee. Self adaptive neuro-fuzzy inference systems for classification applications, IEEE Trans. on Fuzzy Systems, vol. 10, pp. 790 802, December 2002.
13. R. R. Yager, D. P. Filev, Essentials of fuzzy modelling and control. John Wiley & Sons, 1994.

Deterministic Annealing Integrated with ε-Insensitive Learning in Neuro-fuzzy Systems

Robert Czabański

Silesian University of Technology,
Institute of Electronics,
ul Akademicka 16, 44-101 Gliwice, Poland
robert.czabanski@polsl.pl

Abstract. In this paper a new method of parameters estimation for neuro-fuzzy system with parameterized consequents is presented. The novelty of the learning algorithm consists of an application of the deterministic annealing method integrated with ε-insensitive learning. This method allows to improve neuro-fuzzy modeling quality in the sense of an increase in generalization ability and outliers robustness. To demonstrate performance of the proposed procedure two numerical experiments concerning benchmark problems of prediction and identification are given.

1 Introduction

The basic problem while designing fuzzy systems is the determination of their rule base which consists of a set of fuzzy conditional statements. Because there is no standard method of expert knowledge acquisition in process of fuzzy if-then rules determination, automatic methods of rules generation are actively investigated. A set of fuzzy conditional statements may be obtained automatically from numerical data describing input/output system characteristics. A number of fuzzy rules extraction procedures use learning capabilities of artificial neural networks to solve this task [9].

In this work a new learning procedure for Artificial Neural Network Based on Fuzzy Inference System (ANNBFIS) [4] is presented. The main goal of the proposed changes is to allow the enhancement of learning abilities of the neuro-fuzzy system and the same quality of the extracted fuzzy if-then rules.

The ANNBFIS is a fuzzy system with parameterized consequents that generates inference results based on fuzzy if-then rules. Fuzzy sets of linguistic values in rule antecedents have Gaussian membership function and the linguistic connective "**and**" of multi-input rule predicates is represented by algebraic product t-norm. Consequently the firing strength of the i-th rule has the following form [4]:

$$\underset{i=1,2,\ldots,I}{\forall} \quad F^{(i)}\left(\boldsymbol{x}_0\right) = \prod_{j=1}^{t} A_j^{(i)}\left(x_{0j}\right) = \exp\left[-\frac{1}{2}\sum_{j=1}^{t}\left(\frac{x_{0j}-c_j^{(i)}}{s_j^{(i)}}\right)^2\right], \qquad (1)$$

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 220–229, 2006.
© Springer-Verlag Berlin Heidelberg 2006

where I is a number of if-then rules, $A_j^{(i)}$ is a j-th linguistic value of the fuzzy set in rule antecedent, x_{0j} is the element of the input vector $\boldsymbol{x}_0 = [x_{01}, x_{02}, \ldots, x_{0t}]^T$, $c_j^{(i)}$ and $s_j^{(i)}$ for $i = 1, 2, \ldots, I;\ j = 1, 2, \ldots, t$ are membership function parameters, center and dispersion, respectively. Consequents of ANNBFIS fuzzy rules have symmetric triangular membership functions. They are defined using two parameters: width of the triangle base $w^{(i)}$ and center of gravity location $y^{(i)}(\boldsymbol{x}_0)$ determined by linear combinations of fuzzy system inputs:

$$y^{(i)}(\boldsymbol{x}_0) = p_0^{(i)} + p_1^{(i)} x_{01} + \ldots + p_t^{(i)} x_{0t} = \boldsymbol{p}^{(i)T} \boldsymbol{x}_0', \tag{2}$$

where $\boldsymbol{x}_0' = [1, \boldsymbol{x}_0]^T$ is the extended input vector. The above dependency formulates so called parameterized (moving) consequent [4]. The neuro-fuzzy system with parameterized consequents allows both conjunctive and logical interpretations of fuzzy if-then rules [4]. We assume conjunctive interpretation using Larsen's product in the following considerations. It was proven [4] that the neuro-fuzzy system with parameterized consequents based on Larsen fuzzy relation gives equivalent inference results with inference obtained from Reichenbach fuzzy implication. Assuming additionally a normalized arithmetic mean as a aggregation operator and modified indexed center of gravity [4] as a defuzzifier, we can evaluate the final crisp output value of the ANNBFIS system from the following formula:

$$y_0 = \sum_{i=1}^{I} \frac{w^{(i)} F^{(i)}(\boldsymbol{x}_0)}{\sum_{i=1}^{I} w^{(i)} F^{(i)}(\boldsymbol{x}_0)} y^{(i)}(\boldsymbol{x}_0) = \sum_{i=1}^{I} G^{(i)}(\boldsymbol{x}_0)\ y^{(i)}(\boldsymbol{x}_0). \tag{3}$$

The fuzzy system with parameterized consequents can be treated as a radial basis function neural network [4]. Consequently, the unknown neuro-fuzzy system parameters can be estimated using learning algorithms of artificial neural networks. Several solutions of this problem have been introduced in literature [4], [7], [3]. In this work, a new hybrid learning procedure, which connects deterministic annealing method and ε -insensitive learning algorithm by solving a linear inequalities system is presented. For next considerations let us assume that we have N examples of input vectors $\boldsymbol{x}_0(n) \in \mathbb{R}^t$ and the same number of known output values $t_0(n) \in \mathbb{R}$ which formulate the training set.

2 Deterministic Annealing

Our goal is the extraction of the set of fuzzy if-then rules that represents the knowledge of the phenomenon under consideration. The extraction process consists of an estimation of membership function parameters of antecedents as well as consequents $\boldsymbol{\zeta} = \left\{c_j^{(i)}, s_j^{(i)}, p_j^{(i)}, w^{(i)}\right\}$, $\forall i = 1, 2, \ldots, I$, $\forall j = 1, 2, \ldots, t$. The number of rules I is also unknown. We assume that it is pre-set arbitrarily. The number of antecedents t is defined by the size of the input training vector directly. To increase ability to avoid many local minima that traps steepest descent

method used in original ANNBFIS learning algorithm, we employ the technique of deterministic annealing [12] adapted for the sake of learning the neuro-fuzzy system with parameterized consequents [3]. The equation (3) defines the neuro-fuzzy system as a mixture of experts (models). Its global output is expressed as a linear combination of I outputs $y^{(i)}(\boldsymbol{x}_0)$ of local models, each represented by a single fuzzy conditional statement. A randomness of the association between data and local models can be measured using the Shannon entropy:

$$S = -\sum_{n=1}^{N}\sum_{i=1}^{I} G^{(i)}(\boldsymbol{x}_0(n)) \log G^{(i)}(\boldsymbol{x}_0(n)). \tag{4}$$

In deterministic annealing method the objective is minimization of the cost function defined as a squared-error:

$$E = \sum_{n=1}^{N} \frac{1}{2} (t_0(n) - y_0(n))^2, \tag{5}$$

while simultaneously controlling the entropy level of a solution. The deterministic annealing optimization problem is formulated as a minimization procedure of the Lagrangian:

$$L = E - T\,S, \tag{6}$$

where T is the Lagrange multiplier [12].

A connection between the equation presented above and the annealing of solids is essential here. The quantity L can be identified as the Helmholtz free energy of physical system with "energy" E, entropy S and "temperature" T [12]. The procedure involves a series of iterations while the entropy level is reduced gradually. To allow achievement of the cost global optimum the simulated annealing method framework is used. The algorithm starts at a high level of pseudo-temperature T and tracks the solution for lowered values of T. The pseudo-temperature reduction procedure is determined by the annealing schedule function. We use the following decremental rule in the next considerations:

$$T \leftarrow q\,T, \tag{7}$$

where $q \in (0, 1)$ is a pre-set parameter.

At each level of temperature we minimize the Lagrangian iteratively using gradient descent method in L over the parameter space. The parameters of ANNBFIS system are given as:

$$\boldsymbol{\zeta}(k+1) = \boldsymbol{\zeta}(k) - \eta \left.\frac{\partial L}{\partial \boldsymbol{\zeta}}\right|_{\boldsymbol{\zeta}=\boldsymbol{\zeta}(k)}, \tag{8}$$

where k denotes the iteration index and η is the learning rate which can be further expressed using formula proposed by Jang [6]:

$$\eta = \frac{\eta_{\text{ini}}}{\sqrt{\sum_{i=1}^{n_i} \left(\frac{\partial L}{\partial \zeta_i}\right)^2_{\zeta_i=\zeta_i(k)}}}, \tag{9}$$

where η_{ini} denotes initial (constatnt) step size, n_i is a number of optimized parameters: for parameters of membership function of fuzzy sets in antecedents $n_i = 2It$, for parameters of linear function in consequents $n_i = I(t+1)$, and for triangle base widths $n_i = I$.

3 ε-Insensitive Learning

In original ANNBFIS learning method parameters of consequents $\boldsymbol{p}^{(i)}$ are estimated using the least square (LS) method [4]. It accelerates the learning convergence [4]. A novel, tolerant to imprecision method for estimating parameters of consequents – ε-insensitive learning – was presented in [7]. It improves the generalization ability of the neuro-fuzzy system in comparison to the LS algorithm. The ε symbol represents the limiting value of the imprecision tolerance. If the error value is less than ε, then the zero loss is obtained. Three different approaches to solve the ε-insensitive learning problem were proposed in [7] also. In this work we use ε-insensitive Learning by Solving a System of Linear Inequalities (εLSSLI) because of its lowest computational burden which is approximately 3-times grater in comparison to the zero-tolerant learning with LS [7]. The ε-insensitive learning can be formulated using the following equations system:

$$\begin{cases} \boldsymbol{p}^{(i)} = \left(\boldsymbol{X}_{0e}^{'T} \boldsymbol{D}_e \boldsymbol{X}_{0e}^{'} + \frac{\tau}{2} \widetilde{\boldsymbol{I}} \right)^{-1} \boldsymbol{X}_{0e}^{'T} \boldsymbol{D}_e \left(\boldsymbol{t}_{0e} + \boldsymbol{b} \right), \\ \boldsymbol{e} = \boldsymbol{X}_{0e}^{'} \boldsymbol{p}^{(i)} - \boldsymbol{t}_{0e} - \boldsymbol{b} = \boldsymbol{0}. \end{cases} \tag{10}$$

where $\boldsymbol{X}_{0e}^{'} = [\boldsymbol{x}_0^{'}(1), \boldsymbol{x}_0^{'}(2), \ldots, \boldsymbol{x}_0^{'}(N), -\boldsymbol{x}_0^{'}(1), -\boldsymbol{x}_0^{'}(2), \ldots, -\boldsymbol{x}_0^{'}(N)]^T$ is an extended input matrix, $\boldsymbol{t}_{0e} = [t_0(1) - \varepsilon, t_0(2) - \varepsilon, \ldots, t_0(N) - \varepsilon, -t_0(1) - \varepsilon, -t_0(2) - \varepsilon, \ldots, -t_0(N) - \varepsilon]^T$ is an extended output vector, $\widetilde{\boldsymbol{I}} = \text{diag}([0, \mathbb{1}_{t\times 1}^T])$, $\mathbb{1}_{t\times 1}$ is a $(t \times 1)$ dimensional vector with all entries equal to 1 and $\boldsymbol{D}_e$ denotes a diagonal weights matrix, $\boldsymbol{D}_e = \text{diag}(G^{(i)}(\boldsymbol{x}_0(1)) / |e_1|, G^{(i)}(\boldsymbol{x}_0(2)) / |e_2|, \ldots, G^{(i)}(\boldsymbol{x}_0(N)) / |e_N|, G^{(i)}(\boldsymbol{x}_0(1)) / |e_{N+1}|, \ldots, G^{(i)}(\boldsymbol{x}_0(N)) / |e_{2N}|)$ where e_i is the i-th component of the error vector $\boldsymbol{e}$. The regularization parameter $\tau \geq 0$ controls the trade-off between the model matching to the training data and the model generalization ability [7]. A larger τ results in an increase in the model generalization ability. The vector $\boldsymbol{b}$ is called margin vector [7], because its components determine the distance between a datum and the insensitivity region. From the first equation of (10), we can see that solution vector $\boldsymbol{p}^{(i)}$ depends on margin vector. If datum lies in the insensitivity region then the zero error can be obtained by increasing the corresponding distance. Otherwise, the error can be decreased only by decreasing the corresponding margin vector component. The only way to prevent margin vector $\boldsymbol{b}$ from converging to zero is to start with $\boldsymbol{b} > \boldsymbol{0}$ and not allow any of its components to decrease [7]. This problem can be solved using εLSSLI procedure [7] which is an extended version of Ho and Kashyap iterative algorithm [5]. In εLSSLI the margin vector components are modified by corresponding error vector components only if the change results in the margin vector components increase: $\boldsymbol{b}^{[k+1]} = \boldsymbol{b}^{[k]} + \rho\left(\boldsymbol{e}^{[k]} + \left|\boldsymbol{e}^{[k]}\right|\right)$, where

$\rho > 0$ is a parameter. The εLSSLI starts with the pre-set margin vector $\boldsymbol{b}^{[1]} > \mathbf{0}$ and stops if $\left\|\boldsymbol{b}^{[k+1]} - \boldsymbol{b}^{[k]}\right\| > \kappa$, where κ is a pre-set parameter or when the maximal number of iterations $k_{\varepsilon\max}$ is achieved.

4 A New Learning Algorithm

Integration of εLSSLI procedure with deterministic annealing method leads to a hybrid learning algorithm, where the parameters of fuzzy sets from antecedents and consequents of fuzzy if-then rules are adjusted separately. Antecedent parameters $c_j^{(i)}, s_j^{(i)}$, $i = 1, 2, \ldots I$, $j = 1, 2, \ldots, t$, as well as triangle base widths $w^{(i)}$, $i = 1, 2, \ldots, I$ of fuzzy sets in consequents are estimated by means of deterministic annealing method whereas parameters of linear equations from consequents $\boldsymbol{p}^{(i)^T}$, $i = 1, 2, \ldots, I$, are adjusted using ε-insensitive learning and then tuned using deterministic annealing procedure. For decreasing the computational burden of the learning procedure the deterministic annealing method with "freezing" phase (DAF) can be applied [3]. The "freezing" phase consists of the calculation of $\boldsymbol{p}^{(i)}$ using εLSSLI procedure after every decreasing step of pseudo-temperature value while keeping $c_j^{(i)}, s_j^{(i)}$ and $w^{(i)}$ constant. The hybrid learning can be summarized as follows:

1. Set parameters: initial solution $\boldsymbol{\zeta}$, initial pseudo-temperature $T_{\max}$, final pseudo-temperature $T_{\min}$ and annealing schedule function. Set $T = T_{\max}$.
2. Minimize the Lagrangian L using the steepest descent method (8).
3. Check the equilibrium $|\delta S| = \left|\frac{S^{[k-1]} - S^{[k]}}{S^{[k-1]}}\right| > \delta$ or iteration stop condition $k \leq k_{\max}$, where k denotes the iteration index, δ is pre-set parameter and $k_{\max}$ denotes the maximum number of iteration at given level of pseudo-temperature, if one of them is fulfilled go to Step 2.
4. Lower pseudo-temperature according to the annealing schedule.
5. Perform the "freezing" phase i.e. estimate parameters of linear equations from consequents for all rules by means of εLSSLI procedure.
6. If $T \geq T_{\min}$ go to Step 2.
7. Stop the algorithm.

Another problem is estimation of initial values of membership functions for antecedents. It can be solved by means of preliminary clustering of the input training data [4]. We use the fuzzy c-means (FCM) [1] method for this task. The center and dispersion parameters of Gaussian membership functions can be initialized using final FCM partition matrix [4]:

$$\underset{1\leq i\leq I}{\forall}\ \underset{1\leq j\leq t}{\forall} \quad c_j^{(i)} = \frac{\sum_{n=1}^{N} (u_{in})^m x_{0j}(n)}{\sum_{n=1}^{N} (u_{in})^m} \tag{11}$$

and

$$\underset{1\le i\le I}{\forall}\ \underset{1\le j\le t}{\forall}\left(s_j^{(i)}\right)^2=\frac{\sum\limits_{n=1}^{N}\left(u_{in}\right)^m\left(x_{0j}\left(n\right)-c_j^{(i)}\right)^2}{\sum\limits_{n=1}^{N}\left(u_{in}\right)^m}, \tag{12}$$

where u_{in} is the FCM partition matrix element and $m\in[1,\infty)$ is a weighted exponent.

5 Numerical Experiments

To validate the introduced hybrid method of neuro-fuzzy modeling two numerical experiments using benchmark databases were conducted. The first one concerns prediction of sunspots [14]. The data set consists of 280 samples $x(n)$ of sunspots activity measured within a one-year period from 1700 to 1979 A.D. The goal is prediction of a number of sunspots (output value) $y\left(n\right)=x(n)$ using past values combined in the embedded input vector $\left[x\left(n-1\right)\ x\left(n-2\right)\ldots x\left(k-12\right)\right]^T$. The training set consists of the first 100 input-output pairs of data and the testing set contains the remaining 168 pairs.

The learning process was carried out in two phases. In each of the number of if-then rules I was changed from 2 to 6. The generalization ability was determined on the basis of root mean square error (RMSE) values on the testing set. All experiments were conducted in the MATLAB environment. During the first phase of learning only the εLSSLI algorithm was used with triangle base widths set to 1 and initial values of antecedents parameters estimated using FCM clustering results obtained for $m=2$. The clustering was stopped if the maximum number (500) of iterations was achieved or when in sequential iterations the change of the criterion function was less than 10^{-5}. The partition process was repeated 25 times for different random initialization of the partition matrix and results characterized by the minimal value of Xie-Beni validity index [16] were chosen. We sought a set of parameters τ and ε for which the best generalization ability of neuro-fuzzy system was achieved. We set $\boldsymbol{b}^{[1]}=10^{-6}$, $\rho=0.98$, $\kappa=10^{-4}$ and $k_{\varepsilon\,\max}=1000$ and changed values of τ and ε from 0.01 to 0.1 with step 0.01. The quantities for which the lowest RMSE was achieved were chosen (see Table 1).

During the second phase the proposed hybrid learning algorithm (DAF + εLSSLI) was employed. Parameters of the εLSSLI method were set using results obtained during the first learning phase. For the deterministic annealing procedure the following parameters' values were applied: $\eta_{\text{ini}}=0.01$, $T_{\max}\in\left\{10^3,10^2,\ldots,10^{-3}\right\}$, $T_{\min}=10^{-5}T_{\max}$, $\lambda=0.95$, $\delta=10^{-5}$ and $k_{\max}=10$. The obtained results are tabulated in Table 1. The best results are marked. For comparison RMSE values obtained from deterministic annealing with "freezing" phase integrated with least square method (DAF + LS) and results got from original ANNBFIS learning procedure are shown also.

Clearly, in all the examples, the ε-insensitive learning based method demonstrates consistent improvement in generalization ability. However, it must be

Table 1. RMSE of the prediction

	εLSSLI			DAF + εLSSLI		DAF + LS		ANNBFIS
I	RMSE	ε	τ	RMSE	$T_{\min}$	RMSE	$T_{\min}$	RMSE
2	0.0838	0.09	0.10	**0.0728**	$\mathbf{10^{-2}}$	**0.0843**	$\mathbf{10^{-2}}$	**0.0882**
3	**0.0783**	**0.09**	**0.05**	0.0760	10^{0}	0.0847	10^{-2}	0.0921
4	0.0786	0.03	0.06	0.0765	10^{0}	0.0990	10^{2}	0.1112
5	0.0810	0.76	0.01	0.0776	10^{0}	0.1083	10^{0}	0.1122
6	0.0857	0.01	0.10	0.0763	10^{1}	0.1042	10^{-2}	0.1428

noted, that the computational burden of the εLSSLI is approximately 3 times greater comparing to the LS method and the deterministic annealing computational burden is approximately 2 times greater than the steepest descent algorithm used in the original learning procedure of ANNBFIS.

To make a precise comparison of results obtained using the proposed hybrid method with results reported in literature we used the most common partitioning of the sunspots data set with one training and two testing parts. The training set contained sunspots activity measures from years 1700 to 1920, the first testing set measures from 1921 to 1955 and the second from 1956 to 1979. The specification of the learning algorithm was defined the same but additionally, we changed the value of the initial step size η_{ini} in the range from 0.01 to 0.05 with step 0.01. The best prediction quality with the DAF+εLSSLI method was obtained for $I = 2$, $T_{\min} = 10^{-3}$ and $\eta_{\text{ini}} = 0.03$. The comparison results are tabulated in Table 2. It can be seen that DAF + εLSSLI approach improves the generalization ability over the reference methods.

Table 2. Comparison results of sunspots prediction

Author	RMSE_1	RMSE_2
Tong [13]	0.064	0.108
Nowland [10]	0.056	-
Weigend [14]	0.060	0.121
Waterhouse [15]	0.061	0.106
Rementeria [11]	0.057	0.120
DAF+εLSSLI	0.050	0.104

The proposed procedure was also tested for robustness of outliers. For this purpose we added one outlier to the training set: the minimal output sample $y(1)$ equal to 0 was set to doubled value of maximal output sample $2\,y(67)$ equal to 1.6150. Then, we performed second learning stage for $I = 2$ using parameters for which we obtained the best generalization ability without outliers. We obtained the following results $\text{RMSE}_{\text{DAF+}\varepsilon\text{LSSLI}} = 0.0900$, $\text{RMSE}_{\text{DAF+LS}} = 0.1210$, $\text{RMSE}_{\text{ANNBFIS}} = 0.1411$. We can see that the DAF + εLSSLI approach

improves the generalization ability for the sunspots prediction problem in the presence of outliers in the training set over the reference algorithms.

The second numerical experiment concerns the benchmark identification problem of a gas oven. It is based on data originating from G. E. P. Box and G. M. Jenkins work [2]. An input signal consists of measuring samples of methane flow $x(n)$ [ft/min]. Methane is delivered into the gas oven together with air to form a mixture of gases containing carbon dioxide. The samples of CO_2 percentage content form an output signal $y(n)$. The sampling period was 9 sec. The data set consisting of 290 pairs of input vector $[y(n-1)\dots y(n-4)x(n)\dots x(n-5)]^T$ and output value $y(n)$ was divided into two parts: training and testing. The training set consists of the first 100 pairs of data and the testing set contains the remaining 190 pairs.

Analogous to the previous example the whole learning process was split into two phases. The specification of the learning algorithms was defined the same. The results from the first learning phase (εLSSLI) are shown in Table 3.

During the second phase the proposed hybrid learning algorithm (DAF + εLSSLI) was employed. The obtained learning results together with RMSE values on testing data calculated for reference algorithms are tabulated in Table 3.

Table 3. RMSE of the identification

	εLSSLI			DAF + εLSSLI		DAF + LS		ANNBFIS
I	RMSE	ε	τ	RMSE	$T_{\min}$	RMSE	$T_{\min}$	RMSE
2	**0.3507**	**0.01**	**0.01**	**0.3441**	$\mathbf{10^{-1}}$	**0.3491**	$\mathbf{10^{1}}$	**0.3608**
3	0.3656	0.09	0.01	0.3547	10^{1}	0.3669	10^{2}	0.3786
4	0.3936	0.02	0.02	0.3560	10^{3}	0.4284	10^{2}	0.4299
5	0.4000	0.14	0.05	0.3893	10^{3}	0.4282	10^{1}	0.4433
6	0.5094	0.59	0.02	0.3590	10^{-1}	0.5291	10^{3}	0.5390

The results of the second experiment also confirm that ε-insensitive learning leads to better generalization ability in comparison to zero-tolerant learning. The identification error for testing data increases with increase of the number of fuzzy if-then rules. It is due to the overfitting effect of the training set. However, the generalization ability decrease of ε-insensitive based methods is slower in comparison to zero-tolerant learning.

We compared performance of the proposed hybrid algorithm with six different methods using ε-insensitive learning which were presented in [8]. The specification of the DAF+εLSSLI procedure remained the same but additionally we changed the value of the initial step size η_{ini} in the range from 0.01 to 0.05 with step 0.01. The best prediction quality of the proposed hybrid procedure was obtained for $I = 3$, $T_{\min} = 10^2$ and $\eta_{\text{ini}} = 0.04$. The comparison results are tabulated in Table 4. We did not get the generalization ability improvement in comparison with two reference methods denoted as ε-LS-gradient$_g$ and ε-LS-ε-gradient$_g$.

Table 4. Comparison results of gas oven identification

Method	RMSE
ε-LS$_l$	0.3327
ε-LS$_g$	0.3313
ε-LS-gradient$_l$	0.3252
ε-LS-gradient$_g$	0.3162
ε-LS-ε-gradient$_l$	0.3284
ε-LS-ε-gradient$_g$	0.3222
DAF+εLSSLI	0.3231

To test the robustness of outliers for the identification problem we added one outlier to the training set: the minimal output sample $y\,(43)$ equal to 45.6 was set to doubled value of maximal output sample $2\,y\,(82)$ equal to 116.8. Then, analogously to the previous numerical example, we performed the second learning stage for two fuzzy if-then rules using parameters characterized by the best generalization ability without outliers. We obtained the following results $\mathrm{RMSE}_{\mathrm{DAF+\varepsilon LSSLI}} = 0.3649$, $\mathrm{RMSE}_{\mathrm{DAF+LS}} = 2.1698$, $\mathrm{RMSE}_{\mathrm{ANNBFIS}} = 1.4518$. It can be noticed that the DAF + εLSSLI approach significantly improves the generalization ability for the gas oven identification problem in the presence of outliers in the training set over the reference algorithms.

6 Conclusions

In this paper a new learning algorithm of ANNBFIS neuro-fuzzy system was presented. In the proposed procedure parameters of fuzzy sets from antecedents and consequents of fuzzy if-then rules are adjusted separately by means of deterministic annealing with "freezing" phase and $\varepsilon-$insensitive learning by solving a system of linear inequalities method respectively. Experimentation shows the usefulness of the method in extraction of fuzzy if-then rules for signal prediction and system identification problems. The obtained results indicate generalization ability and outliers robustness improvement in comparison with zero-tolerant learning. However, the performance enhancement is achieved through an increase in the computational burden of the learning procedure. Another problem is the necessity of arbitrary selection of learning parameters. Determination of automatic methods for their selection constitutes the cardinal direction of future investigations.

References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1981)
2. Box, G.E.P., Jenkins, G.M.: Time Series Analysis. Forecasting and Control. Holden-Day, San Francisco (1976)

3. Czabański, R.: Automatic Fuzzy If-Then Rules Extraction From Numerical Data. (In Polish) Ph.D. Thesis, Gliwice (2003)
4. Czogała, E., Łęski, J.: Fuzzy and Neuro-Fuzzy Intelligent Systems. Springer-Verlag, Heidelberg (2000)
5. Kashyap, R.L., Ho, Y.C.: An Algorithm for Linear Inequalities and its Applications. IEEE Trans. Elec. Comp. **14** (1965) 683–688
6. Jang, J-S.R.: ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Trans. on Systems Man and Cybernetics **23** (1993) 665-685
7. Łęski, J.: ε-Insensitive Learning Techniques for Approximate Reasoning Systems (Invited Paper). Int. J. of Computational Cognition **1** (2003) 21–77
8. Łęski, J., Czogała, T.: A Fuzzy System with ε-Insensitive Learning of Premises and Consequences of If-Then Rules. Int. J. Appl. Math. Comput. Sci. **15** (2005) 257–273
9. Mitra, S., Hayashi, Y.: Neuro-Fuzzy Rule Generation: Survey in Soft Computing Framework. IEEE Trans. Neural Networks **3** (2000) 748–768
10. Nowland, S., Hinton, G.: Simplifying Neural Networks by Soft Weight-Sharing. Neural Comput. **4** (1992) 473–493
11. Rementeria, S., Olabe, X.B.: Predicting Sunspots with a Self-Configuring Neural System. Proc. 8th Int. Conf. Information Processing Management Uncertainty Knowledge-Based Systems (2000) 1184–1190
12. Rose, K.: Deterministic Annealing for Clustering, Compression, Classification, Regression and Related Optimization Problems. Proceedings IEEE **11** (1998) 2210–2239
13. Tong, H., Lim, K.S.: Threshold Autoregression, Limit Cycle and Cyclical Data. Journal of The Royal Statistical Society B **42** (1980) 245–292
14. Weigend, A.S., Huberman, B.A., Rumelhart, D.E.: Predicting the Future: a Connectionist Approach. Int. J. Neural Syst. **1** (1990) 193–209
15. Waterhouse, S.R., Robinson, A.J.: Non-linear Prediction of Acoustic Vectors Using Hierarchical Mixtures of Experts. In Advances of Neural Information Processing Systems MIT Press Cambridge MA **7** (1995) 835–842
16. Xie, X.L., Beni, G.: A Validity Measure for Fuzzy Clustering. IEEE Trans. Pattern Analysis and Machine Intelligence **13** (1991) 841–847

Transformation Lemma on Analytical Modeling Via Takagi-Sugeno Fuzzy System and Its Applications

Jacek Kluska

Faculty of Electrical and Computer Engineering
Rzeszow University of Technology
35-959 Rzeszow, W. Pola 2, Poland
jacklu@prz.rzeszow.pl

Abstract. The work presents some applications of the transformation lemma on analytical modeling using the Takagi-Sugeno fuzzy rule-based system [7], which can be used for exact fuzzy modeling of some class of conventional systems. The examples are based on recent author's theorems [5], which provide necessary and sufficient conditions for transformation of fuzzy rules into the crisp model of the system and vice-versa. The fuzzy model represents the widely used Takagi-Sugeno fuzzy system with linear membership functions. The main attention is paid for usability of the results for control engineering community.

1 Introduction

There are many advantages in using fuzzy logic by modeling real systems, especially in the area of control systems design [2], [8]. We can define fuzzy rule-based system in the form of a set of linguistic rules. It is known that fuzzy control has the unique ability to successfully accomplish control tasks without knowing the mathematical model of the system, even if it is nonlinear and complex. However, the applications are currently being developed in an *ad hoc* manner requiring significant trial-and-error effort. The fuzzy systems developed are often treated as magic black boxes with little analytical understanding and explanation. There is a need for developing an analytical theory of fuzzy systems to support and accelerate the growth of the technology and eliminate the existing misunderstanding and controversy in the control engineering community [10], [11].

System modeling, with fuzzy rule-based systems, usually comes with two contradictory requirements in the obtained model: the capability to express the behavior of the real system in an understable way (interpretability), and capability to faithfully represent the real system (accuracy) [1]. Thus, the problem in the fuzzy modeling scientific community is: how to assure both interpretability and accuracy, or at least a good balance between them? The answer to this question is given in this contribution. We show some applications of theorems concerning equivalency between the multi-input-single-output Takagi-Sugeno fuzzy rule-based system, and some class of *multivariate polynomials*. Such approach

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 230–239, 2006.
© Springer-Verlag Berlin Heidelberg 2006

combines the advantages of classical control system theory and fuzzy sets theory. We can show that, using the transformation lemma on analytical modeling via Takagi-Sugeno fuzzy system, proved in [5], the important classes of dynamic plants, controllers, or control algorithms are themselves equivalent to a set of Takagi-Sugeno type fuzzy rules.

2 The MISO TS System

We will consider a multi-input-single-output system (MISO, for short), in which any *input variable* $z_k \in [-\alpha_k, \beta_k]$, where $\alpha_k + \beta_k \neq 0$ for $k = 1, 2, \ldots, n$. For every z_k we define two fuzzy sets with *linear membership functions* $N_k(z_k)$ and $P_k(z_k)$, where the fuzzy set P_k is a *complement* to N_k

$$N_k(z_k) = \frac{\beta_k - z_k}{\alpha_k + \beta_k}, \quad P_k(z_k) = \frac{\alpha_k + z_k}{\alpha_k + \beta_k}, \quad k = 1, 2, \ldots, n. \tag{1}$$

Such fuzzy sets are very simple and they preserve clear linguistic interpretation of the inputs of the modeled system. The MISO system with one output S and the inputs $z_1, \ldots, z_n$, will be called MISO Takagi-Sugeno fuzzy system (MISO TS for short). It is defined by 2^n fuzzy rules in the form

$$\text{If } z_1 \text{ is } A_{i_1} \text{ and ... and } z_n \text{ is } A_{i_n}, \text{ then } S = q_{i_1,\ldots,i_n}, \tag{2}$$

where $i_1, \ldots, i_n \in \{1, 2\}$, the fuzzy set A_{i_k} is either $A_{i_k} = N_{i_k}$, if $i_k = 1$, or $A_{i_k} = P_{i_k}$, if $i_k = 2$, for the input variable with the index $k = 1, \ldots, n$. Thus, the system of rules is assumed to be *complete* and *noncontradictory* one in the sense of the work [4]. The indices i_k we can uniquely identify with linguistic *labels*. If it will not be stated differently, we assume that the consequents $q_{i_1,\ldots,i_n}$ of the rules in (2) do not depend on the input variables.

For any n-tuple of indices $(i_1, \ldots, i_n) \in \{1, 2\}^n$, we define the *corresponding index* v, which is formally a function of the labels $(i_1, \ldots, i_n)$

$$v = 1 + \sum_{k=1}^{n} 2^{n-k} (i_k - 1), \qquad i_k \in \{1, 2\}, \qquad k = 1, \ldots, n. \tag{3}$$

Each index $v \in \{1, 2, 3, \ldots, 2^n\}$, corresponds exactly to only one antecedent and consequent of the fuzzy "If–then" rule. Because of the bijection (3), we will write $v \leftrightarrow (i_1, \ldots, i_n)$ for short. By such indexing, the rule-based system can be rewritten in a compact form

$$\text{If } P_v, \text{ then } S = q_v, \qquad v \leftrightarrow (i_1, \ldots, i_n). \tag{4}$$

The output of the MISO TS system $S = S(z_1, \ldots, z_n)$ is defined by the formula

$$S = \frac{\sum_{v=1}^{2^n} \top (A_{i_1}(z_1), \ldots, A_{i_n}(z_n))_v \times q_v}{\sum_{v=1}^{2^n} \top (A_{i_1}(z_1), \ldots, A_{i_n}(z_n))_v}, \qquad v \leftrightarrow (i_1, \ldots, i_n), \tag{5}$$

where $A_{i_k} \in \{N_k, P_k\}$, $(i_k \in \{1, 2\}, k = 1, \ldots, n)$ and $\top$ denotes the algebraic t–norm: $\top(x, y) = xy$ [9].

3 Transformation Lemma

Let us denote the *hyperrectangle* $D^n = \times_{k=1}^{n} [-\alpha_k, \beta_k]$ and its *vertices* $\Gamma^n = \times_{k=1}^{n} \{-\alpha_k, \beta_k\}$, where $\times$ denotes the Cartesian product. All vertices of D^n are the vectors $\boldsymbol{\gamma}_v = [(i_1 - 1)(\alpha_1 + \beta_1) - \alpha_1, \ldots, (i_n - 1)(\alpha_n + \beta_n) - \alpha_n]^T$, for $v = 1$, $v = 2$, ..., $v = 2^n$, where $v \leftrightarrow (i_1, \ldots, i_n)$, according to (3). The set of vertices $\Gamma^n = \{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{2^n}\}$, is ordered so that $\boldsymbol{\gamma}_1 \prec \boldsymbol{\gamma}_2 \prec \ldots \prec \boldsymbol{\gamma}_{2^n}$.

Lemma 1 (Basic transformation lemma for zero-order MISO TS system)**.** *Let us define for the variable* $\mathbf{z} = [z_1, \ldots, z_n]^T$*, the function* $f_0 : D^n \to \mathbb{R}$ *as follows*

$$f_0(\mathbf{z}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{z}), \tag{6}$$

where

$$\boldsymbol{\theta} = [\theta_0, \theta_1, \theta_2, \ldots, \theta_{1,2,\ldots,n}]^T \in \mathbb{R}^{2^n}, \tag{7}$$

$$\boldsymbol{\phi}(\mathbf{z}) = [1, \ldots, (z_1^{p_1} \cdots z_n^{p_n}), \ldots, (z_1 \cdots z_n)]^T, \tag{8}$$

with $p_k \in \{0, 1\}$*, and* $k = 1, \ldots, n$*. For every function of the type (6), there exists a zero-order MISO TS system, such that* $S(\mathbf{z}) = f_0(\mathbf{z})$ *for all points from the hyperrectangle* $\mathbf{z} \in D^n$*, and*

(i) *the inputs of the system are components of* $\mathbf{z} \in D^n$*, and its output is* S*,*
(ii) *each component of* $\mathbf{z}$ *corresponds to two the membership functions as in (1),*
(iii) *the MISO TS system is defined by* 2^n *fuzzy rules in the form of (2).*

One can find the consequents q_1*,* q_2*, ...,* q_{2^n} *of the fuzzy rules by solving the system of* 2^n *linear equations. The unique solution always exists.*

Proof of the lemma is given in [5]. For the given $\mathbf{z}$, the vector $\boldsymbol{\phi}(\mathbf{z})$ is called a *generator* that consists of *monomials*. One can prove that the following 2^n linear equations

$$\mathbf{q} = \mathbf{Z}^T \boldsymbol{\theta}, \qquad \mathbf{Z} = \left[\, \boldsymbol{\phi}(\boldsymbol{\gamma}_1) \ldots \boldsymbol{\phi}(\boldsymbol{\gamma}_{2^n}) \,\right]_{2^n \times 2^n}, \tag{9}$$

are satisfied, where vector $\mathbf{q} = [q_1, \ldots, q_{2^n}]^T$ is the vector of the consequents of the rules [5]. The matrix $\mathbf{Z}^T$ given by (9) is nonsingular one, if and only if, the universe of discourse for any input z_k is a nondegenerated interval, and therefore, it is called the *fundamental matrix*. The equation (9) can be equivalently written as follows

$$\mathbf{q}^T = \left[\boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{\gamma}_1), \ldots, \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{\gamma}_{2^n})\right] = [f_0(\boldsymbol{\gamma}_1), \ldots, f_0(\boldsymbol{\gamma}_{2^n})]. \tag{10}$$

The equations (9)-(10) formulate necessary and sufficient conditions, under which the system of fuzzy rules is equivalent to the function $f_0(\mathbf{z})$ given by (6). Lemma 1 says that we can always obtain an equivalent TS system for the given function (6).

4 Examples of MISO TS Fuzzy Systems

Example 1. Let us consider a TS system with $n = 2$ inputs, which models a logical system. In such case the inputs z_1, z_2 are from the unity interval $[0, 1]$, i.e. the boundaries are $\alpha_1 = \alpha_2 = 0$, and $\beta_1 = \beta_2 = 1$. The function f_0 is $f_0(\mathbf{z}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{z})$, where $\boldsymbol{\theta}$, the generator $\boldsymbol{\phi}(\mathbf{z})$, and the fundamental matrix $\mathbf{Z}$, are as follows

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_{1,2} \end{bmatrix}, \ \boldsymbol{\phi}(\mathbf{z}) = \begin{bmatrix} 1 \\ z_1 \\ z_2 \\ z_1 z_2 \end{bmatrix}, \ \mathbf{Z}^T = \begin{bmatrix} 1 & -\alpha_1 & -\alpha_2 & \alpha_1\alpha_2 \\ 1 & -\alpha_1 & \beta_2 & -\alpha_1\beta_2 \\ 1 & \beta_1 & -\alpha_2 & -\alpha_2\beta_1 \\ 1 & \beta_1 & \beta_2 & \beta_1\beta_2 \end{bmatrix}.$$

There are four fuzzy rules:

R_1 : If z_1 is N_1 and z_2 is N_2, then S is q_1,
R_2 : If z_1 is N_1 and z_2 is P_2, then S is q_2,
R_3 : If z_1 is P_1 and z_2 is N_2, then S is q_3,
R_4 : If z_1 is P_1 and z_2 is P_2 , then S is q_4.

From Lemma 1 we obtain $\boldsymbol{\theta} = \left(\mathbf{Z}^T\right)^{-1} [q_1, q_2, q_3, q_4]^T$ and the output S of the system $S(\mathbf{z}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{z})$. We define a strong negation as $n(x) = 1 - x$ for $x \in [0, 1]$. Taking into account boundary conditions we obtain that, for a generalized AND operation (t-norm), the consequents of the rules have to be $q_1 = 0$, $q_2 = 0$, $q_3 = 0$, and $q_4 = 1$. We immediately obtain $\theta_0 = \theta_1 = \theta_2 = 0$ and $\theta_{1,2} = 1$. Thus, the function expression is $S = z_1 z_2$ - a probabilistic t-norm. Analogously, we obtain analytical expressions for co-t-norm function: "$z_1 + z_2 - z_1 z_2$" and the other functions: implication, equivalency, etc. (see Table 1). Observe that, interpretation of all fuzzy rules and logical functions, is obvious.

Table 1. Multivalued-logic functions of two variables as the TS fuzzy system with $n = 2$ inputs for Example 1

t-norm	co-t-norm	implication	equivalency
$q_1 = 0$	$q_1 = 0$	$q_1 = 1$	$q_1 = 1$
$q_2 = 0$	$q_2 = 1$	$q_2 = 1$	$q_2 = 0$
$q_3 = 0$	$q_3 = 1$	$q_3 = 0$	$q_3 = 0$
$q_4 = 1$	$q_4 = 1$	$q_4 = 1$	$q_4 = 1$
$z_1 z_2$	$z_1 + z_2 - z_1 z_2$	$1 - z_1 + z_1 z_2$	$1 - z_1 - z_2 + 2 z_1 z_2$

Example 2. The discrete-time NARX model (Nonlinear AutoRegressive with the eXtra input) is given by the following equation (see [12])

$$\begin{aligned} y(k+1) = {} & \theta_0 + \theta_1 y(k) + \theta_2 y(k-1) + \theta_3 u(k) \\ & + \theta_{1,2} y(k)\, y(k-1) + \theta_{1,3} y(k)\, u(k) + \theta_{2,3} y(k-1)\, u(k) \\ & + \theta_{1,2,3} y(k)\, y(k-1)\, u(k), \end{aligned}$$

in which $y(k)$, $y(k-1) \in [-a, a]$ and $u(k) \in [-b, b]$. Assume that:
- The inputs of the TS system are $z_1 = y(k)$, $z_2 = y(k-1)$ and $z_3 = u(k)$,
- The system's output is $S = y(k+1)$,
- The vector of coefficients $\boldsymbol{\theta}^T = [\theta_0, \theta_1, \theta_2, \theta_3, \theta_{1,2}, \theta_{1,3}, \theta_{2,3}, \theta_{1,2,3}]$.

For the three input TS system, the generator is

$$\boldsymbol{\phi}^T(\mathbf{z}) = \left[1,\, z_1,\, z_2,\, z_3,\, z_1 z_2,\, z_1 z_3,\, z_2 z_3,\, z_1 z_2 z_3\right]$$

and the fundamental matrix

$$\mathbf{Z} = \begin{bmatrix} 1 & -\alpha_1 & -\alpha_2 & -\alpha_3 & \alpha_1\alpha_2 & \alpha_1\alpha_3 & \alpha_2\alpha_3 & -\alpha_1\alpha_2\alpha_3 \\ 1 & -\alpha_1 & -\alpha_2 & \beta_3 & \alpha_1\alpha_2 & -\alpha_1\beta_3 & -\alpha_2\beta_3 & \alpha_1\alpha_2\beta_3 \\ 1 & -\alpha_1 & \beta_2 & -\alpha_3 & -\alpha_1\beta_2 & \alpha_1\alpha_3 & -\alpha_3\beta_2 & \alpha_1\alpha_3\beta_2 \\ 1 & -\alpha_1 & \beta_2 & \beta_3 & -\alpha_1\beta_2 & -\alpha_1\beta_3 & \beta_2\beta_3 & -\alpha_1\beta_2\beta_3 \\ 1 & \beta_1 & -\alpha_2 & -\alpha_3 & -\alpha_2\beta_1 & -\beta_1\alpha_3 & \alpha_2\alpha_3 & \alpha_2\beta_1\alpha_3 \\ 1 & \beta_1 & -\alpha_2 & \beta_3 & -\alpha_2\beta_1 & \beta_1\beta_3 & -\alpha_2\beta_3 & -\alpha_2\beta_1\beta_3 \\ 1 & \beta_1 & \beta_2 & -\alpha_3 & \beta_1\beta_2 & -\beta_1\alpha_3 & -\alpha_3\beta_2 & -\beta_1\alpha_3\beta_2 \\ 1 & \beta_1 & \beta_2 & \beta_3 & \beta_1\beta_2 & \beta_1\beta_3 & \beta_2\beta_3 & \beta_1\beta_2\beta_3 \end{bmatrix}^T$$

Table 2. Look-up-table for the TS fuzzy system from Example 2

$z_1, z_2 \;\downarrow$	$z_3 \rightarrow$ N_3	P_3
N_1, N_2	q_1	q_2
N_1, P_2	q_3	q_4
P_1, P_2	q_7	q_8
P_1, N_2	q_5	q_6

By assumptions: $\alpha_1 = a$, $\beta_1 = a$, $\alpha_2 = a$, $\beta_2 = a$, $\alpha_3 = b$ and $\beta_3 = b$, we obtain the coefficients of the vector $\boldsymbol{\theta}$ for the given collection of conclusions q_j of the rules, which are given in Table 2, by computing $\boldsymbol{\theta} = \left(\mathbf{Z}^T\right)^{-1} [q_1, q_2, \ldots, q_8]^T$, and finally, the output of the system $S(\mathbf{z}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{z})$. The NARX model becomes the linear ARX one (without offset), iff the conclusions of the rules are as follows

$$\mathbf{q} = \mathbf{Z}^T \begin{bmatrix} 0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -a\theta_1 - a\theta_2 - b\theta_3 \\ -a\theta_1 - a\theta_2 + b\theta_3 \\ -a\theta_1 + a\theta_2 - b\theta_3 \\ -a\theta_1 + a\theta_2 + b\theta_3 \\ a\theta_1 - a\theta_2 - b\theta_3 \\ a\theta_1 - a\theta_2 + b\theta_3 \\ a\theta_1 + a\theta_2 - b\theta_3 \\ a\theta_1 + a\theta_2 + b\theta_3 \end{bmatrix}.$$

The same results were obtained in [12], but our approach is general, sistematic and seems to be very simple.

5 Modeling of n-Dimensional Dynamical One-Input System

Our goal is to model a nonlinear dynamical system $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t))$ by the fuzzy rules expresssed by (2), with linear membership functions, the consequents beeing linear combinations of the state vector $\mathbf{x}$, and a scalar control u.

Let us define for the TS system the following input vector $\mathbf{z}=\mathbf{x}=[x_1,\ldots,x_n]^T$, where $x_i = x_i(t)$ is the state variable of a modeled dynamical system. We define the output of the (dynamical) TS system by

$$S = \frac{dx_i}{dt} = \dot{x}_i, \qquad i = 1,\ldots,n,$$

where the index i is arbitrarily choosen. Finally, let us assume, that the jth consequent of the rule has the following form

$$q_j = \frac{dx_i}{dt} = \mathbf{a}_{i,(j)}^T \mathbf{x} + b_{i,(j)} u,$$

where $\mathbf{a}_{(j)} \in \mathbb{R}^{n\times 1}$, $b_{(j)} \in \mathbb{R}$ for $i = 1,\ldots,n$ and $j = 1,\ldots,2^n$. In other words, the TS system consits of 2^n fuzzy rules

$$\begin{aligned} R_j: \quad & \text{If } x_1 \text{ is } X_1 \text{ and } \ldots \text{ and } x_n \text{ is } X_n, \\ & \text{then } \dot{x}_1 = \mathbf{a}_{1,(j)}^T \mathbf{x} + b_{1,(j)} u \text{ and } \ldots \text{ and } \dot{x}_n = \mathbf{a}_{n,(j)}^T \mathbf{x} + b_{n,(j)} u, \end{aligned} \tag{11}$$

where $X_i \in \{N_i, P_i\}$ for $i = 1,\ldots,n$ and $j = 1,\ldots,2^n$. The above rule is equivalent to

$$R_j: \quad \text{If } x_1 \text{ is } X_1 \text{ and } \ldots \text{ and } x_n \text{ is } X_n, \text{ then } \dot{\mathbf{x}} = \begin{bmatrix} \mathbf{A}_{(j)} & \mathbf{b}_{(j)} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}, \tag{12}$$

and $\mathbf{A}_{(j)}$ is the local state matrix and $\mathbf{b}_{(j)}$ is the local control matrix (as a vector) from the jth region

$$\mathbf{A}_{(j)} = \begin{bmatrix} \mathbf{a}_{1,(j)}^T \\ \vdots \\ \mathbf{a}_{n,(j)}^T \end{bmatrix} = \begin{bmatrix} a_{11,(j)} & \cdots & a_{1n,(j)} \\ \vdots & \ddots & \vdots \\ a_{n1,(j)} & \cdots & a_{nn,(j)} \end{bmatrix}, \quad \mathbf{b}_{(j)} = \begin{bmatrix} b_{1,(j)} \\ \vdots \\ b_{n,(j)} \end{bmatrix}, \quad j = 1,\ldots,2^n.$$

Thus, in the region number j we obtain the consequent of the jth rule:

$$q_j = \dot{x}_i = \mathbf{a}_{i,(j)}^T \mathbf{x} + b_{i,(j)} u = \mathbf{M}_i \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix}, \qquad j = 1,\ldots,2^n.$$

For the i-th state variable $\dot{x}_i$ we define the following vector of all consequents of the rules

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{2^n} \end{bmatrix} = \begin{bmatrix} \dot{x}_i \\ \dot{x}_i \\ \vdots \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{i,(1)}^T & b_{i,(1)} \\ \mathbf{a}_{i,(2)}^T & b_{i,(2)} \\ \vdots & \vdots \\ \mathbf{a}_{i,(2^n)}^T & b_{i,(2^n)} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} = \mathbf{M}_i \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix},$$

for $i = 1, \ldots, n$. From the Lemma 1 the output of the defined TS system is $S = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x})$, and according to (9), the inferred model for the derivative of the state variable $\dot{x}_i$ is given by

$$\dot{x}_i = \mathbf{q}^T \mathbf{Z}^{-1} \boldsymbol{\phi}(\mathbf{x}) = \left[\mathbf{x}^T \; u\right] \mathbf{M}_i^T \mathbf{Z}^{-1} \boldsymbol{\phi}(\mathbf{x}) \tag{13}$$

where

$$\mathbf{M}_i^T = \begin{bmatrix} \mathbf{a}_{i,(1)}^T & b_{i,(1)} \\ \mathbf{a}_{i,(2)}^T & b_{i,(2)} \\ \vdots & \vdots \\ \mathbf{a}_{i,(2^n)}^T & b_{i,(2^n)} \end{bmatrix}^T = \begin{bmatrix} \mathbf{a}_{i,(1)} & \mathbf{a}_{i,(2)} & \cdots & \mathbf{a}_{i,(2^n)} \\ b_{i,(1)} & b_{i,(2)} & \cdots & b_{i,(2^n)} \end{bmatrix}. \tag{14}$$

Finally, for every state variable x_i we obtain

$$\dot{x}_i = \left[\mathbf{x}^T \; u\right] \mathbf{M}_i^T \mathbf{Z}^{-1} \boldsymbol{\phi}(\mathbf{x}), \qquad i = 1, \ldots, n.$$

Theorem 1. *Suppose we model a nonlinear dynamical system* $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u)$, *(*$\mathbf{x} \in \mathbb{R}^n$, $u \in \mathbb{R}$*), using linear membership functions (1) for every state variable* x_i, *(*$i = 1, \ldots, n$*), using* 2^n *fuzzy rules of the form (12).*
1) Such a TS system is equivalent to the following nonlinear dynamical system

$$\dot{\mathbf{x}} = \mathbf{W} \boldsymbol{\Phi}(\mathbf{x}),$$

where $\mathbf{W} \in \mathbb{R}^{n \times \dim \boldsymbol{\Phi}(\mathbf{x})}$ *and* $\boldsymbol{\Phi}(\mathbf{x})$ *is the "generator of the dynamical TS system". The elements of this generator are the same, as the components of the expanded form of the polynomial*

$$\left(\sum_{k=1}^{n} x_k + u\right) \prod_{k=1}^{n} (1 + x_k), \tag{15}$$

by neglecting coefficients of the sum in (15). The elements of the matrix $\mathbf{W}$ *depend on boundaries* α_i, β_i *(*$i = 1, \ldots, n$*), and the elements of* $\mathbf{A}_{(j)}$, *and* $b_{(j)}$,*(*$j = 1, \ldots, 2^n$*).*
2) The length of the generator $\boldsymbol{\Phi}(\mathbf{z})$ *is equal to*

$$\dim \boldsymbol{\Phi}(\mathbf{x}) = 2^{n-1}(n+4) - 1. \tag{16}$$

Proof. The first part of Theorem 1 is rather simple and will be omitted. We prove the relation (16). The length of $\boldsymbol{\Phi}(\mathbf{x})$, denoted by $\dim \boldsymbol{\Phi}(\mathbf{x})$, is for $n = 1$ equal to $\dim \boldsymbol{\Phi}(\mathbf{x})|_{\dim \mathbf{x}=1} = 4$. Next, one can show that the following recurrency

$$\dim \boldsymbol{\Phi}(\mathbf{x})|_{\dim \mathbf{x}=k+1} = 2 \dim \boldsymbol{\Phi}(\mathbf{x})|_{\dim \mathbf{x}=k} + 2^k + 1 \quad \text{by} \quad \dim \boldsymbol{\Phi}(\mathbf{x})|_{\dim \mathbf{x}=1} = 4$$

holds. This implies that for $\dim \mathbf{x} = n$ we have $\dim \boldsymbol{\Phi}(\mathbf{x}) = 2^{n-1}(n+4) - 1$. □

For example, for $n = 2$ the generator of the dynamical TS system is

$$\boldsymbol{\Phi}(\mathbf{x}) = \left[x_1, x_2, u, x_1 x_2, x_1^2, x_2^2, x_1 x_2^2, x_1^2 x_2, u x_1, u x_2, u x_1 x_2\right]^T. \tag{17}$$

The length of $\boldsymbol{\Phi}(\mathbf{x})$ grows faster than $\dim \boldsymbol{\phi}(\mathbf{x})$ – see Table 3.

Table 3. Length of the generators $\boldsymbol{\phi}$ and $\boldsymbol{\Phi}$ for the dynamical TS fuzzy system with n state variables

$n = \dim(\mathbf{x})$	$\dim \boldsymbol{\phi}(\mathbf{x})$	$\dim \boldsymbol{\Phi}(\mathbf{x})$
1	2	4
2	4	11
3	8	27
4	16	63
5	32	143
6	64	319
7	128	703
8	256	1535
9	512	3327
10	1024	7167

Example 3. Let us consider a one-input two-dimensional ($n = 2$) dynamical system

$$\begin{cases} \dot{x}_1 = f_1(x_1, x_2, u) \\ \dot{x}_2 = f_1(x_1, x_2, u) \end{cases}$$

which is modeled by 4 fuzzy rules as in (12):

$$R_{1:} \quad \text{If } x_1 \text{ is } N_1 \text{ and } x_2 \text{ is } N_2, \text{ then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11,(1)}, a_{12,(1)}, b_{1,(1)} \\ a_{21,(1)}, a_{22,(1)}, b_{2,(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix},$$

$$R_{2:} \quad \text{If } x_1 \text{ is } N_1 \text{ and } x_2 \text{ is } P_2, \text{ then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11,(2)}, a_{12,(2)}, b_{1,(2)} \\ a_{21,(2)}, a_{22,(2)}, b_{2,(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix},$$

$$R_{3:} \quad \text{If } x_1 \text{ is } P_1 \text{ and } x_2 \text{ is } N_2, \text{ then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11,(3)}, a_{12,(3)}, b_{1,(3)} \\ a_{21,(3)}, a_{22,(3)}, b_{2,(3)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix},$$

$$R_{4:} \quad \text{If } x_1 \text{ is } P_1 \text{ and } x_2 \text{ is } P_2, \text{ then } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11,(4)}, a_{12,(4)}, b_{1,(4)} \\ a_{21,(4)}, a_{22,(4)}, b_{2,(4)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix}.$$

According to equations (13)-(14) we obtain

$$\dot{x}_1 = \begin{bmatrix} x_1 & x_2 & u \end{bmatrix} \begin{bmatrix} a_{11,(1)} & a_{11,(2)} & a_{11,(3)} & a_{11,(4)} \\ a_{12,(1)} & a_{12,(2)} & a_{12,(3)} & a_{12,(4)} \\ b_{1,(1)} & b_{1,(2)} & b_{1,(3)} & b_{1,(4)} \end{bmatrix} \mathbf{Z}^{-1} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix},$$

$$\dot{x}_2 = \begin{bmatrix} x_1 & x_2 & u \end{bmatrix} \begin{bmatrix} a_{21,(1)} & a_{21,(2)} & a_{21,(3)} & a_{21,(4)} \\ a_{22,(1)} & a_{22,(2)} & a_{22,(3)} & a_{22,(4)} \\ b_{2,(1)} & b_{2,(2)} & b_{2,(3)} & b_{2,(4)} \end{bmatrix} \mathbf{Z}^{-1} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix},$$

where the inverse of the fundamental matrix is

$$\mathbf{Z}^{-1} = \frac{1}{V_2}\begin{bmatrix} \beta_2\beta_1 & -\beta_2 & -\beta_1 & 1 \\ \beta_1\alpha_2 & -\alpha_2 & \beta_1 & -1 \\ \beta_2\alpha_1 & \beta_2 & -\alpha_1 & -1 \\ \alpha_2\alpha_1 & \alpha_2 & \alpha_1 & 1 \end{bmatrix}, \qquad V_2 = (\alpha_1+\beta_1)(\alpha_2+\beta_2).$$

After computations we obtain

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,11} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,11} \end{bmatrix} \boldsymbol{\Phi}(\mathbf{x}),$$

where $\boldsymbol{\Phi}(\mathbf{x})$ is given by (17) and the elements of the matrix $\mathbf{W} \in \mathbb{R}^{2\times 11}$ are

$$\begin{aligned}
w_{i,1} &= \left(\beta_1\beta_2 a_{i1,(1)} + \alpha_2\beta_1 a_{i1,(2)} + \alpha_1\beta_2 a_{i1,(3)} + \alpha_1\alpha_2 a_{i1,(4)}\right)/V_2, \\
w_{i,2} &= \left(\beta_1\beta_2 a_{i2,(1)} + \alpha_2\beta_1 a_{i2,(2)} + \alpha_1\beta_2 a_{i2,(3)} + \alpha_1\alpha_2 a_{i2,(4)}\right)/V_2, \\
w_{i,3} &= \left(\beta_1\beta_2 b_{i,(1)} + \alpha_2\beta_1 b_{i,(2)} + \alpha_1\beta_2 b_{i,(3)} + \alpha_1\alpha_2 b_{i,(4)}\right)/V_2, \\
w_{i,4} &= \left(-\beta_1 a_{i1,(1)} + \beta_1 a_{i1,(2)} - \alpha_1 a_{i1,(3)} + \alpha_1 a_{i1,(4)}\right)/V_2 \\
&\quad + \left(-\beta_2 a_{i2,(1)} - \alpha_2 a_{i2,(2)} + \beta_2 a_{i2,(3)} + \alpha_2 a_{i2,(4)}\right)/V_2, \\
w_{i,5} &= \left(-\beta_2 a_{i1,(1)} - \alpha_2 a_{i1,(2)} + \beta_2 a_{i1,(3)} + \alpha_2 a_{i1,(4)}\right)/V_2, \\
w_{i,6} &= \left(-\beta_1 a_{i2,(1)} + \beta_1 a_{i2,(2)} - \alpha_1 a_{i2,(3)} + \alpha_1 a_{i2,(4)}\right)/V_2, \\
w_{i,7} &= \left(a_{i2,(1)} - a_{i2,(2)} - a_{i2,(3)} + a_{i2,(4)}\right)/V_2, \\
w_{i,8} &= \left(a_{i1,(1)} - a_{i1,(2)} - a_{i1,(3)} + a_{i1,(4)}\right)/V_2, \\
w_{i,9} &= \left(-\beta_2 b_{i,(1)} - \alpha_2 b_{i,(2)} + \beta_2 b_{i,(3)} + \alpha_2 b_{i,(4)}\right)/V_2, \\
w_{i,10} &= \left(-\beta_1 b_{i,(1)} + \beta_1 b_{i,(2)} - \alpha_1 b_{i,(3)} + \alpha_1 b_{i,(4)}\right)/V_2, \\
w_{i,11} &= \left(b_{i,(1)} - b_{i,(2)} - b_{i,(3)} + b_{i,(4)}\right)/V_2,
\end{aligned}$$

for $i = 1, 2$.

Theorem 2. *The TS dynamical system is a linear one if and only if all local matrices are the same. In this case the columns of the matrix* $\mathbf{W}$ *with numbers* $j = n+1, n+2, \ldots, 2^{n-1}(n+4)-1$ *are zero.*

Proof. (Sufficiency) From assumption $\mathbf{A}_{(1)} = ... = \mathbf{A}_{(2^n)} = \mathbf{A}$ and $\mathbf{b}_{(1)} = \mathbf{b}_{(2)} = ... = \mathbf{b}_{(2^n)} = \mathbf{b}$. Let us take a variable x_i for an arbitrarily given index i

$$\dot{x}_i = \begin{bmatrix} \mathbf{x}^T & u \end{bmatrix} \mathbf{M}_i^T \mathbf{Z}^{-1} \boldsymbol{\phi}(\mathbf{x}), \qquad \mathbf{M}_i^T = \begin{bmatrix} \mathbf{a}_i & \mathbf{a}_i & \cdots & \mathbf{a}_i \\ b_i & b_i & \cdots & b_i \end{bmatrix} \in \mathbb{R}^{(n+1)\times 2^n}.$$

Thus, we obtain

$$\mathbf{M}_i^T \mathbf{Z}^{-1} = \begin{bmatrix} \mathbf{a}_i & \mathbf{0} & \cdots & \mathbf{0} \\ b_i & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{(n+1)\times 2^n},$$

and finally

$$\dot{x}_i = \begin{bmatrix} \mathbf{x}^T\mathbf{a}_i + b_i u & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_1 x_2 \ldots x_n \end{bmatrix} = \mathbf{a}_i^T \mathbf{x} + b_i u.$$

The result is valid for any index $i \in \{1, \ldots, n\}$. This ends the proof of sufficiency. (Necessity) The necessary condition one can easily prove by construction a counterexample, for say $n = 2$. □

6 Conclusions

Using our approach we have a detailed insight into the TS fuzzy rule-based system with linear membership functions. The methodology is completely analytic and practically useful: we use 2 linear membership functions for every variable and the fuzzy rules have clear interpretation. Necessary and sufficient conditions, which have been formulated of how to transform some class of nonlinear systems into the fuzzy rules and vice-versa, deliver an exact relationships between fuzzy models and their classical counterparts. Transformation lemma can be easily applied for both some class of discrete- and continuous-time dynamical systems, or logical systems. Among others, using this lemma, we derived a TS model, which is equivalent to some nonlinear dynamical system of nth order with a scalar input. We used linear membership functions for state variables and fuzzy rules with consequents beeing velocities of the state variables, which are linear functions of state variables and the scalar input. We obtained an exact model of such nonlinear dynamical system. The proposed method can be easily extended to multi-input-multi-output systems.

References

1. Cassillas J., Cordón O., Herrera F., Magdalena L. (eds.): Interpretability Issues in Fuzzy Modeling. Studies in Fuzziness and Soft Computing. Vol. 128. Springer-Verlag, Berlin Heidelberg New York (2003)
2. Dubois D., Nguyen H.T., Prade H., and Sugeno M.: Fuzzy Systems: Modeling and Control, Introduction: The real contribution of fuzzy systems. In: Nguyen H.T., Sugeno M. (eds.): Norwell, MA: Kluwer, (1998) 1–17
3. Galichet S., Boukezzoula R., Foulloy L.: Explicit analytical formulation and exact inversion of decomposable fuzzy systems with singleton consequents, Fuzzy Sets and Systems, 146, (2004) 421–436
4. Kluska J.: Fuzzy Control (in Polish), ZNPRz, **12** (104), Rzeszów (1992)
5. Kluska J.: New results in Analytical Modeling Using Takagi-Sugeno Expert System. In: Issues in Intelligent Systems Paradigms, Hryniewicz O., Kacprzyk J., Koronacki J., Wierzchoń S.T. (Eds.), EXIT, (2005) 79-94
6. Kluska J.: Exact fuzzy modeling of conventional control systems. Proc. of the 12th Zittau Fuzzy Colloquium, Sep. 21-23, (2005) Wissenschaftliche Berichte, Heft 84/2005, No 2090-2131, 113-125
7. Takagi T., Sugeno M.: Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Systems, Man, Cybernet., vol. SMC-15, no. 1, (1985) 116–132
8. Wang L.X.: A Course in Fuzzy Systems and Control. Upper Saddle River, NJ: Prentice-Hall (1997)
9. Wygralak M.: Cardinality of Fuzzy Sets. Springer-Verlag, Berlin Heidelberg (2003)
10. Ying H.: Theory and application of a novel fuzzy PID controller using a simplified Takagi-Sugeno rule-scheme, Information Sciences, vol. 123, (2000) 281–293
11. Ying H.: Fuzzy control and modeling. Analytical foundations and applications. IEEE Press, New York (2000)
12. H. Ying: Structure and Stability Analysis of General Mamdani Fuzzy Dynamic Models, Int. J. of Intelligent Systems, vol. 20, (2005) 103–125

Combining Logical-Type Neuro-fuzzy Systems*

Marcin Korytkowski[1], Robert Nowicki[1,2],
Leszek Rutkowski[1,2], and Rafał Scherer[1,2]

[1] Department of Computer Engineering, Częstochowa University of Technology
Al. Armii Krajowej 36, 42-200 Częstochowa, Poland
marcink@kik.pcz.czest.pl, rnowicki@kik.pcz.czest.pl,
lrutko@kik.pcz.czest.pl, rafal@ieee.org
http://kik.pcz.pl
[2] Department of Artificial Intelligence, WSHE University in Łódź
ul. Rewolucji 1905 nr 64, Łódź, Poland
http://www.wshe.lodz.pl

Abstract. Boosting is one of the most popular methods of multiple classification. In the paper we propose a method for merging several logical-type neuro-fuzzy systems that come from boosting ensemble into one neuro-fuzzy system. Thanks to this we can use all rule-bases as one system.

1 Introduction

Classifiers can be combined to improve accuracy. A variety of known classifiers [3,8,10,13] and other learning systems [1, 4, 5, 14, 15, 16, 17, 19] can be combined in ensembles to improve accuracy. By combining intelligent learning systems, the model robustness and accuracy is nearly always improved, comparing to single-model solutions. Combined systems are developed under different names: blending, combining models, bundling, ensemble of classifiers, committee of experts. Classifiers can be combined at the level of features, data subsets, using different classifiers or different combiners, see Figure 1. Popular methods are bagging and boosting [3, 8, 9, 18] which are meta-algorithms for learning different classifiers. They assign weights to learning samples according to their performance on earlier classifiers in the ensemble. Thus subsystems are trained with different datasets. There are many variations of boosting with the most known - AdaBoost, where every learning vector has a weight assigned. According to the AdaBoost algorithm, consecutive classifiers should be learned with the greatest impact from the samples with the highest weight values. These weights have influence on the learning process by changing the learning rate. In the paper we develop a method for merging fuzzy rule-bases from several logical-type neuro-fuzzy systems constituting an ensemble trained by AdaBoost and the backpropagation algorithm. The method leads to the merged neuro-fuzzy

* This work was supported in part by the Foundation for Polish Science (Professorial Grant 2005-2008) and the Polish State Committee for Scientific Research (Grant Nr T11C 04827 and Grant T11A Nr 01427).

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 240–249, 2006.
© Springer-Verlag Berlin Heidelberg 2006

system characterized by interpretability and possibility to reduce its size. In the next section we describe the AdaBoost algorithm and in the Section 3 a logical-type neuro-fuzzy system constituting the boosting ensemble and a new method allowing merging fuzzy rule-bases. In Section 4 we verify the method on a well known data benchmark.

2 AdaBoost Agorithm

In this section we describe the AdaBoost algorithm which is the most popular boosting method [3][9][18]. Let us denote the l-th learning vector by $\mathbf{z}^l = [x_1^l, ..., x_n^l, y^l]$, $l = 1...m$ is the number of a vector, n is a size of input vector x^l, and y^l is the learning class label. Weights, assigned to learning vectors, have to fulfill the following conditions

$$\text{(i) } 0 < d^l < 1 \ , \tag{1}$$

$$\text{(ii) } \sum_{l=1}^{m} d^l = 1 \ . \tag{2}$$

The weight d^l is the information how well classifiers learned in consecutive steps of an algorithm for a given input vector x^l. Vector $\mathbf{d}$ for all input vectors is initialized according to the following equation

$$d_t^l = \frac{1}{m}, \quad \text{for } t = 0 \tag{3}$$

where t is the number of a boosting iteration (and a number of a classifier in the ensemble). Let $\{h_t(\mathbf{x}) : t = 1, ..., T\}$ denotes a set of hypotheses obtained in consecutive steps t of the algorithm being described. For simplicity we limit our problem to a binary classification (dichotomy), i.e. $y \in \{-1, 1\}$ or $h_t(\mathbf{x}) = \pm 1$. Similarly to learning vectors weights, we assign a weight c_t for every hypothesis, such that

$$\text{(i) } 0 < c_t \ , \tag{4}$$

$$\text{(ii)} \sum_{t} c_t = 1 \ . \tag{5}$$

Now in the AdaBoost algorithm we repeat steps 1-4 for $t = 1, \ldots, T$:
1. Create hypothesis h_t and train it with a data set with respect to a distribution d_t for input vectors.
2. Compute the classification error ε_t of a trained classifier h_t according to the formula

$$\varepsilon_t = \sum_{l=1}^{m} d_t^l(z^l) I(h_t(\mathbf{x}^l) \neq y^l) \ , \tag{6}$$

where I is the indicator function

$$I(a \neq b) = \begin{cases} 1 \text{ if } a \neq b \\ 0 \text{ if } a = b \end{cases} . \tag{7}$$

If $\varepsilon_t = 0$ or $\varepsilon_t \geq 0.5$, stop the algorithm.
3. Compute the value

$$\alpha_t = \log \frac{0.5(1 - \varepsilon_t)}{\varepsilon_t(1 - 0.5)} . \tag{8}$$

4. Modify weights for learning vectors according to the formula

$$d_{t+1}(\mathbf{z}^l) = \frac{d_t(\mathbf{z}^l) \exp\{-\alpha_t \mathbf{I}(h_t(\mathbf{x}_l) = y^l)\}}{N_t} , \tag{9}$$

where N_t is a constant such that

$$\sum_{l=1}^{m} d_{t+1}(\mathbf{z}^l) = 1 . \tag{10}$$

To compute the overall output of the ensemble of classifiers trained by AdaBoost algorithm the following formula is used

$$f(\mathbf{x}) = \sum_{t=1}^{T} c_t h_t(\mathbf{x}) , \tag{11}$$

where

$$c_t = \frac{\alpha_t}{\sum_{t=1}^{T} |\alpha_t|} \tag{12}$$

is classifier importance for a given training set. The AdaBoost algorithm is a meta-learning algorithm and does not determine the way of learning for classifiers in the ensemble.

3 Logical-Type Neuro-fuzzy Ensemble

In this section we propose a new method to build one neuro-fuzzy system, consisting of many logical-type neuro-fuzzy systems using boosting algorithm. The final output of an ensemble classifiers in AdaBoost, as was mention above, is

$$f(\overline{\mathbf{x}}) = \sum_{t=1}^{T} c_t h_t(\overline{\mathbf{x}}) . \tag{13}$$

So if classifiers were neuro-fuzzy systems and we used the Center of Area definition of defuzzifier then we could write the output of the t-th member of the committee in the discrete form

$$h_t = \frac{\sum\limits_{r=1}^{N} \overline{y}_t^r \cdot \mu_{B_t'}(\overline{y}_t^r)}{\sum\limits_{r=1}^{N} \mu_{B_t'}(\overline{y}_t^r)} . \tag{14}$$

Putting (14) into (13), the final output of a boosting algorithm $f(\overline{\mathbf{x}})$ is defined by

$$f(\overline{\mathbf{x}}) = \sum_{t=1}^{T} \left(c_t \frac{\sum\limits_{r=1}^{N} \overline{y}_t^r \cdot \mu_{B_t'}(\overline{y}_t^r)}{\sum\limits_{r=1}^{N} \mu_{B_t'}(\overline{y}_t^r)} \right) . \tag{15}$$

It is evident from eq. (15) that to make one neuro-fuzzy system we need to transform it in a such way that in the denominator there should be only the sum of antecedent rules for all members of the committee. One way to obtain that is assuming that

$$\forall t = 1, \ldots, T, \sum_{r=1}^{N} \mu_{B_t'}(\overline{y}_t^r) = 1 \, . \tag{16}$$

Neuro-fuzzy systems can be divided in two categories: Mamdami neuro-fuzzy systems with "engineering implications" and logical-type neuro-fuzzy systems with fuzzy implications [14]. First group consists of fuzzy systems where fuzzy relation is defined by a t-norm. Special transformation for merging this type of system ensemble is shown in [6]. In the second, logical-type approach, all rules aggregated using a t-norm to achieve one output fuzzy set $\mu_{B_t'}(\overline{y}_t^r)$ of the t-th fuzzy classifier

$$\mu_{B_t'}(\overline{y}_t^r) = \underset{k=1}{\overset{N}{T}} \left\{ \mu_{B_t^k}(\overline{y}_t^r) \right\} = \underset{k=1}{\overset{N}{T}} \left\{ I_{\text{fuzzy}}(\mu_{A_t^k}(\overline{\mathbf{x}}), \mu_{B_t^k}(\overline{y}_t^r)) \right\} \, , \tag{17}$$

where I_{fuzzy} denotes fuzzy implication [14]. In this context we can rewrite formula (14) in the form

$$h_t = \frac{\sum\limits_{r=1}^{N} \overline{y}_t^r \cdot \underset{k=1}{\overset{N}{T}} \left\{ I_{\text{fuzzy}}(\tau_t^k, \mu_{B_t^k}(\overline{y}_t^r)) \right\}}{\sum\limits_{r=1}^{N} \underset{k=1}{\overset{N}{T}} \left\{ I_{\text{fuzzy}}(\tau_t^k, \mu_{B_t^k}(\overline{y}_t^r)) \right\}} \, , \tag{18}$$

where $\tau_t^k = (\mu_{A_t^k}(\overline{\mathbf{x}}) = \underset{i=1}{\overset{n}{T}} \left\{ \mu_{A_{t,i}^k}(\overline{x}_i) \right\}$ is the level of the firing strength of the k-rule in the t-th subsystem. Our assumption given by (16) forces us to normalize values of denominators for all members of the committee. It can not be done in the same way as we have shown in [6] for Mamdani type systems. If we use a fuzzy implication e.g. binary or Lukasiewicz, then the final output of t-th system can be written in the form

$$h_t = \frac{\sum\limits_{k=1}^{N} \overline{y}_t^k \cdot \underset{\substack{j=1 \\ j \neq k}}{\overset{N}{T}} \left[1 - \mu_{A_t^j}(\overline{\mathbf{x}}) \right]}{\sum\limits_{k=1}^{N} \underset{\substack{j=1 \\ j \neq k}}{\overset{N}{T}} \left[1 - \mu_{A_t^j}(\overline{\mathbf{x}}) \right]} \, . \tag{19}$$

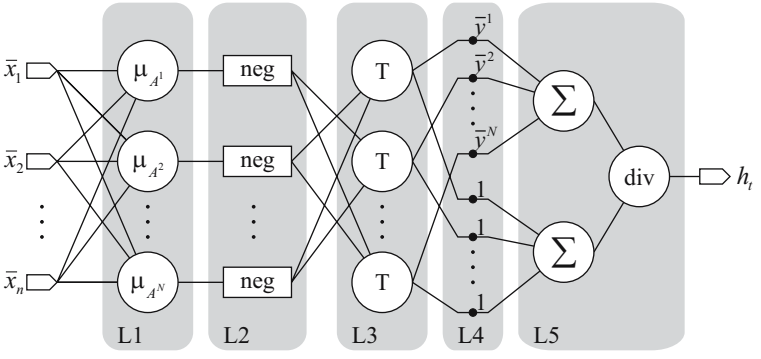


Fig. 1. The general form of a logical-type neuro-fuzzy system

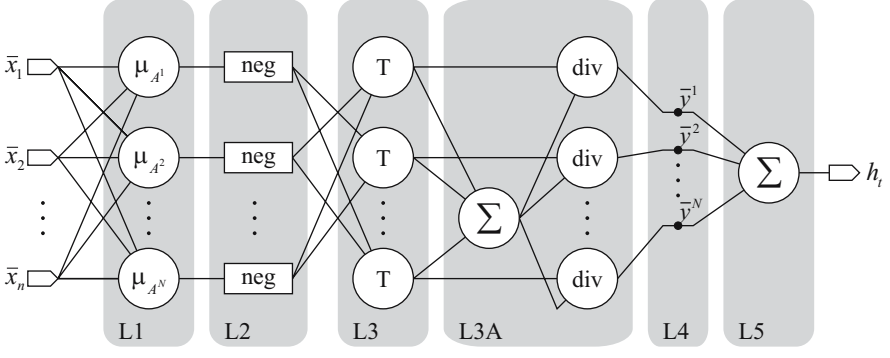


Fig. 2. The general form of a logical-type neuro-fuzzy system with additional layer L3A added for normalization purposes

Graphical representation of this kind of fuzzy systems is presented in Figure 1, where index t is ommited in the structure description for clarity of presentation. In our method we add an additional layer denoted L3A. Single member of the committee after our modification is shown in Figure 2. Assumption (16) in a neuro-fuzzy system can be also fulfilled in a different way than one showed in Fig. 2. In case of using selected triangular norms it can be fulfilled by modifying functions realized by blocks for fuzzy sets of rule antecedents. For aggregation realized by product t-norm, membership functions $\mu_{A^j}(\overline{\mathbf{x}})$ have to be replaced by $\zeta_{A^j}(\overline{\mathbf{x}})$ of the form

$$\zeta_{A^j}(\overline{\mathbf{x}}) = 1 - \frac{1 - \mu_{A^j}(\overline{\mathbf{x}})}{\sqrt[N-1]{\sum_{k=1}^{N} \prod_{\substack{r=1 \\ r \neq k}}^{N} [1 - \mu_{A^r}(\overline{\mathbf{x}})]}} . \tag{20}$$

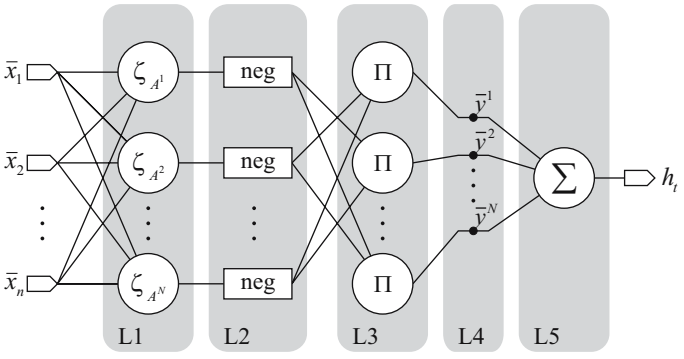


Fig. 3. Simplified structure with modified antecedent sets

It should be emphasized that function $\zeta_{A^j}(\overline{\mathbf{x}})$ can not be interpreted as a membership function. It is easy to prove the correctness of proposition (20) by proving the following formula

$$\sum_{k=1}^{N} \prod_{\substack{j=1 \\ j\neq k}}^{N} [1 - \zeta_{A^j}(\overline{\mathbf{x}})] = 1 \,. \tag{21}$$

Substituting (21) to (20) we obtain

$$\sum_{k=1}^{N} \prod_{\substack{j=1 \\ j\neq k}}^{N} \left[1 - 1 - \frac{1 - \mu_{A^j}(\overline{\mathbf{x}})}{\sqrt[N-1]{\sum_{q=1}^{N} \prod_{\substack{r=1 \\ r\neq q}}^{N} [1 - \mu_{A^r}(\overline{\mathbf{x}})]}} \right] = 1 \,, \tag{22}$$

and in a few steps we obtain

$$\frac{\sum_{k=1}^{N} \prod_{\substack{j=1 \\ j\neq k}}^{N} [1 - \mu_{A^j}(\overline{\mathbf{x}})]}{\sum_{q=1}^{N} \prod_{\substack{r=1 \\ r\neq q}}^{N} [1 - \mu_{A^r}(\overline{\mathbf{x}})]} = 1 \,. \tag{23}$$

In a very simple way we can make further simplifications by eliminating blocks marked "neg" that realize operation $y = 1 - x$. It can be done by merging blocks realizing the function $\zeta_{A^j}(\overline{\mathbf{x}})$ with "neg" operations, introducing blocks that realize function $\xi_{A^j}(\overline{\mathbf{x}})$ defined by

$$\xi_{A^j}(\overline{\mathbf{x}}) = 1 - \zeta_{A^j}(\overline{\mathbf{x}}) = \frac{1 - \mu_{A^j}(\overline{\mathbf{x}})}{\sqrt[N-1]{\sum_{k=1}^{N} \prod_{\substack{r=1 \\ r\neq k}}^{N} [1 - \mu_{A^r}(\overline{\mathbf{x}})]}} \,. \tag{24}$$

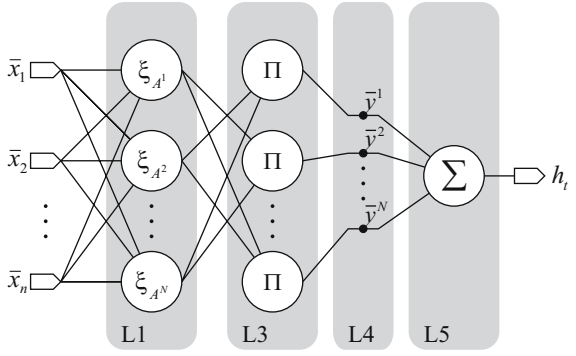


Fig. 4. The structure with blocks neg removed

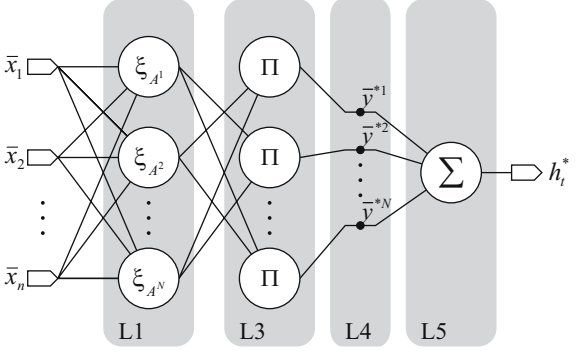


Fig. 5. Structure with parameters $\overline{y}_t^{*r} = c_t \overline{y}_t^r$

The neuro-fuzzy structure using blocks with $\xi_{A^j}(\overline{\mathbf{x}})$ is shown in Figure 4. In the system described by (15) there exists a weight c_t which determines importance of each subsystem. The number of parameters in the neuro-fuzzy system after learning can be reduced by replacing $\overline{y}_t^r$ with parameters $\overline{y}_t^{*r} = c_t \overline{y}_t^r$ and eliminating the weight c_t, see Fig. 5. The neuro-fuzzy system (15) is described then by

$$f(\overline{\mathbf{x}}) = \sum_{t=1}^{T} \left(\frac{\sum\limits_{r=1}^{N} \bar{y}_t^{*r} \cdot \mu_{B_t'}(\overline{y}_t^r)}{\sum\limits_{r=1}^{N} \mu_{B_t'}(\overline{y}_t^r)} \right), \tag{25}$$

and is shown in the Figure 6.

4 Numerical Simulations

Numerical simulations were carried out on the Pima Indians Diabetes problem [2]. The subsystems were trained by the backpropagation algorithm, starting

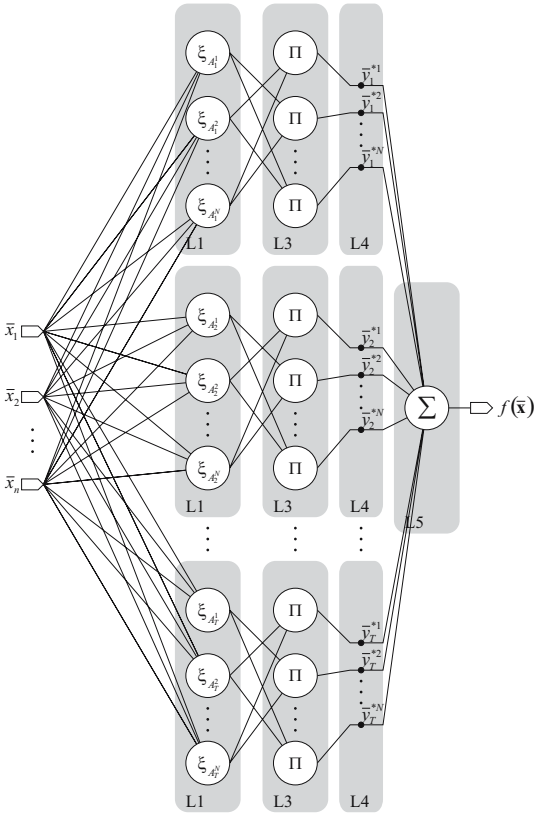


Fig. 6. Merged neuro-fuzzy structure

with random values of all parameters. The dataset were converted such that class label $y(\mathbf{x}) = \pm 1$, which is requirement in the boosting learning algorithm used in the paper. Everytime the training was stopped when $\varepsilon_t \geq 0.5$. The Pima Indians Diabetes (PID) data [2][14] contains two classes, eight attributes

- number of times pregnant
- plasma glucose concentration in an oral glucose tolerance test
- diastolic blood pressure (mm Hg)
- triceps skin fold thickness (mm)
- 2-hour serum
- insulin (mu U/ml)
- body mass index (weight in kg/(height in m)2)
- diabetes pedigree function, age (years)).

We consider 768 instances, 500 (65.1%) healthy and 268 (34.9%) diabetes cases. All patients were females at least 21 years old of Pima Indian heritage, living near Phoenix, Arizona. In our experiments, all sets are divided into a learning sequence (576 sets) and a testing sequence (192 sets).

Table 1. Simulation results for Pima Indians Diabetes problem

Classifier number	Number of fuzzy rules	c_t coefficient	MSE testing error
1	3	0.32	0.34
2	3	0.45	0.22
3	3	0.16	0.56

The training was stopped after the fifth classifier has been trained, because during creating the second classifier value of ε_t did not drop below 0.5. The classification accuracy is 79%. Table 1 shows results of each classifier in the boosting ensemble.

5 Conclusions

The paper introduces a new method for merging logical-type neuro-fuzzy systems constituting a boosting ensemble. The merging is possible thanks to special modification of neuro-fuzzy systems which are parts of the ensemble. The new structures were tested on a popular benchmark.

References

1. R. Babuska, *Fuzzy Modeling For Control*, Kluwer Academic Press, Boston, 1998.
2. C.L. Blake, C.J. Merz, *UCI Repository of machine learning databases*, `www.ics.uci.edu/~mlearn/MLRepository.html`, Irvine, University of California, Department of Information and Computer Science, 1998.
3. L. Breiman, *Bias, variance, and arcing classifiers*, Technical Report 460, Statistics Department, University of California, July 1997.
4. K. Cpałka, "A Flexible Connectionist System" in: *Lecture Notes in Computer Science* 3019, Parallel Processing and Applied Mathematics, Roman Wyrzykowski, Jack Dongarra, Marcin Paprzycki, Jerzy Waniewski (Eds.), Springer-Verlag NewYork, pp. 618-625, 2004.
5. R. J.-S. Jang, C.-T. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, Upper Saddle River, 1997.
6. M. Korytkowski, R. Nowicki, L. Rutkowski, R. Scherer, "Merging Ensemble of Neuro-fuzzy Systems", *2006 IEEE International Conference on Fuzzy Systems, IEEE World Congress on Computational Intelligence*, Vancouver, BC, Canada, (2006).
7. L. I. Kuncheva, *Combining Pattern Classifiers, Methods and Algorithms*, John Wiley & Sons 2004.
8. L. I. Kuncheva, *Fuzzy Classifier Design*, Physica Verlag, Heidelberg, New York, 2000.
9. R. Meir and G. Rätsch, "An introduction to boosting and leveraging", in *Advanced Lectures on Machine Learning*, LNAI 2600, edited by S. Mendelson and A. Smola, Springer, pp. 119-184, 2003.

10. D. Nauck, F. Klawon, R. Kruse, *Foundations of Neuro - Fuzzy Systems*, Chichester, U.K., John Wiley, 1997.
11. D. Nauck, R. Kruse, "How the Learning of Rule Weights Affects the Interpretability of Fuzzy Systems", Proc. of 1998 IEEE World Congress on Computational Intelligence, FUZZ-IEEE, Alaska, pp. 1235–1240, 1998.
12. W. Pedrycz, *Fuzzy Control and Fuzzy Systems*, Research Studies Press, London, 1989.
13. B.D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.
14. L. Rutkowski, *Flexible Neuro-Fuzzy Systems*, Kluwer Academic Publishers, 2004.
15. L. Rutkowski and K. Cpałka, "Designing and learning of adjustable quasi triangular norms with applications to neuro fuzzy systems", *IEEE Trans. on Fuzzy Systems*, vol. 13, Feb 2005, pp. 140-151.
16. L. Rutkowski, K. Cpałka, "Flexible neuro-fuzzy systems", *IEEE Transactions on Neural Networks*, vol. 14, May 2003, pp. 554-574.
17. R. Scherer, L. Rutkowski, "Connectionist Fuzzy Relational Systems", in *Studies in Computational Intelligence, Computational Intelligence for Modelling and Control*, Edited by S. Hagamuge, L.P. Wang, Springer-Verlag, pp. 35–47, 2005.
18. R.E. Schapire, "A brief introduction to boosting", Proc. of the Sixteenth International Joint Conference on Artificial Intelligence, 1999.
19. L.-X. Wang, *Adaptive Fuzzy Systems And Control*, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1994.

On Fuzzy Number Calculus and Some Application

Witold Kosiński

Polish-Japanese Institute of Information Technology
Research Center
ul. Koszykowa 86, 02-008 Warsaw, Poland
wkos@pjwstk.edu.pl
Kazimierz Wielki University of Bydgoszcz
Institute of Environmental Mechanics and Applied Computer Science
ul. Chodkiewicza 30, 85-064 Bydgoszcz, Poland

In memory of the late Professor Ernest Czogała

Abstract. The concept of ordered fuzzy numbers (OFN) is revised in order to handle with general fuzzy inputs in a quantitative way, exactly in the same way as with real numbers. Referring to α cuts and interval arithmetic method the algebraic structure is introduced to allow counting with more general type of membership curves (relations). Application is given to a problem of economics and finance in which fuzzy equation appears.

1 Introduction

Fuzzy numbers are of great importance in fuzzy systems. The fuzzy numbers usually used in applications are the triangular (or triangular shaped) and the trapezoidal (or trapezoidal shaped) fuzzy numbers. This is due to the commonly accepted theory of fuzzy numbers [5] set up by Dubois and Prade [7] in 1978. They proposed a restricted class of membership functions, called (L, R)–numbers with two so-called shape functions: L and R. Then the arithmetic of fuzzy numbers was developed using both the Zadeh's extension principle [28,29] and the α-cut with interval arithmetic method [10].

It is well-known [3] that as long as one works with fuzzy numbers that possess continuous membership functions the two procedures: the extension principle and the α-cut and interval arithmetic method give the same results. However, approximations of fuzzy functions and operations are needed, if one wants to follow the extension principle and stay within (L, R)–numbers. It leads to some drawbacks as well as to unexpected and uncontrollable results of repeatedly applied operations [25,26].

Classical fuzzy numbers are very special fuzzy sets defined on the universe of all real numbers that satisfy three conditions (compare [3,4,6,10,25]): a) the core of a fuzzy number A is nonempty [1], b) α-cuts of A are closed, bounded intervals, and c) the support of A, i.e. supp $A = \{x \in \mathbf{R} : \ \mu_A(x) > 0\}$ is bounded.

[1] A core of A is the set of those $x \in \mathbf{R}$ for which its membership function $\mu_A(x) = 1$.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 250–259, 2006.
© Springer-Verlag Berlin Heidelberg 2006

For the fuzzy number A the α-cut, written $A[\alpha]$ is defined as a classical set $\{x \in \mathbf{R} : \mu_A(x) \geq \alpha\}$. Notice that no assumption about continuity of the membership function μ_A of the fuzzy number has been made. In this way all crisp numbers are fuzzy numbers, as well. However, in most cases one assumes that membership function of a fuzzy number A satisfy convexity assumptions. It was Nguyen [23] who introduced so-called convex fuzzy numbers requiring from all α-cuts to by convex subsets[2] of $\mathbf{R}$. Its generalization has been discussed by Drewniak in several papers, cf. [6], as well as by Klir [11] and Wagenknecht [25].

Those fuzzy numbers are convenient as far as a simple interpretation in the set-theoretical language is concerned [27], however, to operate on them using either the extension principle or α-cut and interval arithmetic method, is not easy task. Moreover, the results of multiply operations on the fuzzy numbers are leading to the large grow of the fuzziness, and, what is especially unpleasant, they depend on the order of operations since the distributive law, which involves the interaction of addition and multiplication, does hold in neither procedures.

Main observations made in our first paper [15] were:

- a kind of quasi-invertibility of membership functions is crucial, and
- if one wants to have the classical arithmetic with crisp numbers within a more general one of that dealing with fuzzy numbers that possess membership functions which are locally invertible then we may (or stronger - we have to) define arithmetic operations on their inverse parts.

Invertibility allows to define two functions a_1, a_2 on $[0,1]$ that give lower and upper bounds of each α-cut of the membership function μ_A of the number A

$$A[\alpha] := \{x \in \mathbf{R} : \mu_A(x) \geq \alpha\} = [a_1(\alpha), a_2(\alpha)], \tag{1}$$

where boundary points are given for each $\alpha \in [0,1]$ by

$$a_1(\alpha) = \mu_A|^{-1}_{incr}(\alpha) \text{ and } a_2(\alpha) = \mu_A|^{-1}_{decr}(\alpha)\ . \tag{2}$$

In (2) the symbol $\mu_A|^{-1}_{incr}$ denotes the inverse function of the increasing part of the membership function $\mu_A|_{incr}$, the other symbol refers to the decreasing part $\mu_A|_{decr}$ of μ. Then we can define the core of A as the interval $[a_1(1), a_2(1)]$, and we can see that the membership function μ_A of A is completely defined by two functions $a_1 : [0,1] \to \mathbf{R}$ and $a_2 : [0,1] \to \mathbf{R}$. In terms of them arithmetic operations for fuzzy numbers can be defined: if A and B are two (convex) fuzzy numbers with the corresponding functions a_1, a_2 and b_1, b_2 for A and B, respectively, which describe (according to (1)-(2)) the boundary points of their α-cuts, then the result C of addition $A + B$ is defined [3,4,23] in terms of their α-cuts and the functions as follows:

$$C[\alpha] = A[\alpha] + B[\alpha],\ C[\alpha] = [a_1(\alpha) + b_1(\alpha),\ a_2(\alpha) + b_2(\alpha)],\ \alpha \in [0,1]. \tag{3}$$

One can do the same for subtraction, however, according to the interval arithmetic [10] if $D = A - B$, then the difference of two intervals is defined

$$D[\alpha] = [a_1(\alpha) - b_2(\alpha),\ a_2(\alpha) - b_1(\alpha)],\ \alpha \in [0,1], \tag{4}$$

[2] Notice that only convex subsets of $\mathbf{R}$ are intervals.

because of the general formula for two arbitrary closed intervals I and J

$$I \star J = \{s \in I, t \in J\} \tag{5}$$

where $\star$ stands for one of operations $+, -, \cdot, \div$. For example, if $I = [2, 4]$ and $J = [3, 5]$ then $[2, 4] - [3, 5] = [2 - 5, 4 - 3] = [-3, 1]$.

Two next operations: multiplication and division may be defined accordingly. Notice, that in subtraction of the same fuzzy number A, i.e. for $C = A - A$, we get $C[\alpha] = [a_1(\alpha) - a_2(\alpha), a_2(\alpha) - a_1(\alpha)]$ which represents non-crisp, fuzzy zero, unless $a_1(\alpha) = a_2(\alpha)$ for each α.

However, when the classical denotation for independent and dependent variables of the membership functions, namely x and y is used, and we look once more at (1)-(2) and Fig.1 when two trapezoidal fuzzy numbers A and B are added, then we can see that for A its corresponding functions are $a_1(\alpha) = 1 + \alpha$ and $a_2(\alpha) = 4 - \alpha$. Now if we put $y = \alpha$ and x for the values of a_1 and a_2, then we will get for two "wings" of the graph of A two possible representations:

$$x = 1 + y \text{ and } x = 4 - y \ , \ y \in [0, 1] \tag{6}$$

or

$$y = x - 1 \text{ if } x \in [1, 2] \text{ and } y = 4 - x \text{ if } x \in [3, 4], \text{ and } \ y = 1 \text{ if } x \in [2, 3]. \tag{7}$$

We can see that in the second representation (7) we have the membership

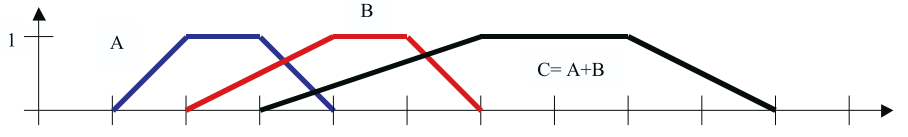


Fig. 1. Addition of two trapezoidal fuzzy numbers A and B

function of A, while the first contains two functions giving the boundary points of all α-cut of A. For the number B in Fig. 1we have

$$x = 2(1 + y) \text{ and } x = 6 - y \ , \ y \in [0, 1] \ , \tag{8}$$

and for the sum C we will have, according to the interval arithmetic method (3)

$$x = (1+y)+2(1+y) = 3(1+y) \text{ and } x = (4-y)+(6-y) = 10-2y \ , \ y \in [0, 1]. \tag{9}$$

We can see that the obtained representation is in agreement with the graph of C in Fig. 1.

In what follows we will use the first approach (6) in the representation of "new' fuzzy numbers, so-called ordered fuzzy numbers which can be identified with pairs of continuous functions defined on the interval $[0, 1]$. In this paper we summarize recent concepts related to the algebra of ordered fuzzy numbers which becomes the efficient tool in dealing with unprecise, fuzzy quantitative terms. Then some applications of fuzzy equations to problems of economics and finance will be given.

2 Ordered Fuzzy Numbers

In the series of papers [12,14,15,16,17,18,19,20] we have introduced and then developed main concepts of the space of ordered fuzzy numbers. In our approach the concept of membership functions [5] has been weakened by requiring a mere *membership relation*. Following our observations made in Introduction a fuzzy number A was originally identified with an ordered pair of continuous real functions defined on the interval $[0,1]$, i.e.

Definition 1. *By an ordered fuzzy number A we mean an ordered pair (f,g) of functions such that $f,g:[0,1]\to \boldsymbol{R}$ are continuous.*

Notice that the continuous functions f and g are exactly the same as already introduced in (1)-(2) the functions a_1 and a_2, respectively. We call the corresponding elements: f – the up-part and g – the down-part of the fuzzy number A.

The continuity of both parts implies their images are bounded intervals, say UP and $DOWN$, respectively (Fig. 1a). We have used symbols to mark boundaries for $UP=[l_A,1_A^+]$ and for $DOWN=[1_A^-,p_A]$.

In general, the functions f,g need not to be invertible as functions of $y\in[0,1]$, only continuity is required. If we assume, however, that
1)they are monotonous: f is increasing, and g is decreasing, and such that
2) $f\le g$ (pointwise), we may define the membership function

$$\begin{aligned}\mu(x)=f^{-1}(x),\ \text{if } x\in[f(0),f(1)]=[l_A,1_A^+],\\ \mu(x)=g^{-1}(x),\ \text{if } x\in[g(1),g(0)]=[1_A^-,p_A]\ \text{ and}\\ \mu(x)=1\ \text{ when } x\in[1_A^+,1_A^-]\ .\end{aligned}\tag{10}$$

In this way we have obtained the membership function $\mu(x), x\in\mathbf{R}$ which is the general representation corresponding to the second form given by (7) in the case of the number A.

Notice that for the representation of the membership function μ of the classical, convex fuzzy number one can attach two monotonous functions $\mu_{up}:=f^{-1}$ and $\mu_{down}:=g^{-1}$ defined on the corresponding intervals $[f(0),f(1)]$ and $[g(1),g(0)]$, respectively.

In the part c) of Fig.2 to the ordered pair of two continuous functions (here just two affine functions) f and g corresponds a membership function of a convex fuzzy number with an extra arrow which denotes the orientation of the closed curve formed below. This arrow shows that we are dealing with the ordered pair of functions and we can see that we have appointed an extra feature, namely an orientation.

Notice that if some of the conditions formulated above are not satisfied the construction of the classical membership function is not possible. However, in the $x-y$ plane the graphs of f and g (both functions of y) can be drawn together with the constant function of x on the interval $[f(1),g(1)]$, equal to 1. Consequently, obtained in this way graphs of three functions form together a curve, which can be called the membership curve of an ordered fuzzy number (f,g) .

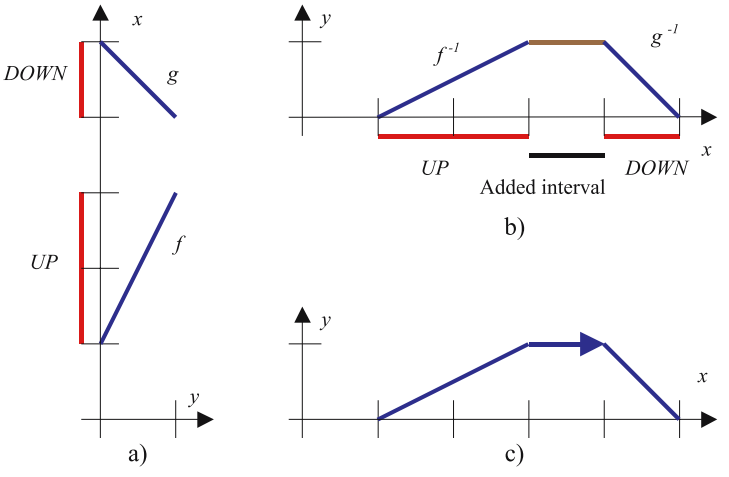


Fig. 2. a) Ordered fuzzy number, b) Ordered fuzzy number with membership function, c) Arrow denotes the order of inverted functions and the orientation

The original definition of the ordered fuzzy numbers [16,17,18] has been recently generalized in [13] by admitting for the pair (f, g) to be functions of bounded variation.

Definition 2. *By an ordered fuzzy number A we mean an ordered pair (f, g) of functions such that $f, g : [0, 1] \to \boldsymbol{R}$ are of bounded variation.*

The necessity of the above generalization follows from some limitations if we pass from the concept of ordered fuzzy numbers (OFN) represented by ordered pairs of continuous functions to the theory of convex fuzzy numbers represented by their membership functions. This is due to the fact that some membership functions already known in the classical theory of fuzzy numbers (cf. [4,5]) cannot be obtained by taking inverses of continuous functions f and g, in the process described above. We think here about such membership functions which are piecewise constant (cf. Fig. 3)), i.e. such μ is one of them if its branches μ_{up} and μ_{down} are not strictly monotonous.

The existence of constancy subintervals implies the inverse functions to μ_{up} and μ_{down}, regarded as functions of y, do not exist in the classical sense. To solve this problem we may assume that both functions μ_{up} and μ_{down} possess at most - countable number of such constancy subintervals, and the inverse functions, say f and g, respectively, exist in a generalized sense, i.e. they are piecewise continuous and monotonous with at most - countable number of discontinuity points. Those discontinuity points are *of the first order*, i.e. at each such a point one–sided limits of the functions exist [21], however, they may be different. Then each jump of discontinuity in the y variable corresponds to a constancy subinterval in the x variable. The class of functions with that property are real–valued functions of *bounded* (finite) *variation* [21] and obtained ordered fuzzy numbers are called of class BV (compare [13] for more details).

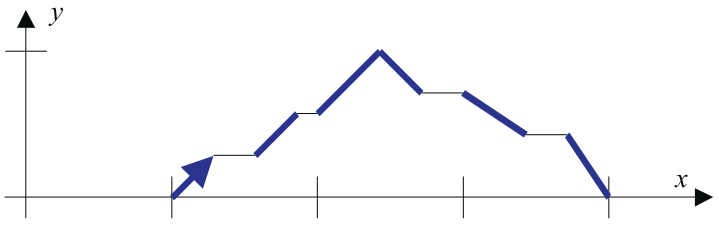


Fig. 3. Membership function of an ordered fuzzy number of class BV

3 Operations

Now, in the most natural way, the operation of addition between two pairs of such functions has been defined (cf. our main definition from [20]), as the pairwise addition of their elements, i.e. if (f_1, g_1) and (f_2, g_2) are two ordered fuzzy numbers, then $(f_1 + f_2, g_1 + g_2)$ will be just their sum. This is exactly the same as operation defined in Introduction on α-cuts of A and B, cf. (3). Notice that as long as we are adding ordered fuzzy numbers which possess their classical counterparts in the form of trapezoidal type membership functions, and moreover, are of the same orientation, the results of addition is in agrement with the α-cut and interval arithmetic. However, this does not hold, in general, if the numbers have opposite orientations, for the result of addition may lead to improper intervals as far as some α-cuts are concerned. In this way we are close to the Kaucher arithmetic [9] with improper intervals, i.e. such $[n, m]$ where n may be greater than m.

Notice that if we stay within ordered fuzzy numbers represented by pairs of affine functions of the variable y then there are pairs to which a trapezoidal type membership function does not correspond (cf. the requirement of the invertibility of f and g and the conditions 1) - 2) formulated above Eq. (10), some of them are improper (as it was noticed already in [20]) like in Fig.4.

Operations are introduced almost identical to that used in our previous publications [12,14,15,16,17,18,19,20].

Definition 3. *Let $A = (f_A, g_A), B = (f_B, g_B)$ and $C = (f_C, g_C)$ are mathematical objects called ordered fuzzy numbers. The sum $C = A + B$, subtraction $C = A - B$, product $C = A \cdot B$, and division $C = A \div B$ are defined by formula*

$$f_C(y) = f_A(y) \star f_B(y) \qquad \wedge \qquad g_C(y) = g_A(y) \star g_B(y) \tag{11}$$

where "$\star$" works for "$+$", "$-$", "$\cdot$", and "$\div$", respectively, and where $A \div B$ is defined, if the functions $|f_B|$ and $|g_B|$, are bounded from below by a positive number.

As it was already noticed in the previous section the subtraction of B is the same as addition of the opposite of B, i.e. the number $(-1) \cdot B$. If we want to add

two pairs of affine functions (i.e. two particular type of ordered fuzzy numbers) defined on $[0,1]$ the final result is easy to obtain, if we apply a mnemotechnic method known in the interval analysis and pointed out by the author in the last paper [13]. If for any pair of affine functions (f,g) of $y \in [0,1]$ we form a quaternion (tetrad) of real numbers according to the rule $[f(0), f(1), g(1), g(0)]$ (which correspond to the presented in Eq. (11) four numbers $[l_A, 1_A^+, 1_A^-, p_A]$), then this tread uniquely determines the ordered fuzzy number A. If $(e,h) =: B$ is another pairs of affine functions then the sum $A + B = (f + e, g + h) =: C$ will be uniquely represented by the tread

$$[f(0) + e(0), f(1) + e(1), g(1) + h(1), g(0) + h(0)] \; . \tag{12}$$

In the assumed definitions (cf. [18]) the operation of subtraction is compatible with a linear structure of OFN's, i.e. $A - B := A + (-1)B$, and the representations (3) and (12) are for our disposal to find the result $D = A - B$ in the form of the corresponding tread.

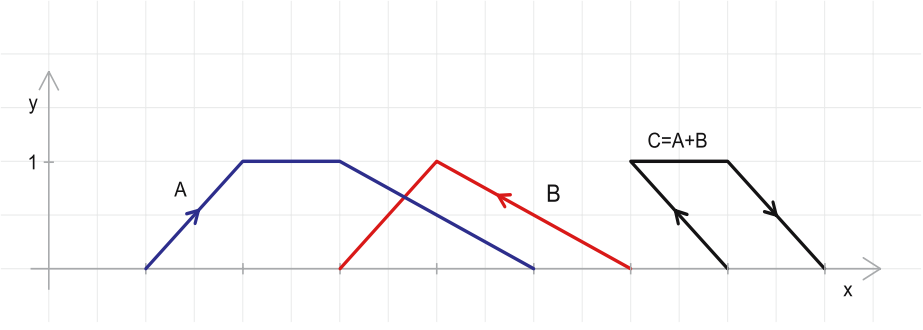


Fig. 4. Sum of two convex OFN's is an improper convex number

Notice that the present operation of subtraction is not the same as the subtraction copied in Introduction from the α-cut and interval arithmetic method, since now, if we use the same denotation as in (4), we will have

$$D[\alpha] = [a_1(\alpha) - b_1(\alpha),\ a_2(\alpha) - b_2(\alpha)],\ \alpha \in [0,1]. \tag{13}$$

Thanks to this definition we will have $A - A = 0$, where 0 is the crisp zero.

If for $A = (f,g)$ we define its complement $\bar{A} = (-g, -f)$ (please note that $\bar{A} \neq (-1) \cdot A$), then the sum $A + \bar{A}$ gives a fuzzy zero $0 = (f - g, -(f - g))$ in the sense of the classical fuzzy number calculus. For better presentation of the advantages of the new operations on OFN we are adding extra figure for the sum.In Fig. 4 we can follow the operation of addition using the tread representation of two trapezoidal ordered fuzzy numbers.

Additionally, more set-theoretic operations can be defined, cf. [13]. The Fuzzy Calculator has been already created as a calculation tool, by my co-worker

Mr.Roman Koleśnik [19]. It lets an easy future use of all mathematical objects described as ordered fuzzy numbers.

Algebraic operations on OFN give a unique possibility to define new types of *compositional rules of fuzzy inference* and new methods of aggregation of premise parts of fuzzy *If–Then*, cf. [24]. The original case of OFN's with continuous elements (f, g) allows to define a set of defuzzyfications operators thanks to the Riesz-Kakutami-Banach theorem, cf. [12].

In $\mathcal{R}$ - the set of all OFN's of class BV a different norm from that of sup (cf. [12,14] can be introduced, compare [13]. Finally, $\mathcal{R}$ is a Banach algebra with the unity $(1^{\dagger}, 1^{\dagger})$, where $1^{\dagger}(y) = 1,\ y \in [0, 1]$.

The relation of partial order in $\mathcal{R}$ can be introduced by defining the subset of those ordered fuzzy numbers which are bigger or equal to zero. We say that the fuzzy number $A = (f, g)$ is not less than zero, and write $A \geq 0$, iff $f \geq 0$ and $g \geq 0$. Hence for two ordered fuzzy numbers B, C the relation $B \geq C$ if $B - C \geq 0$ which makes $\mathcal{R}$ a partial ordered ring.

4 Application

The internal rate of return is one of commonly used methods of comparing mutually exclusive investment alternatives. Let a given, or estimated, net cash flow of a proposed investment project over n periods of time be represented by the finite sequence $A_0, A_1, A_2, ..., A_n$ of positive values, where A_0 is the initial outlay. The internal rate of return (IRR) for this cash flow is any solution $r > -1$ to

$$\sum_{i=1}^{n} A_i (1 + r)^{-i} = A_0 \ . \tag{14}$$

It is a well-known fact from the theory of polynomials, that Eq. (14) possesses a unique solution $r > -1$ if all A_i's are positive real numbers. Notice, that (14) says the present value of all future returns discounted at rate r must equal the initial outlay A_0. If the cost of capital to the firm is r_0 and assuming all projects have a unique IRR, then those projects with IRR $> r_0$ are ranked from highest to lowest according to their IRR value, and the firm accepts these projects in this order until its investment capital is depleted.

Since future returns are always uncertain one can modelled the net cash flow using positive ordered fuzzy numbers, and try to solve Eq. (14) as a fuzzy equation. So one can look for ordered fuzzy number $X = (f_r, g_r)$, which a unique positive root of the fuzzy polynomial

$$W(X) := \sum_{i=1}^{n} A_i X^i - A_0 \ , \tag{15}$$

where all coefficients $A_0, A_1, ..., A_n$ are positive ordered fuzzy numbers. The variable X is related to the previous r from (14) by the relation $X = (1 + r)^{-1}$.

Theorem 1. *If the above assumptions concerning coefficients in (15) are satisfied then there exists a unique positive ordered fuzzy number X_r which is the root of the polynomial $W(X)$, i.e. $W(X_r) = 0$, where 0 is the crisp zero.*

The proof is a simple application of the operations defined on the partial ordered ring $\mathcal{R}$. The above result generalizes those obtained for example in [2], where it was assumed that coefficients are triangular fuzzy numbers.

References

1. Alexiewicz A. (1969), *Functional Analysis* (In Polish: Analiza funkcjonala), Monografie Matematyczne, Tom 49, PWN, Warszawa.
2. Buckley James J. (1992), Solving fuzzy equations in economics and finance, *Fuzzy Sets and Systems*, **48**, 289–296.
3. Buckley James J. and Eslami E. (2005), *An Introduction to Fuzzy Logic and Fuzzy Sets*, Physica-Verlag, A Springer-Verlag Company, Heidelberg.
4. Chen Guanrong, Pham Trung Tat, (2001), *Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*, CRS Press, Boca Raton, London, New York, Washington, D.C.
5. Czogała E., Pedrycz W. (1985), *Elements and Methods of Fuzzy Set Theory* (in Polish), PWN, Warszawa, Poland.
6. Drewniak J.(2001), Fuzzy numbers (In Polish), in: *Fuzzy Sets and their Applications*(In Polish),J. Chojcan, J. Łęski (Eds.), WPŚ, Gliwice, Poland, pp. 103–129.
7. Dubois D., H. Prade H. (1978), Operations on fuzzy numbers, *Int. J. System Science*, **9** (6), 613–626.
8. Goetschel R. Jr., Voxman W. (1986), Elementary fuzzy calculus, *Fuzzy Sets and Systems*, **18** (1), 31-43.
9. Kaucher E. (1980), Interval analysis in the extended interval space IR, *Computing, Suppl.* **2**, 33-49 .
10. Kaufman A. and Gupta M. M. (1991), *Introduction to Fuzzy Arithmetic*, Van Nostrand Reinhold, New York.
11. Klir G.J. (1997), Fuzzy arithmetic with requisite constraints, *Fuzzy Sets and Systems*, **91**(2), 165–175.
12. Kosiński W. (2004), On defuzzyfication of ordered fuzzy numbers, in: *ICAISC 2004, 7th Int. Conference, Zakopane, Poland, June 2004*, L. Rutkowski, Jörg Siekmann, Ryszard Tadeusiewicz, Lofti A. Zadeh (Eds.) LNAI, vol. 3070, pp. 326–331, Springer-Verlag, Berlin, Heidelberg.
13. Kosiński W. (2006), On fuzzy number calculus, *Int. J. Appl. Math. Comput. Sci.*, **16** (1), 51–57.
14. Kosiński W., Prokopowicz P. (2004), Algebra of fuzzy numbers (In Polish: Algebra liczb rozmytych), *Matematyka Stosowana. Matematyka dla Społeczeństwa*, **5** (46), 37–63 .
15. Kosiński W., Piechór K., Prokopowicz P. (2001), Tyburek K.: *On algorithmic approach to operations on fuzzy numbers*, in: Methods of Artificial Intelligence in Mechanics and Mechanical Engineering, T. Burczyński, W. Cholewa (Eds.),pp. 95–98, PACM, Gliwice, Poland.
16. Kosiński W., Prokopowicz P. (2002), Ślęzak D.: Fuzzy numbers with algebraic operations: algorithmic approach, in: *Intelligent Information Systems 2002*, M. Klopotek, S.T. Wierzchoń, M. Michalewicz(Eds.) Proc.IIS'2002, Sopot, June 3-6, 2002, Poland, pp. 311-320, Physica Verlag, Heidelberg.

17. Kosiński W., Prokopowicz P., Ślezak D. (2002), Drawback of fuzzy arithmetics - new intutions and propositions, in: *Proc. Methods of Aritificial Intelligence*, T. Burczyński, W. Cholewa, W. Moczulski(Eds.), pp. 231-237, PACM, Gliwice, Poland.
18. Kosiński W., P. Prokopowicz P., Ślęzak D. (2003), On algebraic operations on fuzzy numbers, in *Intelligent Information Processing and Web Mining*, Proc. of the International IIS: IIPWM,03 Conference held in Zakopane, Poland, June 2-5,2003, M. Klopotek, S.T. Wierzchoń, K. Trojanowski(Eds.), pp. 353-362, Physica Verlag, Heidelberg.
19. Koleśnik R., Prokopowicz P., Kosiński W. (2004), Fuzzy Calculator – usefull tool for programming with fuzzy algebra, in *Artficial Intelligence and Soft Computing - ICAISC 2004, 7th Int. Conference, Zakopane, Poland, June 2004*, L. Rutkowski, Jörg Siekmann, Ryszard Tadeusiewicz, Lofti A. Zadeh (Eds.) LNAI, vol. 3070, pp. 320–325, Springer-Verlag, Berlin, Heidelberg.
20. Kosiński W., Prokopowicz P., Ślęzak D.(2003), Ordered fuzzy numbers, *Bulletin of the Polish Academy of Sciences*, Sér. Sci. Math., **51** (3), 327-338.
21. Łojasiewicz S. (1973), *Introduction to the Theory of Real Functions* (In Polish) *Wstęp do teorii funkcji rzeczywistych*, Biblioteka Matematyczna, Tom 46, PWN, Warszawa.
22. Martos B. (1983), *Nonlinear Programming – Theory and Methods*, PWN, Warszawa, Poland (Polish translation of the English original published by Akadémiai Kiadó, Budapest, 1975).
23. Nguyen H.T. (1978), A note on the extension principle for fuzzy sets, *J. Math. Anal. Appl.* **64**, 369-380 .
24. Prokopowicz P. (2005), *Algorithmization of Operations on Fuzzy Numbers and its Applications* (In Polish: Algorytmizacja działań na liczbach rozmytych i jej zastosowania), Ph. D. Thesis, IPPT PAN, kwiecień 2005.
25. Wagenknecht M. (2001), On the approximate treatment of fuzzy arithmetics by inclusion, linear regression and information content estimation, in: *Fuzzy Sets and their Applications* (In Polish), J. Chojcan, J. Łęski (eds.), Wydawnictwo Politechniki Śląskiej, Gliwice, 291-310.
26. Wagenknecht M., Hampel R., Schneider V.(2001), Computational aspects of fuzzy arithmetic based on archimedean t-norms, *Fuzzy Sets and Systems*, **123** (1), 49–62.
27. Zadeh L. A.(1965), Fuzzy sets, *Information and Control*, **8** (3), 338–353.
28. Zadeh L. A. (1975), The concept of a linguistic variable and its application to approximate reasoning, Part I, *Information Sciences*, **8**(3), 199–249.
29. Zadeh L. A.(1983), The role of fuzzy logic in the management of uncertainty in expert systems, *Fuzzy Sets and Systems*, **11**(3), 199–227.

Combination of Fuzzy TOPSIS and Fuzzy Ranking for Multi Attribute Decision Making

Mohammad Reza Mehregan and Hossein Safari

Department of Industrial Management, Faculty of Management, University of Tehran,
Nasr Bridge, North Kargar St., Tehran, Iran
hsafari@ut.ac.ir

Abstract. TOPSIS is a multiple criteria method to identify solutions from a finite set of alternatives based upon simultaneous minimization of distance from an ideal positive point and maximization of distance from a negative point. Owing to vague concepts frequently represented in decision data, the crisp value is inadequate to model real-life situations. In this paper, the scoring of each alternative and the weight of each criterion are described by linguistic terms which can be expressed in triangular fuzzy numbers. Then, the ratings and weights assigned by decision makers are averaged and normalized into a comparable scale. A coefficient of variation is defined to determine the ranking order of alternatives by calculating the mean value and standard deviation. A numerical example demonstrates the feasibility of the proposed method.

1 Introduction

Among the numerous approaches available for conflict management, one of the most prevalent is multicriteria decision making. Decision-making problem is the process of finding the best option from all of the feasible alternatives. In classical MCDM methods, the ratings and the weights of the criteria are known precisely. MCDM may be considered as a complex and dynamic process including one managerial level and one engineering level [10].

Technique for order performance by similarity to ideal solution (TOPSIS), one of the known classical MCDM methods, was first developed by Hwang and Yoon [4]. It bases upon the concept that the chosen alternative should have the shortest distance from the positive ideal solution and the farthest from the negative ideal solution. In the process of TOPSIS, the performance ratings and the weights of the criteria are given as crisp values.

Under many conditions, crisp data are inadequate to model real-life situations. Since human judgments including preferences are often vague and cannot estimate his preference with an exact numerical value.

A more realistic approach may be to use linguistic assessments instead of numerical values, that is, to suppose that the ratings and weights of the criteria in the problem are assessed by means of linguistic variables [5, 9, 13, 16]. Decision making

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 260 – 267, 2006.
© Springer-Verlag Berlin Heidelberg 2006

often takes place in a fuzzy environment where the information available is imprecise or uncertain. For fuzzy decision problems of prioritizing or evaluating a finite set of alternatives involving multiple criteria, the application of fuzzy set theory to multicriteria analysis models under the framework of utility theory has proven to be an effective approach [2, 11, 17, 19]. In fuzzy TOPSIS, attribute values are represented by fuzzy numbers.

In this paper, we further extended the concept of TOPSIS to develop a methodology for solving multi-person multi-criteria decision-making problems in fuzzy environment. In order to develop the fuzzy TOPSIS method, the paper is organized as follows. Next section introduces the basic definitions and notations of the fuzzy number and linguistic variable. Section 3 presents the proposed fuzzy TOPSIS method in group decision making and the choice process. And then, the proposed method is illustrated with an example. Finally, some conclusions are pointed out in the end of this paper.

2 Preliminary

In the following, we briefly review some basic definitions of fuzzy sets from [1, 7, 14]. These basic definitions and notations below will be used throughout the paper until otherwise stated.

Definition 1. A real fuzzy number A is described as a fuzzy subset of the real line R with membership function f_A which possesses the following properties [6]:

1. f_A is a continuous mapping from R to the closed interval [0,1].
2. $f_A(x)$=0, for all $x \le a$.
3. f_A is strictly decreasing on [a, b].
4. f_A=1, for all $b \le x \le c$.
5. f_A is strictly decreasing on [c, d].
6. $f_A(x)$=0, for all $x \ge d$.

Where a, b, c, and d are real numbers. It may be $a = -\infty$, $a = b$, or $b = c$, or $c = d$, or $d = +\infty$.

The membership function f_A of the fuzzy number A can also be expressed as [15]:

$$f_A = \begin{cases} f_A^L & a \le x \le b \\ 1 & b \le x \le c \\ f_A^R & c \le x \le d \\ 0 & otherwise \end{cases} \tag{1}$$

A triangle fuzzy number A can be defined by a triplet (a,b,c) shown in Fig. 1. For this fuzzy number, f_A^L is left membership function and is equal to $(x-a)/(b-a)$ and f_A^R is right membership function and is equal to $(x-c)/(b-c)$.

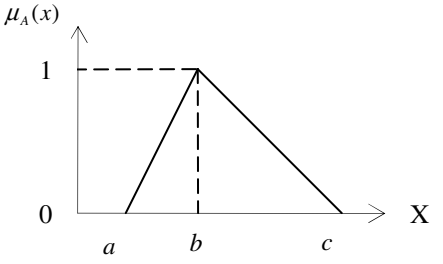


Fig. 1. A triangular fuzzy number

Definition 2. The -cut of fuzzy number A is defined as [10]:

$$A^\alpha = \{x | f_A(x) \geq a\},\ x \in R,\ 0 \leq \alpha \leq 1 \tag{2}$$

A^α is a non-empty bounded closed interval contained in R and it can be denoted by $A^\alpha = [A_l^\alpha, A_u^\alpha]$, where A_l^α and A_u^α are the lower and upper bounds of the closed interval, respectively [1, 11]. For triangle fuzzy number A, -cut can be expressed as:

$$A^\alpha = [A_l^\alpha, A_u^\alpha] = [(b-a)\alpha + a, (b-c)\alpha + c] \tag{3}$$

If A is a fuzzy number and $A^\alpha > 0$ for $0 \leq \alpha \leq 1$, then A is called a positive fuzzy number [12, 13].

Given any two positive fuzzy numbers A and B subset of the R^+, the -cut of them are $A^\alpha = [A_l^\alpha, A_u^\alpha]$ and $B^\alpha = [B_l^\alpha, B_u^\alpha]$. By interval arithmetic, some main operations of A and B can be expressed as following [1]:

$$(A(+)B)^\alpha = [A_l^\alpha + B_l^\alpha, A_u^\alpha + B_u^\alpha] \tag{4}$$

$$(A(-)B)^\alpha = [A_l^\alpha - B_l^\alpha, A_u^\alpha - B_u^\alpha] \tag{5}$$

$$(A(\times)B)^\alpha = [A_l^\alpha B_l^\alpha, A_u^\alpha B_u^\alpha] \tag{6}$$

$$(A(\div)B)^\alpha = [A_l^\alpha \div B_u^\alpha, A_u^\alpha \div B_l^\alpha] \tag{7}$$

$$\left(A(\times)r\right)^{\alpha} = \left[A_l^{\alpha} r, A_u^{\alpha} r\right], r \in R^+ \tag{8}$$

$$\left(A(\times)r\right)^{\alpha} = \left[A_l^{\alpha} r, A_u^{\alpha} r\right], r \in R^+ \tag{9}$$

$$\left(A^{\alpha}\right)^{-1} = \left[1 \div A_u^{\alpha}, 1 \div A_l^{\alpha}\right] \tag{10}$$

Definition 4. According to the Lee and Li [8], mean and standard deviation of the triangular fuzzy number $A = (a,b,c)$ are defined as:

$$\mu_A = \frac{1}{4}(a + 2b + c) \tag{11}$$

$$\sigma_A = \frac{1}{80}\left(3a^2 + 4b^2 + 3c^2 - 4ab - 4bc - 2ac\right) \tag{12}$$

3 Proposed Method

This method is very suitable for solving the group decision-making problem under fuzzy environment. In this paper, the importance weights of various criteria and the score of qualitative criteria are considered as linguistic variables.

At whole, an algorithm of the multi-person multicriteria decision making with fuzzy set approach is given in the following:

Step 1- Identifying the evaluation criteria and the appropriate linguistic variables for the importance weight of the criteria and the linguistic score for alternatives with respect to criteria.
Step 2- Constructing the normalized fuzzy decision matrix (NFDM).
Step 3- Constructing the weighted mean normalized fuzzy decision matrix (WMNFDM).
Step 4- Determining the mean, standard deviation, coefficient variation, and ranking of each alternative.

3.1 Identifying the Evaluation Criteria and the Appropriate Linguistic Variables

Assume that a committee of k decision makers $(D^1, D^2, ..., D^k)$ is responsible for evaluating m alternatives $(A_1, A_2, ..., A_m)$ under n criteria $(C_1, C_2, ..., C_n)$. Criteria are classified into benefit (B) and cost (C).

Let $x_{ij}^t = \left(a_{ij}^t, b_{ij}^t, c_{ij}^t\right)$; $x_{ij}^t \in R^+$; $i = 1,2,...,m$; $j = 1,2,...,n$; $t = 1,2,...,k$, is a triangular fuzzy number and the score assigned to alternative A_i by decision maker D^t for criterion C_j. In addition, let $w_j^t = \left(e_j^t, f_j^t, g_j^t\right)$; $w_j^t \in R^+$; $j = 1,2,...,n$;

$t = 1, 2, ..., k$ is a triangular fuzzy number and the weight assigned to criterion C_j by decision maker D^t.

3.2 Constructing the NFDM

According to before step, the importance of the criteria and the rating of alternatives with respect to each criterion can be calculated as [20]:

$$x_{ij} = \frac{1}{k}(\times)\left[x_{ij}^1(+)x_{ij}^2(+)...(+)x_{ij}^k\right];\ a_{ij} = \sum_{t=1}^{k} a_{ij}^t \Big/ k\ ;\ b_{ij} = \sum_{t=1}^{k} b_{ij}^t \Big/ k\ ;\ c_{ij}^t = \sum_{t=1}^{k} a_{ij}^t \Big/ k \quad (13)$$

$$w_j = \frac{1}{k}(\times)\left[w_j^1(+)w_j^2(+)...(+)w_j^k\right];\ e_j = \sum_{t=1}^{k} e_j^t \Big/ k\ ;\ f_j = \sum_{t=1}^{k} f_j^t \Big/ k\ ;\ g_j^t = \sum_{t=1}^{k} g_j^t \Big/ k \quad (14)$$

To ensure compatibility between averaged scores and averaged weights, they must normalize into a comparable scale. Then the normalization formula used in classical TOPSIS, was used. Therefore, we can obtain the normalized fuzzy decision matrix denoted by $\tilde{U}$ [20].

$$\begin{aligned}
&x_{ij}^+ = \underset{i}{Max}\, x_{ij} = \left(\underset{i}{Max}\ a_{ij}, \underset{i}{Max} b_{ij}, \underset{i}{Max} c_{ij}\right) \quad , \quad j \in C;\\
&x_{ij}^- = \underset{i}{Min}\, x_{ij} = \left(\underset{i}{Min}\ a_{ij}, \underset{i}{Min} b_{ij}, \underset{i}{Min} c_{ij}\right) \quad , \quad j \in B;\\
&\tilde{u}_{ij} = x_{ij} \Big/ \sum_{i=1}^{m} x_{ij} = \left(a_{ij} \Big/ \sum_{i=1}^{m} a_{ij},\ b_{ij} \Big/ \sum_{i=1}^{m} b_{ij},\ c_{ij} \Big/ \sum_{i=1}^{m} c_{ij}\right) \quad , \quad j \in B;\\
&\tilde{u}_{ij} = \left(x_{ij}^+ + x_{ij}^- - x_{ij}\right) \Big/ \sum_{i=1}^{m} x_{ij} = \left\{\left(\underset{i}{Max}\ a_{ij} + \underset{i}{Min}\ a_{ij} - a_{ij}\right) \Big/ \sum_{i=1}^{m} a_{ij}, \left(\underset{i}{Max}\ b_{ij} + \underset{i}{Min}\ b_{ij} - b_{ij}\right) \Big/ \sum_{i=1}^{m} b_{ij},\right.\\
&\qquad \left.\left(\underset{i}{Max}\ c_{ij} + \underset{i}{Min}\ c_{ij} - c_{ij}\right) \Big/ \sum_{i=1}^{m} c_{ij}\right\} \quad , \quad j \in C
\end{aligned} \quad (15)$$

According to the normalization method mentioned above, the ranges of normalized triangular fuzzy numbers belong to [0, 1].

3.3 Constructing the WMNFDM

According to the following equation, we can construct the weighted mean normalized fuzzy decision matrix.

$$\tilde{V} = \sum \tilde{U}(\times)\tilde{W}\ ;\ v_{ij} = \left\{\left(\sum_{j=1}^{n} u_{ij}^a (\times) w_j^a\right), \left(\sum_{j=1}^{n} u_{ij}^b (\times) w_j^b\right), \left(\sum_{j=1}^{n} u_{ij}^c (\times) w_j^c\right)\right\} \quad (16)$$

3.4 Ranking

In order to choose a best alternative, we need a method for building a crisp total ordering from fuzzy numbers. Many methods for ranking of fuzzy numbers have been suggested. Each method appears to have some advantages as well as disadvantages

[18]. We can compare them on the basis of the standard deviation or mean value. From the concept of statistics, the standard deviation and mean value cannot be the sole basis for comparing two fuzzy numbers, respectively. Furthermore, according to Lee and Li [3], higher mean value and at the same time lower spread is ranked higher. However, when higher mean value and at the same time higher spread/or lower mean value and at the same time lower spread, it is not easy to compare the orderings clearly. Therefore, we propose an efficient index [3] that is, using the coefficient of variation. It is defined as

$$CV = \sigma\ (S\tan dard\ Deviation) \,/\, \mu\ (Mean) \quad ; \quad \mu \neq 0\ ,\ \sigma > 0 \tag{17}$$

For calculation of Standard deviation and mean value, Lee and Li method is used (Eq. 10, 11).

4 An Illustrative Example

In this section, an example of [7] has been solved. According to the fuzzy TOPSIS steps, we have:

Step 1- In this example, we have three decision makers, three alternatives, and five criteria. Assume that decision maker use the linguistic score set S = {VL, L, ML, M, MH, H, VH}, where VL = very low = (0, 0, 0.1), L = low = (0, 0.1, 0.3), ML = medium low = (0.1, 0.3, 0.5), M = medium = (0.3, 0.5, 0.7), MH = medium high = (0.5, 0.7, 0.9), H = high = (0.7, 0.9, 1), VH = very high = (0.9, 1, 1) to evaluate the suitability of each alternative under each of the criteria. Also assume that the decision makers employ a linguistic weighting set W = {VP, P, MP, F, MG, G, VG}, where VP = very poor = (0, 0, 1), P = poor = (0, 1, 3), MP = medium poor = (1, 3, 5), F = (3, 5, 7), MG = medium good = (5, 7, 9), G = good = (7, 9, 10), VG = very good = (9, 10, 10) to assess the importance of all the criteria.

Step 2- Table 1 and 2, represent The importance weight of the criteria and The ratings of the three candidates by decision makers under all Criteria, respectively. By Eq. (13), the averaged suitability rating of each alternative A_i with respect to criterion C_j from the decision making committee can be obtained, as shown in Table 1. By Eq. (14), the averaged weights of the criteria C_j from the decision making committee can be obtained, as shown in Table 2.

Table 1. The importance weight of the criteria

Criteria	Decision Maker			Average Weights
	D1	D2	D3	
C_1^+	H	VH	MH	(0.7, 0.87, 0.97)
C_2^-	VH	VH	VH	(0.9, 1, 1)
C_3^+	VH	H	H	(0.77, 0.93, 1)
C_4^-	VH	VH	VH	(0.9, 1, 1)
C_5^+	MH	MH	MH	(0.5, 0.7, 0.9)

Table 2. The ratings of the three candidates by decision makers under all criteria

Criteria	Candidates	Decision Maker			Average Weights
		D1	D2	D3	
C_1^+	A1	MG	G	MG	(5.67, 7.67, 9.33)
	A2	G	G	MG	(6.33, 8.33, 9.67)
	A3	VG	G	F	(6.33, 8, 9)
C_2^-	A1	G	MG	F	(5, 7, 8.67)
	A2	VG	VG	VG	(9, 10, 10)
	A3	MG	G	VG	(7, 8.67, 9.67)
C_3^+	A1	F	G	G	(5.67, 7.67, 9)
	A2	VG	VG	G	(8.33, 9.67, 10)
	A3	G	MG	VG	(7, 8.67, 9.67)
C_4^-	A1	VG	G	VG	(8.33, 9.67, 10)
	A2	VG	VG	VG	(9, 10, 10)
	A3	G	VG	MG	(7, 8.67, 9.67)
C_5^+	A1	F	F	F	(3, 5, 7)
	A2	VG	MG	G	(7, 8.67, 9.67)
	A3	G	G	MG	(6.33, 8.33, 9.67)

Step 3- According to the Eq. (15), weighted normalized fuzzy matrix is as Table 3.

Table 3. The weighted normalized fuzzy decision matrix

	C_1^+	C_2^-	C_3^+	C_4^-	C_5^+
A1	(0.20, 0.32, 0.51)	(0.32, 0.39, 0.48)	(0.20, 0.29, 0.43)	(0.19, 0.26, 0.36)	(0.11, 0.23, 0.43)
A2	(0.23, 0.35, 0.53)	(0.18, 0.27, 0.41)	0.29, 0.37, 0.48)	(0.17, 0.25, 0.36)	(0.27, 0.39, 0.59)
A3	(0.23, 0.33, 0.49)	(0.25, 0.32, 0.43)	(0.24, 0.33, 0.48)	(0.24, 0.29, 0.37)	(0.24, 0.38, 0.59)

Step 4- According to the Eq. (16), the weighted mean normalized fuzzy decision matrix is as Table 4. Afterwards, by Eq. (11, 12, 17), final ranking is as Table 4.

Table 4. The weighted mean, mean value, standard deviation, and coefficient variation

	WMNEM	Mean	Standard Deviation	CV	RANK
A1	(0.81, 1.36, 2.14)	1.42	0.04	0.0317	2
A2	(0.82, 1.44, 2.29)	1.50	0.05	0.0361	3
A3	(0.90, 1.48, 2.27)	1.53	0.05	0.0307	1

5 Conclusion

In this paper, a linguistic decision process is proposed to solve the multi person multi criteria decision-making problem under fuzzy environment. Considering the fuzziness in the decision data and group decision-making process, linguistic variables are used to assess the weights of all criteria and the ratings of each alternative

with respect to each criterion. In this paper, at first, we converted the decision matrix into a fuzzy decision matrix. Then according to the concept of fuzzy TOPSIS, a normalized fuzzy decision matrix was constructed. In the next step, a weighted mean normalized fuzzy decision matrix was made. Finally, for alternatives ranking, we calculate mean value, standard deviation, and the coefficient of variation. The best alternative has the lowest coefficient of variation.

References

1. A. Kaufmann, M.M. Gupta.: Introduction to Fuzzy Arithmetic: Theory and Applications, Van Nostrand Reinhold, New York, 1985.
2. C. Carlsson, R. Fuller.: Fuzzy multiple criteria decision making: recent developments, Fuzzy Sets and Systems 78 (1996) 139–153.
3. C.H. Cheng.: A new approach for ranking fuzzy numbers by distance method, Fuzzy Sets and Systems 95 (1998) 307- 317.
4. C.L. Hwang, K. Yoon.: Multiple Attributes Decision Making Methods and Applications, Springer, Berlin Heidelberg (1981).
5. C.T. Chen.: Extensions of the TOPSIS for group decision-making under fuzzy environment, Fuzzy Sets and Systems 114 (2000) 1-9.
6. D. Dubis and H. Prade.: Operations on fuzzy numbers, International journal of Systems Science 9 (1978) 613-626.
7. D.S. Negi.: Fuzzy analysis and optimization, Ph.D. Thesis, Department of Industrial Engineering, Kansas State University, 1989.
8. E.S. Lee and R.L. Li.: Comparison of fuzzy numbers based on the probability measure of fuzzy events, Computer Mathematics Application 15 (1988) 887-896.
9. F. Herrera, E. Herrera-Viedma, J.L. Verdegay.: A model of consensus in group decision making under linguistic assessments, Fuzzy Sets and Systems 78 (1996) 73-87.
10. H. Deng, C. H., Yeh, R. J., Willis.: Inter-company comparison using modified TOPSIS with objective weights, Computers & Operations Research 27 (10) (2000) 963–973.
11. H.-J. Zimmermann.: Fuzzy Set Theory and Its Applications, Kluwer, Boston, 1996.
12. H.M. Hsu, C.T. Chen.: Fuzzy credibility relation method for multiple criteria decision-making problems, Information Science 96 (1997) 79-91.
13. H.M. Hsu, C.T. Chen.: Fuzzy credibility relation method for multiple criteria decision-making problems, Information Sciences 96 (1997) 79-91.
14. J.J. Buckley.: Fuzzy hierarchical analysis, Fuzzy Sets and Systems 17 (1985) 233-247.
15. P. J. M. van Laarhoven and Q. Pedryez.: A fuzzy extension of Satty's priority theory, Fuzzy Sets and Systems 11(1983) 229-241.
16. R.E. Bellman, L.A. Zadeh.: Decision-making in a fuzzy environment, Management Sci.17 (4) (1970) 141-164.
17. S. Murakami, S. Maeda, S. Imamura.: Fuzzy decision analysis on the development of centralized regional energy control system, Proceedings of IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis, 1983.
18. S.-J. Chen and C-L. Hwang.: Fuzzy multiple attribute decision making methods and applications, Lecture Notes in Economics and Mathematical Systems, Springer, New York, 1992.
19. S.J. Chen, C.L. Hwang.: Fuzzy Multiple Attribute Decision Making: Methods and Applications, Springer-Verlag, New York, 1992.
20. T. C. Chu.: Selecting Plant Location via a Fuzzy TOPSIS Approach, International Journal of Manufacturing Technology 20 (2002) 859-864.

Flow Graphs and Decision Tables with Fuzzy Attributes

Alicja Mieszkowicz-Rolka and Leszek Rolka

Department of Avionics and Control,
Rzeszów University of Technology,
ul. W. Pola 2, 35-959 Rzeszów, Poland
{alicjamr, leszekr}@prz.edu.pl

Abstract. This paper is concerned with the issue of design and analysis of fuzzy decision systems, basing on recorded process data. A concept of fuzzy flow graphs is proposed to allow representation of decision tables with fuzzy attributes. Basic notions of the crisp flow graph approach are generalized. Satisfaction of flow graph properties, with respect to fuzzy connectives used in calculations, is taken into account. Alternative definitions of the path's certainty and strength are introduced. In an illustrative example a decision table with fuzzy attributes is analyzed and interpreted in terms of flow graphs.

1 Introduction

An appropriate utilization of recorded process data and decision examples, for creating a set of relevant fuzzy decision rules, is an important problem in applications of fuzzy inference systems [3,9]. The used data can be conveniently represented in the form of a decision table with fuzzy attributes. It might be advantageous to carry out an analysis of this kind of decision table, by applying the fuzzy rough set model [4].

A hybrid approach to decision algorithms, proposed by Pawlak [5,6,7], combines the idea of flow graphs with the crisp rough set model. Greco, Pawlak and Słowiński [1] proved that relaxation of mutual exclusion of decision rules does not violate basic properties of flow graphs, and every decision algorithm can be associated with a flow graph.

The main goal of this paper consists in developing a fuzzy flow graph approach, which is suitable for representing and analyzing fuzzy decision systems. First of all, we want to address crucial issues of the generalized flow graph approach, concerning especially its connection to fuzzy inference systems. We concentrate on the aspect of representing and selecting fuzzy decision rules with the help of flow graphs. The problem of a correct choice of fuzzy connectives is considered with the aim to retain the flow conservation equations. Furthermore, we give new definitions of the path's certainty and strength, by respecting only the relevant part of the flow, i.e by disregarding the flow components which come from other paths.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 268–277, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Decision Tables with Fuzzy Attributes

In further discussion, we apply fuzzy decision tables in the form introduced in [4]. We consider a finite universe U with N elements: $U = \{x_1, x_2, \ldots, x_N\}$. Every element x of the universe U will be described with the help of fuzzy attributes, which are divided into a subset of n condition attributes $C = \{c_1, c_2, \ldots, c_n\}$, and a subset of m decision attributes $D = \{d_1, d_2, \ldots, d_m\}$.

We assign a set of linguistic values to every fuzzy attribute. Let us denote by $V_i = \{V_{i1}, V_{i2}, \ldots, V_{in_i}\}$ the family of linguistic values of the condition attribute c_i, and by $W_j = \{W_{j1}, W_{j2}, \ldots, W_{jm_j}\}$ the family of linguistic values of the decision attribute d_j, where n_i and m_j is the number of the linguistic values of the i-th condition and the j-th decision attribute, respectively, $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.

For any element $x \in U$, its membership degrees in all linguistic values of the condition attribute c_i (or decision attribute d_j) should be determined. This is accomplished during the fuzzification stage, basing on the recorded crisp value of a particular attribute of x. The value of an attribute for a given element x is a fuzzy set on the domain of all linguistic values of that attribute.

We denote by $V_i(x)$ the fuzzy value of the condition attribute c_i for any x, as a fuzzy set on the domain of the linguistic values of c_i

$$V_i(x) = \{\mu_{V_{i1}}(x)/V_{i1},\ \mu_{V_{i2}}(x)/V_{i2},\ \ldots,\ \mu_{V_{in_i}}(x)/V_{in_i}\}\,. \tag{1}$$

$W_j(x)$ denotes the fuzzy value of the decision attribute d_j for any x, as a fuzzy set on the domain of the linguistic values of d_j

$$W_j(x) = \{\mu_{W_{j1}}(x)/W_{j1},\ \mu_{W_{j2}}(x)/W_{j2},\ \ldots,\ \mu_{W_{jm_j}}(x)/W_{jm_j}\}\,. \tag{2}$$

If the linguistic values of an attribute have the form of singletons or disjoint intervals, with membership degree equal to 1 on the original domain of the attribute, then only one linguistic value can be assigned to that attribute. In that case we get a classical crisp decision table. In general, we obtain a non-zero membership of x to more than one linguistic value of an attribute.

Table 1. Decision table with fuzzy attributes

	c_1	c_2	$\cdots$	c_n	d_1	d_2	$\cdots$	d_m
x_1	$V_1(x_1)$	$V_2(x_1)$	$\cdots$	$V_n(x_1)$	$W_1(x_1)$	$W_2(x_1)$	$\cdots$	$W_m(x_1)$
x_2	$V_1(x_2)$	$V_2(x_2)$	$\cdots$	$V_n(x_2)$	$W_1(x_2)$	$W_2(x_2)$	$\cdots$	$W_m(x_2)$
					$\cdots$			
x_N	$V_1(x_N)$	$V_2(x_N)$	$\cdots$	$V_n(x_N)$	$W_1(x_N)$	$W_2(x_N)$	$\cdots$	$W_m(x_N)$

Furthermore, we assume that for any element $x \in U$, all linguistic values $V_i(x)$ and $W_j(x)$ $(i = 1, 2, \ldots, n\,,\ \ j = 1, 2, \ldots m)$ satisfy the requirments

$$\text{power}(V_i(x)) = \sum_{k=1}^{n_i} \mu_{V_{ik}}(x) = 1\,, \qquad \text{power}(W_j(x)) = \sum_{k=1}^{m_j} \mu_{W_{jk}}(x) = 1\,. \tag{3}$$

This assumption allows us to generalize the flow graph approach and use it for analysis of fuzzy information system.

Decision tables with fuzzy values of attributes will be applied for examining all possible decision rules generated by using the Cartesian product of sets of the linguistic values.

Let us denote by R_k the k-th decision rule from the set consisting of r possible decision rules ($r = \prod_{i=1}^{n} n_i \prod_{j=1}^{m} m_j$)

$$\begin{aligned} R_k\colon\ & \text{IF } c_1 \text{ is } V_1^k \text{ AND } c_2 \text{ is } V_2^k \ \ldots \text{ AND } c_n \text{ is } V_n^k \\ & \text{THEN } d_1 \text{ is } W_1^k \text{ AND } d_2 \text{ is } W_2^k \ \ldots \text{ AND } d_m \text{ is } W_m^k \end{aligned} \tag{4}$$

where $k = 1, 2, \ldots, r\,,\ \ V_i^k \in V_i\,,\ \ i = 1, 2, \ldots n\,,\ \ W_j^k \in W_j\,,\ \ j = 1, 2, \ldots, m.$

When we use the fuzzy Cartesian products $C^k = V_1^k \times V_2^k \ldots \times V_n^k$ and $D^k = W_1^k \times W_2^k \ldots \times W_m^k$, the k-th decision rule can be written in the form of a fuzzy implication $C^k \rightarrow D^k$.

We need to select a subset of those decision rules which are relevant to the considered decision process. To this end, we determine to what degree any element $x \in U$, corresponding to a single row of the decision table, confirms particular decision rules. We should calculate the truth value of the decision rule's antecedent and the truth value of the decision rule's consequent, by determining the conjunction of the respective membership degrees of x in the linguistic values of attributes.

In the case of a decision table with crisp attributes, a decision rule is confirmed for some x, if the result of conjunction is equal to 1, both for the rule's premise and the rule's conclusion. Otherwise, the element x does not confirm the considered decision rule. The set of those elements $x \in U$, which confirm a decision rule, is called the support of the decision rule.

In order to determine the confirmation degree of fuzzy decision rules, in the case of decision tables with fuzzy attributes, we need to apply a T-norm operator. By $\mathrm{cd}(x, k)$, we denote the confirmation degree of the k-th decision rule by the element $x \in U$

$$\mathrm{cd}(x, k) = \mathrm{T}(\mathrm{cda}(x, k), \mathrm{cdc}(x, k))\,, \tag{5}$$

where $\mathrm{cda}(x, k)$ denotes the confirmation degree of the decision rule's antecedent

$$\mathrm{cda}(x, k) = \mathrm{T}(\mu_{V_1^k}(x), \mu_{V_2^k}(x), \ldots, \mu_{V_n^k}(x))\,, \tag{6}$$

and $\mathrm{cdc}(x, k)$ the confirmation degree of the decision rule's consequent

$$\mathrm{cdc}(x, k) = \mathrm{T}(\mu_{W_1^k}(x), \mu_{W_2^k}(x), \ldots, \mu_{W_m^k}(x))\,. \tag{7}$$

By determining the confirmation degrees (6), (7) and (5), we get the following fuzzy sets on the domain U:
the support of the decision rule's antecedent

$$\mathrm{support}(\mathrm{cda}(x, k)) = \{\mathrm{cda}(x_1, k)/x_1,\ \mathrm{cda}(x_2, k)/x_2,\ \ldots,\ \mathrm{cda}(x_N, k)/x_N\}\,, \tag{8}$$

the support of the decision rule's consequent

$$\text{support}(\text{cdc}(x,k)) = \{\text{cdc}(x_1,k)/x_1,\ \text{cdc}(x_2,k)/x_2,\ \ldots,\ \text{cda}(x_N,k)/x_N\}\,, \quad (9)$$

and the support of the decision rule R_k, respectively

$$\text{support}(R_k) = \{\text{cd}(x_1,k)/x_1,\ \text{cd}(x_2,k)/x_2,\ \ldots,\ \text{cd}(x_N,k)/x_N\}\,. \quad (10)$$

The introduced notions will be used in the next section to define strength, certainty, and coverage factors of a decision rule.

3 Flow Graphs

The idea of using flow graphs in the framework of rough sets, for discovering the statistical properties of crisp decision algorithms, was proposed by Pawlak [5,6,7]. We want to utilize and extend this concept with the aim of applying flow graphs to analysis of fuzzy information systems. First, we recall basic notions of the crisp flow graph approach.

A flow graph is given in the form of directed acyclic final graph $G = (\mathcal{N}, \mathcal{B}, \varphi)$, where $\mathcal{N}$ is a set of nodes, $\mathcal{B} \subseteq \mathcal{N} \times \mathcal{N}$, is a set of directed branches, $\varphi\colon \mathcal{B} \to \mathrm{R}^+$ is a flow function with values in the set of non-negative reals R^+.

For any $(X,Y) \in \mathcal{B}$, X is an input of Y and Y is an output of X. The quantity $\varphi(X,Y)$ is called the throughflow from X to Y.

$I(X)$ and $O(X)$ denote an input and an output of X, respectively. The input $I(G)$ and output $O(G)$ of a graph G are defined by

$$I(G) = \{X \in \mathcal{N}\colon\ I(X) = \emptyset\}\,, \qquad O(G) = \{X \in \mathcal{N}\colon\ O(X) = \emptyset\}\,. \quad (11)$$

Every node $X \in \mathcal{N}$ of a flow graph G is characterized by its inflow

$$\varphi_+(X) = \sum_{Y \in I(X)} \varphi(Y,X)\,, \quad (12)$$

and by its outflow

$$\varphi_-(X) = \sum_{Y \in O(X)} \varphi(X,Y)\,. \quad (13)$$

For any internal node X, the equality $\varphi_+(X) = \varphi_-(X) = \varphi(X)$ is satisfied. The quantity $\varphi(X)$ is called the flow of the node X.

The flow for the whole graph G is defined by

$$\varphi(G) = \sum_{x \in I(G)} \varphi_-(X) = \sum_{x \in O(G)} \varphi_+(X)\,. \quad (14)$$

By using the flow $\varphi(G)$, the normalized troughflow $\sigma(X,Y)$ and the normalized flow $\sigma(X)$ are determined as follows

$$\sigma(X,Y) = \frac{\varphi(X,Y)}{\varphi(G)}\,, \qquad \sigma(X) = \frac{\varphi(X)}{\varphi(G)}\,. \quad (15)$$

For every branch of a flow graph G the certainty factor is defined by

$$\mathrm{cer}(X,Y) = \frac{\sigma(X,Y)}{\sigma(X)}\,. \tag{16}$$

The coverage factor for every branch of a flow graph G is defined by

$$\mathrm{cov}(X,Y) = \frac{\sigma(X,Y)}{\sigma(Y)}\,. \tag{17}$$

The certainty and coverage factors satisfy the following properties

$$\sum_{Y\in O(X)} \mathrm{cer}(X,Y) = 1\,, \qquad \sum_{X\in I(Y)} \mathrm{cov}(X,Y) = 1\,. \tag{18}$$

The measures (16) and (17) are useful for analysis of decision algorithms [2].

Let us now consider the possibility of applying flow graphs in the case of fuzzy decision algorithms. Any decision table with fuzzy attributes can be conveniently expressed as a flow graph. We can assume, without losing the generality of consideration, that only one decision attribute will be used. Each attribute is represented by a layer of nodes. The nodes of the input and hidden layers correspond to linguistic values of the condition attributes, whereas the output layer nodes correspond to linguistic values of the decision attribute.

Let us denote by $\tilde{X}$ a fuzzy set on the universe U, which describes membership degree of particular elements $x \in U$ in the linguistic value represented by X. The membership degrees of all x in the set $\tilde{X}$ can be found in a respective column of the considered decision table.

The flow $\varphi(X,Y)$ for the branch (X,Y) is equal to power (fuzzy cardinality) of the product of fuzzy sets $\tilde{X}$ and $\tilde{Y}$. However, the T-norm operator `prod` should be used for determining the product of sets, in order to satisfy the following equation for the input and internal layer nodes

$$\varphi_-(X) = \mathrm{power}(\tilde{X}) = \sum_{Y\in O(X)} \varphi(X,Y) = \sum_{Y\in O(X)} \mathrm{power}(\tilde{X}\cap\tilde{Y})\,. \tag{19}$$

An analogous equation can be given for the output and internal layer nodes

$$\varphi_+(X) = \mathrm{power}(\tilde{X}) = \sum_{Y\in I(X)} \varphi(Y,X) = \sum_{Y\in I(X)} \mathrm{power}(\tilde{X}\cap\tilde{Y})\,. \tag{20}$$

Similarly, the equality $\varphi_+(X) = \varphi_-(X) = \varphi(X)$ is satisfied for any internal node X, when the T-norm operator `prod` is used. The above equations do not hold in general, if we use another T-norm operator, e.g. `min`. This is because the total normalized inflow (outflow) of each layer does depend on the form of T-norm operator used in calculations. In order to satisfy (19) and (20), the total normalized inflow (outflow) of a layer should be equal to 1. By applying the property (3), we can show that this is fulfilled, when we choose the T-norm operator `prod`.

In the special case of crisp decision tables, the formulae (19) and (20) become equivalent to (12) and (13).

The input and hidden layers of the flow graph can be merged into a single layer, which contains nodes representing all possible combinations of linguistic values of the condition attributes. Let us denote by X^* a node of the resulting layer. The node X^* corresponds to antecedent of a certain decision rule R_k. Support of the antecedent of the decision rule R_k is determined by using (8).

The decision rule R_k is represented by a branch (X^*,Y), where Y denotes a node of the output layer.

Power of the support of the rule R_k, defined by (10), is equal to the flow between the nodes X^* and Y

$$\varphi(X^*, Y) = \text{power}(\text{support}(R_k)) \,. \tag{21}$$

By using the formulae (8), (9) and (10), we can determine, for every decision rule R_k, the certainty factor $\text{cer}(X^*, Y)$, coverage factor $\text{cov}(X^*, Y)$, and strength of the rule $\sigma(X^*, Y)$

$$\text{cer}(X^*, Y) = \text{cer}(R_k) = \frac{\text{power}(\text{support}(R_k))}{\text{power}(\text{support}(\text{cda}(x, k)))} \,, \tag{22}$$

$$\text{cov}(X^*, Y) = \text{cov}(R_k) = \frac{\text{power}(\text{support}(R_k))}{\text{power}(\text{support}(\text{cdc}(x, k)))} \,, \tag{23}$$

$$\sigma(X^*, Y) = \text{strength}(R_k) = \frac{\text{power}(\text{support}(R_k))}{\text{card}(U)} \,. \tag{24}$$

Every decision rule can be represented by a sequence of nodes $[X_1 \ldots X_n]$, i.e. by a path from the 1-th to the n-th layer of the flow graph G. For a given path $[X_1 \ldots X_n]$, the resulting certainty and strength can be defined. In contrast to the definitions presented in [5,6,7], in which the statistical properties of flow are taken into account, we introduce an alternative form of the path's certainty and strength

$$\text{cer}[X_1 \ldots X_n] = \prod_{i=1}^{n-1} \text{cer}(X_1 \ldots X_i, X_{i+1}) \,, \tag{25}$$

$$\sigma[X_1 \ldots X_n] = \sigma(X_1)\, \text{cer}[X_1 \ldots X_n] \,, \tag{26}$$

where

$$\text{cer}(X_1 \ldots X_i, X_{i+1}) = \frac{\text{power}(\tilde{X}_1 \cap \tilde{X}_2 \cap \ldots \cap \tilde{X}_{i+1})}{\text{power}(\tilde{X}_1 \cap \tilde{X}_2 \cap \ldots \cap \tilde{X}_i)} \,. \tag{27}$$

With the help of equation (25), we can determine what part of the flow of the starting node X_1 reaches the final node X_n, passing through all nodes of the path $[X_1 \ldots X_n]$.

4 Example

We consider a decision table with fuzzy attributes (Table 2). There are two condition attributes A and B, and one decision attribute D. All attributes have three linguistic values. We use the same labels for both the linguistic values of the attributes and the corresponding nodes of the flow graph. As we stated in previous section, we use the T-norm operator `prod` in our calculations.

Table 2. Decision table with fuzzy attributes

	A			B			D		
	A_1	A_2	A_3	B_1	B_2	B_3	D_1	D_2	D_3
x_1	0.2	0.8	0.0	0.0	0.9	0.1	0.0	0.9	0.1
x_2	0.9	0.1	0.0	1.0	0.0	0.0	0.0	0.1	0.9
x_3	0.0	0.2	0.8	0.0	0.1	0.9	0.9	0.1	0.0
x_4	0.2	0.8	0.0	0.0	0.8	0.2	0.0	1.0	0.0
x_5	0.0	0.8	0.2	0.9	0.1	0.0	0.0	0.1	0.9
x_6	0.9	0.1	0.0	0.0	0.2	0.8	1.0	0.0	0.0
x_7	0.1	0.9	0.0	0.0	0.9	0.1	0.1	0.9	0.0
x_8	0.0	0.1	0.9	0.8	0.2	0.0	0.0	0.0	1.0
x_9	0.0	0.1	0.9	0.0	0.1	0.9	0.9	0.1	0.0
x_{10}	0.1	0.9	0.0	0.2	0.8	0.0	0.0	1.0	0.0

The values of normalized flow, for nodes representing condition attributes, are given in Table 3. We can easy check that the flow conservation equations (19) and (20) are satisfied, for example,

$$\sigma_-(A_1) = \frac{\text{power}(\widetilde{A_1})}{\text{card}(U)} = \sum_{i=1}^{3} \sigma(A_1, B_i) = 0.240\,,$$

$$\sigma_+(B_1) = \frac{\text{power}(\widetilde{B_1})}{\text{card}(U)} = \sum_{i=1}^{3} \sigma(A_i, B_1) = 0.290\,.$$

In the next step, we merge the layers corresponding to condition attributes into a resulting layer, which represents all possible linguistic values in the antecedences of decision rules. We determine the degrees of satisfaction of the rules' antecedences for particular elements $x \in U$. For the antecedence represented by A_1B_1, we get:

$$\widetilde{A_1B_1} = \tilde{A}_1 \cap \tilde{B}_1 = \{\ 0.00/x_1,\ \ 0.90/x_2,\ \ 0.00/x_3,\ \ 0.00/x_4,\ \ 0.00/x_5,\ \ 0.00/x_6,\ \ 0.00/x_7,\ \ 0.00/x_8,\ \ 0.00/x_9,\ \ 0.02/x_{10}\},$$

$$\varphi(A_1, B_1) = \text{power}(\widetilde{A_1B_1}) = 0.92, \quad \sigma(A_1, B_1) = \frac{\varphi(A_1, B_1)}{\text{card}U} = 0.092.$$

Table 3. Normalized flow between nodes of condition attributes' layers

	$\sigma(A_i, B_j)$			
	B_1	B_2	B_3	Σ
A_1	0.092	0.069	0.079	0.240
A_2	0.108	0.304	0.068	0.480
A_3	0.090	0.037	0.153	0.280
Σ	0.290	0.410	0.300	1.000

Table 4. Normalized flow between resulting and output layer

	$\sigma(A_iB_j, D_k)$			
	D_1	D_2	D_3	Σ
A_1B_1	0.0000	0.0110	0.0810	0.0920
A_1B_2	0.0189	0.0483	0.0018	0.0690
A_1B_3	0.0721	0.0067	0.0002	0.0790
A_2B_1	0.0000	0.0262	0.0818	0.1080
A_2B_2	0.0128	0.2748	0.0164	0.3040
A_2B_3	0.0332	0.0340	0.0008	0.0680
A_3B_1	0.0000	0.0018	0.0882	0.0900
A_3B_2	0.0153	0.0019	0.0198	0.0370
A_3B_3	0.1377	0.0153	0.0000	0.1530
Σ	0.2900	0.4200	0.2900	1.0000

Table 5. Certainty factor for branches between resulting and output layer

	$\mathrm{cer}(A_iB_j, D_k)$			
	D_1	D_2	D_3	Σ
A_1B_1	0.0000	0.1196	0.8804	1.00
A_1B_2	0.2740	0.7000	0.0260	1.00
A_1B_3	0.9127	0.0848	0.0025	1.00
A_2B_1	0.0000	0.2426	0.7574	1.00
A_2B_2	0.0421	0.9039	0.0539	1.00
A_2B_3	0.4882	0.5000	0.0118	1.00
A_3B_1	0.0000	0.0200	0.9800	1.00
A_3B_2	0.4140	0.0510	0.5350	1.00
A_3B_3	0.9000	0.1000	0.0000	1.00

The results of calculation of normalized flow between nodes of the resulting layer and nodes of the output layer are given in Table 4. The values of normalized outflow $\sigma_-(A_iB_j)$, $i, j \in \{1, 2, 3\}$, (column $\sum$ in Table 4) are equal to the respective values of normalized troughflow $\sigma(A_i, B_j)$, given in Table 3, e.g.,

Table 6. Coverage factor for branches between resulting and output layer

	$\mathrm{cov}(A_iB_j, D_k)$		
	D_1	D_2	D_3
A_1B_1	0.0000	0.0262	0.2793
A_1B_2	0.0652	0.1150	0.0062
A_1B_3	0.2486	0.0159	0.0007
A_2B_1	0.0000	0.0624	0.2821
A_2B_2	0.0441	0.6543	0.0566
A_2B_3	0.1145	0.0810	0.0028
A_3B_1	0.0000	0.0043	0.3040
A_3B_2	0.0528	0.0045	0.0683
A_3B_3	0.4748	0.0364	0.0000
Σ	1.0000	1.0000	1.0000

Table 7. Decision rules with the largest value of certainty factor

decision rule	certainty	coverage	strength [%]
$A_1B_1 \rightarrow D_3$	0.8804	0.2793	8.10
$A_1B_2 \rightarrow D_2$	0.7000	0.1150	4.83
$A_1B_3 \rightarrow D_1$	0.9127	0.2486	7.21
$A_2B_1 \rightarrow D_3$	0.7574	0.2821	8.18
$A_2B_2 \rightarrow D_2$	0.9039	0.6543	27.48
$A_3B_1 \rightarrow D_3$	0.9800	0.3040	8.82
$A_3B_3 \rightarrow D_1$	0.9000	0.4748	13.77

$$\sigma_{-}(A_1B_1) = \sigma(A_1, B_1) = \sum_{j=1}^{3} \sigma(A_1B_1, D_j) = 0.0920\,.$$

Thus, the flow conservation equations are satisfied. This is due to applying the T-norm operator `prod`.

For branches connecting the resulting and output layers, the certainty and coverage factors are determined according to (16) and (17). The results are given in Tables 5 and 6. They correspond to certainty and coverage factors of the decision rules $A_iB_j \rightarrow D_k$, $i, j, k \in \{1, 2, 3\}$, expressed by the formulae (22) and (23). For example, $\mathrm{cer}(A_1B_1, D_3) = 0.8804$ means that 88.04% of outflow from the node A_1B_1 reaches the decision node D_3, $\mathrm{cov}(A_1B_1, D_3) = 0.2793$ means that 27.93% of inflow to the decision node D_3 comes from the node A_1B_1.

Another important measure is the strength of decision rule expressed by (24). For example, the strength of the rule $A_1B_1 \rightarrow D_3$ is equal to 8.1%. We can say that the troughflow (A_1B_1, D_3) constitutes 8.1% of the total flow of the considered graph.

Fuzzy decision rules with the largest values of certainty factor (Table 7) can be included in the final fuzzy inference system. The respective values of coverage factor are useful for explaining these decision rules.

5 Conclusions

The proposed approach to fuzzy flow graphs is suitable for representing and analyzing decision tables with fuzzy attributes. Every layer of a flow graph corresponds to a particular attribute, and all nodes of a layer correspond to linguistic values of the attribute. For calculating the flow between nodes, the T-norm operator `prod` was chosen in order to satisfy the flow conservation equations. New definitions of the path's certainty and strength were given with the aim to correctly determine the change of the original flow along the paths. Future work should consider the problem of generating optimal flow graphs, by taking into account the properties of crisp or fuzzy decision tables (e.g. significance of attributes). This can be done by applying the methods of the rough sets theory. In particular, the variable precision fuzzy rough sets model seems to be a promising tool for reduction of fuzzy flow graphs.

References

1. Greco, S., Pawlak, Z., Słowiński, R.: Generalized Decision Algorithms, Rough Inference Rules, and Flow Graphs. In: Alpigini, J., Peters, J.F., Skowron, A., Zhong, N., (eds.): Rough Sets and Current Trends in Computing. Lecture Notes in Artificial Intelligence, Vol. 2475. Springer-Verlag, Berlin Heidelberg New York (2002) 93–104
2. Greco, S., Pawlak, Z., Słowiński, R.: Bayesian Confirmation Measures within Rough Set Approach. In: Tsumoto, S., et al., (eds.): Rough Sets and Current Trends in Computing. Lecture Notes in Artificial Intelligence, Vol. 3066. Springer-Verlag, Berlin Heidelberg New York (2004) 264–273
3. Klir, G.J., Folger, T.A.: Fuzzy Sets, Uncertainty, and Information. Prentice Hall, Englewood, New Jersey (1988)
4. Mieszkowicz-Rolka, A., Rolka, L.: Variable Precision Fuzzy Rough Sets Model in the Analysis of Process Data. [8] 354–363
5. Pawlak, Z.: Decision Algorithms, Bayes' Theorem and Flow Graphs. In: Rutkowski, L., Kacprzyk, J., (eds.): Advances in Soft Computing. Physica-Verlag, Heidelberg (2003) 18–24
6. Pawlak, Z.: Flow Graphs and Data Mining. In: Peters, J.F., et al., (eds.): Transactions on Rough Sets III. Lecture Notes in Computer Science (Journal Subline), Vol. 3400. Springer-Verlag, Berlin Heidelberg New York (2005) 1–36
7. Pawlak, Z.: Rough Sets and Flow Graphs. [8] 1–11
8. Ślęzak, D., et al., (eds.): Rough Sets and Current Trends in Computing. Lecture Notes in Artificial Intelligence, Vol. 3641. Springer-Verlag, Berlin Heidelberg New York (2005)
9. Yager, R.R., Filev, D.P.: Essentials of Fuzzy Modelling and Control. John Wiley & Sons, Inc., New York (1994)

Elements of the Type-2 Semantics in Summarizing Databases

Adam Niewiadomski and Michał Bartyzel

Institute of Computer Science, Technical University of Łódź
ul. Wólczańska 215, 93-005, Łódź, Poland
aniewiadomski@ics.p.lodz.pl

Abstract. Modern and effective methods of knowledge extraction from databases and information systems are required to provide rather linguistic than numerical information. The so-called *linguistic summaries of databases* by Yager [1], exemplified by *Many children like sweet ice cream*, and further improvements by George and Srikanth [2] and by Kacprzyk and Yager [3], are discussed in this paper. The use of type-2 fuzzy sets is an original contribution to the domain, since only ordinary fuzzy sets have been originally employed. Elements of type-2 semantics are shown to handle effectively pieces of imprecise information (e.g. fuzzy sets) stored in databases. An application on sample data is provided.

1 Type-2 Fuzzy Sets

The type-1 fuzzy sets, i.e. the Zadeh sets, are 40 years old [4]. Thanks to their semantics it is possible to create models of uncertain pieces of knowledge and information, like *big car*, *heavy ship*, etc. The so-called *full, partial*, or *none belongingness* of an element to a fuzzy set A in a universe of discourse $\mathcal{X}$ is determined via *a membership function* $\mu_A(x)\colon \mathcal{X} \to [0,1]$ (in the example below, $\mathcal{X}$ is discreet).

$$\begin{pmatrix} x_1, & x_2, & \ldots, & x_n \\ \mu(x_1), & \mu(x_2), & \ldots, & \mu(x_n) \end{pmatrix} \tag{1}$$

Type-2 fuzzy sets were mentioned, at first, by Zadeh, too. Not many works have appear since there, e.g. [5,6,7], until Karnik and Mendel presented the results of their research on theory and applications [8,9,10]. The crucial extension of Zadeh's idea is that type-2 membership levels are fuzzy sets themselves, in contrary to traditional membership levels which are crisp numbers. Let $\mathcal{F}(Y)$ be the set of all fuzzy sets over the universe Y. A type-2 fuzzy set $\tilde{A}$ is a collection of ordered pairs $< x, \tilde{\mu}(x,u) >$, where $x \in \mathcal{X}$ and $u \in [0,1]$ – *a primary membership level* for x. Hence, $\tilde{\mu}\colon \mathcal{X} \to \mathcal{F}([0,1])$:

$$\begin{pmatrix} x_1 & & x_2 & & \cdots & x_n \\ \begin{pmatrix} u_1(x_1) & \cdots & u_n(x_1) \\ \mu_{x_1}(u_1) & \cdots & \mu_{x_1}(u_n) \end{pmatrix} & \begin{pmatrix} u_1(x_2) & \cdots & u_n(x_2) \\ \mu_{x_2}(u_1) & \cdots & \mu_{x_2}(u_n) \end{pmatrix} & \cdots & \begin{pmatrix} u_1(x_n) & \cdots & u_n(x_n) \\ \mu_{x_n}(u_1) & \cdots & \mu_{x_n}(u_n) \end{pmatrix} \end{pmatrix} \tag{2}$$

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 278–287, 2006.
© Springer-Verlag Berlin Heidelberg 2006

where $u_j(x_i) \in J_x$ is a primary and $\mu_{x_i}(u_j)$ is a secondary membership, and $J_x \subseteq [0,1]$ – the set of primary membership degrees of $x \in \mathcal{X}$. Secondary membership levels may be viewed as *weights* or as *possibility levels*, cf. [9]. A membership function for a type-2 fuzzy set can be treated as a *fuzzy-valued function* [11]. Naturally, each type-1 set is a special case of a type-2 set.

Karnik and Mendel in [8,10] under the assumption that *different words can mean different things to different people* propose to increase the number of degrees of freedom in fuzzy logic systems. The very analogous situation from the field of the probability theory using higher-than-first-order moments is recalled in [8]:

> To just use the first-order moments [of probability density function (pdf)] would not be very useful, because random uncertainty requires understanding of dispersion about the mean and this information is provided by the variance. So, our accepted probabilistic modelling of random uncertainty focuses on (...) methods, that use *at least* the first two moments of a pdf.[1]

It must be emphasised, that the quotation **is not** the interpretation of fuzziness by means of probability, but only and simply the analogy which is for sure familiar to the reader. The conclusion given by the authors of [8] is that, although it is impossible to use an infinite number of degrees of freedom in both probabilistic and fuzzy manners of handling uncertainty, the introduction of at least one additional degree of freedom to fuzzy logic systems may provide a measure of dispersion for totally certain type-1 membership functions.

Various applications have been introduced [12,13,14] and useful comments and remarks on operations on fuzzy sets of type-2 [15] and on groups of fuzzy sets of type-2 (as Interval Type-2 Fuzzy Sets) have been given [16]. Also some works of de Korvin et al. concern rules in type-2 fuzzy logic systems [17]. Cardinality concepts for type-2 fuzzy sets are presented in [11].

2 Linguistic Summarization of Databases

A linguistic summary of a database $\mathcal{D} = \{d_1, d_2, \cdots, d_m\}$ is a semi-natural language sentence: Q P are/have S $[T]$, where Q is a quantity in agreement, expressed with the so-called *linguistic quantifier* represented by a fuzzy set [18], e.g. *more than half*, *about 100*; P is a subject of the summary, the set of objects the data of which are stored in the summarized base, e.g. *cars*, *workers*; S is a summarizer – a property of objects expressed with a fuzzy set, e.g. *big car*, *young worker*; T is a degree of truth of a summary, a real number from $[0,1]$. T is computed via

$$T = \mu_Q\left(\frac{\sum_{i=1}^{m}\mu_S(d_i)}{M}\right) \tag{3}$$

[1] [8], p.1.

where $\mu_S(d_i)$ is a membership level of the d_i record to S. In addition, m is a number of records in the base, and $M = m$ if Q is a relative fuzzy quantifier, cf. [18], or $M = 1$, if Q is absolute.

The presented in [2] improvement of the method for linguistic summarization, allows to generate summaries with *composite summarizers*: Q P are/have S_1 and S_2 and ... and S_n (e.g. *few cars are cheap and well-equipped*), where the *and* connective is modelled by the minimum operation:

$$\mu_S(d_i) = \min_{j=1,..,n} \mu_{S_j}(d_i) \tag{4}$$

and the method for computing the T index is analogous to (3).

The method presented in (4) can influence, however, on the high computational cost of a summary, especially when m and n are large numbers. The algorithm which helps to reduce this cost is presented by Kacprzyk and Yager [3] and Kacprzyk, Yager and Zadrożny [19], and is based on a preselected feature, the so-called *query*, denoted as w_g, $g = 1, ..., n$. The task of this method is to summarize these records only, which manifest w_g at a non-zero grade. The μ_S function is given as

$$\mu_S(d_i) = \min_{j=1,2,...,n} \left\{ \mu_{S_j}(d_i) \; t \; \mu_{w_g}(d_i) \right\}, i = 1, 2, ..., m \tag{5}$$

and T is computed as

$$T = \mu_Q \left(\frac{\sum_{i=1}^{m'} \mu_S(d_i)}{\sum_{i=1}^{m'} \mu_{w_g}(d_i)} \right) \tag{6}$$

Since the denominator in (6) is a cardinality of the w_g fuzzy set, Q must be relative, cf. [3]. Additionally, it must be explained, that it is necessary to preselect a database $\mathcal{D}' \subseteq \mathcal{D}$ consisting of those records d_i only for which $\mu_{w_g}(d_i) > 0$, and $|\mathcal{D}'| = m'$; otherwise, the computation via (5) and (6) would be more, instead of less, costly.

3 Type-2 Fuzzy Sets in Summarization

3.1 Type-2 Linguistic Variables

The concept of *linguistic variable* was introduced by Zadeh in 1975 [20]. An L linguistic variable is defined as an ordered quintuple $< L, H, \mathcal{X}, G, M >$. H, or $H(L)$, is a set of linguistic values of L – these values are modelled by fuzzy sets in $\mathcal{X}$ generated according to G (syntactic) and M (semantic) rules.

The main reason for extending the idea of linguistic variable is that ordinary fuzzy sets describe imprecise information with crisp numbers, which may be seen contradictory. Moreover, in many situations membership degrees are described by people with words again, instead of crisp numbers. Hence, the type-2-fuzzy-set-based extension of linguistic variable is defined:

Definition 1. *A type-2 linguistic variable is an ordered quintuple* $< L, H, \mathcal{X}, G, M >$*, where:*

L is the name of the variable,
H or H(L) is the term-set of linguistic values of L,
$\mathcal{X}$ is the universe of discourse,
G is a syntactic rule which generates the terms (labels) in L,
M is a semantic rule which associates a term from L with a type-2 fuzzy set in $\mathcal{X}$.

The definition is a generalization of the Zadeh definition, since a type-2 set is a generalization of a type-1 set. In particular, in linguistic summarization, at least three possibilities of employing type-2 linguistic variable may be enumerated: 1) as a summarizer, 2) as a linguistic quantifier (in this case $\mathcal{X} \subseteq \mathcal{R}^+ \cup \{0\}$), and 3) as a query w_g. The idea of 1) and 2) cases has already been introduced in [21], nevertheless, this paper additionally provides technical details and a real application.

The model of the AND connective for two or more labels associated to a type-2 fuzzy set each, can be found via the *meet* operation on their their membership functions, which produces the intersection for type-2 fuzzy sets [15]:

$$\mu_{\tilde{A}\cap\tilde{B}}(x) = \mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x) = \int_{u_{\tilde{A}}} \int_{u_{\tilde{B}}} \frac{\mu_{\tilde{A}}(x, u_{\tilde{A}}) \; t_1 \; \mu_{\tilde{B}}(x, u_{\tilde{B}})}{u_{\tilde{A}} \; t_2 \; u_{\tilde{B}}} \tag{7}$$

where $u_{\tilde{A}}$, $u_{\tilde{B}}$ are primary membership degrees for x in $\tilde{A}$, $\tilde{B}$, respectively, and t_1, t_2 – t-norms (in a discreet $\mathcal{X}$, integrals can be replaced by summations). The result is interpreted as $\tilde{A}$ AND $\tilde{B}$, e.g. INTELLIGENT AND GOOD-LOOKING.

Similarly, the union is defined by means of *the join* of membership functions of type-2; it is denoted by $\mu_{\tilde{A}}(x, u_{\tilde{A}}) \sqcup \mu_{\tilde{B}}(x, u_{\tilde{B}})$, $x \in \mathcal{X}$ and expressed as

$$\mu_{\tilde{A}\cup\tilde{B}}(x) = \mu_{\tilde{A}}(x, u_{\tilde{A}}) \sqcup \mu_{\tilde{B}}(x, u_{\tilde{B}}) = \int_{u_{\tilde{A}}} \int_{u_{\tilde{B}}} \frac{\mu_{\tilde{A}}(x, u_{\tilde{A}}) \; t_1 \; \mu_{\tilde{B}}(x, u_{\tilde{B}})}{u_{\tilde{A}} \; s \; u_{\tilde{B}}} \tag{8}$$

The complement of $\tilde{A}$ is denoted by $\tilde{A}^c$ and defined as

$$\mu_{\tilde{A}^c}(x) = \int_{u_{\tilde{A}}} \frac{\mu_{\tilde{A}}(x, u_{\tilde{A}})}{1 - u_{\tilde{A}}} \tag{9}$$

Equations (7), (8), and (9) are valid also for type-1 fuzzy sets, since they are special cases of type-2 fuzzy sets. In that case, formula (8) is rewritten as

$$\mu_{A\cup B}(x) = \int_{\mathcal{X}} \frac{1}{u_A \; s \; u_B} \tag{10}$$

and (7), (9) – analogously, because the primary membership levels u_A and u_B are single numbers which describe grades of belonging of x to the ordinary fuzzy sets A and B, respectively, and therefore the secondary membership levels are replaced by unities in both sets. The forms of intersection, union, and complement for fuzzy sets of type-1, at the same time as the analogous operations for

interval-valued fuzzy sets are obtained as special cases of formulae (8)–(9). The operations are applied as models for the OR, AND and NOT connectives respectively, in representing linguistic information.

If only common operations on type-1 and type-2 summarizers are needed, we propose to express an ordinary fuzzy set A as a type-2 fuzzy set $\tilde{A}$ in which all secondary degrees equal 1:

$$A = \int_{x \in \mathcal{X}} \frac{\mu_A(x)}{x} \longrightarrow \tilde{A} = \int_{x \in \mathcal{X}} \int_{u \in J_x} \frac{1}{x, u(x)} \tag{11}$$

where $\mu_A(x) = u(x)$.

3.2 Type-1 Fuzzy Quantification of Type-2 Fuzzy Propositions

The canonical forms of linguistically quantified propositions are presented by Zadeh in [18]. Let us extend the first and the second canonical forms with the use of type-2 fuzzy sets as a representation of imprecise linguistic expressions:

$$Q \text{ objects are } \widetilde{S_1} \tag{12}$$

and

$$Q \text{ objects being } \widetilde{S_2} \text{ are } \widetilde{S_1} \tag{13}$$

where $\widetilde{S}_1$ and $\widetilde{S}_2$ are the labels associated with type-2 fuzzy sets in a crisp universe of discourse $\mathcal{X}$, and Q is a linguistic quantifier represented by a type-1 fuzzy set. Degrees of truth of propositions (12) and (13) are real numbers computed as

$$T\left(Q \text{ objects are } \widetilde{S_1} \right) = \mu_Q \left(\frac{card(\widetilde{S_1})}{M} \right) \tag{14}$$

where $card(\widetilde{S_1})$ is a real number – the result of (17), $M = card(\mathcal{X})$ if Q is relative, or $M = 1$ if Q is absolute, and

$$T\left(Q \text{ objects being } \widetilde{S_2} \text{ are } \widetilde{S_1} \right) = \mu_Q \left(\frac{card(\widetilde{S_1} \cap \widetilde{S_2})}{card(\widetilde{S_2})} \right) \tag{15}$$

where the $\widetilde{S_1} \cap \widetilde{S_2}$ intersection is computed via (7). Similarly to the propositions represented by type-1 fuzzy sets, only relative quantification is possible in (13), and $\widetilde{S_2}$ can be interpreted as the importance.

Finally, it must be noticed that quantified propositions (14) and (15) are reduced to the forms proposed by Zadeh, if $\widetilde{S_1}$ and $\widetilde{S_2}$ are type-1 fuzzy sets.

3.3 Type-2 Summarizers and Their Cardinalities

Some basic ideas and implementations of extended summarization methods have already been given by the authors [21,22,23,24]. The quoted publications concern, however, mostly interval-valued fuzzy sets [5,6,7] with interval-valued cardinalities and compatibility levels. The approach presented in this paper is a bit

different from those earlier attempts, since it concerns type-2 fuzzy sets in general, and, even if an interval-valued fuzzy set may be seen as an interval type-2 fuzzy set, its cardinality and membership values are considered as reals.

When a type-2 summarizer is used, the algorithm for computing the T index must be enriched with a cardinality of a type-2 fuzzy set (which represents a summarizer). Jang and Ralescu point to at least two possible forms of cardinalities of type-2 fuzzy sets: a fuzzy set or a crisp number [11]. The former variant requires much more complicated computations, therefore, the latter is chosen, since the ordinary fuzzy quantifiers are employed in summarization. In particular, we use *non-fuzzy sigma count*, which produces the cardinality of a type-2 fuzzy set $\tilde{A}$ as a real number. This approach assumes that values of a type-2 membership function are fuzzy numbers with membership functions in the form of:

$$\mu_{\tilde{A}}(x_i, u_i) = \begin{cases} 1, & x_i \in [r_i, s_i] \neq \emptyset \\ L_i(u_i), & x_i < r_i \\ R_i(u_i), & x_i > s_i \end{cases} \tag{16}$$

where $[r_i, s_i]$ are closed intervals, L_i are continuous and monotonically non-decrea-sing functions, and R_i are continuous and monotonically non-increasing functions. Non-fuzzy sigma count of $\tilde{A}$ (nfσ-count($\tilde{A}$)), where values of membership function of $\tilde{A}$ are given by (16) is considered as:

$$card(\widetilde{A}) = \text{nf}\sigma\text{-count}(\tilde{A}) = \sum_{x \in \mathcal{X}} \max\{u \in J_x \colon \mu_{\tilde{A}}(x, u) = 1\} \tag{17}$$

In particular, the max operation can be omitted, if $r_i = s_i$ in (16). In case of a summarizer composed of two or more type-2 fuzzy sets, see (4), (5) and (7), the *meet* operation on their membership functions, $\mu_{\tilde{A}}$ and $\mu_{\tilde{B}}$, must be performed:

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \sqcap \mu_{\tilde{B}}(x) \tag{18}$$

Let $\mu_{x,\tilde{A}}, \mu_{x,\tilde{B}}$ be the secondary membership functions of $\tilde{A}, \tilde{B}$, respectively, and $\theta_{\tilde{A}}, \theta_{\tilde{B}}$ are primary memberships such a $\mu_{x,\tilde{A}}(\theta_{\tilde{A}}) = \mu_{x,\tilde{B}}(\theta_{\tilde{B}}) = 1$. Thus, secondary membership of the intersection $\tilde{A}$ and $\tilde{B}$ for given primary membership grades:

$$\mu_{\tilde{A} \cap \tilde{B}}(x, \theta) = \begin{cases} \max\{\mu_{\tilde{A}}(x, \theta), \mu_{\tilde{B}}(x, \theta)\} & \theta < \theta_{\tilde{A}} \\ \mu_{\tilde{A}}(x, \theta) & \theta_{\tilde{A}} \leq \theta \leq \theta_{\tilde{B}} \\ \min\{\mu_{\tilde{A}}(x, \theta), \mu_{\tilde{B}}(x, \theta)\} & \theta > \theta_{\tilde{A}} \end{cases} \tag{19}$$

The cardinality of the intersection can be computed via replacing $\tilde{A}$ with $\tilde{A} \cap \tilde{B}$ in (16)–(18).

3.4 Degrees of Truth for Type-2 Summaries

We compute the degree of truth of a summary with a type-2 fuzzy summarizer $\tilde{S}$ as:

$$T\left(Q\ P \text{ are/have } \tilde{S}\right) = \mu_Q\left(\frac{\text{nf}\sigma\text{-count}(\tilde{S})}{M}\right) \tag{20}$$

where $\tilde{S}$ is a single or a composite summarizer, and $M = 1$ if Q is an absolute quantifier, or $M = m$, if Q is relative; cf. [1,18].

If a type-2 summary with a $\widetilde{w}_g$ query is to be obtained, its degree of truth is

$$T\left(Q\ P \text{ being/having } \widetilde{w}_g \text{ are/have } \tilde{S}\right) = \mu_Q\left(\frac{\text{nf}\sigma\text{-count}(\tilde{S} \cap \widetilde{w}_g)}{\text{nf}\sigma\text{-count}(\widetilde{w}_g)}\right) \tag{21}$$

which is valid for relative quantifiers only. Similarly to the summaries presented in [3,19], the second canonical form is to be a template for more sophisticated summaries, which additionally can be obtained at a lower computational cost, see Section 2.

4 Implementation: Summarizing Fuzzy Values

The database used in the experiment consists of 200 records which describe reviews of articles submitted to the scientific conference AWIC 2005. 6 attributes in the database, i.e. RELEVANCE, ORIGINALITY, SIGNIFICANCE; USEFULNESS, TECHNICAL SOUNDNESS, REFERENCE TO THE RELATED LITERATURE, and PRESENTATION are evaluated with linguistic values BAD, WEAK, FAIR, GOOD, EXCELLENT, and each of them is related to a fuzzy set in $[0, 1]$, e.g.

$$\mu_{\text{BAD}}(x) = \begin{cases} -10x + 1, & \text{if } x \in [0, 0.1] \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

The reason for modelling of scores via fuzzy sets, instead of crisp numbers (e.g. BAD=0, GOOD=0.8, etc.), is that the necessity of handling rather wide range of experts' intuitions on linguistic descriptions of quality of papers is usually observed. According to Mendel's explanation, that *different words mean different things for different people* [8,10], type-2 fuzzy sets provide much more flexible and "safe" notation of expressing the scores which are being given in rather subjective circumstances.

The degree of truth of the summary for a chosen $\tilde{S}$ (e.g. FAIR SIGNIFICANCE or EXCELLENT RELEVANCE AND ORIGINALITY) and with $\widetilde{w}_g = \emptyset$ is computed:

1. For each $d_i \in D$ compute the (fuzzy) value of $\mu_{\tilde{S}}(d_i)$ summarizer $\tilde{S}$

$$\tilde{S} = \begin{pmatrix} d_1 & d_2 & \dots & d_m \\ \mu_{\tilde{S}}(d_1) & \mu_{\tilde{S}}(d_2) & \dots & \mu_{\tilde{S}}(d_m) \end{pmatrix} \tag{23}$$

2. Determine Q and compute the degree of truth for "$Q\ P$ are $\tilde{S}$" via (20).

The membership of a record to the type-2 fuzzy set is a type-1 fuzzy set – the intersection of a value of a chosen attribute (fuzzy number) of the record and of the fuzzy set appearing as a value of summarizer's membership function, i.e.

$$\mu_{\tilde{S}}(d_i) = V_C(d_i) \cap N \tag{24}$$

where $V_C(d_i)$ is value of the V_C attribute for d_i, and $N \in \{$ BAD, WEAK, FAIR, GOOD, EXCELLENT$\}$. The sample results are:

ABOUT_HALF of papers are of the EXCELLENT RELEVANCE [0.81]
ABOUT_HALF of papers are of the GOOD ORIGINALITY [0.99]
FEW of papers are of the WEAK PRESENTATION [0.93]
ABOUT_HALF of papers are of the EXCELLENT USEFULNESS [0.97]

If $\widetilde{w}_g$ appears in the summary to be generated, the computation proceeds:

1. Determine $D' \subseteq D$ which collects only those records d_i for which $\mu_{\widetilde{w}_g}(d_i) \neq \emptyset$
2. For each $d_i \in D'$ compute $\mu_{\widetilde{S}}(d_i) \sqcap \mu_{\widetilde{w}_g}(d_i)$ via (7).
3. Compute T for a chosen relative quantifier Q via (21).

The sample results obtained this way are:

ALMOST_NONE of papers having FAIR relevance are of BAD ORIGINALITY [1.0]
ABOUT_HALF of papers having GOOD TECHNICAL SOUNDNESS have WEAK REFERENCES [0.87]

It must be said that the results would be impossible to obtain via the Yager summarization methods. Since it enables handling mostly crisp values, the fuzzy data on the papers, e.g. originality, technical soundness, etc., would have to be converted to crisp values, defuzzified, for instance, and the loss of naturalness in such processed experts (reviewers) opinions would be observed. Thus, the type-2 fuzzy sets applied to represent objects' features in linguistic summarization, provide more flexible, more general, and more human consistent manner of describing uncertain data.

5 Conclusions and Future Work

The chosen aspects of applying type-2 fuzzy sets to linguistic summarization of databases which contain fuzzy values, have been presented. In particular, the linguistic variable by Zadeh has been redefined based on type-2 fuzzy sets. Thanks to the used semantics, summarizing imprecise and subjective linguistic opinions is possible at the reasonable computational cost.

Another possibility is to represent linguistic quantifiers as type-2 fuzzy sets. The canonical forms (12) and (13) should be extended to

$$\widetilde{Q} \text{ objects are } \widetilde{S_1} \tag{25}$$

and

$$\widetilde{Q} \text{ objects being } \widetilde{S_2} \text{ are } \widetilde{S_1} \tag{26}$$

where $\widetilde{Q}$ is represented by a type-2 fuzzy set in $[0, 1]$. Next, the use of type-2 (instead of type-1) membership functions of $\widetilde{Q}$ will cause that the degrees of truth of these propositions are expressed by fuzzy numbers, instead of crisp, compare (14) and (15). Moreover, definitions of convex and normal type-2 fuzzy sets,

should be introduced, to provide a generalization which includes type-1 fuzzy quantification as a special case. Naturally, these future formulae must generalize (12) and (13).

The type-2-fuzzy-set-based extensions of other quality measures and applying these sets as fuzzy quantifiers are currently being developed by the authors.

References

1. Yager, R.R.: A new approach to the summarization of data. Information Sciences **28** (1982) 69–86
2. George, R., Srikanth, R.: Data summarization using genetic algorithms and fuzzy logic. In Herrera, F., Verdegay, J., eds.: Genetic Algorithms and Soft Computing. Physica–Verlag, Heidelberg (1996) 599–611
3. Kacprzyk, J., Yager, R.R.: Linguistic summaries of data using fuzzy logic. International Journal of General Systems **30** (2001) 133–154
4. Zadeh, L.A.: Fuzzy sets. Information and Control **8** (1965) 338–353
5. Turksen, I.B.: Interval-valued fuzzy sets based on normal forms. Fuzzy Sets and Systems (1986) 191–210
6. Gorzalczany, M.B.: A method of inference in approximate reasoning based on interval-valued fuzzy sets. Fuzzy Sets and Systems **21** (1987) 1–17
7. Gorzalczany, M.B.: An interval-valued fuzzy inference method in approximate reasoning. Fuzzy Sets and Systems **31** (1989) 243–251
8. Karnik, N.N., Mendel, J.M.: An Introduction to Type-2 Fuzzy Logic Systems. University of Southern California, Los Angeles (1998)
9. Karnik, N.N., Mendel, J.M.: Type-2 fuzzy logic systems. IEEE Transactions on Fuzzy Systems **7** (1999) 643–658
10. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, Upper Saddle River, NJ (2001)
11. Jang, L.C., Ralescu, D.: Cardinality concept for type-two fuzzy sets. Fuzzy Sets and Systems **118** (2001) 479–487
12. Liang, Q., Mendel, J.M.: Equalization of non-linear time-varying channels using type-2 fuzzy adaptive filters. IEEE Transactions on Fuzzy Systems **8** (2000) 551–563
13. de Tre, G., de Caluwe, R.: Level-2 fuzzy sets and their usefulness in object-oriented database modelling. Fuzzy Sets and Systems **140** (2003) 29–49
14. Wu, H., Mendel, J.M.: Uncertainty bounds and their use in the design of interval type–2 fuzzy logic systems. IEEE Transactions on Fuzzy Systems **10** (2002)) 622–639
15. Karnik, N.N., Mendel, J.M.: Operations on type-2 fuzzy sets. Fuzzy Sets and Systems **122** (2001) 327–348
16. Liang, Q., Mendel, J.M.: Interval type-2 fuzzy logic systems. theory and design. IEEE Transactions on Fuzzy Systems **8** (2000) 535–550
17. de Korvin, A., Hu, C., Sirisaengtaksin, O.: On firing rules of fuzzy sets of type ii. International J. of Applied Mathematics **3** (2000) 151–159
18. Zadeh, L.A.: A computational approach to fuzzy quantifiers in natural languages. Computers and Maths with Applications **9** (1983) 149–184
19. Kacprzyk, J., Yager, R.R., Zadrożny, S.: Fuzzy linguistic summaries of databases for an efficient business data analysis and decision support. In Abramowicz, W., Żurada, J., eds.: Discovery for Business Information Systems. Kluwer Academic Publisher B. V., Boston (2001) 129–152

20. Zadeh, L.A.: The concept of linguistic variable and its application for approximate reasoning (i). Information Sciences **8** (1975) 199–249
21. Niewiadomski, A.: On two possible roles of type-2 fuzzy sets in linguistic summaries. Lecture Notes on Artificial Intelligence **3528** (2005) 341–347
22. Niewiadomski, A.: Interval-valued linguistic variables. an application to linguistic summaries. In Hryniewicz, O., Kacprzyk, J., Koronacki, J., Wierzchoń, S.T., eds.: Issues in Intelligent Systems. Paradigms. EXIT Academic Press, Warsaw (2005) 167–184
23. Niewiadomski, A.: Interval-valued quality measures for linguistic summaries. In Grzegorzewski, P., Krawczak, M., Zadrożny, S., eds.: Issues in Soft Computing. Theory and Applications. EXIT Academic Press, Warsaw (2005) 211–224
24. Niewiadomski, A., Ochelska, J., Szczepaniak, P.S.: Interval-valued linguistic summaries of databases. Control and Cybernetics (2005) (in print)

Isolines of Statistical Information Criteria for Relational Neuro-fuzzy System Design*

Agata Pokropińska[1], Robert Nowicki[2,3], and Rafał Scherer[2,3]

[1] Institute of Mathematics and Computer Science, Jan Długosz University
al. Armii Krajowej 13/15, 42-200 Częstochowa, Poland
a.pokropinska@ajd.czest.pl
http://www.imi.ajd.czest.pl
[2] Department of Computer Engineering, Częstochowa University of Technology
Al. Armii Krajowej 36, 42-200 Częstochowa, Poland
rafal@ieee.org, rnowicki@kik.pcz.czest.pl
http://kik.pcz.pl
[3] Department of Artificial Intelligence
Academy of Humanities and Economics in Łódź
ul. Rewolucji 1905 nr 64, Łódź, Poland
http://www.wshe.lodz.pl

Abstract. The paper concerns designing relational neuro-fuzzy systems as a multicriteria optimization problem. Relational neuro-fuzzy systems have additional relation making rules to have more flexible form. A method for designing neuro-fuzzy systems by using information criteria and criteria isolines is used to find the optimal relational system for a given problem.

1 Introduction

There are many diferrent neuro-fuzzy systems [1][4][5][6][8][9][10][11][16]. Most of them are based on intelligible fuzzy rules. These fuzzy rules are obtained through expert knowledge, some heuristic methods or by learning from numerical data, using so called data-driven learning as in neuro-fuzzy systems. The structure of neuro-fuzzy systems is similar to a neural network but elements constituting the structure reflect fuzzy sets and operations performed on fuzzy sets like t-norms or t-conorms. Thanks to this, the system can be presented in a graphical form and it can be learned by a gradient algorithm. A learning algorithm can be used to approximate any n-dimensional function [16]. And yet the structure consists of fuzzy rules which are easy to extract unlike in the case of neural networks. In traditional fuzzy systems the input-output mapping is defined by the relation built from fuzzy rules and input and output fuzzy sets. Usually, output fuzzy sets are singleton sets in neuro-fuzzy systems. One kind of fuzzy systems are relational fuzzy systems [3][8][14], where there is an additional relation binding

* This work was supported in part by the Foundation for Polish Science (Professorial Grant 2005-2008) and the Polish State Committee for Scientific Research (Grant Nr T11C 04827).

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 288–296, 2006.
© Springer-Verlag Berlin Heidelberg 2006

input and output fuzzy sets. In this case we obtain fuzzy rules with additional weights that can be regarded as a kind of rule weights. In relational neuro-fuzzy systems [12][13] there is a possibility to train the system by the backpropagation algorithm. This kind of fuzzy systems has more flexible fuzzy rules which improves fitting the system for data. That is possible thanks to more adjustable parameters. Because the systems have more parameters it is very important to keep a good trade-off between the size of the system and its performance. In machine learning it is important to check after learning the performance of the system on testing and validating data set. Good results on a learning set do not guarantee satisfactory performance on testing or unseen before data. This ability to generalize data can be provided by maintaining suitable number of parameters in the learning system.

In the paper we propose the use information criteria [7] to evaluate the quality of relational neuro-fuzzy systems. We also use so called criteria isolines as a basis for system designing. The paper is organized as follows. In Section 2 relational fuzzy systems are described, and in Section 3 the idea of using information criteria for system evaluation is presented. Finally, in Section 4, we evaluate the system for function approximation for various numbers of parameters.

2 Fuzzy Relational Systems

Fuzzy relational systems can be perceived as extended Mamdani-type systems. In such systems fuzzy rules are more flexible because of the form of the consequents. Fuzzy rules in a MISO relational model have the following form

$$\begin{aligned} &R^k: \text{ IF } x_1 \text{ is } A_1^k, \ldots, \ x_n \text{ is } A_n^k, \text{ THEN} \\ &y \text{ is } B^1 \left(r_{k1}\right), y \text{ is } B^m \left(r_{km}\right), \ldots \\ &\ldots, y \text{ is } B^L \left(r_{kL}\right) \ , \end{aligned} \tag{1}$$

where A_i^k is an antecedent (input) fuzzy set of the k-th rule and the i-th input, B^m is a consequent (output) fuzzy set and r_{km} is a weight, responsible for the strength of connection between input and output fuzzy sets. Relational fuzzy systems store associations between the input and the output linguistic values, represented by sets A and B defined further in the text, in the form of a discrete fuzzy relation

$$\mathbf{R}\left(A, B\right) \in [0, 1] \ . \tag{2}$$

In case of a multi-input multi-output system (MIMO), the relation $\mathbf{R}$ is a matrix containing degree of connection for every possible combination of input and output fuzzy sets. If we would consider a fuzzy system with multidimensional input linguistic values, where input fuzzy sets are common for all classes, we would have only one set A of fuzzy linguistic values

$$A = \left\{A^1, A^2, ..., A^K\right\} \ . \tag{3}$$

Thus the relational matrix $\mathbf{R}$ is only two-dimensional. Output variable y has a set of L linguistic values B^m with membership functions $\mu_{B^m}\left(y\right)$, for $m = 1, ..., L$

$$B = \left\{B^1, B^2, ..., B^L\right\}. \tag{4}$$

Sets A and B are related to each other with a certain degree by the $K \times L$ binary relation matrix

$$\mathbf{R}_c = \begin{bmatrix} r_{11} & r_{11} & \cdots & r_{1M} \\ r_{21} & r_{22} & \cdots & r_{2M} \\ \vdots & \vdots & r_{km} & \vdots \\ r_{K1} & r_{K2} & \cdots & r_{KL} \end{bmatrix} . \tag{5}$$

For simplicity we use a system where sets A^k are replaced by t-norm of n input fuzzy sets A_i^k

$$\mu_{A^k} = \tau^k = \prod_{i=1}^{n} \mu_{A_i^k}(x_i) . \tag{6}$$

Having given vector $\bar{A}$ of K membership values $\mu_{A^k}(\bar{\mathbf{x}})$ for a crisp observed feature values $\bar{\mathbf{x}}$, vector $\bar{B}$ of L crisp memberships μ_m is obtained through a fuzzy relational composition

$$\bar{B} = \bar{A} \circ \mathbf{R} , \tag{7}$$

implemented element-wise by a generalized form of sup-min composition [1], i.e. s-t composition

$$\mu_{mc} = \underset{k=1}{\overset{K}{\mathrm{S}}} \left[\mathrm{T}\left(\mu_{A^k}(\bar{\mathbf{x}}), r_{km}\right) \right] . \tag{8}$$

The crisp output of the relational system is computed by the weighted mean

$$\bar{y} = \frac{\sum_{m=1}^{L} \left\{ \bar{y}^m \, \mathrm{S}_{k=1}^{K} \left[\mathrm{T}\left(\mu_{A^k}(\bar{\mathbf{x}}), r_{km}\right) \right] \right\}}{\sum_{m=1}^{L} \mathrm{S}_{k=1}^{K} \left[\mathrm{T}\left(\mu_{A^k}(\bar{\mathbf{x}}), r_{km}\right) \right]} , \tag{9}$$

where $\bar{y}^m$ is a centre of gravity (centroid) of the output fuzzy set B^m. The system is depicted in Figure 1. Antecedent fuzzy sets with t-norms (see (6)) are replaced in the figure with multidimensional fuzzy sets for clarity. Antecedent fuzzy sets have Gaussian membership function

$$\mu_{\mathrm{Gauss}}(x) = \exp\left(- \left(\frac{x - \bar{x}}{\sigma} \right)^2 \right) , \tag{10}$$

where $\bar{x}$ is responsible for its centre and σ responsible for its width. Using algebraic product T-norm and the arithmetic mean (the boundary case of the OWA operator [18][19]) as t-conorm in s-t composition, the output of the relational system becomes

$$\bar{y} = \frac{\sum_{m=1}^{M} \left\{ \bar{y}^m \frac{1}{K} \sum_{k=1}^{K} \left\{ \mu_{A^k}(\bar{\mathbf{x}}) \cdot r_{km} \right\} \right\}}{\sum_{m=1}^{L} \left\{ \frac{1}{K} \sum_{k=1}^{K} \left\{ \mu_{A^k}(\bar{\mathbf{x}}) \cdot r_{km} \right\} \right\}} , \tag{11}$$

and reducing K, we obtain

$$\bar{y} = \frac{\sum_{m=1}^{L} \left\{ \bar{y}^m \sum_{k=1}^{K} \left\{ \mu_{A^k}(\bar{\mathbf{x}}) \cdot r_{km} \right\} \right\}}{\sum_{m=1}^{L} \sum_{k=1}^{K} \left\{ \mu_{A^k}(\bar{\mathbf{x}}) \cdot r_{km} \right\}} , \tag{12}$$

and

$$\bar{y} = \frac{\sum_{m=1}^{M} \sum_{k=1}^{K} \{\bar{y}^m \cdot \mu_{A^k}(\bar{\mathbf{x}}) \cdot r_{km}\}}{\sum_{m=1}^{L} \sum_{k=1}^{K} \{\mu_{A^k}(\bar{\mathbf{x}}) \cdot r_{km}\}} \,. \tag{13}$$

This special case of OWA operators would further simplify the neuro-fuzzy relational system.

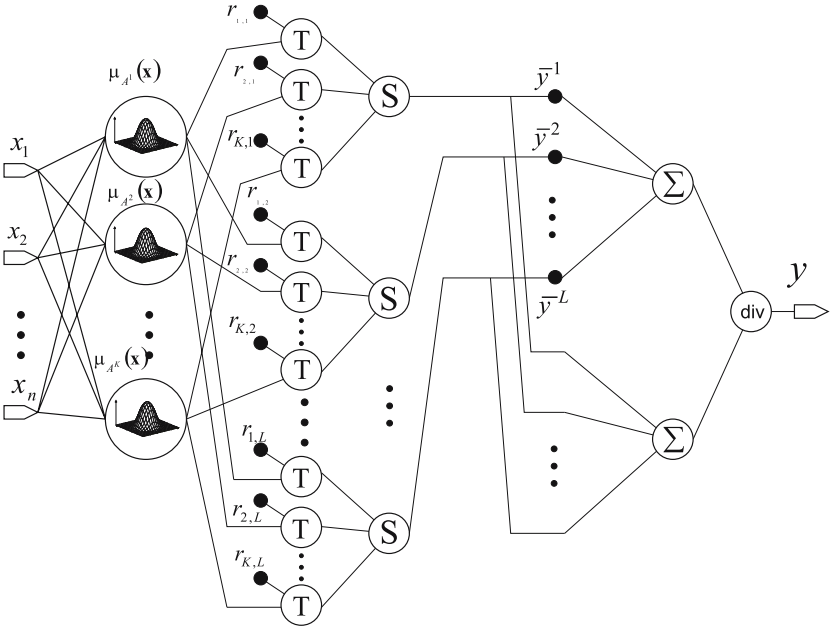


Fig. 1. Neuro-fuzzy relational system. Antecedent fuzzy sets (linguistic terms) with t-norms (see (6)) are replaced in the figure with multidimensional fuzzy sets for clarity.

3 Evaluating Relational System Quality

In case of regression tasks the most important factor is error Q. The most commonly used error measure is the root mean square error

$$RMSE = \sqrt{\frac{1}{M} \sum_{l=1}^{M} (d_l - y_l)^2} \,, \tag{14}$$

where M is the size of the learning set, d_l id the output value from the learning set and y_l is the system output. In case of classification the quality measure is the number of misclassified patterns. The evaluation of relational neuro-fuzzy systems will be made using some error measure against the number of adjustable parameters, which "store" the knowledge. The number of parameters in case of relational neuro-fuzzy systems can be computed as follows

$$p = 2nK + KL + L \,. \tag{15}$$

Increasing the number of parameters improves learning error but after some treshold the testing error gets higher. We have problem of searching for the optimal set utilizing two criteria p and Q. This leads to determining the whole set of optimal neuro-fuzzy systems. To evaluate the quality of neuro-fuzzy systems we use information criteria. The Akaike Information Criterion (AIC) is defined as follows

$$\mathrm{AIC}(p) = M \ln \hat{Q}_p^f + 2p\,, \tag{16}$$

where $\hat{Q}_p^f$ is an estimation of average square error of prediction, M is the size of the considered problem and p is the estimated order of autoregression. Final Prediction Error is defined by

$$\mathrm{FPE}(p) = \frac{M+p}{M-p}\hat{Q}_p^f\,. \tag{17}$$

The Schwarz criterion is defined as follows

$$S(p) = M \ln \hat{Q}_p^f + p \ln M\,. \tag{18}$$

The Södeström and Stoica criterion is defined as follows

$$H(p) = M \ln \hat{Q}_p^f + 2pc \log\left(\log M\right)\,, \tag{19}$$

where c is assumed 1. Seeing an analogy between autoregression and neuro-fuzzy systems we can treat p, M and Q_p^f as the number of parameters, the size of the problem and average error. The size of the problem was assumed as product of the learning dataset size and number of input variables. We use values of the information criteria to graph criteria isolines, where coordinate p is the number of the learning system parameters and coordinate Q is the testing RMSE error. Isolines show different values of criteria. Thanks to this it is easy to find an optimal relational neuro-fuzzy system. The next section shows the application of the information criteria to relational neuro-fuzzy selection for a given approximation problem.

4 Numerical Simulations

4.1 Nonlinear Dynamic Plant Problem (NDP)

We consider the second-order nonlinear plant from [17]

$$y\left(k\right) = g\left(y\left(k-1\right), y\left(k-2\right) + u\left(k\right)\right)\,, \tag{20}$$

$$g\left(y\left(k-1\right), y\left(k-2\right)\right) = \frac{y\left(k-1\right) y\left(k-2\right)\left(y\left(k-1\right)-0.5\right)}{1+y^2\left(k-1\right)+y^2\left(k-2\right)}\,, \tag{21}$$

and the goal is to approximate nonlinear component $g\left(y\left(k-1\right), y\left(k-2\right)\right)$ of the plant with a neuro-fuzzy relational model. In [17], 400 simulated data were generated from the plant model (2). Starting from the equilibrium state (0,0), 200 samples of identification data were obtained with a random input signal $u(k)$ uniformly distributed in $[-1.5, 1.5]$, followed by 200 samples of evaluation data obtained using a sinusoidal input signal $u\left(k\right) = \sin\left(2\pi k/25\right)$.

4.2 Nonlinear Function Aproximation Problem

Numerical simulations were carried out on a double-input and a single-output static system

$$y = \left(1 + x_1^{-2} + x_2^{-1,5}\right)^2, \quad 1 \leqslant x_1, x_2 \leqslant 5\ , \tag{22}$$

taken from [15]. From the evenly distributed grid point of the input range $x, y \in [1, 5]$ of the preceding equation, 50 training data pairs were obtained.

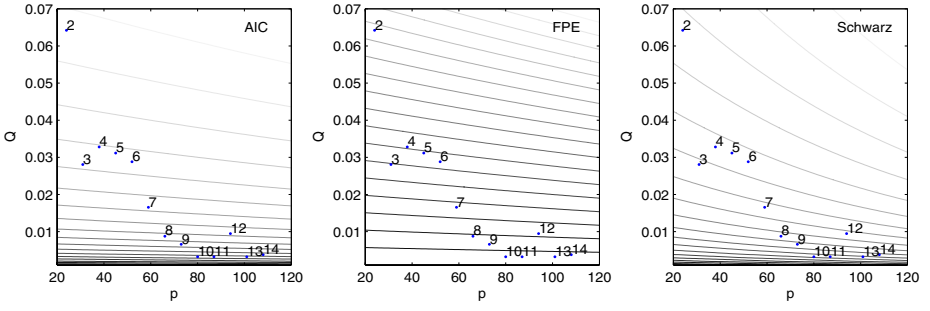


Fig. 2. Criteria isolines for NDP for L=3

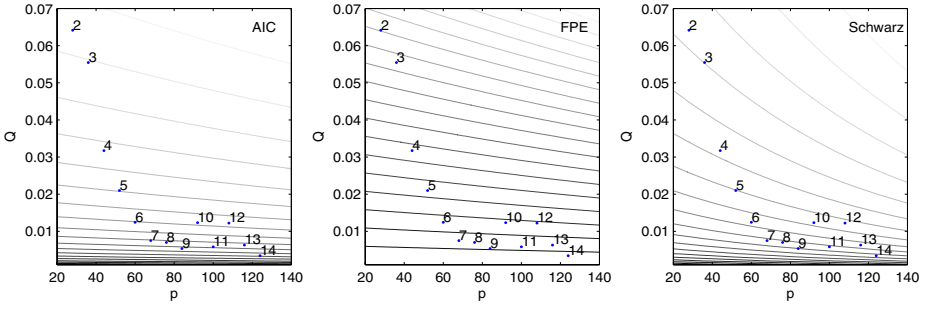


Fig. 3. Criteria isolines for NDP for L=4

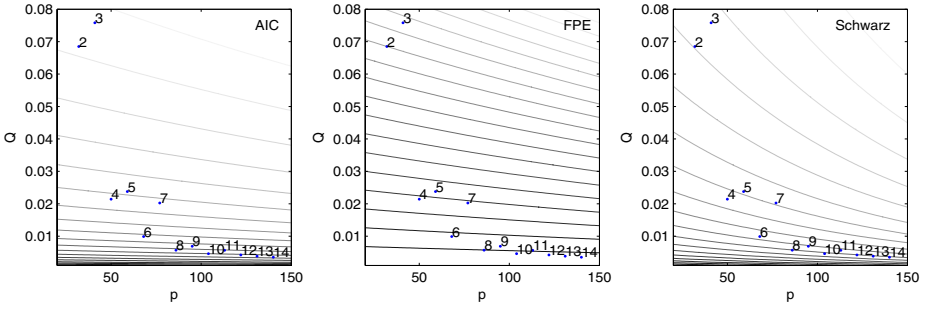


Fig. 4. Criteria isolines for NDP for L=5

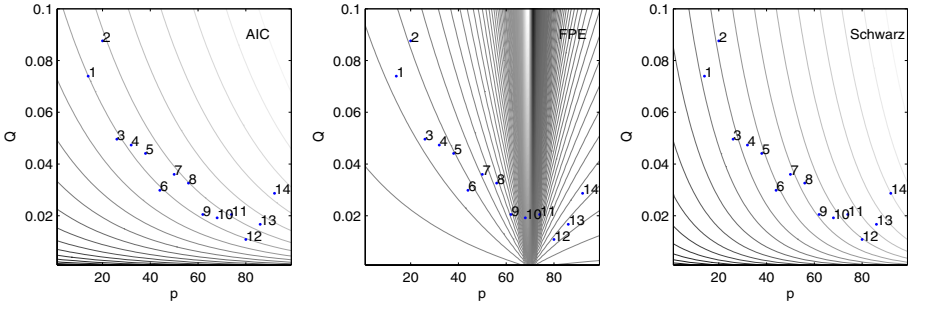


Fig. 5. Criteria isolines for nonlinear function aproximation for $L = 2$

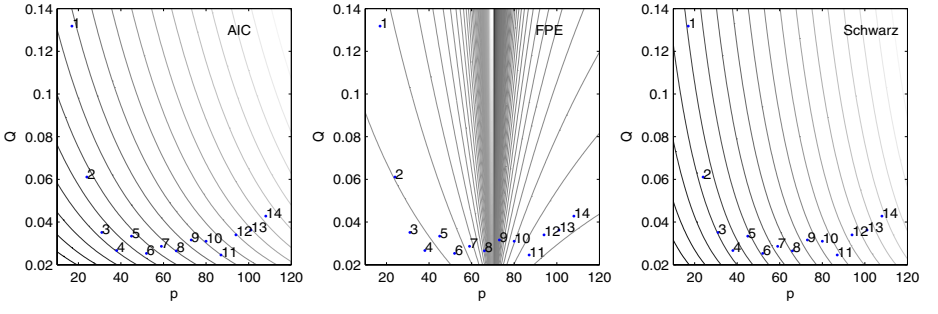


Fig. 6. Criteria isolines for nonlinear function aproximation for $L = 3$

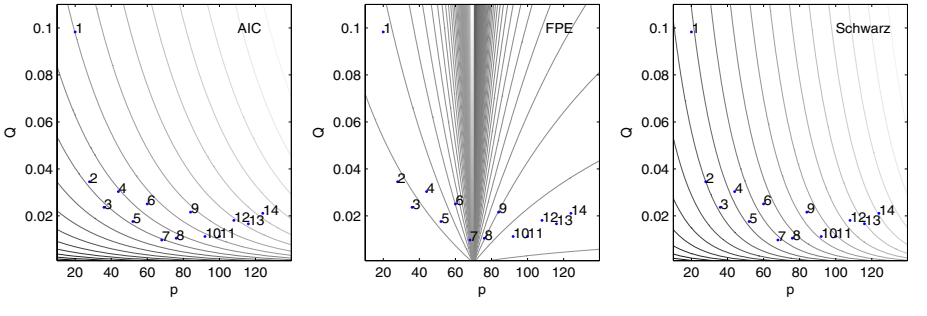


Fig. 7. Criteria isolines for nonlinear function aproximation for $L = 4$

All parameters were tuned by the backpropagation algorithm. By choosing the system with the smallest value of a criterion we obtain the system with the highest quality. Approximately all criteria choose systems with similar number of parameters.

5 Conclusions

In the paper information criteria and criteria isolines are used to design relational neuro-fuzzy systems. Rules in such systems are more flexible because of the additional weights in rule consequents. Because of the higher number of adjustable parameters it is beneficial to evaluate the systems against their size and performance on testing error. The system design was made on two well-known benchmarks.

References

1. R. Babuska, *Fuzzy Modeling For Control*, Kluwer Academic Press, Boston, 1998.
2. C.L. Blake, C.J. Merz, *UCI Repository of machine learning databases*, `www.ics.uci.edu/~mlearn/MLRepository.html`, Irvine, University of California, Department of Information and Computer Science, 1998.
3. P.J.C. Branco, J.A. Dente, "A Fuzzy Relational identification Algorithm and its Application to Predict the Behaviour of a Motor Drive System", *Fuzzy Sets and Systems*, vol. 109, pp. 343–354, 2000.
4. K. Cpałka, "A Flexible Connectionist System" in: *Lecture Notes in Computer Science* 3019, Parallel Processing and Applied Mathematics, Roman Wyrzykowski, Jack Dongarra, Marcin Paprzycki, Jerzy Waniewski (Eds.), Springer-Verlag NewYork, pp. 618-625, 2004.
5. R. J.-S. Jang, C.-T. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing*, A Computational Approach to Learning and Machine Intelligence, Prentice Hall, Upper Saddle River 1997.
6. D. Nauck, F. Klawon, R. Kruse, *Foundations of Neuro-Fuzzy Systems*, Chichester, U.K., John Wiley, 1997.
7. R. Nowicki, A. Pokropińska, "Information Criterions Applied to Neuro-Fuzzy Architectures Design", ICAISC 2004, Artificial Intelligence and Soft Computing, L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L. A. Zadeh (Eds.), LNAI 3070, Springer-Verlag Berlin Heidelberg, 332-337 (2004)
8. W. Pedrycz, *Fuzzy Control and Fuzzy Systems*, Research Studies Press, London, 1989.
9. L. Rutkowski, K. Cpałka, "Flexible neuro-fuzzy systems", *IEEE Transactions on Neural Networks*, vol. 14, May 2003, pp. 554-574.
10. L. Rutkowski, *Flexible Neuro-Fuzzy Systems*, Kluwer Academic Publishers, 2004.
11. L. Rutkowski and K. Cpałka, "Designing and learning of adjustable quasi triangular norms with applications to neuro fuzzy systems", *IEEE Trans. on Fuzzy Systems*, vol. 13, Feb 2005, pp. 140-151.
12. R. Scherer, L. Rutkowski, "Neuro-Fuzzy Relational Systems", Proc. of 2002 International Conference on Fuzzy Systems and Knowledge Discovery, November 18-22, Singapore (CD-ROM), 2002.
13. R. Scherer, L. Rutkowski, "Connectionist Fuzzy Relational Systems", in *Studies in Computational Intelligence, Computational Intelligence for Modelling and Control*, Edited by S. Hagamuge, L.P. Wang, Springer-Verlag, pp. 35–47, 2005.
14. M. Setness, R. Babuska, "Fuzzy Relational Classifier Trained by Fuzzy Clustering", *IEEE Transactions on Systems, Man and Cybernetics* - Part B: Cybernetics, Vol. 29, No. 5, October, 1999, pp.619–625.

15. M. Sugeno, T. Yasukawa, *A Fuzzy-Logic-Based Approach to Qualitative Modeling*, IEEE Transactions on Fuzzy Systems, vol. 1, no. 1, pp. 7-31, 1993.
16. L.-X. Wang, *Adaptive Fuzzy Systems And Control*, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1994.
17. L. Wang,J. Yen, "Application of statistical information criteria for optimal fuzzy model construction",*IEEE Transactions on Fuzzy Systems*, vol. 6, August (1998), pp. 353-362.
18. R.R. Yager, D.P. Filev, *Essentials of Fuzzy Modeling and Control*, John Wiley Sons, Inc., 1994.
19. R.R. Yager, D.P. Filev, *On a Flexible Structure for Fuzzy Systems Models* in Fuzzy Sets, Neural Networks, and Soft Computing, R.R. Yager, L.A. Zadeh, Eds.,Van Nostrand Reinhold, New York, pp.1-28, 1994.

Adjusting Software-Intensive Systems Developed by Using Software Factories and Fuzzy Features

Silva Robak[1] and Andrzej Pieczyński[2]

[1] Faculty of Mathematics, Computer Science and Econometrics
[2] Faculty of Electrical Engineering, Informatics and Telecommunication
Institute of Control and Computation Engineering,
University of Zielona Góra,
ul. Podgórna 50, 65-246 Zielona Góra, Poland
A.Pieczynski@issi.uz.zgora.pl, S.Robak@wmie.uz.zgora.pl
http://www.uz.zgora.pl

Abstract. Growing expectations for low priced software-intensive applications tailored to the individual customers needs require the usage of techniques aimed at the individual development of application systems, which could be attained at the costs of standard software and easily adapted. In the paper the concepts associated with assembling the applications by means of software factories in conjunction with the fuzzy description of the variable product parts are introduced. The approach is demonstrated in the example of a part of an order process.

1 Introduction

In the software engineering industry reuse of the pieces of software has been applied for several decades. Reuse of software components [18] in subsequent versions of a product has been known and applied for a long time in the development of one system at a time. The reuse of components in product versions and various products is the goal in the software product family approach [11]. This kind of component reuse has succeeded in several domains and is at present being addressed by the software product lines [1]. Software product line is a "set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way" [2]. Component reuse in product versions, various products and different organizations and domains, may also be addressed in the future. The contemporary developments in this field are concepts of software factories designated for assembling software-intensive applications with patterns [5], models, frameworks [7] and tools [3].

1.1 Software Factory

Software factory is a special case of a software product line: "A software factory is a software product line that configures extensible tools, processes, and contents using a software factory template based on a software factory schema to automate the development and maintenance of variants of an archetypical product

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 297–305, 2006.
© Springer-Verlag Berlin Heidelberg 2006

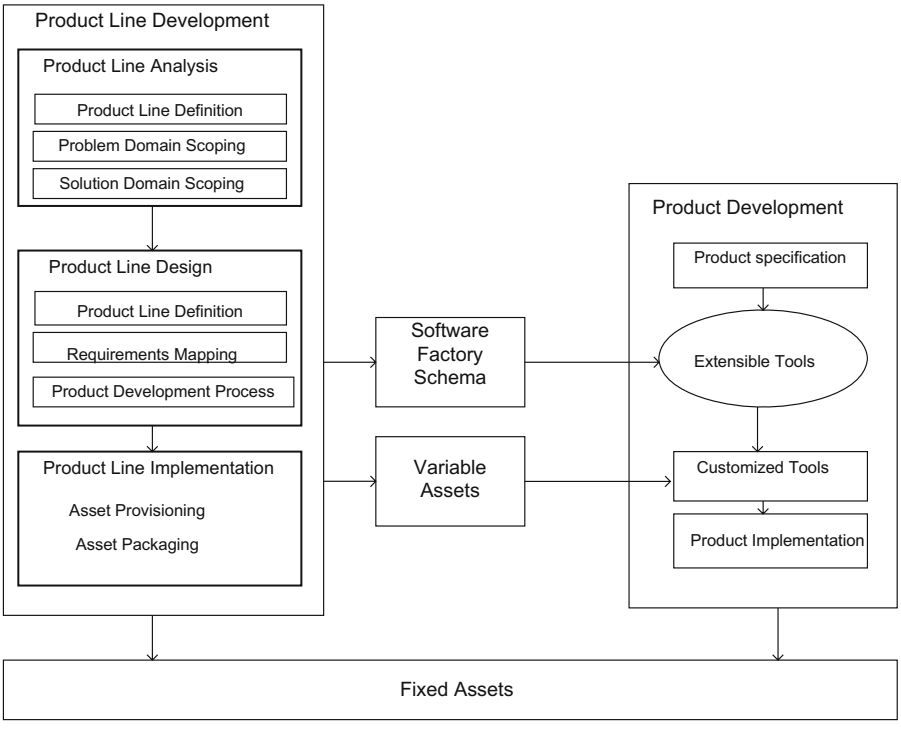


Fig. 1. Schema of a Software Factory [6]

by adapting, assembling, and configuring framework-based components" [6].The software factory schemata (see Fig. 1) is a directed graph whose nodes are viewpoints (perspectives) and whose edges are computable mappings between the viewpoints. A viewpoint provides a pattern for describing the given aspect of a software product [6]. Such a schema for e.g. a business application [4] will contain the viewpoints for the subsystems like customer management, order management or order fulfillment.

The paper addresses the problems associated with the capturing of the variability in variable assets for configuring the software factory schemata. One possible mechanism for product specification and implementation is the feature-based configuration.

The feature is a logical unit of behavior specified by a set of functional and non-functional requirements [1]. A feature may represent many requirements and aggregate them from different points of view for scoping purposes in a software product line. Feature modeling has been introduced [8] and then extended in diverse methodologies [11].

In the paper the extension of the feature modeling in UML [16], [11] with application of the fuzzy weights of some variable features is proposed. The approach is demonstrated in an example of a part of an order process [4].

1.2 Payment Alternatives in an Order Process

The example in this section is based on order processing, which is a part of the Sales and Marketing Information System [9]. Generating and fulfilling an order is a multistep process involving activities performed by sales, manufacturing and production, and accounting functions. The accounting functions include activities such as check credit, approve credit and generate invoice. Considering the diverse possible payment kinds, a payment bears for the trader some risk associated with a possible failure of a customer to make full payment. Therefore a part of an order process example containing some variant features modeled with fuzzy weights will be considered below. The optional sub-features of the Payment feature, i.e., *Credit*, *Pay_by_Bill*, *Credit_Card*, and *Pay_on_Delivery* represent the different risk levels, dependent on the customer paying for the ordered goods (see Fig. 2).

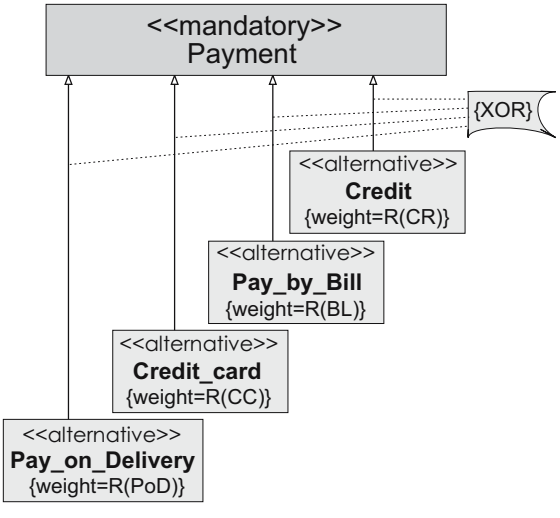


Fig. 2. Part of the Order Process Feature Model with Fuzzy Payment Sub-features [11]

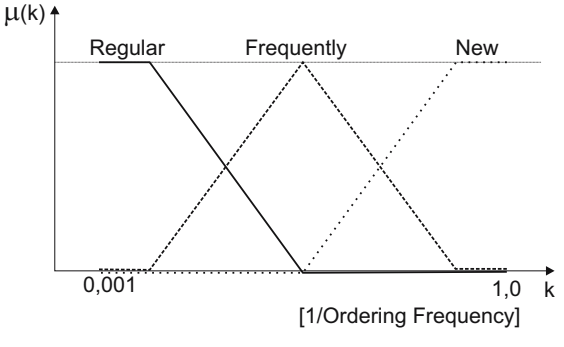


Fig. 3. The fuzzy representation of the clients

Three types of clients have been distinguished for the fuzzy representation. These will be described in the fuzzy representation by using the linguistic values, such as the "*new client*", "*frequent client*" and the "*regular client*" (see Fig. 3).

The belongingness of a client to a fuzzy set depends on a frequency of the buying activities. The membership functions representing those three types of clients are represented by the trapezoidal membership functions. For each of the payment types the experts should define the shape of the membership function for representing the applied fuzzy set. The risk associated with obtaining the payments from the customers is dependent on the particular kind of payment. For each of the payment kinds mentioned above, the fuzzy representation of the risk associated with a possible failure has been applied. For the description of the risk three fuzzy sets [19], i.e., Low (**L**), Medium (**M**) and High (**H**), have been used (see Fig. 4).

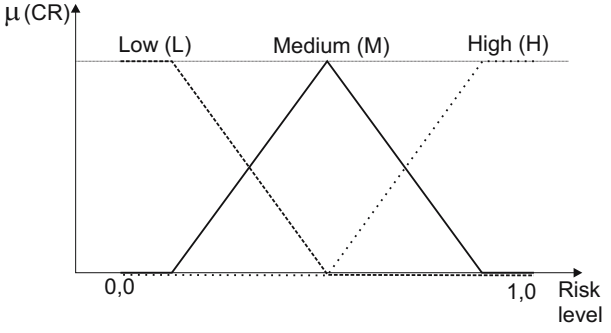


Fig. 4. The description of the risk

2 Inference in Risk Assessment System by Using Fuzzy Features

Let us assume, that according to the above description and the diagram depicted graphically in Fig. 2 there are following payment possibilities: *Credit* (CR), *Pay_by_Bill* (BL), *Credit_Card* (CC), *Pay_on_Delivery* (PoD). The risk associated with obtaining a payment from a customer will be given from the following formula 1:

$$r_i(py) = T(\mu_A(py), \mu(k)) \tag{1}$$

Where: $\mu_A(py)$ stands for a membership function of a belongingness of the payment risk to the fuzzy set $A_i(L, M, H)$;
T stands for the $t - norm$ operator determining the intersection of the set A_i describing the risk and the set defining the client's status.

The number of the sets and a shape of the function defining the payment risks in a particular form depend on the mutual arrangement of the sets defining the risks for the particular payment form and the set describing the status of

the client. The risk of the debt has been calculated for all payment kinds with Mamdani-type min of the $t-norm$ operator:

$$R(py) = \min_{j=CR,\cdots,PoD}(R_j(py)) \tag{2}$$

Where: $R_j(py)$ denotes the risk for the j^{th} kind of payment.

It takes the discrete values given by the following formula:

$$R_j(py) = \sum_{i=L}^{H} r_{i,j}(py) \cdot w_{i,j} = \widetilde{R}_j^T(py)W_j \tag{3}$$

Where: $\widetilde{R}_j^T(py) = [r_{L,j}, r_{M,j}, r_{H,j}]$ stands for a vector with the discrete values attained in the procedure sharpening the fuzzy answer for particular fuzzy sets (L, M, H), which define the risk level for the chosen j^{th} kind of payment.

W_j stands for the vector containing the weights defined by an expert, determining the influence of the particular risks features (L, M, H) on the whole payment risks.

The inference in the risk assessment system for two clients and the different kinds of payment is graphically depicted in Fig. 5. Assume the definitions of the payment risk in the fuzzy representation, given for the different kinds of a payment as shown in Fig. 5. The possible risks weights for the case of the payment on credit have been graphically depicted. Moreover, let us assume, that the new and regular client carries out the payment operation. The process of establishing the risk level value for the particular kinds of a payment and different clients is also graphically depicted in Fig. 5.

The process of obtaining the crisp values for the fuzzy answers of the system will be accomplished for the particular payment forms and for the fuzzy sets (L, M, H) describing the risks weights for the chosen kind of payment. The operation of sharpening has been made by means of the well-known and often used Center of Area (COA) method [19]. The final weighting of the discrete risks levels has been determined with the aid of the following weight vector:

$$W = \begin{bmatrix} 0.2 \\ 0.5 \\ 0.8 \end{bmatrix}. \tag{4}$$

2.1 Example of Risk Payment Calculation

Let the new client and *Credit* payment be used to explain the risk calculation procedure. The risks for the particular payment forms and the whole risks for the payment have been attained by using the formulas (1) - (3). The crisp value of payment risk can be obtained using formula (1) and the COA defuzzification method (Fig. 6). There are the following results of calculating using a different fuzzy sets defining the kinds of payment: high=0.81, medium=0.66 and low=0.

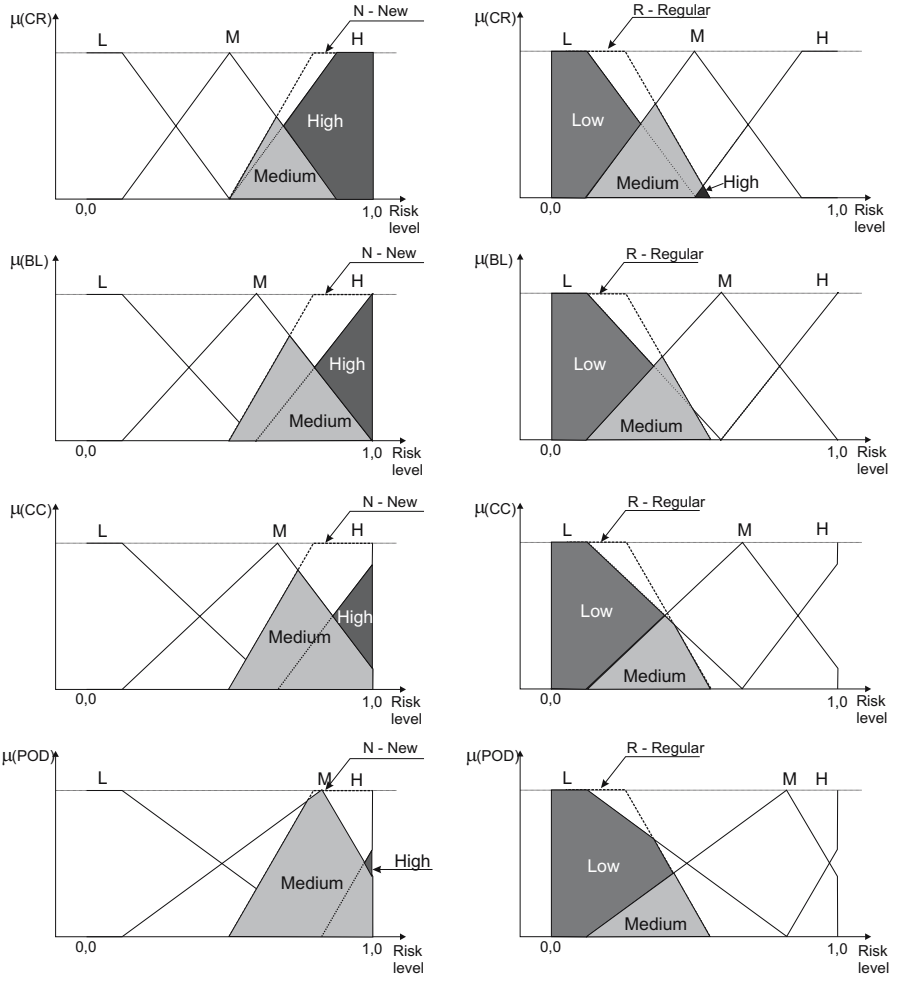


Fig. 5. Inference in Risk Assessment System for the New and Regular Client and Different Kinds of Payment

Using formula (3) and weight vector (4) the *Credit* risk payment has been calculated and it is 0.98. Finally, the risks results attained for the particular kinds of payment for a new client are as follows: $CR = 0.98$, $BL = 1.03$, $CC = 1.10$, $PoD = 1.16$. Therefore the most secure possibility for the trader is by the *Credit* (CR). The highest risks for a new client are in the case of *Payment_on_ Delivery* (PoD). For a regular client (using data from the Fig. 3) we can obtain following results: $CR = 0.63$, $BL = 0.22$, $CC = 0.21$, $PoD = 0.18$. Therefore the most secure possibility for the trader is by the *Pay_on_Delivery* (PoD). The highest risks for a regular client are in the case of *Credit*(CR).

In this section an example of the feature model of a part of the order process, containing some variant features, which have been modeled with fuzzy weights,

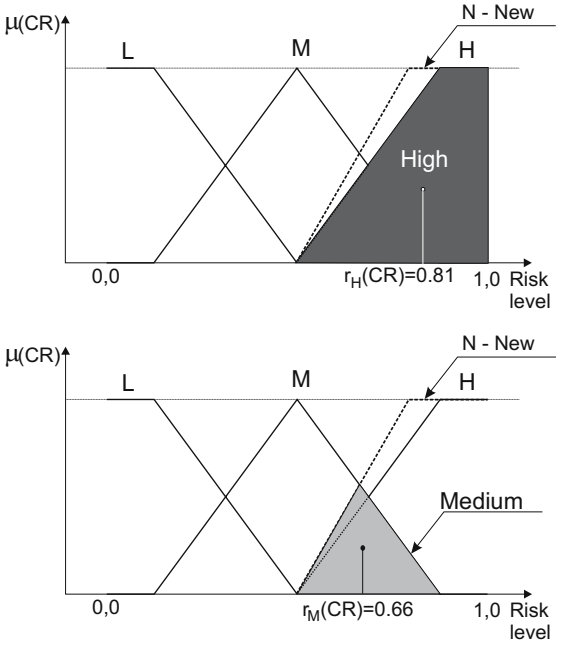


Fig. 6. Risk associated with chosen client and fuzzy set

has been given. The establishment of the risk level values for the particular kind of payment is possible by using the sharpening methods, such as COA. The achieved results of the inference may then be used for the choice of the appropriate kind of payment according to the customer making the order.

3 Conclusion and Related Work

Development of software-intensive systems within a foreseen time and budget has to deal with the factors affecting the quality of the system i.e. complexity and change. Contemporary software systems are growing in size and complexity and so the object-oriented approach, in practice with its reuse concepts alone is not sufficient.

The critical innovations in the paradigm shift in the software engineering embrace systematic reuse, development by assembly and the model driven development. The promising solutions in this field are addressed by the so-called software-factory concepts. Modeling variable assets with feature models can support configuring software factory schemata. The proposed extensions of the feature diagram, which are applied in the description of the system features, can be helpful to satisfy the needs of the customers buying software-intensive products.

The further applications of the fuzzy weighted feature diagram have been presented in [10], [12], [13], [14] and [15]. The implementation of the components

containing the problem domain features should take into account the validation of the component composition [17].

Future research will consider some modification to the representation of fuzzy and crisp knowledge: the customization of the particular elements of the fuzzification system. These are the number of fuzzy sets, the shapes of the member functions and their arrangement, and the appropriate choice of values for the elements of weight vectors. Another open problem is the appropriate choice of fuzzy weights (importance factors), which should be tuned for specific customer profiles.

References

1. Bosch J.: Design and Use of Software Architectures, Adopting and Evolving Product-Line Approach. Addison-Wesley, New York, 2000.
2. Clements P. and Northrop L. M.: Software Product Lines. Practices and Patterns. Addison Wesley, New York, 2002.
3. Czarnecki K. and Eisenecker U.: Generative Programming - Methods, Tools and Application. Addison-Wesley, New York, 2000.
4. Fingar P., Kumar H. and Sharma T.: Enterprise E-Commerce, The Software Component Breakthrough for Business-to-Business Commerce. Megan-Kiffer Press Tampa, FL USA, 2000.
5. Gamma E., Helm R., Johnson R. and Vlissides J.: Design Patterns - Elements of Reusable Object-Oriented Software. Addison-Wesley, New York, 1995.
6. Greenfield J. and Short K.: Software Factories: Assembling Applications with Patterns, Models, Framworks and Tools, John Wiley and Sons 2004.
7. Johnson R.E.: Frameworks = (Components + Patterns). Comm. of the ACM, vol.40, pages 39-42, No.10, 1997.
8. Kang K. C., Cohen S., Hess J., Nowak W. and Peterson S.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report No. CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1990.
9. Laudon K.C. and Laudon J.P.: Management Information Systems. Managing the digital firm. 9th edition. Pearson Education 2006.
10. Pieczyński A., Robak, S. and Walaszek-Babiszewska A.: Features with fuzzy probability. Engineering of Computer-Based Systems - ECBS IEEE pp. 323-328, IEEE Press 2004.
11. Robak, S.: Contribution to the Improvement of the Software Development Process for Product Families. Postdoctoral Thesis, Ilmenau Technical University, 2005.
12. Robak S. and Pieczyński A.: Employment of Fuzzy Logic in Feature Diagrams to Model Variability in Software Families. Journal of Integrated Design and Process Science, vol. 7, no. 3, pp. 79-94, USA September 2003.
13. Robak S. and Pieczyński A.: Employing Fuzzy Logic in Feature Diagrams to Model Variability in Software Product-Lines. In Proceeding of Engineering of Computer-Based Systems - ECBS IEEE, pp. 305-311, IEEE Press 2003.
14. Robak S. and Pieczyński A.: Variant system features modeled with Fuzzy Weights in the Development with Software Factories. Wiss. Fachberichte, Institut fr Prozesstechnik, Prozessautomatisierung und Messtechnik, Heft 84/2005, pp.23-29, 2005.

15. Robak S. and Pieczyński A.: Application of fuzzy weighted feature diagrams to model variability in software families. Lecture Notes in Artificial Intelligence. In Artificial Intelligence and Soft Computing ICAISC 2004, Vol. 3070, pp 370-375. Springer Verlag Berlin, 2004.
16. Robak S., Franczyk B. and Politowicz K.: Extending The UML for Modelling Variability For System Families. International Journal of Applied Mathematics and Computer Science, 12(2), pp. 285-298, 2002.
17. Speck A., Pulvermüller E., Jerger M., Franczyk B.: Component Composition Validation, International Journal of Applied Mathematics and Computer Science, 12 (4), pp. 581-589, 2002.
18. Szyperski C.: Component Software: Beyond Object-Oriented Programming. Addison-Wesley, New York, 1998.
19. Yager F. and Filev D.: Essentials of fuzzy modeling and control. WNT Warszawa, 1995.

Boosting Ensemble of Relational Neuro-fuzzy Systems*

Rafał Scherer

Department of Computer Engineering, Częstochowa University of Technology
Al. Armii Krajowej 36, 42-200 Częstochowa, Poland
rafal@ieee.org
http://kik.pcz.pl
Department of Artificial Intelligence, WSHE University in Łódź
ul. Rewolucji 1905 nr 64, Łódź, Poland
http://www.wshe.lodz.pl

Abstract. In the paper a boosting ensemble of neuro-fuzzy relational systems is created. Rules in relational fuzzy systems are more flexible than rules in linguistic fuzzy systems because of the additional weights in rule consequents. The weights come from an additional binary relation. Thanks to this, input and output fuzzy sets are related to each other with a certain degree. The size of the relations is determined by the number of input fuzzy sets and the number of output fuzzy sets. Simulations performed on popular benchmarks show that the proposed ensemble outperforms other learning systems.

1 Introduction

Fuzzy and neuro-fuzzy systems abound in many variations [1][6][10][14][15][16]. Most of them are based on intelligible fuzzy rules. These fuzzy rules are obtained through expert knowledge or some heuristic methods, but they lack the ability to learn from data as neural networks. Neuro-fuzzy systems are a solution to the problem. They have the ability to learn from numerical data, using so called data-driven learning. The structure of these systems looks like a neural network but its units reflect fuzzy sets and operations performed on fuzzy sets like t-norms or t-conorms. Thanks to this, it is easy to see the construction of the fuzzy system and to learn the system by a gradient algorithm. A learning algorithm can be used to approximate any n-dimensional function. And yet the structure consists of fuzzy rules which are easy to extract unlike in the case of neural networks. In traditional fuzzy systems the input-output mapping is defined by the relation built from fuzzy rules and input and output fuzzy sets. Usually, output fuzzy sets are singleton sets in neuro-fuzzy systems. Neuro-fuzzy systems are used for various tasks, as they are able to approximate any function [21]. In case of classification, a fuzzy system has to approximate discriminant

* This work was supported in part by the Foundation for Polish Science (Professorial Grant 2005-2008) and the Polish State Committee for Scientific Research (Grant Nr T11C 04827).

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 306–313, 2006.
© Springer-Verlag Berlin Heidelberg 2006

functions. Fuzzy classifiers [8][10][19] are developed as an alternative approach to traditional classifiers [13].

One kind of fuzzy systems are relational fuzzy systems [2][12][19], where there is a relation binding input and output fuzzy sets. In this case we obtain fuzzy rules with additional weights that can be regarded as a kind of rule weights [11]. In relational neuro-fuzzy systems [17][18] there is a possibility to train the system by the backpropagation algorithm. This kind of fuzzy system has more flexible fuzzy rules which improves fitting the system for data.

Neuro-fuzzy can be joined into ensembles which improves accuracy and robustness. One of the most popular methods for creating multiple learning systems is boosting, where input patterns obtain weights according to learning system performance on a given input pattern. Then consecutive classifiers are learned with the greatest impact from the learning samples with the highest weight values. The most popular modification of boosting is the AdaBoost algorithm [4][9][20], described in Section 3.

In the paper we propose a boosting ensemble of relational neuro-fuzzy systems. The ensemble has the following advantages comparing to other fuzzy systems:

(i) The system allows learning all its parameters (relation matrix elements and membership function parameters) by the backpropagation algorithm.
(ii) The system has very flexible, yet intelligible form of fuzzy rules.
(iii) The ensemble of such systems outperforms other fuzzy learning systems, not sacrificing transparency (which happens when we introduce more adjustable parameters).

The paper is organized as follows. In Section 2 relational fuzzy systems are described, and in Section 3 the AdaBoost algorithm is presented. Finally, we test the multiple classifier system on popular benchmarks taken from [3].

2 Fuzzy Relational Systems for Classification

Fuzzy relational models can be regarded as a generalization of linguistic fuzzy systems, where each rule has more than one linguistic value defined on the same output variable, in its consequent. Fuzzy rules in a MISO relational model have the following form

$$\begin{aligned} &R^k: \text{ IF } x_1 \text{ is } A_1^k, \ldots, x_n \text{ is } A_n^k, \text{ THEN} \\ &y \text{ is } B^1\ (r_{k1}), y \text{ is } B^m\ (r_{km}), \ldots \\ &\ldots, y \text{ is } B^M\ (r_{kM})\ , \end{aligned} \tag{1}$$

where A_i^k is an antecedent (input) fuzzy set of k-th rule and i-th input, B^m is a consequent (output) fuzzy set r_{km} is a weight, responsible for the strength of connection between input and output fuzzy sets. Relational fuzzy systems store associations between the input and the output linguistic values in the form of a discrete fuzzy relation

$$\mathbf{R}\,(A, B) \in [0, 1]\ . \tag{2}$$

In case of a multi-input multi-output system (MIMO), the relation $\mathbf{R}$ is a matrix containing degree of connection for every possible combination of input and output fuzzy sets. If we would consider a fuzzy system with multidimensional input linguistic values, where input fuzzy sets are common for all classes. Thus, we would have only one set A of fuzzy linguistic values

$$A = \left\{A^1, A^2, ..., A^K\right\} , \tag{3}$$

thus the relational matrix $\mathbf{R}$ is only two-dimensional. Output variable y has a set of M linguistic values B^m with membership functions $\mu_{B^m}(y)$, for $m = 1, ..., M$

$$B = \left\{B^1, B^2, ..., B^M\right\} . \tag{4}$$

Sets A and B are related to each other with a certain degree by the $K \times M$ binary relation matrix

$$\mathbf{R}_c = \begin{bmatrix} r_{11} & r_{11} & \cdots & r_{1M} \\ r_{21} & r_{22} & \cdots & r_{2M} \\ \vdots & \vdots & r_{km} & \vdots \\ r_{K1} & r_{K2} & \cdots & r_{KM} \end{bmatrix} . \tag{5}$$

For simplicity we use a system where sets A^k are replaced by t-norm of n input fuzzy sets A_i^k

$$\mu_{A^k} = \tau^k = \prod_{i=1}^{n} \mu_{A_i^k}(x_i) , \tag{6}$$

In case of classification, $\mathbf{x}$ is a vector of features of an object ν, and $\Omega = \{\omega_1, ..., \omega_C\}$ is a set of classes. The classifier knowledge is represented by a set of K rules in the form

$$\begin{array}{l} R^k : \text{ IF } x_1 \text{ is } A_1^k, \ldots, \ x_n \text{ is } A_n^k, \text{ THEN} \\ \ \nu \in \omega_1(z_1^k), \ \ \nu \in \omega_2(z_2^k), \ldots, \ \nu \in \omega_C(z_C^k) , \end{array} \tag{7}$$

where z_c^k, $c = 1, ..., C$, $k = 1, ..., K$, are interpreted as a "support" for a class ω_c given by a rule R^k. Farther we redefine a description of the fuzzy relational system using this idea. Let us introduce a vector $\mathbf{z} = [z_1, ..., z_C]$, where z_c, $c = 1, ..., C$, is the "support" for a class ω_c given by all C rules. We can scale the support values to the interval $[0, 1]$, so that z_c is the membership degree of an object ν to class ω_c according to all K rules. Now a k-th rule in case of binary classification has the form (1) and this form will be used to design our neuro-fuzzy relational classifier.

Having given vector $\bar{A}$ of K membership values $\mu_{A^k}(\bar{\mathbf{x}})$ for a crisp observed feature values $\bar{\mathbf{x}}$, vector $\bar{B}$ of M crisp memberships μ_m is obtained through a fuzzy relational composition

$$\bar{B} = \bar{A} \circ \mathbf{R} , \tag{8}$$

implemented element-wise by a generalized form of sup-min composition [1], i.e. s-t composition

$$\mu_{mc} = \underset{k=1}{\overset{K}{\mathrm{S}}} \left[\mathrm{T}\left(\mu_{A^k}(\bar{\mathbf{x}}), r_{km}\right)\right] . \tag{9}$$

The crisp output of the relational system is computed by the weighted mean

$$\bar{y} = \frac{\sum_{m=1}^{M} \left\{ \bar{y}^m \, \mathrm{S}_{k=1}^{K} \left[\mathrm{T} \left(\mu_{A^k} \left(\bar{\mathbf{x}} \right), r_{km} \right) \right] \right\}}{\sum_{m=1}^{M} \mathrm{S}_{k=1}^{K} \left[\mathrm{T} \left(\mu_{A^k} \left(\bar{\mathbf{x}} \right), r_{km} \right) \right]} \, , \tag{10}$$

where $\bar{y}^m$ is a centre of gravity (centroid) of the output fuzzy set B^m. The system is depicted in Figure 1.

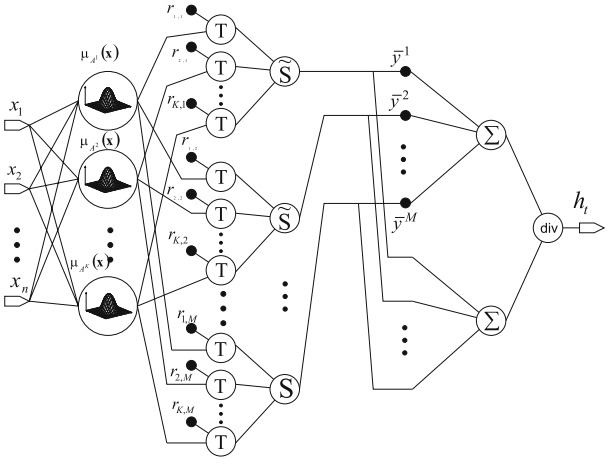


Fig. 1. Neuro-fuzzy relational system used in the ensemble. The index t denoting hypothesis number is omitted for simplicity. Antecedent fuzzy sets (linguistic terms) with t-norms (see (6)) are replaced in the figure with multidimensional fuzzy sets for clarity.

3 AdaBoost Agorithm

In this section we describe the AdaBoost algorithm which is the most popular boosting modification [4][9][20]. Let us denote the l-th learning vector by $\mathbf{z}^l = [x_1^l, ..., x_n^l, y^l]$, $l = 1...m$ is the number of a vector, n is a size of input vector x^l, and y^l is the learning class label. Weights, assigned to learning vectors, have to fulfill the following conditions

$$\text{(i)} \; 0 < d^l < 1 \, , \tag{11}$$

$$\text{(ii)} \; \sum_{l=1}^{m} d^l = 1 \, . \tag{12}$$

The weight d^l is the information how well classifiers learned in consecutive steps of an algorithm for a given input vector x^l. Vector $\mathbf{d}$ for all input vectors is initialized according to the following equation

$$d_t^l = \frac{1}{m}, \quad \text{for } t = 0 \tag{13}$$

where t is the number of a boosting iteration (and a number of a classifier in the ensemble). Let $\{h_t(\mathbf{x}) : t = 1, ..., T\}$ denotes a set of hypotheses obtained in consecutive steps t of the algorithm being described. For simplicity we limit our problem to a binary classification (dichotomy), i.e. $y = -1, 1$ or $h_t(\mathbf{x}) = \pm 1$. Similarly to learning vectors weights, we assign a weight c_t for every hypothesis, such that

$$\text{(i) } 0 < c_t \ , \tag{14}$$

$$\text{(ii)} \sum_t c_t = 1 \ . \tag{15}$$

Now in the AdaBoost algorithm we repeat steps 1-4 for $t = 1, \ldots, T$:
1. Create hypothesis h_t and train it with a data set with respect to a distribution d_t for input vectors.
2. Compute the classification error ε_t of a trained classifier h_t according to the formula

$$\varepsilon_t = \sum_{l=1}^{m} d_t^l(z^l) I(h_t(\mathbf{x}^l) \neq y^l) \ , \tag{16}$$

where I is the indicator function

$$I(a \neq b) = \begin{cases} 1 \text{ if } a \neq b \\ 0 \text{ if } a = b \end{cases} \ . \tag{17}$$

If $\varepsilon_t = 0$ or $\varepsilon_t \geq 0.5$, stop the algorithm.
3. Compute the value

$$\alpha_t = \log \frac{0.5(1 - \varepsilon_t)}{\varepsilon_t(1 - 0.5)} \ . \tag{18}$$

4. Modify weights for learning vectors according to the formula

$$d_{t+1}(\mathbf{z}^l) = \frac{d_t(\mathbf{z}^l) \exp\{-\alpha_t \mathbf{I}(h_t(\mathbf{x}_l) = y^l)\}}{N_t} \ , \tag{19}$$

where N_t is a constant such that

$$\sum_{l=1}^{m} d_{t+1}(\mathbf{z}^l) = 1 \ . \tag{20}$$

To compute the overall output of the ensemble of classifiers trained by AdaBoost algorithm the following formula is used

$$f(\mathbf{x}) = \sum_{t=1}^{T} c_t h_t(\mathbf{x}) \ , \tag{21}$$

where

$$c_t = \frac{\alpha_t}{\sum_{t=1}^{T} |\alpha_t|} \tag{22}$$

is classifier importance for a given training set. The AdaBoost algorithm is a meta-learning algorithm and do not determine the way of learning for classifiers in the ensemble.

4 Numerical Simulations

Numerical simulations were carried out on two binary classfication benchmarks from [3]. The ensembles were created from several relational neuro-fuzzy systems (Figure 1). The subsystems were trained by the backpropagation algorithm, starting with random values of all parameters. The dataset were converted such that class label $y(\mathbf{x}) = \pm 1$, which is requirement in the boosting learning algorithm used in the paper. Everytime the training was stopped when $\varepsilon_t \geq 0.5$.

4.1 MONK-2 Problem

The three monks problems are artificial problems designed to test machine learning algorithms. Each of the three monks problems requires determining whether an object described by six features (head shape, body shape, is smiling, holding, jacket colour, has tie) is a monk or not. In the MONK-2 problem the learning set has 169 cases and the testing set has 432 cases. The obtained classification accuracy is 100% and Table 1 shows hypothesis parameters.

Table 1. Simulation results for MONK-2 problem

Classifier number	Antecedent fuzzy sets	Consequent fuzzy sets	c_t coefficient
1	16	2	2.50
2	16	2	0.99
3	16	2	0.46
4	16	2	0.13

4.2 Pima Indians Diabetes Problem

The Pima Indians Diabetes (PID) data [3][14] contains two classes, eight attributes (number of times pregnant, plasma glucose concentration in an oral glucose tolerance test, diastolic blood pressure (mm Hg), triceps skin fold thickness (mm), 2-hour serum insulin (mu U/ml), body mass index (weight in kg/(height in m)2), diabetes pedigree function, age (years)). We consider 768 instances, 500 (65.1%) healthy and 268 (34.9%) diabetes cases. All patients were females at least 21 years old of Pima Indian heritage, living near Phoenix, Arizona. In our experiments, all sets are divided into a learning sequence (576 sets) and a testing sequence (192 sets).

The relational neuro-fuzzy systems used in the experiment have 2 antecedent fuzzy sets and 2 consequent fuzzy sets. In the PID case the training was stopped after the fifth classifier has been trained, because during creating the second classifier value of ε_t did not dropped below 0.5. The classification accuracy is 82.3%. Table 2 shows results of each classifier in the boosting ensemble.

Table 2. Simulation results for Pima Indians Diabetes problem

Classifier number	Antecedent fuzzy sets	Consequent fuzzy sets	c_t coefficient
1	2	2	0.616
2	2	2	0.308
3	2	2	0.058
4	2	2	0.012
5	2	2	0.015

5 Conclusions

In the paper, we presented a new fuzzy relational system as a part of a boosting ensemble. In fuzzy classifiers we can use vague knowledge, and rules are intelligible and easier to define than in purely data-driven classifiers. Rules in the system are more flexible because of the additional weights in rule consequents. The weights comes from binary relations **R**. The dimension of the relation is determined by the number of input fuzzy sets and the number of output fuzzy sets for a given class. To train the relational system we can set linguistic values in advance and fine-tune only elements of the relation. In the paper we train all the relational neuro-fuzzy systems constituting the ensemble by the backpropagation algorithm with random initial parameter values. The achieved results for Pima Indians Problem are better than achieved so far in the literature [14] and for MONK-2 problem the results are comparable to the best obtained in the literature. The relational neuro-fuzzy systems can be highly transparent, despite having very good performance and ability to fit to the data very accurately.

References

1. R. Babuska, *Fuzzy Modeling For Control*, Kluwer Academic Press, Boston, 1998.
2. P.J.C. Branco, J.A. Dente, "A Fuzzy Relational identification Algorithm and its Application to Predict the Behaviour of a Motor Drive System", *Fuzzy Sets and Systems*, vol. 109, pp. 343–354, 2000.
3. C.L. Blake, C.J. Merz, *UCI Repository of machine learning databases*, `www.ics.uci.edu/~mlearn/MLRepository.html`, Irvine, University of California, Department of Information and Computer Science, 1998.
4. L. Breiman, *Bias, variance, and arcing classifiers*, Technical Report 460, Statistics Department, University of California, July 1997.
5. H. Ischibuchi, T. Nakashima, "Effect of Rule Weights in Fuzzy Rule-Based Classification Systems", *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 506–515, 2001.
6. R. J.-S. Jang, C.-T. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, Upper Saddle River, 1997.
7. L. I. Kuncheva, *Combining Pattern Classifiers, Methods and Algorithms*, John Wiley & Sons 2004.

8. L. I. Kuncheva, *Fuzzy Classifier Design*, Physica Verlag, Heidelberg, New York, 2000.
9. R. Meir and G. Rätsch, "An introduction to boosting and leveraging", in *Advanced Lectures on Machine Learning*, LNAI 2600, edited by S. Mendelson and A. Smola, Springer, pp. 119-184, 2003.
10. D. Nauck, F. Klawon, R. Kruse, *Foundations of Neuro - Fuzzy Systems*, Chichester, U.K., John Wiley, 1997.
11. D. Nauck, R. Kruse, "How the Learning of Rule Weights Affects the Interpretability of Fuzzy Systems", Proc. of 1998 IEEE World Congress on Computational Intelligence, FUZZ-IEEE, Alaska, pp. 1235–1240, 1998.
12. W. Pedrycz, *Fuzzy Control and Fuzzy Systems*, Research Studies Press, London, 1989.
13. B.D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.
14. L. Rutkowski, *Flexible Neuro-Fuzzy Systems*, Kluwer Academic Publishers, 2004.
15. L. Rutkowski and K. Cpałka, "Designing and learning of adjustable quasi triangular norms with applications to neuro fuzzy systems", *IEEE Trans. on Fuzzy Systems*, vol. 13, Feb 2005, pp. 140-151.
16. L. Rutkowski, K. Cpałka, "Flexible neuro-fuzzy systems", *IEEE Transactions on Neural Networks*, vol. 14, May 2003, pp. 554-574.
17. R. Scherer, L. Rutkowski, "Neuro-Fuzzy Relational Systems", Proc. of 2002 International Conference on Fuzzy Systems and Knowledge Discovery, November 18-22, Singapore (CD-ROM), 2002.
18. R. Scherer, L. Rutkowski, "Connectionist Fuzzy Relational Systems", in *Studies in Computational Intelligence, Computational Intelligence for Modelling and Control*, Edited by S. Hagamuge, L.P. Wang, Springer-Verlag, pp. 35–47, 2005.
19. M. Setness, R. Babuska, "Fuzzy Relational Classifier Trained by Fuzzy Clustering", *IEEE Transactions on Systems, Man and Cybernetics* - Part B: Cybernetics, Vol. 29, No. 5, October, 1999, pp.619–625.
20. R.E. Schapire, "A brief introduction to boosting", Proc. of the Sixteenth International Joint Conference on Artificial Intelligence, 1999.
21. L.-X. Wang, *Adaptive Fuzzy Systems And Control*, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1994.

An Application of Intuitionistic Fuzzy Set Similarity Measures to a Multi-criteria Decision Making Problem

Eulalia Szmidt and Janusz Kacprzyk

Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01–447 Warsaw, Poland
{szmidt, kacprzyk}@ibspan.waw.pl

Abstract. We propose a new solution to a multi-criteria decision making problem by using similarity measures for intuitionistic fuzzy sets. We show that the new solution is better than the method proposed in [5] which fails in some situations.

1 Introduction

Intuitionistic fuzzy sets (Atanassov [1], [2]) can be viewed as a tool that may better model and process imperfect information. They have found numerous applications among which a notable example is multi-criteria decision making under imperfectly defined facts and imprecise knowledge.

The idea of using positive and (independently) negative information that is the core of intuitionistic fuzzy sets is natural in any real life human discourse and action, and as an obvious consequence, is well-known and widely studied in psychology and other social sciences [e.g., [8], [6]]. It has also attracted much attention and reserach interest in soft computing. It would be difficult to deal with machine learning (making use of examples and counter-examples), modeling of preferences or voting without taking into account positive and (independently) negative testimonies or opinions. Although from a mathematical point of view intuitionistic fuzzy sets are equipollent to interval-valued fuzzy sets (as noticed by Atanassov and Gargov in 1989 [3]), from the point of view of solving problems (starting from the stage of collecting data), they are different as intuitionistic fuzzy sets force a user to explicitly consider positive and negative information independently. On the other hand, while employing interval-valued fuzzy sets, a user's attention is focused on positive information (in an interval) only. This fact – that is strongly related to a psychological phenomenon called by the Nobel Prize winner Kahneman (cf. Kahneman [6]) a "bounded rationality" (see also [8]), caused among others by the fact that people tend to notice and take into account only most obvious aspects (e.g. advantages only) – may be viewed to places intuitionistic fuzzy sets among up-to-date and promising means of knowledge representation and processing.

In this paper we will use intuitionistic fuzzy sets for a multi-criteria decision making problem. More specifically, we have a set M of options fulfilling a set of criteria C. We wish to rank the options satisfying: C_j, and C_k, ..., and C_p

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 314–323, 2006.
© Springer-Verlag Berlin Heidelberg 2006

or C_s whereas each criterion is fulfilled to some extent μ and is not fulfiiled to some extent ν (when $0 \leq \mu + \nu \leq 1$). The solution of this problem was proposed by Chen and Tan [5] but it does not always give a proper answer as indicated by Liu [7]. We show how to overcome these deficiencies by using some measures of similarity between intuitionistic fuzzy sets.

2 Brief Introduction to Intuitionistic Fuzzy Sets

As opposed to a fuzzy set in X(Zadeh [21]), given by

$$A' = \{< x, \mu_{A'}(x) > | x \in X\} \tag{1}$$

where $\mu_{A'}(x) \in [0,1]$ is the membership function of the fuzzy set A', an intuitionistic fuzzy set (Atanassov [1], [2]) A is given by

$$A = \{< x, \mu_A(x), \nu_A(x) > | x \in X\} \tag{2}$$

where: $\mu_A : X \to [0,1]$ and $\nu_A : X \to [0,1]$ such that $0 \leq \mu_A(x) + \nu_A(x) \leq 1$ and $\mu_A(x)$, $\nu_A(x) \in [0,1]$ denote a degree of membership and a degree of non-membership of $x \in A$, respectively. For details on operations on, properties of, etc. intuitionistic fuzzy sets, see Atanassov [1], [2].

Obviously, each fuzzy set may be represented by:

$$A = \{< x, \mu_{A'}(x), 1 - \mu_{A'}(x) > | x \in X\} \tag{3}$$

For each intuitionistic fuzzy set in X, we call

$$\pi_A(x) = 1 - \mu_A(x) - \nu_A(x) \tag{4}$$

an *intuitionistic fuzzy index* (or a *hesitation margin*) of $x \in A$ and, it expresses a lack of knowledge of whether x belongs to A or not (cf. Atanassov [2]).

Applications of intiutionistic fuzzy sets to group decisions, negotiations and other situations are given in Szmidt and Kacprzyk [10], [11], [13], [15], [17].

2.1 A Geometrical Interpretation of Intuitionistic Fuzzy Sets

Having in mind that for each element x belonging to an intuitionistic fuzzy set A, the values of membership, non-membership and the intuitionistic fuzzy index add up to one, i.e.

$$\mu_A(x) + \nu_A(x) + \pi_A(x) = 1$$

and that each: membership, non-membership, and the intuitionistic fuzzy index are from the interval $[0,1]$, we can imagine a unit cube (Figure 1) inside which there is ABD triangle where the above equations are fulfilled. In other words, ABD triangle represents a surface where coordinates of any element belonging to an intuitionistic fuzzy set can be represented. Each point belonging to ABD triangle is described via three coordinates: (μ, ν, π). Points A and B represent the crisp elements. Point $A(1,0,0)$ represents elements fully belonging to an

intuitionistic fuzzy set as $\mu = 1$. Point $B(0, 1, 0)$ represents elements fully not belonging to an intuitionistic fuzzy set as $\nu = 1$. Point $D(0, 0, 1)$ represents elements about which we are not able to say if they belong or not belong to an intuitionistic fuzzy set (intuitionistic fuzzy index $\pi = 1$). Such an interpretation is intuitively appealing and provides quite powerful and adequate means for the representation of many aspects of imperfect information. Segment AB (where $\pi = 0$) represents elements belonging to classical fuzzy sets ($\mu + \nu = 1$). Any other combination of the values characterizing an intuitionistic fuzzy set can be represented inside the triangle ABD. In other words, each element belonging to an intuitionistic fuzzy set can be represented as a point (μ, ν, π) belonging to the triangle ABD (cf. Figure 1).

It is worth mentioning that this geometrical representation is directly related to the definition of an intuitionistic fuzzy set introduced by Atanassov [1], [2], and it does not need any additional assumptions. More considerations on the geometrical representations of intuitionistic fuzzy sets can be found in [20] and [4].

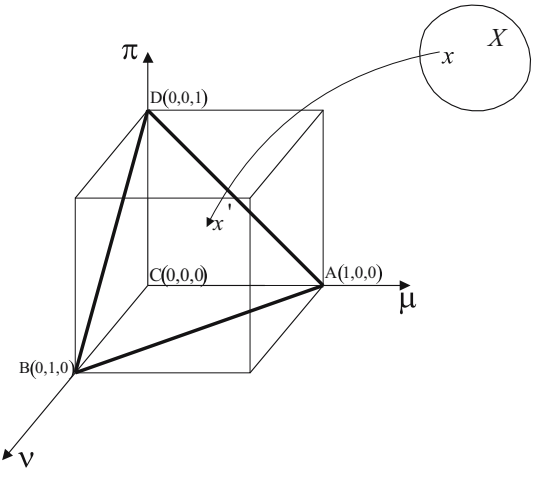


Fig. 1. Geometrical representation

3 A Multicriteria Decision Making Problem

Basically, Chen and Tan's approach [5] to be considered here can be outlined, assuming a intuitionistic fuzzy perspective, as follow. M is a set of options and C is a set of criteria

$$M = \{M_1, M_2, ..., M_m\}, \qquad C = \{C_1, C_2, \ldots, C_n\}$$

where each option M_i is expressed via intuitionistic fuzzy description, namely

$$M_i = \{(C_1, \mu_{i1}, \nu_{i1}), (C_2, \mu_{i2}, \nu_{i2}), \ldots, (C_n, \mu_{in}, \nu_{in})\}, \quad i = 1, 2, \ldots, m$$

where μ_{ij} indicates the degree to which option M_i satisfies criterion C_j, ν_{ij} indicates the degree to which option M_i does not satisfy criterion C_j.

Our goal is to point out the best option (to rank the considered options). The options should satisfy the criteria C_j, C_k,..., and C_p or criterion C_s, i.e.:

$$(C_j \text{ and } C_k \text{ and } , \ldots , \text{ and } C_p) \text{ or } C_s \tag{5}$$

Chen and Tan's [5] solution of the problem is as follows. First, they define an evaluation function E to measure a degree to which option M_i satisfies and does not satisfy the condition given by (5):

$$\begin{aligned} E(M_i) &= ((\mu_{ij}, \nu_{ij}) \wedge (\mu_{ik}, \nu_{ik}) \wedge, \ldots , \wedge (\mu_{ip}, \nu_{ip})) \vee (\mu_{is}, \nu_{is}) \\ &= (\mu_{M_i}, \nu_{M_i}) \end{aligned} \tag{6}$$

where $\wedge$ and $\vee$ denote min and max, and

$$\mu_{M_i} = \max(\min(\mu_{ij}, \mu_{ik}, \ldots , \mu_{ip}), \mu_{is})$$

$$\nu_{M_i} = \min(\max(\nu_{ij}, \nu_{ik}, \ldots , \nu_{ip}), \nu_{is})$$

The score function S gives a degree of satisfaction (5) by option M_i

$$S(E(M_i)) = \mu_{M_i} - \nu_{M_i} \tag{7}$$

where $S(E(M_i)) \in [-1, 1]$. Option M_i is the best one if $S(E(M_i))$ is the largest value among the values $\{S(E(M_i)) | i = 1, 2, \ldots , m\}$.

But as shown in [7], the proposed solution does not work properly in some situations as outlined in the below example.

Example 1. $M = \{M_1, M_2, M_3, M_4, M_5\}$ and $N = \{N_1, N_2\}$ are two sets of options. $C = \{C_1, C_2, C_3\}$ – a set of criteria. The characteristics of option are expressed via the following intuitionistic fuzzy description

$$M_1 = \{(C_1, 0.2, 0.2), (C_2, 0.3, 0.1), (C_3, 0.2, 0.)\}$$

$$M_2 = \{(C_1, 0.3, 0.3), (C_2, 0.2, 0.2), (C_3, 0.3, 0.1)\}$$

$$M_3 = \{(C_1, 0.4, 0.4), (C_2, 0.5, 0.4), (C_3, 0.3, 0.2)\}$$

$$M_4 = \{(C_1, 0.5, 0.3), (C_2, 0.4, 0.4), (C_3, 0.5, 0.3)\}$$

$$M_5 = \{(C_1, 0.4, 0.4), (C_2, 0.6, 0.3), (C_3, 0.6, 0.4)\}$$

and

$$N_1 = \{(C_1, 0.3, 0.5), (C_2, 0.2, 0.4), (C_3, 0.2, 0.3)\}$$

$$N_2 = \{(C_1, 0.3, 0.6), (C_2, 0.2, 0.4), (C_3, 0.4, 0.6)\}$$

and we assume that the following condition concerning the criteria is fulfilled C_1 and C_2 or C_3

From (6) we have:

$$E(M_1) = (0.2, 0), \;\; E(M_2) = (0.3, 0.1), \;\; E(M_3) = (0.4, 0.2)$$

$$E(M_4) = (0.5, 0.3), \;\; E(M_5) = (0.6, 0.4)$$

$$E(N_1) = (0.2, 0.3), \;\; E(N_2) = (0.4, 0.6)$$

Finally, from(7) we obtain:

$$S(E(M_1)) = S(E(M_2)) = S(E(M_3)) = S(E(M_4)) = S(E(M_5)) = 0.2$$

$$S(E(N_1)) = -0.1, \;\; S(E(N_2)) = -0.2$$

what means that it is not possible to single out the best options among M_1, M_2, M_3, M_4, M_5, and that N_1, is better than N_2. But this conclusion is difficult to accept. □

In this article we propose another solution of the problem. Our approach differs from the solution proposed in [5] in two decisive aspects:

1. We evaluate options comparing them to the positive-ideal solution and nega - tive-ideal solution. The best considered option should be as close as possible to the positive-ideal solution and as far as possible to the negative-ideal solution. In our previous works (cf. e.g, [16]) we have shown that looking for the solution (the best option) taking into account only positive-ideal solution can be misleading.
2. We claim that the criterion (7) proposed in [5] is not enough while employing intuitionistic fuzzy sets. The degree of satisfaction (7) takes into account the membership values and non-membership values of the considered options only. But from the point of view of a real decision making it is also important if we have a complete information concerning an option (the hesitation margin is equal to 0) or if there is a considerable lack of knowledge as far as an option is concerned. When comparing the options taking into account (7), we put a sign of equality, for instance between the following obviously different options (μ, ν) because for all of them the criterion (7) gives the same value that is equal to 0.4:
   - $(0.7, 0.3)$ with a lack of knowledge (concerning the membership and non − membership) equal to 0, and
   - $(0.6, 0.2)$ with a lack of knowledge (concerning the membership and non − membership) equal to 0.2, and
   - $(0.5, 0.1)$ with a lack of knowledge (concerning the membership and non − membership) equal to 0.4, and
   - $(0.4, 0.0)$ with a lack of knowledge (concerning the membership and non − membership) equal to 0.6.

   As opposed to the results obtained from (7), it seems obvious that a decision maker would like to consider as quite different the above options, e.g., $(0.7, 0.3)$ and $(0.4, 0.0)$. In the first case we know that the option fulfils 70% of desirable conditions and does not fulfil 30% of desirable conditions. In the

second case we know that the option fulfils 40% of desirable conditions, but we are not able to say anything about the remaining 60% of the considered conditions.

In this article we propose an alternative solution of this problem (i.e. the one considered in [5]) by using a measure of similarity for intuitionistic fuzzy sets which overcomes the two drawbacks mentioned above hence helping attain more adequate and intuitively appealing solutions.

4 A Similarity Measure for Intuitionistic Fuzzy Sets and Its Use in Solving the Multi-criteria Decision Making Problem

We remind here briefly the concept of a similarity measure for intuitionistic fuzzy sets. The starting point is a geometrical interpretation of intuitionistic fuzzy sets (Section 2.1, see also Szmidt and Baldwin [9], Szmidt and Kacprzyk [12],[14]) which implies that any combination of the parameters characteristic for elements belonging to an intuitionistic fuzzy set can be represented inside triangle ABD (Figure 2). In other words, a degree to which any considered criterion C_i is fulfilled can be represented inside the ABD triangle.

In [16], [17] we proposed a general concept of a similarity measure for two elements of an intuitionistic fuzzy set (or sets). Here we give a modified measure of similarity that is meant for any criterion C_i, and element A (Figure 2). Element A ($\mu_A = 1$, $\nu_A = 0$, and $\pi_A = 0$) is our reference element representing the positive-ideal solution, i.e., a criterion which is fully satisfied ($\mu_A = 1$). The proposed measure indicates if a criterion C_i is more similar to A (representing the positive-ideal solution, i.e. a fully satisfied criterion) or to B ($\mu_B = 0, \nu_B = 1$, and $\pi_B = 0$) representing the negative-ideal solution, i.e. a fully dissatisfied criterion. In other words, it may indicate if the criterion considered is more satisfied or more dissatisfied (Figure 2).

Definition 1

$$Sim(C_i, A) = \frac{l_{IFS}(C_i, A)}{l_{IFS}(C_i, B)} \tag{8}$$

where: $l_{IFS}(C_i, A)$ *is a distance from* $C_i(\mu_{C_i}, \nu_{C_i}, \pi_{C_i})$ *to* $A(1, 0, 0)$,
$l_{IFS}(C_i, B)$ *is the distance from* $C_i(\mu_{C_i}, \nu_{C_i}, \pi_{C_i})$ *to* $B(0, 1, 0)$.
The distances $l_{IFS}(C_i, A)$ *and* $l_{IFS}(C_i, B)$ *are calculated from (9) and (10), respectively [12], as:*

$$l_{IFS}(C_i, A) = \frac{1}{2}\sum_{i=1}^{n}(|1 - \mu_{C_i}| + |0 - \nu_{C_i}| + |0 - \pi_{C_i}|) \tag{9}$$

$$l_{IFS}(C_i, B) = \frac{1}{2}\sum_{i=1}^{n}(|0 - \mu_{C_i}| + |1 - \nu_{C_i}| + |0 - \pi_{C_i}|) \tag{10}$$

For (8) we have $0 \leq Sim(C_i, A) \leq \infty$.

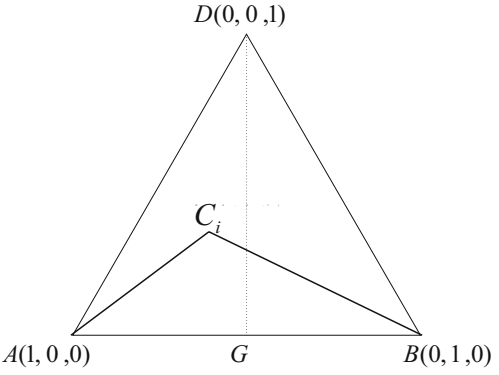


Fig. 2. Idea of similarity measures for intitionistic fuzzy sets

It is worth emphasizing that the distances (9) and (10) used in the definition of similarity take into account all three functions (membership, non-membership and hesitation margin) characterizing intuitionistic fuzzy sets. A motivation for using all three functions and poor effects of omitting one of them (from the point of view of decision making) is given in (Szmidt and Kacprzyk [18], [19]).

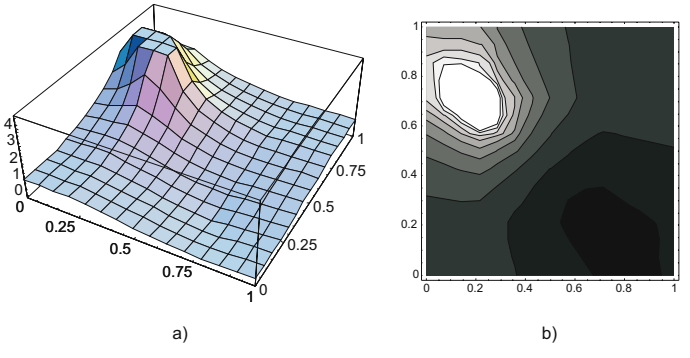


Fig. 3. Values of similarity measure $Sim(C_i, A)$ (8) – a), b) – contour plot

Note that:

- $Sim(C_i, A) = 0$ means the identity of C_i and A,
- $Sim(C_i, A) = 1$ means that C_i is to the same extent similar to A and B (i.e., values bigger than 1 mean in fact a closer similarity of C_i and B to C_i and A),
- when $C_i = B$, i.e. $l_{IFS}(C_i, B) = 0$ means the complete dissimilarity of C_i and A (or the identity of C_i and B), and then $Sim(C_i, A) \to \infty$.

So, when applying the measure (8) to analyse our problem (Section 3), the values $0 \leq Sim(X, F) < 1$ are of interest – the measure (8) was constructed for

selecting objects which are more similar than dissimilar, and well-defined in the sense of possessing (or not) the attributes we are interested in (cf. Szmidt and Kacprzyk [16]).

Now we will show that a measure of similarity (8) is more powerful and gives more intuitively appealing results then a simple distance between an option C_i and the ideal option A (Figure 2), i.e. (8) gives more resonable results than the distance (9) alone.

Example 2. C_1 and C_2 are two options (with the coordinates (μ, ν, π)), $C_1 = (0, 0, 1)$, $C_2 = (0, 1, 0)$ so from (8) we have

$$l_{IFS}(C_1, A) = \frac{1}{2}(|1 - 0| + |0 - 0| + |0 - 1|) = 1 \tag{11}$$

$$l_{IFS}(C_2, A) = \frac{1}{2}(|1 - 0| + |0 - 1| + |0 - 0|) = 1 \tag{12}$$

which means that both options are the same (their distances to the the ideal option A are the same). But when we analyse thoroughly the options, we see that

- option C_1 fulfills at worst 0% of our demands, and at best 100% of our demands,
- option C_2 does not fulfil for sure any demands.

Formulas (11) – (12) inform us only about a sure fulfilment of the considered criteria (the lower bound) as for both C_1 and C_2 the lower bound of the fulfilment of the criteria is equal to 0, both options are treated as equal ones. In other words, while calculating the distance alone puts the equality sign between such different options like C_1 and C_2.

On the other hand, similarity (8) does clearly differentiate between options C_1 and C_2: $Sim(C_1, A) = 1$ whereas $Sim(C_4, A) =\rightarrow \infty$. □

Now we will show that the similarity measure (8) is a good tool to solve the problem formulated in Section 3.

The problem of finding an option M_i satisfying in the best way condition (5) can be solved by evaluating each option M_i

$$E(M_i) = Sim(A, M_i) = \min\{\max[Sim(A, C_j), Sim(A, C_k), \ldots \\ \ldots, Sim(A, C_p)], Sim(A, C_s)\} \tag{13}$$

Condition (13) means that for each M_i we look for the worst satisfied criterion W_i among C_j, C_k,..., and C_p, and next – we look for the better criterion between W_i and C_s). The *worst* means the least similar, and the *best* means the most similar.

The smallest value among $E(M_i)$, $i = 1, \ldots, m$ (13) points out the option which best satisfies condition (5).

Let us now come back to *Example 1.* For M_1 we calculate from (8)

$$Sim(A, C_1) = 1, \quad Sim(A, C_2) = 0.77, \quad Sim(A, C_3) = 0.8 \tag{14}$$

from (13) and (14) we have

$$E(M_1) = Sim(A, M_1) = \min[\max(1, 0.77), 0.8] = 0.8 \tag{15}$$

Repeating the above steps for the rest of options we obtain

$$E(M_2) = 0.77, \quad E(M_3) = 0.875, \quad E(M_4) = 0.714, \quad E(M_5) = 0.666 \tag{16}$$

$$E(N_1) = 1.143, \quad E(N_2) = 1.5 \tag{17}$$

So now, as opposed to the results given in [5], the options can be ordered: M_5 is the best among M_1 – M_5, and N_1 is better than N_2 (although both N_1 and N_2 are bad options which can be easy noticed as for both of them, for each criterion C_1, C_2, C_3, the non-membership degree is bigger than the membership degree).

The difference between the solution proposed in [5] and our method lies in the fact that evaluation of the options in [5] boils down to the consideration of the differences (7) between their membership and non-membership degrees only. It means that not all the information accessible is taken into account. Our solutions differs from [5] because:

- besides the membership and non-membership degrees we also make use of the information concerning a lack of knowledge about the membership and non-membership degrees of the options considered;
- we use the similarity measure (8) which compares each option with both the positive-ideal solution and the negative-ideal solution.

5 Conclusions

We proposed a solution to a multi-decision problem by using a similarity measure for intuitionistic fuzzy sets. The proposed method makes it possible to compare the options in a more intuitively appealing and human consistent way than it was proposed in [5].

References

1. Atanassov K. (1983), Intuitionistic Fuzzy Sets. VII ITKR Session. Sofia (Deposed in Centr. Sci.-Techn. Library of BAS, 1697/84) (in Bulgarian).
2. Atanassov K. (1999) Intuitionistic Fuzzy Sets: Theory and Applications. Physica-Verlag, Heidelberg and New York.
3. Atanassov K. and Gargov G. (1989), Interval-valued intuitionistic fuzzy sets. Fuzzy sets and Systems, 31, 3, 343–349.
4. Atanassov K., Taseva V., Szmidt E., Kacprzyk J. (2005) On the Geometrical Interpretations of the Intuitionistic Fuzzy Sets. In: K. T. Atanassov, J. Kacprzyk, M. Krawczak, E. Szmidt (Eds.): Issues in the Representation and Processing of Uncertain and Imprecise Information. EXIT, Warsaw 2005, 11–24.

5. Chen S.M., Tan J.M. (1994) Handling multi-criteria fuzzy decision making problems based on vague set theory. Fuzzy Sets and Systems 67, 2, 163–172.
6. Kahneman D. (2002) Maps of bounded rationality: a perspective on intuitive judgment and choice. Nobel Prize Lecture, December 8, 2002.
7. Liu H.W. Multi-criteria decision making methods based on intuitionistic fuzzy sets. In press.
8. Sutherland S. (1994) Irrationality. The Enemy Within. Penguin Books.
9. Szmidt E. and Baldwin J. (2003) New Similarity Measure for Intuitionistic Fuzzy Set Theory and Mass Assignment Theory. Notes on IFSs, 9, 60–76.
10. Szmidt E. and Kacprzyk J. (1996b) Remarks on some applications of intuitionistic fuzzy sets in decision making, Notes on IFS, 2, 22–31.
11. Szmidt E. and Kacprzyk J. (1998a) Group Decision Making under Intuitionistic Fuzzy Preference Relations. IPMU'98, 172–178.
12. Szmidt E. and J. Kacprzyk J. (2000) Distances between intuitionistic fuzzy sets, Fuzzy Sets and Systems, 114, 505–518.
13. Szmidt E. and Kacprzyk J. (2000) On Measures on Consensus Under Intuitionistic Fuzzy Relations. IPMU'2000, 1454–1461.
14. Szmidt E., Kacprzyk J. (2001) Entropy for intuitionistic fuzzy sets. Fuzzy Sets and Systems, 118, 467–477.
15. Szmidt E. and Kacprzyk J. (2002) Analysis of Agreement in a Group of Experts via Distances Between Intuitionistic Fuzzy Preferences. IPMU'2002, 1859–1865.
16. Szmidt E. and Kacprzyk J. (2004) Similarity of Intuitionistic Fuzzy Sets and Jaccard Coefficient. IPMU'04, 1405 – 1412.
17. Szmidt E. and Kacprzyk J. (2005) A new concept of a similarity measure for intuitionistic fuzzy sets and its use in group decision making. In V. Torra, Y. Narukawa, S. Miyamoto (Eds.): Modelling Decisions for Artificial Intelligence. LNAI 3558, Springer 2005, 272–282.
18. Szmidt E. and Kacprzyk J. (2006) Distances between intuitionistic fuzzy sets: straightforward approaches may not work. Accepted for publication.
19. Szmidt E. and Kacprzyk J. (2006) A Model of Case Based Reasoning Using Intuitionistic Fuzzy Sets. Accepted for publication.
20. Tasseva V., Szmidt E. and Kacprzyk J. (2005) On one of the geometrical interpretations of the intuitionistic fuzzy sets. Notes on IFS, Vol. 11, No. 3, 21–27.
21. Zadeh L.A. (1965) Fuzzy sets. Information and Control, 8, 338 – 353.

Additive Sequential Evolutionary Design of Experiments

B. Balasko, J. Madar, and J. Abonyi

Department of Process Engineering, University of Veszprem
Veszprem P.O.box 158.,H-8201 HUNGARY
abonyij@fmt.vein.hu
www.fmt.vein.hu/softcomp

Abstract. Process models play important role in computer aided process engineering. Although the structure of these models are *a priori* known, model parameters should be estimated based on experiments. The accuracy of the estimated parameters largely depends on the information content of the experimental data presented to the parameter identification algorithm. Optimal experiment design (OED) can maximize the confidence on the model parameters. The paper proposes a new *additive sequential evolutionary experiment design* approach to maximize the amount of information content of experiments. The main idea is to use the identified models to design new experiments to gradually improve the model accuracy while keeping the collected information from previous experiments. This scheme requires an effective optimization algorithm, hence the main contribution of the paper is the incorporation of Evolutionary Strategy (ES) into a new iterative scheme of optimal experiment design (AS-OED). This paper illustrates the applicability of AS-OED for the design of feeding profile for a fed-batch biochemical reactor.

1 Introduction

Process models play important role in computer aided process engineering since most of advanced process monitoring, control, and optimization algorithms rely on the process model. Unfortunately often some of the parameters of these models are not known *a priori*, so they must be estimated from experimental data.

The accuracy of these parameters largely depends on the information content of the experimental data presented to the parameter identification algorithm [1]. Optimal Experiment Design (OED) can maximize the confidence on model parameters through optimization of the input profile of the system. For parameter identification of different dynamic systems and models, this approach has been already utilized in several studies [2]- [6]. OED is based on an iterative algorithm where the optimal conditions of the experiments or the optimal input of the system depends on the current model, which parameters were estimated based on the result of the previously designed experiment. Consequently, experiment design and parameter estimation are solved iteratively, and both of them are based on nonlinear optimization of cost functions.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 324–333, 2006.
© Springer-Verlag Berlin Heidelberg 2006

That means in practice, the applied nonlinear optimization algorithms have great influence on the whole procedure of OED, because for nonlinear dynamical models the design of the experiment is a difficult task. This problem is usually solved by several gradient-based methods e.g. nonlinear least squares method or sequential quadratic programming. Several gradient computation methods are described in [7]. In [8] extended maximum likelihood theory is applied for optimizing the experiment conditions.

As a population-based effective optimization algorithm, this paper proposes the application of evolutionary strategy (ES). In [9] it has been shown that ES results in more satisfactory parameter values than classical sequential quadratic programming (SQP) or nonlinear least square (NLS) methods.

On the presented application example, a fed-batch biochemical reactor, some results were already shown in e.g. [10], but these results assume that model parameters or model structure are perfectly known.

The drawback of OED is that the experiment design uses only information from the current experiment and parameter identification relies only on one experiment while there are previous experiment information available. In this paper an *additive sequential evolutionary OED* was proposed, which uses the results of the previous experiment designs and parameter estimations. The aim of this paper is to illustrate the usefulness of AS-OED, independently from the model structure, hence a simplified monotonic Monode model was used.

The paper is organized into five sections: the first, second and third sections review the theoretical background of experiment design, additive design and evolutionary strategy, respectively. The fourth section presents an application example and finally conclusions are given in the fifth section.

2 Classical Optimal Experiment Design

The case study considered in this paper belongs to the following general class of process models:

$$\frac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}) \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t)), \tag{2}$$

where $\boldsymbol{u}$ is the vector of the manipulated inputs, $\boldsymbol{y}$ is the output (vector), $\boldsymbol{x}$ represents the state of the system, where $\boldsymbol{p}$ denotes the model parameters. The $\boldsymbol{p}$ parameters are unknown and should be estimated based on data taken from experiments. The estimation of these parameters is based on the minimization of the square error between the output of the system and the output of the model:

$$\min_{\boldsymbol{p}} \left[J_{mse}(\boldsymbol{u}(t), \boldsymbol{p}) = \frac{1}{t_{exp}} \int_{t=0}^{t_{exp}} (\boldsymbol{e}^T(t) \cdot \boldsymbol{Q}(t) \cdot \boldsymbol{e}(t)) dt) \right] \tag{3}$$

$$\boldsymbol{e}(t) = \tilde{\boldsymbol{y}}(\boldsymbol{u}(t)) - \boldsymbol{y}(\boldsymbol{u}(t), \boldsymbol{p}) \tag{4}$$

in which $\tilde{\boldsymbol{y}}(\boldsymbol{u}(t))$ is the output of the system for a certain $\boldsymbol{u}(t)$ input profile, and $\boldsymbol{y}(\boldsymbol{u}(t))$ is the output of the model for the same $\boldsymbol{u}(t)$ input profile with $\boldsymbol{p}$ parameters, $\boldsymbol{Q}$ is a user supplied square weighting matrix that represents the variance measurement error.

The basic element of the experiment design methodology is the Fisher information matrix $\boldsymbol{F}$, which combines information on (i) the output measurement error and (ii) the sensitivity of the model outputs $\boldsymbol{y}$ with respect to the model parameters:

$$\boldsymbol{F}(\boldsymbol{p}^0, \boldsymbol{u}(t)) = \frac{1}{t_{exp}} \int_{t=0}^{t_{exp}} (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{p}}(\boldsymbol{u}(t), \boldsymbol{p})|_{\boldsymbol{p}=\boldsymbol{p}^0})^T \cdot \boldsymbol{Q}(t) \cdot (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{p}}(\boldsymbol{u}(t), \boldsymbol{p})|_{\boldsymbol{p}=\boldsymbol{p}^0}) dt \quad (5)$$

The sensitivities are calculated based on the partial derivatives of the model parameters. As the true parameters $\boldsymbol{p}*$ are unknown during experiment design, the derivatives are calculated near to the so-called nominal parameters $\boldsymbol{p}^o$, which can be given by some initial guess, extracted from literature or estimated from the previous experiments. The optimal design criterion aims the minimization of a scalar function of the $\boldsymbol{F}$ matrix. several optimal criterion exist, we present D-optimal and E-optimal criterion suggested by Bernaerts et al. [1]:

– D-optimal criterion minimizes the determinant of the covariance matrix and thus minimizes the volume of the joint confidence region:

$$J_D = \min_{u(t)}(\det(\boldsymbol{F})) \quad (6)$$

– E-optimal criterion minimizes the condition number of $\boldsymbol{F}$, i.e. the ratio of the largest to the smallest eigenvalue of the Fisher matrix:

$$J_E = \min_{u(t)} \frac{\lambda_{max}(\boldsymbol{F})}{\lambda_{min}(\boldsymbol{F}} \quad (7)$$

If the $\boldsymbol{p}^o$ nominal parameters are far from the $\boldsymbol{p}*$ true parameters, convergence cannot be guaranteed after a first optimal design. So an iterative design scheme is needed to obtain convergence from $\boldsymbol{p}^o$ to $\boldsymbol{p}*$(Fig.2(a)).

Both the parameter estimation and the experiment design steps of this iterative scheme represent a complex nonlinear optimization problem, hence the effectiveness of the applied optimization algorithms have great influence on the performance of the whole procedure. The classical solution is to use nonlinear least squares (NLS) algorithm for parameter estimation eq. (3), and sequential quadratic programming (SQP) for the experiment design eq. (7). In this paper the application of evolutionary strategy (ES) is proposed for this purpose, which can be used for non-linear optimization problems.

The main drawback of the classical iterative approach is that the Fisher information matrix $\boldsymbol{F}$ contains information only about the current experiment regardless of the information content of the previous experiments, and the parameters are identified from this experiment. It is useful to include the information from previous experiments within the parameter estimation and the experiment design steps, too.

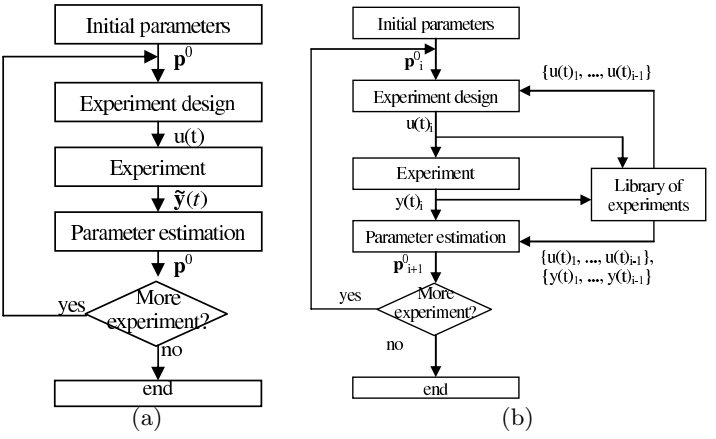


Fig. 1. Design schemes of the classical (a) and the Additive Sequential (b) OED method

3 Additive Sequential Experiment Design

The paper proposes a new additive sequential experiment design approach. The goal of the additive sequential design is to include all available information within the iterative design scheme, so the Fisher information matrix is calculated for the current AND the previous experiments. It means that in every iteration step, the new experiment is designed considering the previous experiments too:

$$\boldsymbol{F}_i(\boldsymbol{p}_i^0, \boldsymbol{u}_i(t)) = \sum_{k=1}^{i} \frac{1}{t_{exp,k}} \int_{t=0}^{t_{exp,k}} (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{p}}(\boldsymbol{u}_k(t), \boldsymbol{p})|_{\boldsymbol{p}=\boldsymbol{p}_i^0})^T \cdot \\ \cdot \boldsymbol{Q}(t) \cdot (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{p}}(\boldsymbol{u}_k(t), \boldsymbol{p})|_{\boldsymbol{p}=\boldsymbol{p}_i^0}) dt \tag{8}$$

where $\boldsymbol{u}_i$ is the input vector of the ith experiment with $t_{exp,i}$ experiment time. In this way, the parameters might be identified very effectively since every available experimental data is used to design a set of new informative (and independent) experiments. The iterative scheme of this Additive Sequential Experiment Design is shown in Fig.2(b).

The main advantage of this new approach is that in this way the experiment design becomes more robust and effective.

4 Evolutionary Strategy

This paper proposes the application of *evolutionary strategy* (ES) instead of the utilization of NLS and SQP. ES is a stochastic optimization algorithm that

uses the model of natural selection [11]. The advantage of ES is that it has proved particularly successful in problems that are highly nonlinear, that are stochastic, and that are poorly understood [11]. The design variables in ES are represented by n-dimensional vector $\boldsymbol{x}_j = [x_{j,1}, \ldots, x_{j,n}]^T \in R^n$, where $\boldsymbol{x}_j$ represents the j-th potential solution, i.e. the j-th the member of the population. The mutation operator adds $z_{j,i}$ normal distributed random numbers to the design variables: $x_{j,i} = x_{j,i} + z_{j,i}$, where $z_{j,i} = N(0, \sigma_{j,i})$ is a random number with $\sigma_{j,i}$ standard deviation. To allow for a better adaptation to the objective functions's topology, the design variables are accompanied by these standard deviation variables, which are so-called strategy parameters. Hence the $\boldsymbol{\sigma}_j$ strategy variables control the step size of standard deviations in the mutation for j-th individual. So an ES-individual $\boldsymbol{a}_j = (\boldsymbol{x}_j, \boldsymbol{\sigma}_j)$ consists of two components: the design variables $\boldsymbol{x}_j = [x_{j,1}, \ldots, x_{j,n}]^T$ and the strategy variables $\boldsymbol{\sigma}_j = [\sigma_{j,1}, \ldots, \sigma_{j,n}]^T$. Before the design variables are changed by mutation operator, the standard deviations $\boldsymbol{\sigma}_j$ are mutated using a multiplicative normally distributed process:

$$\sigma_{j,i}^{(t)} = \sigma_{j,i}^{(t-1)} \exp(\tau' N(0,1) + \tau N_i(0,1)) \,. \tag{9}$$

The $\exp(\tau' N(0,1))$ is a global factor which allows an overall change of the mutability, and the $exp(\tau N_i(0,1))$ allows individuals to change of their mean step sizes $\sigma_{j,i}$. So τ' and τ parameters can be interpreted as global learning rates. Schwefel suggests to set them as [12]:

$$\tau' = \frac{1}{\sqrt{2n}},\ \tau = \frac{1}{\sqrt{2\sqrt{n}}} \,. \tag{10}$$

Throughout this work discrete recombination of the object variables and intermediate recombination of the strategy parameters were used:

$$x'_{j,i} = x_{F,i} \quad or \quad x_{M,i} \tag{11}$$

$$\sigma'_{j,i} = (\sigma_{F,i} + \sigma_{M,i})/2 \,, \tag{12}$$

where F and M denotes the parents, j is index of the new offspring.

The Evolutionary Strategy function in Matlab environment has three important parameters that one has to adjust carefully in order to find the most reliable solution with the least computation time: (i) number of generations, (ii) size of a population in a generation and (iii) number of individuals with the best fit values, which appear unchanged in the next generation.

5 Application Example

This paper illustrates the applicability of the proposed approach for the design of a feeding profile of a fed-batch biochemical reactor with monotonic kinetics.

Prior knowledge on the microbial dynamics is generally lacking, the parameters of the applied kinetic model are usually determined by using a method

which minimizes the difference between the measured response of the reactor and the predicted response of the model. The following equation describes the mass balance of the reactor:

$$\frac{d}{dt}\begin{bmatrix} S \\ X \\ V \end{bmatrix} = \begin{bmatrix} -\sigma X \\ \mu X \\ 0 \end{bmatrix} + \begin{bmatrix} C_{S,in} \\ 0 \\ 1 \end{bmatrix} u \tag{13}$$

where S is the mass of the substrate [g], X is the mass of the micro-organism [g DW], V is the volume [L], u is the inlet flowrate [L/h], $C_{S,in} = 500$ g/L is the substrate concentration in the inlet feed, $\sigma = \mu/Y_{X/S} + m$ is the specific substrate consumption rate, where, $Y_{X/S} = 0.47$ g DW/g, $m = 0.29$ g/g DW h, while μ [1/h] is the kinetic rate. The initial conditions: $S(t = 0) = 500$ g, $X(t = 0) = 10.5$ g DW, $V(t = 0) = 7$ L. The maximum volume is $V_{max} = 10$ L, and the maximum inlet flowrate is $u_{max} = 0.3$ L/h.

The following monotonic kinetic (*Monode*) model of the μ kinetic rate was considered:

$$\mu^M(C_S) = \mu_{max}\frac{C_S}{K_S + C_S} \tag{14}$$

The system was simulated with $\mu_{max} = 0.1$ 1/h and $K_S = 1$ g/L as real parameter values, i.e. $\boldsymbol{p}^* = [0.1\ 1]^T$. The goal was to find the unknown parameters of the model: μ_{max} and K_S (the other parameters were assumed to be accurately known). The μ_{max} and K_S parameters were estimated by optimization, see eq. (3).

It was assumed that only the substrate concentration C_S is measured. Consequently, in this application example, the system output is $\tilde{y} = \frac{\tilde{S}[g]}{\tilde{V}[L]}$, which was generated by the simulation of (13) and (14).

We applied the iterative OED methodology to design the feeding profile $u(t)$ with E-optimal criterion equations (5) (8) and (7). It has been shown in [9] that ES seems to be the best optimization algorithm for this non-linear problem. The aim of the paper is to prove that an additive (memory-effect type) evolutionary experiment design can accord better results for this application.

The parameter error (Err) was defined as the sum of the absolute difference between the ending output parameter results and the 'true' μ_{max}=0.1 1/h and K_S=1 g/L values, normalized with the true values(Eq.(15)) where n_{par} denotes the number of optimized parameters.

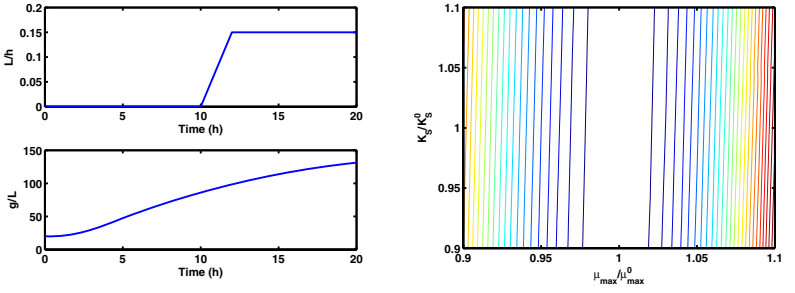
$$Err = \sum_{i=1}^{n_{par}} \left| \frac{\boldsymbol{p}^0 - \boldsymbol{p}^*}{\boldsymbol{p}^*} \right| \tag{15}$$

Both the classical and the additive evolutionary ED was initialized with the $p_{init} = [0.15\ 0.5]$ parameter vector (50 percent error for each), 3 independent runs were made with 9 iteration during each of them. The length of one experiment

was 20 h, the sample time was 1 h. Ten input profile values were optimized with linear extrapolation between them.

The parameters of ES function were adjusted to 40 generations with a population size of 25 individuals and the best 10 individuals appear unchanged in the next generation. With all these experimental parameters the runtime of the algorithm is ca. 2 minutes.

Note, that our task here is not to maximize the system output, i.e. the biomass concentration but to find an input profile which maximizes the precision of the parameter estimation. For comparison, Fig.2 (a) shows a manually selected input step profile and the response of the system. The ES optimized parameters for this profile are $p_{manual} = [0,0997\ 0,8968]$ with an Err value of 0.1053. The parameter space does not converge to a global minima if this input profile is selected as shown in Fig.2 (b). Identification cost function is defined as the square error of the model output with respect to its parameters around the p^0 nominal parameters.

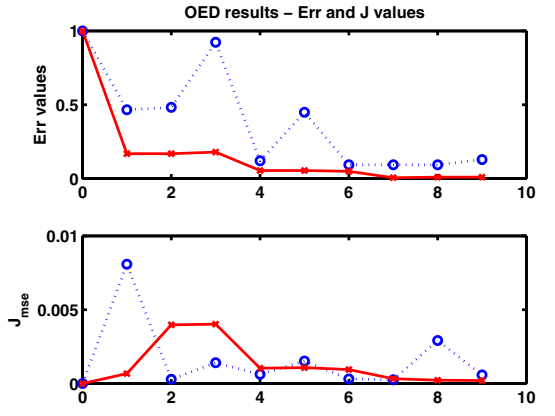


(a) (b)

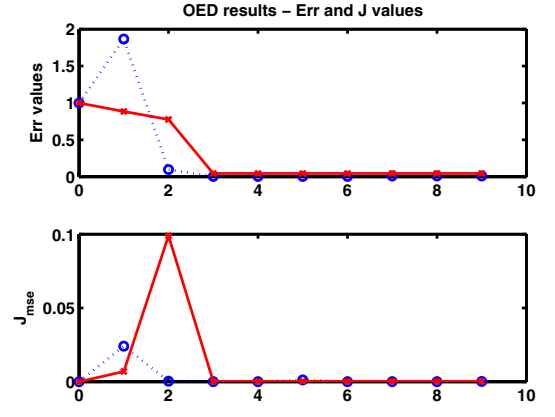
Fig. 2. Manually selected input profile and system output (a) and contour plot of the identification cost (b)

The results are shown in Table 1., Fig.3. and Fig.4. Table 1. shows the sum of Err values of the ending parameters for the 3 independent runs, and their mean values; p_{init} is shown on the figure as the 0th experiment.

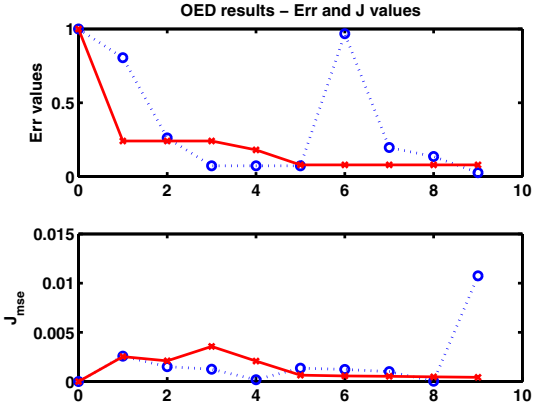
As one can see, the additive sequential evolutionary experiment design results in better parameters at the end of the experiments and almost in every iteration step, it is more robust and computationally just a bit more expensive. The iterative sequential design has its uncertainty in the output because of the large searching space and the lack of information from previous experiments. As Fig.3. shows, in additive design, less iteration cycles and evolutionary computation would result also in reliable parameters, hence computation time excess could be spared without any quality loss in parameter estimation. E-optimized input profiles are presented on Fig.4. for the runs with the best ending parameter estimations.



(a)



(b)

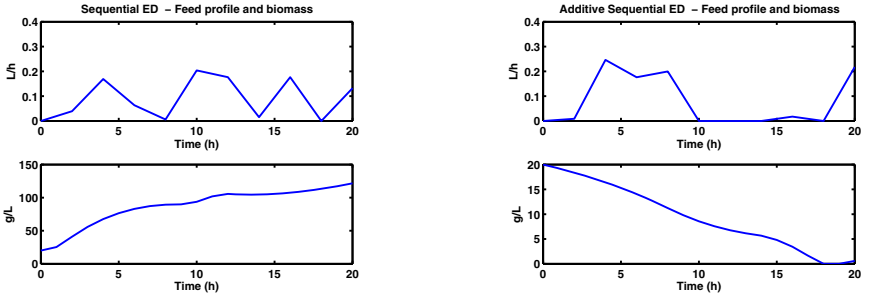


(c)

Fig. 3. *Err* and J_{mse} values (dotted line with 'o' marker for classical and continous line with 'x' marker for AS-OED) for the three independent runs

Table 1. Sum of error of the ending parameters and mean values

	1st run	2nd run	3rd run	$\overline{Err}$
Sequential Ev. Design	0.1283	0.0094	0.0256	0.0544
Additive Seq.Ev.Design	0.0091	0.0449	0.0785	0.0441



(a) (b)

Fig. 4. E-Optimized input profiles and system outputs generated by (a) classical sequential OED and (b) Additive Sequential OED

6 Conclusion

The aim of this paper was to develop a robust and effective additive evolutionary experiment design strategy and to demonstrate its usefulness on an industrial example. This approach uses Evolutionary Strategy for both the experiment design and the parameter identification steps of the iterative scheme. Additive Sequential Evolutionary Experiment Design uses information from the previous experiments and although it is computationally just a little more expensive than the ES aided classical Optimal Experiment Design because Fisher matrices and the input profiles of the previous experiments already exist so they do not need to be calculated again. Computation time could be lowered by decreasing the number of iteration cycles what does not worsen parameter estimation quality. It has been proven that this new approach results in more reliable and confidential values of the identified parameters.

References

1. BERNAERTS, K., SERVAES, R.D., KOOYMAN, S. and VERSYCK, K.J. and VAN IMPE, J.F.: Optimal temperature design for estimation of the Square Root model parameters: parameter accuracy and model validity restrictions, Int. Jour. of Food Microbiology, 73, 145-157, 2002
2. BERNAERTS, K., VAN IMPE, J.F.: Optimal dynamic experiment design for estimation of microbial growth kinetics at sub-optimal temperatures: Modes of implementation, Simulation Modelling Practice and Theory, 13, 129138, 2005

3. VERSYCK, K.J., BERNAERTS, K., GEERAERD, A.H., VAN IMPE, J.F.: Introducing optimal experimental design in predictive modeling: A motivating example, International Journal of Food Microbiology 51, 3951, 1999
4. BERNAERTS, K., GYSEMANS, K.P.M., MINH, T.N., VAN IMPE, J.F.: Optimal experiment design for cardinal values estimation: guidelines for data collection, International Journal of Food Microbiology, 100, 153 165, 2005
5. CHEN, B.H., BERMINGHAM, S., NEUMANN, A.H., KRAMER, H.J.M., and ASPREY, S.P.: On the Design of Optimally Informative Experiments for Dynamic Crystallization Process Modeling, Ind. Eng. Chem. Res. 43, 4889-4902, 2004
6. COHN, D.A.: Neural Network Exploration Using Optimal Experiment Design, Neural Networks, Vol. 9, Issue 6, 1071-1083, 1996
7. POINT, N. VANDE WOUWER, A., and REMY, M.: Practical Issues in Distributed Parameter Estimation: Gradient Computation and Optimal Experiment Design, Control Engineering Practice, Vol. 4, Issue 11, 1553-1562, 1996
8. A.F. EMERY, A.F., NENAROKOMOV, A.V., FADALE, T.D.: Uncertainties in parameter estimation: the optimal experiment design, International Journal of Heat and Mass Transfer, 43, 3331-3339, 2004
9. MADAR, J., SZEIFERT, F. AND ABONYI, J.: Evolutionary Strategy in Iterative Experiment Design, the Hungarian Journal of Industrial Chemistry, Special Issue on Recent advances in Computer Aided Process Engineering
10. SMETS, I.Y.M, VERSYCK, K.J.E and VAN IMPE, J.F.: Optimal control theory: A generic tool for identification and control of (bio-)chemical reactors, Annual Reviews in Control, 26, 57-73, 2002
11. MADAR, J. and ABONYI, J.: Evolutionary Algorithms, Chapter 2.10. in Instrument Engineers' Handbook, 4th Edition, Volume 2 - Process Control, editor: B. Liptak, CRC Press, 2005
12. SCHWEFEL, H.: Numerical optimization of computer models. Wiley, Chichester, 1995

A New Inter-island Genetic Operator for Optimization Problems with Block Properties

Wojciech Bożejko[1] and Mieczysław Wodecki[2]

[1] Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
Janiszewskiego 11-17, 50-372 Wrocław, Poland
wojciech.bozejko@pwr.wroc.pl
[2] Institute of Computer Science
University of Wrocław
Przesmyckiego 20, 51-151 Wrocław, Poland
mwd@ii.uni.wroc.pl

Abstract. Combinatorial optimization problems of scheduling belongs in most cases to the NP-hard class. In this paper we propose a very effective method of construct parallel algorithms based on the island model of coevolutionary algorithm. We apply block properties, which enable the inter-island genetic operator to distribute calculations and shorten communication between processors.

1 Introduction

Many methods of algorithms construction consist in looking through (directly or indirectly) either all or a part of a set of feasible solutions. Such a mechanism is based on generating from a current (base) solution a next solution, or a set of solutions (so called neighborhood), from which the best solution is chosen. This solution is the base solution in the next iteration. Such a mechanism can be met (among others) in Branch and Bound (B&B) method and in many other algorithms which deal with improving the solution, as well as in the best (nowadays) approximate algorithms, metaheuristics: tabu search and simulated annealing methods. The same method is used in the path-relinking method, which is used in the local search genetic operators, such as Multi – Step Crossover Fusion (MSXF) of Reeves and Yamada [19]. Quality of these algorithm's solutions depends on: the number of iteration, the method of neighborhood describing and its reviewing. The time of computations can be shorten by its realization in a multiprocessor environment. Unfortunately parallelization of the sequential algorithms directly (for example by using parallel compiler) does not give satisfactory speedup. In this paper we propose some new elements of parallel local search algorithms. Partitioning solution (permutation) into blocks (subpermutations – subsequences of jobs) enables to decrease the neighborhood size and its division into separated subsets. It makes possible its generating and reviewing independently on a parallel machine. Many classic scheduling problems have blocking partitioning property, i.e.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 334–343, 2006.
© Springer-Verlag Berlin Heidelberg 2006

1. Flow shop problem – subsets of jobs from the same machine on critical path are blocks, see Nowicki and Smutnicki [16], Grabowski and Wodecki [9].
2. Job shop problem – subsets of jobs from the same machine on critical path are blocks, see Nowicki and Smutnicki [17], Grabowski and Wodecki [10] .
3. Classic single machine total weighted tardiness problem (TWTP) – subsets of jobs made on time and made after deadline are blocks, see Bożejko, Grabowski and Wodecki [5].
4. Single machine earliness/tardiness scheduling problem – subsets of jobs made on time, made before the earliest moment of finishing and made after deadline are blocks (three types of blocks; considered in this paper).

We consider a single machine earliness/tardiness problem in this paper. Methods proposed here can be directly applied to all of the problems presented above.

Many authors have studied the earliness and tardiness (E/T) problem which was first introduced by Kanett [13]. An useful review of early/tardy scheduling is provided in Baker, Scudder [1] and the book T'kindt and Billant [22]. In paper [1] Baker and Scudder proved that there can be an idle time in an optimal solution (jobs need not be processed directly one after another). Solving the problem amounts to establishing a sequence of jobs and their starting times. Hoogeven and van de Velde [12] proposed an algorithm based on a branch and bound method. Because of the exponentially growing computation's time, this algorithm can be used only to solve cases, where the number of jobs is not bigger than 20. Therefore, in practice almost always the approximate algorithms are used. The best of them are based on the artificial intelligence methods. Calculations are performed in two stages:

1. determining the scheduling of jobs (with no idle times).
2. establishing jobs' optimal starting times.

There is an algorithm in the paper of Wan and Yen [24] based on this scheme. To determine scheduling a tabu search algorithm is used. Algorithms for the optimal job sequencing are relatively less studied. Szwarc [20] proposed a Branch and Bound algorithm based on some adjacent ordering conditions for jobs with distinct penalty weights. Lee and Choi [14] proposed a genetic algorithm and Yano and Kim [25] studied several dominance properties for sequencing jobs with penalty weights proportional to processing times.

The most important results for unrestrictive and restrictive scheduling E/T problems are presented in the papers of Bank and Werner [2], Gordon et al. [11], Valente and Alves [23], and Feldmann and Biskup [8].

In this paper we consider *TWET* problem, additionally assuming that the machine begins the execution of jobs in time zero and it works with no idle (*TWET-no-idle* problem). We present new elements of the neighborhood search method. Partition of permutation into blocks (subsequences) and replacing sets of moves with its representatives significantly decrease size of the neighborhood, eliminating 'bad' moves and speeding up the calculations. Computational experiments shows that such a neighborhood is not bigger (in average) than the size of classical neighborhood generated only by the insert moves. We use a

parallel computer for executing coevolutionary genetic algorithm with inter-island genetic operator based on local search procedure with blocks.

2 Blocks in Solutions

For the *TWET-no-idle* problem, each schedule of jobs can be represented by permutation $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$ of the set of jobs J. Let $\Phi(n)$ denotes the set of all such permutations. The total cost $\pi \in \Phi(n)$ is $F(\pi) = \sum_{i=1}^{n} f_{\pi(i)}(C_{\pi(i)})$, where $C_{\pi(i)}$ is a completed time of the job $\pi(i)$, $\quad C_{\pi(i)} = \sum_{j=1}^{i} p_{\pi(j)}$. The job $\pi(i)$ is considered *early* if it is completed before its earliest moment of finishing ($C_{\pi(i)} < e_{\pi(i)}$), *on time* if $e_{\pi(i)} \leq C_{\pi(i)} \leq d_{\pi(i)}$, and *tardy* if the job is completed after its due date (i.e. $C_{\pi(i)} > d_{\pi(i)}$). An expression $u_i E_i + w_i T$ is the *cost* of the job execution, where u_i and w_i are the nonnegative coefficients of a goal function. The objective is to find a sequence of jobs that minimize the following non-regular function:

$$\sum\nolimits_{i=1}^{n} (u_i E_i + w_i T_i).$$

This problem became more popular after the introduction of JIT manufacturing philosophy, where jobs are desired to be completed as close as possible to their due dates. In the classical scheduling notation in literature the problem is denoted by $1||\sum (u_i E_i + w_i T_i)$ and it belongs to a strongly NP-hard class (if we assume that $u_i = 0$, $i = 1, 2, \ldots, n$, we obtain a problem $1||\sum w_i T_i$, which is strongly NP-hard, see Lenstra et al. [15]).

Each permutation $\pi \in \Phi(n)$ is decomposed into subpermutations (subsequences of jobs) $\Omega = [B^1, B^2, \ldots, B^v]$ called *blocks* in π, each of them contains the jobs, where:

1. $B^i = (\pi(a^i), \pi(a^i + 1), \ldots, \pi(b^i - 1), \pi(b^i))$, and $a^i = b^{i-1} + 1, \;\; 1 \leq i \leq v, \;\; a^0 = 0, \; b^v = n.$
2. All the jobs $j \in B_i$ satisfy the following condition:
   $e_j > C_{\pi(b^i)}$, or (C1)
   $e_j \leq C_{\pi(b^{i-1})} + p_j$ and $d_j \geq C_{\pi(b^i)}$, or (C2)
   $d_j < C_{\pi(b^{i-1})} + p_j$. (C3)
3. B_i are maximal subsequences of π in which all the jobs satisfy either Condition $C1$ or Condition $C2$ or Condition $C3$.

By definition, there exist three types of blocks implied by either C1 or C2 or C3. To distinguish them we will use the *E-block*, *O-block* and *T-block* notions respectively.

Theorem 1. *For any permutation $\pi \in \Phi(n)$ there are partitions into blocks (subsequences), such that every of them is:*

i. *$\mathcal{E}$-block, or*
ii. *$\mathcal{O}$-block, or*
iii. *$\mathcal{T}$-block.*

Proof. Let us assume that a permutation π is partitioned into blocks. From definition of $\mathcal{E}$, $\mathcal{O}$ and $\mathcal{T}$ blocks, if $C_{\pi(1)} < e_{\pi(1)}$, then $\pi(1)$ belongs to the first $\mathcal{E}$-block in the opposite case if $C_{\pi(1)} > d_{\pi(1)}$, then $\pi(1)$ belongs to the first $\mathcal{T}$-block or (on the contrary) to the first $\mathcal{O}$-block. We sequentially consider jobs $\pi(2), \pi(3), \ ..., \ \pi(n)$. For the job $\pi(i), \ \ 2 \leq i \leq n$, let the previous job $\pi(i-1)$ belongs to a block B.

Let us assume, that $C_{\pi(i)} < e_{\pi(i)}$. If

1. B is $\mathcal{O}$ or $\mathcal{T}$-block, then $\pi(i)$ is the first job of the next $\mathcal{E}$-block.
2. B is $\mathcal{E}$-block and $B \cup \{\pi(i)\}$ is $\mathcal{E}$-block, then job $\pi(i) \in B$, and on the contrary $\pi(i)$ is the first job of the next $\mathcal{E}$-block.

We can consider the case of $C_{\pi(i)} > d_{\pi(i)}$ and $e_{\pi(i)} \leq C_{\pi(i)} \leq d_{\pi(i)}$ similarly. ■

The definition of blocks and Theorem 1 presents, that after partitioning of a permutation:

1) every job belongs to some $\mathcal{E}$ or $\mathcal{O}$ or $\mathcal{T}$ block,
2) blocks are disjoin sets of jobs.

For any block B in a partition Ω of permutation $\pi \in \Phi(n)$, let

$$F_B(\pi) = \sum_{i \in B} (u_i E_i + w_i T_i).$$

Therefore, the value of a goal function takes the form of

$$F(\pi) = \sum_{i=1}^{n} (u_i E_i + w_i T_i) = \sum_{B \in \Omega} F_B(\pi).$$

If B is a $\mathcal{T}$-block, then every job which belongs to it is early. Therefore, in the permutation π, an optimal sequence of jobs within B (which is minimizing $F_B(\pi)$) can be obtained using the well-known Weighted Shortest Processing Time (*WSPT*) rule proposed by Smith [8]. The *WSPT* rule creates an optimal sequence of jobs in the non-increasing order of the ratios w_j/p_j. Similarly, if B is an $\mathcal{E}$-block, then an optimal sequence of jobs can be obtained using the Weighted Longest Processing Time (*WLPT*) rule which creates a sequence of jobs in non-decreasing order of the ratios u_j/p_j.

Partition Ω of the permutation π is *ordered*, if there are jobs scheduled by the *WSPT* rule in any $\mathcal{T}$-block and jobs scheduled by the *WLPT* rule in any $\mathcal{E}$-block.

Lemma 1. *If a permutation $\pi \in \Phi(n)$ is ordered, then changing the order of jobs in any block does not generate permutation with less cost of the goal function.*

Proof. Let $B = (\pi(a), \pi(a+1), \ ... \ , \pi(b)), \ 1 \leq a < b \leq n$, be a block in a partition of ordered permutation $\pi \in \Phi(n)$. Let us assume that permutation β was generated from π by changing the order of jobs in block B. Therefore

$$\beta(i) = \pi(i), \ \ i = 1, 2, \ ... \ , a-2, \ a-1, b+1, b+2, \ ... \ , n$$

and sets of jobs fulfill the equality

$$\{\beta(j):\ j=a,a+1,\ \ldots,b\}=\{\pi(j):\ j=a,a+1,\ \ldots,b\}.$$

We should consider two cases:

1. B is $\mathcal{E}$-block. If jobs from the set $\{\beta(a),\beta(a+1),\ \ldots,\beta(b)\}$ does not fulfill the *WLPT* rule in permutation β, then $F(\beta)\geq F(\pi)$.
2. B is $\mathcal{O}$-block. From the definition of $\mathcal{O}$-block, every job $\beta(j),\ j=a,a+1,\ \ldots,b$ is on time in permutation β, so $F(\beta)=F(\pi)$.
3. B is $\mathcal{T}$-block. If jobs from the set $\{\beta(a),\beta(a+1),\ \ldots,\beta(b)\}$ does not fulfill the *WSPT* rule in permutation β, then $F(\beta)\geq F(\pi)$. ■

Next theorem is a base of the neighborhood's construction in the local search algorithms.

Theorem 2. *For each ordered permutation $\pi\in\Phi(n)$, if a permutation $\beta\in\Phi(n)$ was obtained from π by any interchange of its elements and*

$$F(\beta)<F(\pi),$$

then in the permutation β at least one job of some block of π was moved before the first or after the last job of this block.

Proof. Let $[B^1,B^2,\ \ldots,B^v]$ be a partition of ordered permutation $\pi\in\Phi(n)$ into blocks. Each block is a subsequence of jobs

$$B^i=(\pi(a^i),\pi(a^i+1),\ \ldots,\pi(b^i)),\quad i=1,2,\ \ldots,v,$$

$$1\leq a^1\leq b^1<a^2\leq b^2<,\ \ldots,<a^v\leq b^v.$$

By

$$Y^i(\pi)=\{\pi(a^i),\pi(a^i+1),\ \ldots,\pi(b^i)\}$$

we represent the set of jobs from the block B^i.

Let permutation $\beta\in\Phi(n)$ and $F(\beta)<F(\pi)$. Let us assume on the contrary, that in permutation β any job from any block $B^1,B^2,\ \ldots,B^v$ has not been moved before the first or after the last job of this block. Therefore

$$Y^i(\pi)=Y^i(\beta),\quad i=1,2,\ \ldots,v.$$

Then for $i=1,2,\ldots,v$ subsequences $(\pi(a^i),\pi(a^i+1),\ \ldots,\pi(b^i))$ in permutation π and $(\beta(a^i),\beta(a^i+1),\ \ldots,\beta(b^i))$ in β are permutations of the same subset of jobs $\{\pi(a^i),\pi(a^i+1),\ \ldots,\pi(b^i)\}$. Lemma 1 presents, that $F(\beta)\geq F(\pi)$, which contradicts the assumption on the contrary. ■

Let us notice that Theorem 1 provides the necessary condition to obtain a permutation β from π such, that $F(\beta)<F(\pi)$.

Let $\Omega=[B_1,B_2,\ \ldots,B_v]$ be an ordered partition of the permutation $\pi\in\Phi(n)$ into blocks. If a job $\pi(j)\in B_i$ $(B_i\in\Omega)$, therefore existing moves, which can

improve the goal function value, consist in reordering a job $\pi(j)$ before the first or after the last job of this block. Let $\mathcal{M}_j^{bf}$ and $\mathcal{M}_j^{af}$ be sets of such moves (obviously $\mathcal{M}_1^{bf} = \mathcal{M}_v^{af} = \oslash$). Therefore, the neighborhood of the permutation $\pi \in \Phi(n)$ has the form of

$$\mathcal{M}(\pi) = \bigcup_{j=1}^{n} \mathcal{M}_j^{bf} \cup \bigcup_{j=1}^{n} \mathcal{M}_j^{af}. \tag{1}$$

A move $\hat{m}$ is a *representative* of moves from the set $M \subseteq \mathcal{M}$, if

$$\forall\, m \in M,\ \ F(m(\pi)) \geq F(\hat{m}(\pi)).$$

Sets of moves $\mathcal{M}_x^{bf}$ and $\mathcal{M}_x^{af}$ are presented in Fig. 1. Looking inside the block, the job $\pi(f_k)$ is the first one, and $\pi(l_k)$ is the last one.

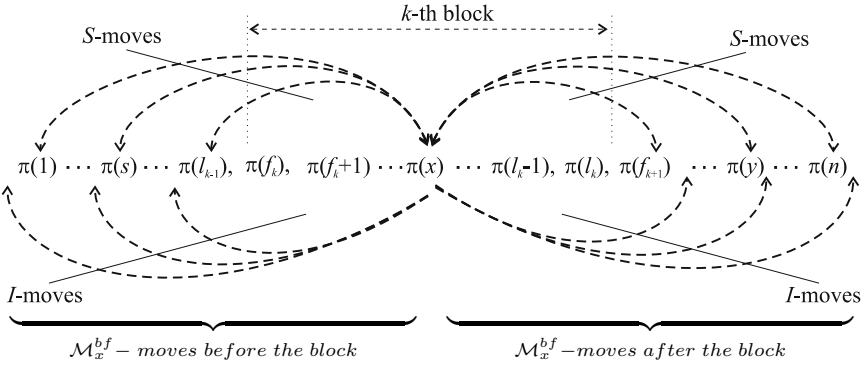


Fig. 1. Moves of the job $\pi(j)$

3 Parallel Genetic Algorithm

There are three basic types of parallelization strategies which can be applied to the genetic algorithm: global, diffusion model and island model (migration model).

Algorithms based on the island model divide the population into a few subpopulations. Each of them is assigned to a different processor which performs a sequential genetic algorithm based on its own subpopulation. The crossover involves only individuals within the same population. Occasionally, the processor exchanges individuals through a migration operator. The main determinants of this model are: (1) size of the subpopulations, (2) topology of the connection network, (3) number of individuals to be exchanged, (4) frequency of exchanging.

The island model of parallel genetic algorithm is characterized by a significant reduction of the communication time, compared to the global model (with distributed computations of the fitness function only). Shared memory is not required, so this model is also more flexible. Bubak and Sowa [7] developed an

implementation of the parallel genetic algorithm for the TSP problem using the island model. Belding [3] extended a previous work on the distributed genetic algorithm of Tanese [21], focusing on migration rates and intervals.

Below, a parallel genetic algorithm is proposed. The algorithm is based on the island model of parallelism. There is the MSXF (Multi – Step Crossover Fusion) operator used to extend the process of researching for better solutions of the problem. MSXF has been described by Reeves and Yamada [19]. Its idea is based on local search, starting from one of the parent solutions, to find a new good solution where the other parent is used as a reference point. Additionally, block properties were used to make the search process more effective – to prevent changes inside the block which are unprofitable from the fitness function's point of view. Such a proceedings is consistent with an idea of not making changes between genes of different chromosomes. In such a way a MSXF+B (MSXF with blocks) operator was created.

The neighborhood $N(\pi)$ of the permutation (individual) π is defined as a set of new permutations that can be achieved from π by exactly one adjacent pairwise exchange operator which exchanges the positions of two adjacent jobs of a problem's solution connected with permutation π. The distance measure $d(\pi,\sigma)$ is defined as a number of adjacent pairwise exchanges needed to transform permutation π into permutation σ. Such a measure is known as Kendall's τ measure.

Algorithm 1. Multi-Step Crossover Fusion with Blocks (MSXF+B)
Let π_1, π_2 be parent solutions. Set $x = q = \pi_1$;
repeat
For each member $y_i \in N(\pi)$, calculate $d(y_i, \pi_2)$;
Sort $y_i \in N(\pi)$ in ascending order of $d(y_i, \pi_2)$;
repeat
Select y_i from $N(\pi)$ with a probability inversely proportional to the index i; Calculate $C_{sum}(y_i)$;
Accept $y_i$with probability 1 if $C_{sum}(y_i) \leq C_{sum}(x)$, and with probability $P_T(y_i) = exp((C_{sum}(x) - C_{sum}(y_i)) / T)$ otherwise (T is temperature);
Change the index of y_i from i to n and the indices of y_k, $k = i+1,...,n$ from k to $k-1$;
until y_i is accepted;
$x \leftarrow y_i$; **if** $C_{sum}(x) < C_{sum}(q)$ **then** $q \leftarrow x$;
until *some termination condition is satisfied*;
q is the offspring.

In our implementation, MSXF+B is an inter-island (i.e. inter-subpopulation) crossover operator which constructs a new individual by using the best individuals of different islands connected with subpopulations on different processors. The condition of termination consisted in exceeding of 100 iterations by the MSXF+B function.

Algorithm 2. Parallel genetic algorithm

parfor $j = 1, 2, ..., p$ { *p is number of processors* }
  $i \leftarrow 0$;
  $P_j \leftarrow$ random subpopulation connected with processor j;
  $p_j \leftarrow$ number of individuals in j subpopulation;
  repeat
    Selection(P_j, P'_j); Crossover(P'_j, P''_j); Mutation(P''_j);
    if (k mod $R = 0$) **then** {*every R iteration*}
      $r :=$ *random*(1, p); MSXF+B($P'_j(1), P_r(1)$);
    end if;
    $P_j \leftarrow P''_j$; $i \leftarrow i + 1$;
    if *there is no improvement of the average* C_{sum} **then** {*Partial restart*}
      $r :=$ *random*(1,p);
      Remove $\alpha = 90$ percentage of individuals in subpopulation P_j.;
      Replenish P_j by random individuals;
    end if;
    if (k mod $S = 0$) **then** {*Migration*}
      $r :=$ *random*(1,p);
      Remove $\beta = 20$ percentage of individuals in subpopulation P_j;
      Replenish P_j by the best individuals from subpopulation P_r
      taken from processor r;
    end if;
  until *Stop_Condition*;
end parfor

The frequency of communication between processors (MSXF+B operator and migration) is very important for the parallel algorithm performance. It must not be too frequent because of the relative long time of communication between processors, comparing to the time of communication inside the program of a one processor. In this implementation the processor gets new individuals quite rarely, every $R = 20$ (MSXF+B operator) or every $S = 35$ (migration) iterations.

4 Computer Simulations

The algorithm was implemented in the Ada95 language and ran on 4-processors Sun Enterprise 4 x 400 MHz under the Solaris 7 operating system. Tasks of the Ada95 language were executed in parallel as system threads. Tests were based on 125 instances with 40,50 and 100 jobs taken from the OR-Library [18]. The results were compared to the best known, also taken from [18].

The computational results can be found in Table 1. The number of iterations was counted as a sum of iterations on processors, and was permanently set to 800. For example, 4-processor implementations make 200 iterations on each of the 4 processors, so we can obtain comparable costs of computations. As we can see, the parallel versions of the algorithm have much better results of the average and maximal relative deviation from the optimal (or the best known) solutions,

Table 1. Relative deviation of solutions of sequence and parallel genetic algorithms compared to the best known solutions

n	1 processor		4 processors	
	aver. dev.	max. dev.	aver. dev.	max. dev.
40	2.907	99.963	0.057	1.534
50	4.035	167.576	0.064	1.362
100	0.005	1.054	0.004	0.103
average	**2.317**	**89.531**	**0.042**	**0.999**

working (parallel) in a shorter time. Because of a small cost of the communication the speedup parameter of the parallel algorithms is almost linear.

5 Conclusions

We have discussed a new approach to the optimization problems with block properties based on the new inter-island genetic operator for the parallel asynchronous coevolutionary algorithm. Compared to the sequential algorithm, the parallelization shortens the computation's time and it improves quality of the obtained solutions. The advantage of the parallel algorithm is especially visible for large cases of the problem.

References

1. Baker K.R., Scudder C.D., Sequencing with earliness and tardiness penalties: a review, Operations Research **38** (1990) 22–36.
2. Bank J., Werner F., Heuristic algorithm for unrelated parallel machine scheduling with a common due date, relase dates, and linear earliness and tardiness penalties, Mathematical and Computer Modelling **33** (2001) 363–383.
3. Belding T.C, The distributed genetic algorithm revisited, Proceedings of the Sixth International Conference on Genetic Algorithms (L. Eschelman, ed.), Morgan Kaufmann (1995) 114–121
4. Bożejko W., Wodecki M., Parallel genetic algorithm for minimizing total weighted completion time, Lecture Notes in Computer Science No. **3070**, Springer Verlag (2004) 400–405.
5. Bożejko W., Grabowski J., Wodecki M., Block approach-tabu search algorithm for single machine total weighted tardiness problem, Computers & Industrial Engineering (2006) (in press).
6. Bożejko W., Wodecki M., Parallel genetic algorithm for the flow shop scheduling problem, Lecture Notes in Computer Science No. **3019**, Springer Verlag (2004) 566–571.
7. Bubak M., Sowa M., Objectoriented implementation of parallel genetic algorithms, in High Performance Cluster Computing: Programming and Applications (R. Buyya, ed.), **2**, Prentice Hall (1999) 331–349.
8. Feldmann M., Biskup D., Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches, Computers & Industrial Engineering **44** (2003) 307–323.

9. Grabowski J., Wodecki M., A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion, Computers and Operations Research, **31** (2004) 1891–1909.
10. Grabowski J., Wodecki M., A very fast tabu search algorithm for job shop problem, in: Rego C., Alidaee B. (editors), Adaptive memory and evolution: tabu search and scatter search, Kluwer Academic Publishers (2004).
11. Gordon V., Proth J.P., Chu C., A survey of the state-of-art of common due date assignment and scheduling research, European Journal of Operational Research **139** (2002) 1–25.
12. Hoogeveen J.A., Van de Velde L.S., A branch and bound algorithm for single-machine earliness-tardiness scheduling with idle time, INFORMS Journal on Computing **8** (1996) 402–412.
13. Kanett J.J., Minimizing the average deviation of job completion times about a common due date. Naval Research Logistics **28** (1981) 643–651.
14. Lee C.Y., Choi J.Y., A generic algorithm for job sequencing problem with distinct due dates and general early-tardy penalty weights, Computers and Operations Research **22** (1995) 857–869.
15. Lenstra J.J., Rinnoy Kan A.H.G., Brucker P., Complexity of machine scheduling problems, Annals of Discrete Mathematics **1** (1977) 343–362.
16. Nowicki E., Smutnicki C., A fast tabu search algorithm for the permutation flow shop problem, European Journal of Operational Research, **91** (1996) 160–175.
17. Nowicki E., Smutnicki C., A fast tabu search algorithm for the job shop problem, Management Science, **42** (1996) 797–813.
18. OR-Library: http://people.brunel.ac.uk/ mastjjb/jeb/info.html
19. Reeves C. R., Yamada T., Solving the Csum Permutation Flowshop Scheduling Problem by Genetic Local Search, IEEE International Conference on Evolutionary Computation (1998) 230–234.
20. Szwarc W., Adjacent ordering in single machine scheduling with earliness and tardiness penalties. Neval Research Logistics **40** (1993) 229–243.
21. Tanese R., Distribued genetic algorithms, Proc. of the Third Intern. Conf. on Genetic Algorithms (J.D. Schaffer, ed.), Morgan Kaufmann (1989) 434–439
22. T'kindt V., Billaut J-C., Multicriteria scheduling: theory, models and algorithms. Springer Verlag (2002).
23. Valente J.M.S., Alves R.A.F.S., Fildered and recovering beam search algorithms for the early/tardy scheduling problem with no idle time, Computers & Industrial Engineering **48(2)** (2005) 363–375.
24. Wan G., Yen B.P.C., Tabu search for single machine scheduling with distinct due windows and weighted earliness/tardiness penalties, European Journal of Operational Research **142** (2002) 271–281.
25. Yano C.A., Kim Y.D., Algorithms for a class of single machine weighted tardiness and earliness problems, European Journal of Operational Research **52** (1991) 167–178.

Multiobjective Design Optimization of Electrostatic Rotary Microactuators Using Evolutionary Algorithms

Paolo Di Barba[1] and Sławomir Wiak[2]

[1] Department of Electrical Engineering University of Pavia
via Ferrata 1, I-27100 Pavia, Italy
paolo.dibarba@unipv.it
[2] Institute of Mechatronics and Information Systems Technical University of Łódź
ul. Stefanowskiego 18/22, 90-924 Łódź, Poland
wiakslaw@p.lodz.pl

Abstract. An example of soft computing applied to electrical engineering is presented and discussed. Attention is focused on the design techniques of MicroElectroMechanical Systems. In particular, the criterion of Pareto optimality is used to identify the optimal shape design of a rotary electrostatic microactuator. Accordingly, two algorithms of evolutionary optimization are presented and compared. The requirements in terms of know-how and computing facilities fit the resources of a research-and-development center of an industrial company.

1 Introduction

MicroElectroMechanical Systems (MEMS) technology has generated an impressive deal of academic and industrial research, opening the relevant market. Although the power delivered by an electrostatic motor is modest, the absence of field coils and ferromagnetic material makes these devices competitive, with respect to the conventional electromagnetic ones, in applications where a weak torque and small dimensions are required. This remark, together with the far-dating experience in fabricating electronic microsystems on Silicon, has stimulated, since at least ten years, the development of mechanical microsystems on Silicon, in which the motion-generating force is electrostatic. By exploiting the mechanical properties of Silicon - as extraordinary as the electrical ones - using the technology already achieved for integrated circuit manufacturing, and integrating power supply, motion actuation and control functions, it is possible, nowadays, to make MEMS smaller than one square millimeter, for several applications. Actually, pressure and acceleration microsensors, as well as microswitches, microvalves, micromirrors, microresonators, linear and rotary microactuators and microrobots, have already been developed, for application in medicine, space and automobile industry, and precise mechanical industry. The design of a MicroElectroMechanicalDevice (MEMD) needs semiconductors and related technologies: VLSI, polycrystalline silicon and intelligent materials; generally, the supply system is integrated into the device on the same board. Critical

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 344–353, 2006.
© Springer-Verlag Berlin Heidelberg 2006

components of MEMD are represented by microactuators, for which severe requirements in terms of performance, reliability and cost are prescribed. [1,2,3]

MEMS are characterized by a multi-physics domain: structural mechanics, electrostatics, fluid flow, optical phenomena interact and give rise to coupled fields at the device level. These requirements demand for a sophisticated design procedure. The field of microengineering is leading to radical changes in strategies of computer modeling simulation for high-technology MEMS. Modeling and simulation of MEMS is of vital importance to develop innovative products and to reduce time-to-market at lower total costs.

2 Soft Computing and Automated Optimal Design: A Review

In all engineering fields, and especially in electrical and electronic engineering, the optimized computer-based design has reached a crucial role. In literature many non-deterministic methods are available, especially those ones based on the principle of natural evolution. In these ones, the natural law "survival of the fittest will win" is the way to determine the best design set fulfilling constraints. In nature, individuals of a population in an assigned domain mate and reproduce with the aim of adjusting to new surroundings, so that those chromosomes more prone to adaptation survive in the new generation. Similarly, in a simulation environment, a population of different candidate design solutions, after reproduction, combines their own most favourable features, thus giving rise to a new population of solutions that best fit design criteria.

Another class of widely used non-deterministic methods is based on the so-called "swarm intelligence", in contrast to the intellective ability of the single individual. The most studied swarm is the ant colony: ants indirectly communicate by modifying the natural environment, with the aim of reaching food from the nest through the shortest path. The imitation of such a behaviour leads to the implementation of optimization methods ("Ant Colony Optimization") based on software agents (artificial ants), which are singularly equipped with a small computational ability and work in a distributed way without a centralized coordination. These are reliable and robust techniques, because of the little importance of the single agent within the swarm; furthermore, the small computational ability of the single agent allows a cheap implementation of the algorithms. The reproduction of flocks of birds and of antibodies of a living organism immune system are two other important themes of recent studies, resulting in "Particle Swarm Optimization" and "Artificial Immune System" techniques, respectively.

Developed methods allow to cast the automated design of systems and devices as an optimization problem of an objective function dependent on design variables and whose values must fulfil an assigned number of constraints. Unfortunately, many real-life engineering problems exhibit an increased complexity due to the presence of a high number of design criteria; actually, a vector of mutually conflicting objective functions characterizes the full formulation of the

design problem. In other words, in a design problem from real-life engineering, the presence of a single design criterion is rather an exception than a rule; often, the designer has to cope with the minimization of two or more conflictual objectives. It is reasonable to state that design — or inverse — problems are multiobjective problems by their very nature and imply the simultaneous minimization of a number of objectives being in mutual conflict. [4]

Traditionally, multiobjective problems are reduced to single-objective problems, for instance by means of one of the following procedures:

- the use of a penalty function composed of the various objectives;
- the separate solutions of single-objective problems and their trade-off;
- the solution of a single-objective problem, with the other objectives as constraints.

This approach leads to classical methods of multiobjective optimization and gives a solution which is supposed to be the optimum.

A more satisfactory way to tackle the problem of multiobjective optimisation consists of applying the theory of Pareto optimality in connection with a suitable minimization algorithm. In this respect, the key concept is the dominance criterion: a solution x is said to dominate another solution y, if both the following conditions hold:

- x is not worse than y for all objectives;
- x is strictly better than y for at least one objective.

Basically, the procedure of automated optimal design consists of searching for the non-dominated solutions of the problem. In this context, several non-deterministic methods are available in the literature for Pareto multiobjective optimisation. Most of these methods have been developed for solving problems in which the computational cost of the objective functions is moderate. When dealing with design optimisation of electromagnetic devices, the evaluation of the objectives requires at least a field simulation based on finite-element method (FEM); field simulation, in turn, may have an inherent complexity due to various reasons: complicated shape of the device implying two- or three-dimensional models, coupled-field analysis, non-linear material properties, transient conditions. Typical amount of runtime required by the direct problem limits severely the use of non-deterministic methodologies, even resorting to powerful computing facilities.

To clarify this point, let us imagine a case in which an objective implies the FEM simulation of the torque-angle curve featuring a rotating electrical device; in the case of e.g. a twelve-pole device, taking a rotation step of 1 deg and considering symmetry, 15 FEM analyses are necessary to find out the torque-angle curve for a given geometry of the device. If, additionally, the non-linear characteristic of ferromagnetic material is taken into account and 10 Newton-Raphson iterations are involved, the number of FEM analyses grows up to 150 per geometry. If the optimisation of the geometry relies on a standard algorithm with a population of 30 individuals and 100 iterations are required to converge, the total number of field analyses becomes 450,000; even if a single FEM analysis

lasts only 10 s for a three-dimensional model, the resulting runtime is 4,500,000 s (nearly 52 days) for finding 30 non-dominated solutions, which is an unaffordable duration for industrial design timing.

Indeed, one might think of more sophisticated strategies to reduce the cost, e.g. those based on parallel computing, but this kind of resources is still loosely available in a typical design center of an electrical or electromechanical company. Consequently, it has been decided to focus the attention on cost-effective algorithms, i.e. algorithms where the number of calls to the objectives is reduced for a given degree of accuracy in approximating the non-dominated solutions of the problem.

Moving from this background, a cost-effective procedure for the automated shape design of a MEMD has been developed. [5]

3 An Evolutionary Method for the Multiobjective Shape Design of an Electrostatic Microactuator

A simplified flow-chart of the four-step procedure that has been devised and implemented is shown in Fig.1.

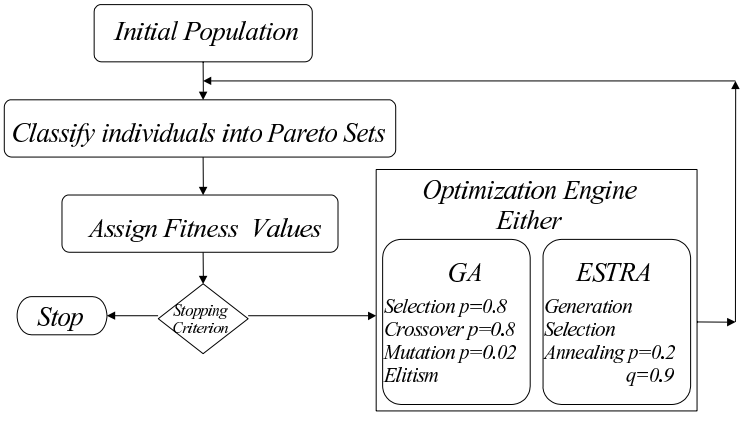


Fig. 1. Flowchart of NSESA and NSGA algorithms

In the *first step* of the algorithm an initial population of individuals is randomly generated in the search domain; generation fulfils constraints the design variables are subject to.

In the *second step* individuals are ranked into local Pareto fronts. In particular, the criterion of dominance is first applied to the whole population; this way, all non-dominated individuals are identified in order to set up the first Pareto front. The individuals belonging to the first front are then removed from the population and the above criterion is applied again in order to obtain the second front and so on. The procedure ends when all individuals have been ranked into local fronts; an example of sorting is shown in Fig. 2.

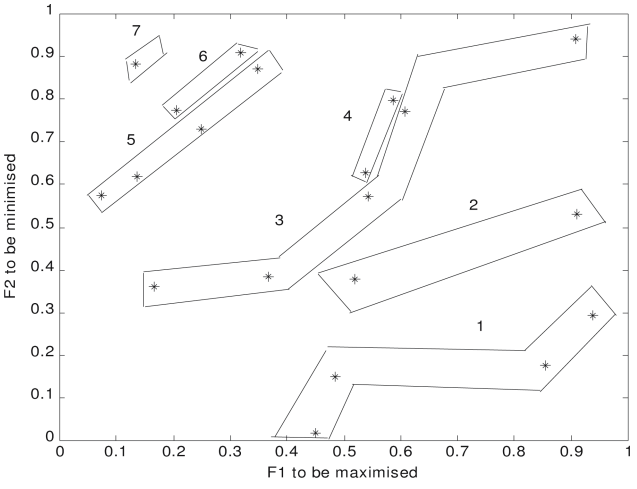


Fig. 2. Sorting a population in local Pareto fronts

The *third step* consists of assigning a suitable value of fitness to each individual; two criteria must be followed in this step: forcing convergence to the global Pareto front; forcing diversity among solutions. In order to do this, the fitness value of each individual depends on the local front which it belongs to; successively, a sharing procedure is implemented in order to reduce clustering of solutions.

Finally, as the *fourth step*, if the stopping criterion is not fulfilled, a stochastic algorithm produces a new generation of individuals; then, steps 2 and 3 are repeated.

A code has been developed and implemented, in which two minimizers have been implemented giving rise to two different algorithms of multiobjective optimisation based on non-dominated sorting, respectively: Non-dominated Sorting Genetic Algorithm (NSGA) if the minimizer is a GA; Non-dominated Sorting Evolution Strategy Algorithm (NSESA) if the minimizer is an ESA (see Fig. 1). As far as GA is concerned, three classical genetic operators have been implemented, i.e. selection, crossover and mutation; an elitism procedure has been added to guarantee the survival of the best individual coming from the previous generation. The values of probability p are listed in Fig. 1 for the three operators; real coding is adopted. On the other hand, as concerns ESA, standard evolutionary operators (generation, mutation, annealing) of a (1+1) evolution strategy are implemented; the parameter values used (namely, probability of successful iteration p and annealing amount q) are listed in Fig. 1 as well.

4 Case Study

The optimal shape design of a variable-capacitance rotary microactuator with radial field has been carried out by means of the multiobjective strategy described

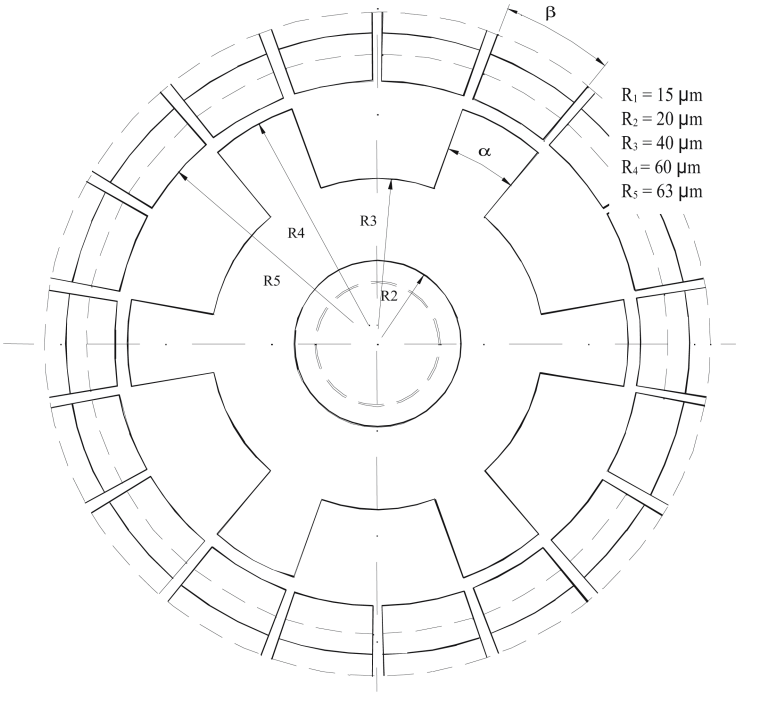


Fig. 3. Cross section of the microactuator

above. The device, etched on a polySilicon structure, is characterized by 18 stator electrodes and 6 rotor teeth, respectively (see Fig.3). One of the three-phase system of square voltages, having amplitude equal to $V = 100V$, is switched on; denoting the tooth width as $x_1 = \alpha$ and the electrode width as $x_2 = \beta$ the equivalent capacitance $C_{eq}(\phi, x_1, x_2)$ is considered, where ϕ is the rotor angular position. The simplest model for computing both no-load commutation torque T_0 and static torque T_S is the following:

$$T_0\left(x_1, x_2\right) = \frac{1}{2} V^2 N_S \frac{\left[C_{eqA}\left(x_1, x_2\right) - C_{eqB}\left(x_x, x_2\right)\right]}{2\pi} \tag{1}$$

$$T_S\left(\square, x_1, x_2\right) = \frac{1}{4} V^2 N_R \left[C_{max}\left(x_1, x_2\right) - C_{min}\left(x_1, x_2\right)\right] \sin\left(N_R \square\right) \tag{2}$$

where N_S and N_R are the number of stator electrodes and rotor teeth, respectively; $C_{eqA}(x_1, x_2)$ is the capacitance of the maximum coenergy configuration, when the axis of the supplied electrode is coincident with the axis of the rotor tooth, while $C_{eqB}(x_1, x_2)$ is the equivalent capacitance when the rotor position is the same as before, but the supply has been switched to the next phase. In turn, $C_{max}(x_1, x_2)$ and $C_{min}(x_1, x_2)$ are maximum and minimum capacitances with respect to the rotor position ϕ, keeping the supplied electrode fixed. Two objective functions, namely:

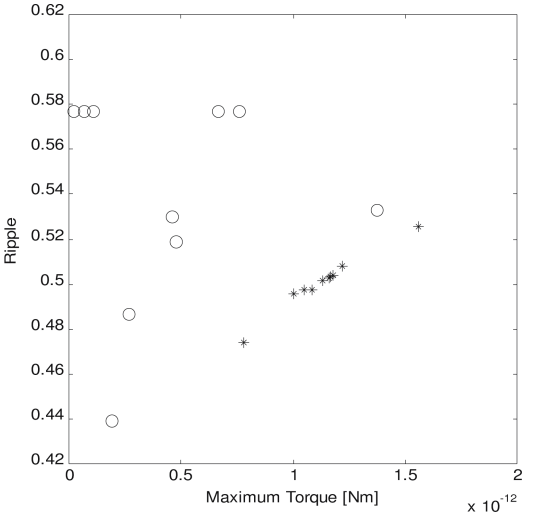


Fig. 4. NSESA: starting (°) and final (*) populations in objective space (10 individuals, 150 iterations)

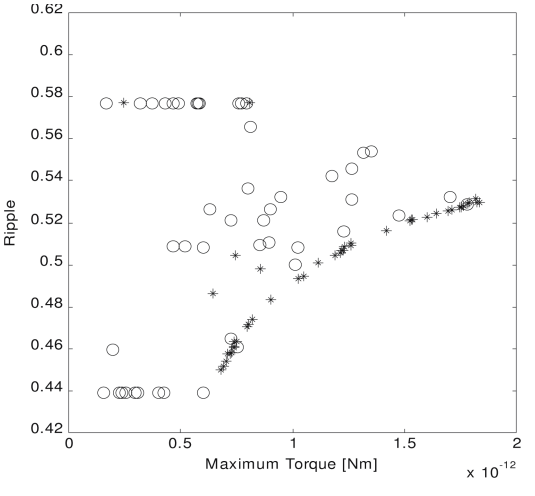


Fig. 5. NSGA: starting (°) and final (*) populations in objective space (50 individuals, 150 iterations)

maximum static torque, to be maximized

$$f_1(x_1, x_2) = C_{max}(x_1, x_2) - C_{min}(x_1, x_2) \tag{3}$$

and torque ripple, to be minimized

$$f_2(x_1, x_2) = 1 - \frac{T_0(x_1, x_2)}{f_1(x_1, x_2)} \tag{4}$$

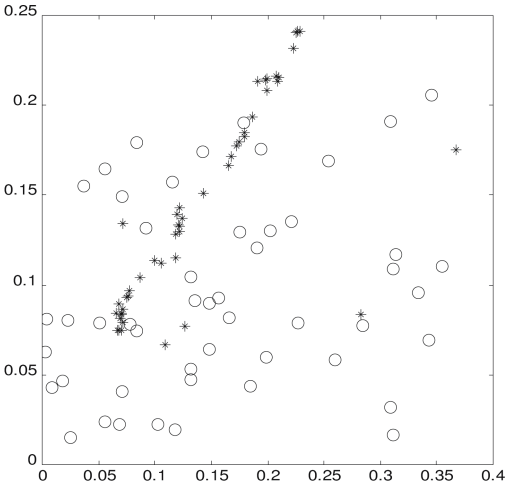


Fig. 6. NSGA: starting (°) and final (*) populations in design space (50 individuals, 150 iterations)

are considered; the design variables (x_1, x_2) which values should fulfill the geometric congruency of the model. In particular, f_1 represents the performance of the actuator, while f_2 accounts for the side effect. Therefore, the design problem can be cast as follows: *starting from a population of individuals randomly generated in the design space* (x_1, x_2)*, find an approximation of the global Pareto front, such that* f_1 *is maximum and* f_2 *is minimum, subject to the problem constraints.*

In order to obtain an approximation of the global Pareto front, different solutions obtained by means of either NSESA (see Fig.4) or NSGA (see Fig.5) are reported, respectively.

The genetic algorithm proves to be particularly suited for large population size (more than 30 individuals, as a rule of thumb), while the quality of solution is poor when a smaller population is adopted.

In Fig.6 the geometries corresponding to the Pareto front are represented in the design space (x_1, x_2); the diversity of shape can be noted.

The evolution of the initial population towards the Pareto optimal front has been monitored in the objective space for NSGA; Fig. 7 shows six frames of the evolution in the case of 50 individuals, from initial guess (first frame) to convergence (last frame).

A final remark on computational burden is worth of consideration. In order to estimate the maximum cost *mc* of the evolutionary strategy implemented, the following formula holds

$$mc = niter \times npop \times nobj \times \Delta t \tag{5}$$

where *nobj* is the number of objective functions (2 through 3), *niter* is the maximum number of iterations to convergence (say 300) at a given accuracy (say 10-9), *npop* is the number of individuals (5 through 20); finally, Δt is the

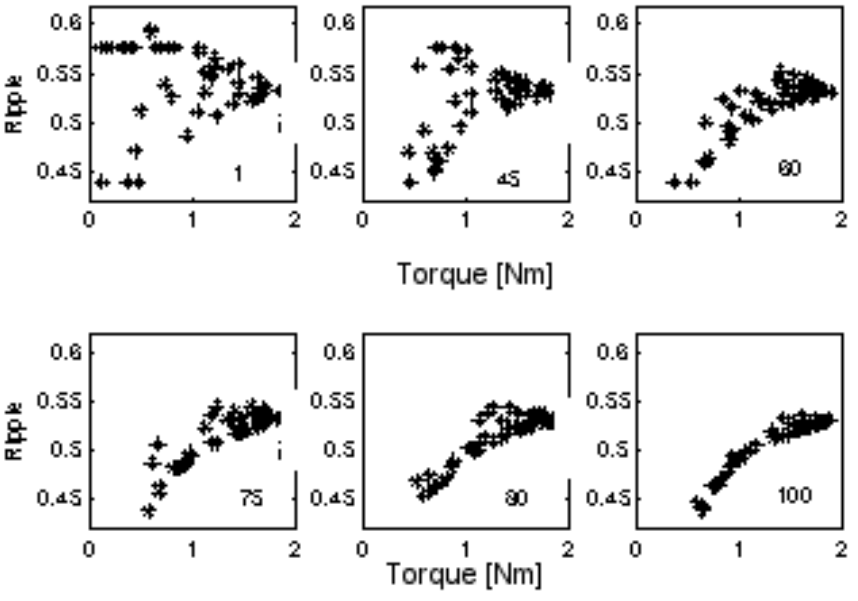


Fig. 7. NSGA: evolution of the solutions in the objective space

runtime of a single analysis. The latter is the crucial factor influencing the overall cost of optimisation.

5 Conclusion

Referring to the algorithms implemented, the following features should be emphasized:

- both NSGA and NSESA are conservative strategies; in other words, the total number of individuals is constant and equal to the initial population (no individual expires during the optimisation procedure);
- in both NSGA and NSESA each elementary operator can be implemented in parallel, whereas ranking a population into Pareto fronts and assigning fitness to each individual are sequential operations by their very nature;
- the intrinsic operation of NSGA implies the continuous interaction between pairs of individuals: as a rule of thumb, different tenths of individuals (say 30 through 50) are necessary in order to achieve a satisfactory convergence;
- in the frame of NSESA individuals do not interact during the optimisation procedure: convergence does not depend on the number of individuals (the procedure converges for a single individual too).

As a consequence, it is reasonable to state that NSESA can be applied when a small population (say 5 to 10 individuals) is available, leading to a remarkable

reduction of computational cost; the drawback is a poor approximation of the Pareto optimal front.

More generally, it should be noted that the definition of Pareto front enhances the diversity of optimal solutions; consequently, non-trivial designs that are *a priori* unpredictable can be highlighted after investigating the front. This is a method of soft computing which represents a powerful way to promote the innovation of the industrial design. The numerical implementation of a procedure of automated optimal design, like that described in the paper, is compatible with the standard resources of a research-and-development center of an industrial company.

References

1. Wiak, S., Smółka, K., Rudnicki, M.: Modelling and Optimisation of an Intelligent Electrostatic Comb Accelerometer. (S. Wiak, A. Krawczyk, M. Trlep editors): Computer Engineering in Applied Electromagnetism, Springer, 2005), 99-104
2. Wiak, S., Cader, A., Drzymała, P., Welfle, H.: Virtual Modeling and Optical Design of Intelligent Micro-accelerometers; 7th International Conference on Artificial Intelligence and Soft Computing - ICAISC 2004; Zakopane, June 2004, Springer, Subseries of Lecture Notes in Computer Science, (Ed.s: L. Rutkowwski, J. Siekmann, R. Tadesusiewicz, L.A. Zadeh), 942-947
3. Gardner, J.W., Varadan, V.K., Awadelkarim, O.O.: Microsensor MEMS and Smart Devices, Wiley (2001)
4. Deb, K.: Multi-Objective Optimisation using Evolutionary Algorithms, Wiley (2001)
5. Di Barba, P.: Silicon Electrostatic Microactuators: Numerical Models and Design Optimization. PhD Dissertation, Technical University of Ł?dź, Academic Year 2000-2001

Evolutionary Learning of Mamdani-Type Neuro-fuzzy Systems*

Marcin Gabryel[1] and Leszek Rutkowski[1,2]

[1] Department of Computer Engineering, Częstochowa University of Technology
Al. Armii Krajowej 36, 42-200 Częstochowa, Poland
{marcing, lrutko}@kik.pcz.czest.pl
http://kik.pcz.pl
[2] Department of Artificial Intelligence, WSHE University in Łódź
ul. Rewolucji 1905 nr 64, Łódź, Poland
http://www.wshe.lodz.pl

Abstract. In this paper we present an evolutionary method for learning fuzzy rule base systems as an alternative to gradient methods. It is known that the backpropagation algorithm can be trapped in local minima. We use evolutionary strategies (μ, λ) with a novel method for generating an initial population. The results of simulations illustrate efficiency of our method.

1 Introduction

In the last decade various architectures of neuro-fuzzy systems have been developed [13,14]. They can be divided as follows:

1. Mamdani-type neuro-fuzzy systems - in this approach antecedents and consequences in the individual rules are connected by a t-norm.
2. logical-type neuro-fuzzy systems - in this approach antecedents and consequences in the individual rules are connected by a fuzzy implication satisfing Fodor's definition [5], e.g. an S-implication.
3. Takagi Sugeno neuro-fuzzy systems - which are characterized by a functional dependence between consequents and inputs.

Recently, a concept of flexible neuro-fuzzy systems has been developed [11,13,14]. The main idea is based on the incorporation of various parameters into construction of such systems leading to high accuracy in problems of modelling and classification.

We can use successfully evolutionary methods for designing different types of fuzzy systems [3]. For example a genetic algorithm optimizes values of parameters which define membership functions. In the second case it creates the whole knowledge base, i.e. form of membership functions, antecedents and consequences of rules and their number. In this paper we present evolutionary learning

* This work was supported in part by the Foundation for Polish Science (Professorial Grant 2005-2008) and the Polish State Committee for Scientific Research (Grant Nr T11E 03627 and Grant Nr T11C 04827).

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 354–359, 2006.
© Springer-Verlag Berlin Heidelberg 2006

process for learning Mamdani type neuro-fuzzy systems. We use the evolutionary strategy (μ, λ) with a novel method for generating an initial population. Our initialization method leads to smaller populations and less number of generations comparing with previous methods [12]. The method can be used as an alternative to the backpropagation algorithm which is frequently adopted to learn neuro-fuzzy systems. Unfortunately, the backpropagation algorithm can be trapped in local minima. Experiments and simulations presented in section 4 show that our evolutionary learning procedure with novel initialization outperforms gradient methods [11].

2 Description of Fuzzy Systems

In this paper, we consider a multi input, single output neuro fuzzy system mapping $\mathbf{X} \to \mathbf{Y}$ where $\mathbf{X} \subset \mathbf{R}^n$ and $\mathbf{Y} \subset \mathbf{R}$. The fuzzy rule base consists of a collection of N fuzzy IF-THEN rules in the form:

$$R^{(k)}: \text{ IF } x_i \text{ is } A_1^k \text{ AND} \ldots \text{AND } x_n \text{ is } A_n^k \text{ THEN } y \text{ is } B^k \tag{1}$$

where $\mathbf{x} = [x_1, \ldots, x_n]$ are input variables, n - number of inputs, y - output value, fuzzy sets $A_1^k, A_2^k, \ldots, A_n^k$ and B^k are characterized by membership functions $\mu_{A_i^k}(x_i)$ and $\mu_{B^k}(y)$, respectively, $k = 1, ..., N$, $i = 1, ..., n$. This system is based on the Mamdani-type reasoning, where antecedents and consequences in the individual rules are connected by the product t-norm. We use the most common singleton fuzzifier for mapping crisp values of input variables into fuzzy sets [11]. The defuzzification process is made by the COA (center of area) method [11]. We choose as membership functions $\mu_{A_i^k}(x_i)$ and $\mu_{B^k}(y)$ the Gaussian functions

$$\mu_{A_i^k}(x_i) = \exp\left[-\left(\frac{x_i - \overline{x}_i^k}{\sigma_i^k}\right)^2\right], \tag{2}$$

$$\mu_{B_i^k}(y) = \exp\left[-\left(\frac{y - \overline{y}^k}{\sigma^k}\right)^2\right]. \tag{3}$$

The following neuro-fuzzy system [11] will be investigated:

$$y = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \max\limits_{1 \le k \le N} \left\{ \prod_{i=1}^{n} \exp\left[-\left(\frac{x_i - \overline{x}_i^k}{\sigma_i^k}\right)^2\right] \cdot \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k}\right)^2\right] \right\}}{\sum_{r=1}^{N} \max\limits_{1 \le k \le N} \left\{ \prod_{i=1}^{n} \exp\left[-\left(\frac{x_i - \overline{x}_i^k}{\sigma_i^k}\right)^2\right] \cdot \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k}\right)^2\right] \right\}}. \tag{4}$$

This system has been trained using the idea of the backpropagation method [11]. In the next section we will develop an evolutionary algorithm to train system (4).

3 Genetic Fuzzy System

Let $\overline{\mathbf{x}} \in \mathbf{X} \subset \mathbf{R}^n$, $y(t) \in \mathbf{Y} \subset \mathbf{R}$ and $d(t) \in \mathbf{Y} \subset \mathbf{R}$, $t = 1, ..., K$. Based on the learning sequence $((\mathbf{x}(1), d(1)), (\mathbf{x}(2), d(2)), ..., (\mathbf{x}(K), d(K)))$ we wish to determine all parameters $\overline{x}_i^k$, σ_i^k, $\overline{y}^k$, σ^k, $i = 1, ..., n$, $k = 1, ..., N$, such that

$$e(t) = \frac{1}{2}\left[f\left(\mathbf{x}(t)\right) - d(t)\right]^2 \tag{5}$$

is minimized.

We will solve the problem (5) by using the evolutionary strategy (μ, λ) (see [1][4]). It is well known that evolution strategies are distinguished by self-adaptation of additional strategy parameters, which enables them to adapt the evolutionary optimization process to the structure of the fitness landscape [2]. It is assumed that the chromosome of an individual is formed by a pair of real-valued vectors $(\mathbf{X}, \boldsymbol{\sigma})$. The strategy vector $\boldsymbol{\sigma}$ is subject to a random mutation according to

$$\sigma_i' = \sigma_i \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)} \tag{6}$$

where $\tau' = \frac{1}{\sqrt{2L}}$, $\tau = \frac{1}{\sqrt{2\sqrt{L}}}$, $i = 1, \ldots, L$, and L is the length of the chromosome. The mutation

$$X_i' = X_i + \sigma' \cdot N_i(0,1) \tag{7}$$

replaces the parent $\mathbf{X}'$ with the parent $\mathbf{X}$. We extend the standard evolution strategy based on mutation by making use of a uniform recombination operator [9].

3.1 Encoding

In a fuzzy system described by formula (4) membership functions (2) and (3) are determined by two parameters $(\overline{x}_i^k, \sigma_i^k)$ and $(\overline{y}^k, \sigma^k)$, respectively. Thus each of the rules will be encoded in a piece of the chromosome $\mathbf{X}_j$ denoted by $\mathbf{X}_{j,k}$, $k = 1, \ldots, N$, in following way:

$$\mathbf{X}_{j,k} = \left(\overline{x}_1^k, \sigma_1^k, \overline{x}_2^k, \sigma_2^k, \ldots, \overline{x}_n^k, \sigma_n^k, \overline{y}^k, \sigma^k\right) \tag{8}$$

where $j = 1, ..., \mu$ or $j = 1, ..., \lambda$ where μ and λ are parametrers of the evolutionary strategy (μ,λ). The complete rule base is represented by chromosome $\mathbf{X}_j$

$$\mathbf{X}_j = (\mathbf{X}_{j,1}, \mathbf{X}_{j,2}, \ldots, \mathbf{X}_{j,N}) \tag{9}$$

or more in detail

$$\mathbf{X}_j = \begin{array}{l} \left(\overline{x}_1^1, \sigma_1^1, \overline{x}_2^1, \sigma_2^1, \ldots, \overline{x}_n^1, \sigma_n^1, \overline{y}^1, \sigma^1,\right. \\ \overline{x}_1^2, \sigma_1^2, \overline{x}_2^2, \sigma_2^2, \ldots, \overline{x}_n^2, \sigma_n^2, \overline{y}^2, \sigma^2, \\ \vdots \\ \left.\overline{x}_1^N, \sigma_1^N, \overline{x}_2^N, \sigma_2^N, \ldots, \overline{x}_n^N, \sigma_n^N, \overline{y}^N, \sigma^N\right). \end{array} \tag{10}$$

3.2 Initialization

In this section we will describe the first step of evolutionary strategy, i.e. process of initialization of the first population. When we have a learning sequence $(\mathbf{x}(t), d(t))$, for every input variable x_i we define range $\left[x_i^-, x_i^+\right]$. Boundary values x_i^- and x_i^+ can be computed as follows

$$\begin{aligned} x_i^- &= \min_{1\le t\le K} \left(x_i(t)\right), \\ x_i^+ &= \max_{1\le t\le K} \left(x_i(t)\right). \end{aligned} \tag{11}$$

In the same way we find range $[d^-, d^+]$ for output values y, i.e.

$$\begin{aligned} d^- &= \min_{1\le t\le K} \left(d(t)\right), \\ d^+ &= \max_{1\le t\le K} \left(d(t)\right). \end{aligned} \tag{12}$$

On the ranges defined above uniform fuzzy partition are made, where every region has a membership function assigned. The number of pieces for every partition $\left[x_i^-, x_i^+\right]$ and $[d^-, d^+]$ is equal N. Therefore width a_i for input variables $(i = 1, \ldots, n)$ has value $a_i = \frac{x_i^+ - x_i^-}{N}$, $i = 1, ..., n$. For output variable we have $a_{n+1} = \frac{d^+ - d^-}{N}$. Initial parameters of membership functions $\mu_{A_i^k}(x_i)$ (see (2)) take values

$$\overline{x}_i^k = x_i^- + b_k \cdot a_i - \frac{a_i}{2}, \tag{13}$$

$$\sigma_i^k = \frac{a_i}{2} \tag{14}$$

where $\mathbf{b} = [b_1, ..., b_N]$ is a vector with randomly chosen components $b_k \in \{1, ..., N\}$, $b_k \neq b_l$, $k, l = 1, ..., N$. Similarly, intial parameters of membership functions $\mu_{B^k}(y)$ (see (3)), $k = 1, ..., N$ assume values

$$\overline{y}^k = d^- + b'_k \cdot a_{n+1} - \frac{a_{n+1}}{2}, \tag{15}$$

$$\sigma^k = \frac{a_{n+1}}{2} \tag{16}$$

where b'_k is determined analogously to b_k, $k = 1, ..., N$. Numbers b_k and b'_k, determined in the random way, give us the method of generating parameters of membership functions in the initial population.

We will present a method of initialization of the standard deviation vector $\boldsymbol{\sigma}$. This problem is not clearly described in literature and most often elements of vector $\boldsymbol{\sigma}$ are chosen experimentally. We use information which is encoded in chromosome $\mathbf{X}_j$. Therefore the vector of standard deviation $\boldsymbol{\sigma}$ will be initiated by using values of a_i and a_{n+1} as follows

$$\boldsymbol{\sigma}_j = \begin{matrix} (h \cdot a_1, h \cdot a_1, h \cdot a_2, h \cdot a_2, ..., h \cdot a_n, h \cdot a_n, h \cdot a_{n+1}, h \cdot a_{n+1}, \\ \vdots \\ h \cdot a_1, h \cdot a_1, h \cdot a_2, h \cdot a_2, ..., h \cdot a_n, h \cdot a_n, h \cdot a_{n+1}, h \cdot a_{n+1}) \end{matrix} \tag{17}$$

where $h = 0.1$ is a constant, which value was selected experimentally.

4 Simulation Result

In this section we present results of evolutionary learning of the Mamdani-type system. We consider classification problems taken from [15]: Iris problem, Ionosphere problem, the Pima Indians Diabetes, Glass Identification, Wisconsin Breast Cancer and Wine Recognition problem.

In the first experiment we show a comparison of diffrent initialization methods of an initial population. We assume the following parameters of simulations: $\mu = 10$, $\lambda = 50$, maximal number of generations is 100, each simulation is repeated five times and we take the RMSE (root means square error) as a fitness function

$$fitness = \sqrt{\frac{1}{K} \sum_{i=1}^{K} (y - d(i))^2}. \tag{18}$$

The experimental results for the Pima Indians Diabetes problem are shown in Table 1. In the first column we show average results of learning by evolutionary strategies presented in section 3. In the next two columns we present average results assuming that all elements of stadard deviation vector $\boldsymbol{\sigma}$ was initialized by values 0.1 and 1, respectively.

In the second experiment we show effectiveness of our learning method. We assume the following parameters of simulations: $\mu = 10$, $\lambda = 50$, maximal num-

Table 1. Simulation results of evolutionary learning

our method		$\sigma = 1$		$\sigma = 0.1$	
training	testing	training	testing	training	testing
79.4%	76.6%	73.0%	71.2%	73.8%	71.6%

Table 2. Comparation of effectiveness diffrent methods

problem	[11]	[8]	our method
Iris	97.8%	95.7%	100%
IO	94.25%	91.8%	95.2%
PID	80.2%	-	79.7%
GI	98.4%	-	98.4%
WBC	98.5%	-	98.3%
Wine	100%	93.8%	97%

ber of generations is 200 and we take (18) as a fitness function. The experimental results and comparisons for six benchmark problems are shown in Table 2.

5 Final Remarks

In the paper we presented a learning method of Mamdani-type neuro-fuzzy systems. The evolutionary strategy was initialized by a new algorithm. We have obtained comparable or better results than those presented in [11] and [8]. It seems that our approach leads to obtaining smaller populations and faster convergence.

References

1. J. Arabas, "Lectures on Evolutionary Algorithms" (in Polish), WNT, Warsaw, 2001.
2. T. Back, "Evolutionary Algorithms in Theory and Practice", Oxford University Press, Oxford, 1996.
3. O. Cordon, F. Herrera, F. Hoffman, L. Magdalena, "Genetic Fuzzy System. Evolutionary Tunning and Learning of Fuzzy Knowledge Bases", World Scientific, Singapur, 2000.
4. A.E. Eiben, J.E. Smith, "Introduction to Evolutionary Computing", Springer, 2003.
5. J. C. Fodor. "On fuzzy implication operators", Fuzzy Sets and Systems, vol. 42, pp. 293-300, 1991.
6. M. Gabryel, K. Cpalka, L. Rutkowski, "Evolutionary strategies for learning of neuro-fuzzy systems", in I Workshop on Genetic Fuzzy Systems, pp. 119-123, Genewa, 2005.
7. M. Gabryel, L. Rutkowski, "Evolutionary method for learning neuro-fuzzy systems with applications to medical diagnosis" (in Polish), XIV Krajowa Konferencja Naukowa Biocybernetyka i Inynieria Biomedyczna, pp. 960-965, Czstochowa, 2005.
8. A. Gonzalez and R. Perez, "SLAVE: a genetic learning system based on an iterative approach", IEEE trans. on Fuzzy Systems, vol. 7, pp. 176-191, April 1999.
9. Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs" 3rd edition, Springer-Verlag, 1996.
10. D. Rutkowska, "Neuro Fuzzy Architectures and Hybrid Learning", Springer Verlag 2002.
11. L. Rutkowski, "Flexible Neuro Fuzzy Systems", Kluwer Academic Publishers, 2004.
12. L. Rutkowski, "Methods and Techniques of Artificial Inteligence" (in Polish), Wydawnictwo Naukowe PWN, Warsaw, 2005.
13. L. Rutkowski, K. Cpalka, Flexible neuro-fuzzy systems, IEEE Transactions on Neural Networks, vol. 14, pp. 554-574, 2003.
14. L. Rutkowski, K. Cpalka. Designing and learning of adjustable quasi triangular norms with applications to neuro fuzzy systems, IEEE Trans. on Fuzzy Systems, vol. 14, pp. 140-151, 2005.
15. UCI respository of machine learning databases, C. J. Mertz, P. M. Murphy. Available online: `http://www.ics.uci.edu/pub/machine-learning-databases`.

Study of Objective Functions in Fuzzy Job-Shop Problem

Inés González-Rodríguez, Camino R. Vela, and Jorge Puente*

Dept. of Computer Science, University of Oviedo, Spain
inesgr@uniovi.es, camino@aic.uniovi.es, puente@aic.uniovi.es

Abstract. We consider the *fuzzy job-shop problem*, a job-shop scheduling problem with uncertain task durations and flexible due-date constraints. We propose different definitions of the objective function and analyse solutions obtained for each alternative using a genetic algorithm.

1 Introduction

In the last decades, scheduling problems have been subject to intensive research due to their multiple applications in areas of industry, finance and science [1]. Some of these applications involve modelling the uncertainty and vagueness pervading real-life situations; this has resulted in a particular branch of scheduling, known as fuzzy scheduling [2],[3]. Here we find a great variety of approaches, connected with the three semantics of fuzzy sets. They range from representing incomplete or vague states of information to using fuzzy priority rules with linguistic qualifiers or preference modelling. It is also possible to find models combining more than one of these approaches. Although the first applications of fuzzy scheduling date back to the 1970's, it has not been until recently that it has received an increasing attention. In particular, little work has been done with more realistic and complex problems such as open-shop and job-shop. In classical scheduling, the high complexity of such problems means that practical approaches to solving them usually involve heuristic strategies. Among these, genetic algorithms have proved to be a powerful tool to tackle this kind of problems, due to their ability to cope with huge search spaces involved in optimising schedules [4]. All the above motivates our description of a fuzzy job-shop problem and of a genetic algorithm to solve it.

2 Description of the Problem

The *job shop scheduling problem*, also denoted *JSSP*, consists in scheduling a set of jobs $\{J_1, \ldots, J_n\}$ on a set of physical resources or machines $\{M_1, \ldots, M_m\}$, subject to a set of constraints. There are *precedence constraints*, so each job J_i, $i = 1, \ldots, n$, consists of m tasks $\{\theta_{i1}, \ldots, \theta_{im}\}$ to be sequentially scheduled. Also, there are *capacity constraints*, whereby each task θ_{ij} requires the uninterrupted

* All authors supported by MCYT-FEDER Grant TIC2003-04153.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 360–369, 2006.
© Springer-Verlag Berlin Heidelberg 2006

and exclusive use of one of the machines for its whole processing time. In addition, we may consider *due-date constraints*, where each job has a maximum completion time and all its tasks must be scheduled to finish before this time. The goal is twofold: we need to find a *feasible* schedule, so that all constraints hold and then we want this schedule to be *optimal*, in the sense that its *makespan* (i.e., the time it takes to finish all jobs) is minimal.

2.1 Uncertain Processing Times and Flexible Constraints

In real-life applications, it is often the case that the exact duration of a task is not known in advance. For instance, in ship-building processes, some tasks related to piece cutting and welding are performed by a worker and, depending on his/her level of expertise, the task will take a different time to be processed. Hence, it is impossible to know a priori the exact duration of this task. However, based on previous experience, an expert may have some knowledge about the duration, thus being able to estimate, for instance, an interval for the possible processing time or its most typical value. Clearly, classical job shop problems, are not adequate to deal with this type of situations. Instead, it is necessary to somehow model uncertain processing times and thus take advantage of the expert's knowledge.

It is possible to find many examples in the literature where fuzzy numbers are used to represent uncertain processing times. Fuzzy sets proved an alternative to probability distributions, which require a deeper knowledge of the problem and usually yield a complex calculus. When there is little knowledge available, the crudest representation for uncertain processing times would be a human-originated confidence interval. If some values appear to be more plausible than others, a natural extension is a fuzzy interval or a fuzzy number. The simplest model of fuzzy interval is a *triangular fuzzy number* or *TFN*, using only an interval $[a^1, a^3]$ of possible values and a single plausible value a^2 in it. That is, for a TFN A, denoted $A = (a^1, a^2, a^3)$, the membership function takes a triangular shape completely determined by the three real numbers, $a^1 \leq a^2 \leq a^3$ as follows:

$$\mu_A(x) = \begin{cases} 0 & : x < a^1 \\ \frac{x-a^1}{a^2-a^1} & : a^1 \leq x \leq a^2 \\ \frac{x-a^3}{a^2-a^3} & : a^2 < x \leq a^3 \\ 0 & : a^3 < x \end{cases} \tag{1}$$

To compute the completion time of a given task, it is necessary to add the task's duration to its starting time. This can be done using *fuzzy number addition*, which in the case of TFNs $A = (a^1, a^2, a^3)$ and $B = (b^1, b^2, b^3)$ is reduced to adding three pairs of real numbers as follows:

$$A + B = (a^1 + b^1, a^2 + b^2, a^3 + b^3) \tag{2}$$

A consequence of this operation is that completion times are TFNs as well.

Another situation where the need of fuzzy number arithmetic arises is when the starting time for a given task θ must be found. Here, it is necessary to find the

maximum between two TFNs, the completion time of the task preceding θ in its job J and that preceding θ in its resource M. Now, given two TFNs $A = (a^1, a^2, a^3)$ and $B = (b^1, b^2, b^3)$, the *maximum* $A \vee B$ is obtained by extending the lattice operation max on real numbers using the Extension Principle. However, the calculation of the resulting membership function might be quite complex. Also, the result of such an operation, while still being a fuzzy number, is not guaranteed to be a TFN. For these reasons, we approximate $A \vee B$ by a TFN, $A \sqcup B$, given by:

$$A \vee B \approx A \sqcup B = (a^1 \vee b^1, a^2 \vee b^2, a^3 \vee b^3) \tag{3}$$

The approximation $\sqcup$, proposed in [5] for 6-point fuzzy numbers, may coincide with the maximum $\vee$. Even if this is not the case, the support of both fuzzy sets $A \vee B$ and $A \sqcup B$ is exactly the same and the unique point x with full membership in $A \sqcup B$ also has full membership in $A \vee B$.

Using the addition and the maximum $\sqcup$, it is possible to find the completion time for each job. The fuzzy makespan C_{max} would then correspond to the greatest of these TFNs. Unfortunately, neither the maximum $\vee$ nor its approximation $\sqcup$ can be used to find such TFN, because they do not define a total ordering in the set of TFNs. Instead, it is necessary to use a method for *fuzzy number ranking* [6]. The chosen method consists in obtaining three real numbers $C_1(A), C_2(A), C_3(A)$ from each TFN A as follows:

$$\begin{aligned} C_1(A) &= \frac{a^1 + 2a^2 + a^3}{4}, \\ C_2(A) = a^2, \quad C_3(A) &= a^3 - a^1 \end{aligned} \tag{4}$$

Using real number comparisons, it is then possible to establish a total ordering in any set of TFNs according to Algorithm 1.

1: order the *TFNs* using C_1
2: **if** there are TFNs with identical value of C_1 **then**
3: order these TFNs using C_2
4: **if** there are TFNs with identical value of C_1 and C_2 **then**
5: rank them using C_3

Algorithm 1. Ranking Method for TFNs

In practice, if due-date constraints exist, they are often flexible. For instance, a customer may have a preferred delivery date d^1, but some delay will be allowed until a later date d^2, after which the order will be cancelled. We would then be completely satisfied if the job finishes before d^1 and after this time our level of satisfaction would decrease, until the job surpasses the later date d^2, after which date we will be clearly dissatisfied. The satisfaction of a due-date constraint becomes a matter of degree, our degree of satisfaction that a job is finished on a certain date. A common approach to modelling such satisfaction levels is to use a fuzzy set D with linear decreasing membership function:

$$\mu_D(x) = \begin{cases} 1 & : x \leq d^1 \\ \frac{x-d^2}{d^1-d^2} & : d^1 < x \leq d^2 \\ 0 & : d^2 < x \end{cases} \tag{5}$$

Such membership function expresses a flexible threshold "less than", representing the satisfaction level $sat(t) = \mu_D(t)$ for the ending date t of the job [2]. However, when dealing with uncertain task durations, the job's completion time is no longer a real number t, but a TFN C. In this case, the degree to which a completion time C satisfies the due-date constraint D may be measured using the following *agreement index* [8],[7]:

$$AI(C, D) = \frac{area(D \cap C)}{area(C)} \tag{6}$$

The intuition behind this definition is to measure the degree to which C is contained in D (the degree of subsethood).

2.2 Definition of the Objective Function

Once we have established a means of modelling uncertain duration times and flexible due-dates, we can find a schedule for a given problem. Let us assume that resource and precedence constraints hold (otherwise, the schedule is unfeasible and hence is not a solution). Every job J_i, $i = 1, \ldots, n$ has a fuzzy completion time C_i; a fuzzy makespan C_{max} may be obtained from these completion times and, in the case that a due date D_i exists for job J_i, the agreement index $AI_i = AI_i(C_i, D_i)$ measures to what degree the due date is satisfied. Based on this information, it is necessary to decide on the quality of this schedule.

If flexible due-date constraints exist, the degree of feasibility of the given schedule may be obtained by combining the satisfaction degrees AI_i, $i = 1, \ldots, n$. If the aim is that due dates be satisfied in average, the degree to which a schedule s is feasible is given by:

$$AI_{av} = \frac{1}{n} \sum_{i=1}^{n} AI_i \tag{7}$$

A more restrictive approach is to expect that all due dates be satisfied, so satisfaction degrees are combined using the minimum aggregation operator as follows:

$$AI_{min} = \min_{i=1,\ldots,n} AI_i \tag{8}$$

The value of AI_{av} can be seen as the probability $\Pr(F)$ of the fuzzy event F "*the schedule s is feasible*" over the finite domain of jobs $D = \{J_1, \ldots, J_n\}$, provided that the membership of job J_i in F is $\mu_F(J_i) = AI_i$, $i = 1, \ldots, n$. Similarly, AI_{min} corresponds to the necessity measure $N(F)$ of the fuzzy event F over the finite domain D [9]. Clearly, $AI_{av}, AI_{min} \in [0, 1]$ and $AI_{min} \leq AI_{av}$. Both measure the degree to which due-date constraints are satisfied by the schedule,

and our aim should be to maximise them. However, the two measures model different requirements for due dates and encourage different behaviours.

Regarding makespan, the "smaller" C_{max} is, the better the schedule is. Now, because C_{max} is a TFN, it is not totally clear what is meant by "smaller". If we consider the total ordering defined by the ranking method, it would mean a smaller $C_1(C_{max})$ and our goal should be to minimise this quantity.

Given both measures of feasibility and the makespan and depending on the final goal of the job-shop scheduling problem, we may define different objective functions.

If no due-date constraint is considered and the only goal is to find a schedule with minimum makespan, the objective function will be given by:

$$f_1 = \frac{1}{C_1(C_{max})} \tag{9}$$

When due-date constraints are present and the only goal is to find a feasible schedule, the objective function of the job-shop problem may be defined alternatively as:

$$f_2 = AI_{av} \quad f_3 = AI_{min} \tag{10}$$

Finally, even if due-date constraints hold, we may also want to minimise the makespan. Here, the degree of feasibility, given by AI_{av} or AI_{min}, must be maximised and, at the same time, the makespan (in fact, $C_1(C_{max})$) must be minimised. The combination of the two goals yield the following objective functions (depending on the feasibility measure used):

$$f_4 = \frac{AI_{av}}{C_1(C_{max})} \quad f_5 = \frac{AI_{min}}{C_1(C_{max})} \tag{11}$$

Having proposed the above objective functions, we may define the *Fuzzy Job Shop Scheduling Problem* or *FJSSP* as the problem of maximising $f_i, i = 1, \ldots, 5$, subject to precedence and capacity constraints.

To our knowledge, despite their simplicity, the above objective functions for the FJSSP have not been yet considered in the literature. For instance, the FJSSP is considered in [10] and [7], but the definition of the objective function, based on fuzzy decision making, is completely different. A job-shop problem is also considered [5], but uncertain durations are modelled using 6-point fuzzy numbers, there are no due dates, and the only objective of minimising makespan is achieved based on fuzzy number comparison. Finally, similar objective functions appear in [8], but in the setting of a different and less complex problem, the fuzzy flow-shop problem.

3 Using Genetic Algorithms to Solve FJSSP

In classical JSSP, the search for an optimal schedule is usually limited to the space of active schedules. One of the best-known algorithms to find active schedules is the *G&T Algorithm* [11], which allows to use complementary techniques

1: $A = \{\theta_{i1},\ i = 1, \ldots, n\}$; /*first task of each job*/
2: **while** $A \neq \emptyset$ **do**
3: Find the task $\theta' \in A$ with minimum earliest completion time /*$CT(\theta)^1$*/;
4: Let M' be the machine required by θ' and B the subset of tasks in A requiring machine M';
5: Delete from B any task that cannot overlap with θ'; /*$ST(\theta)^1 > CT(\theta')^3$*/
6: Select $\theta^\star \in B$ (according to some criteria) to be scheduled;
7: Remove $\theta^\star$ from A and, if $\theta^\star$ is not the last task of its job, insert in A the task following $\theta^\star$ in the job;

Algorithm 2. Fuzzy G&T

to reduce the search space [12]. Also, it can be used as a basis for efficient genetic algorithms (GA), successful in solving classical job-shop problems. We describe a possible extension of G&T for the FJSSP (see Algorithm 2) and a GA to solve the FJSSP based on this algorithm. Both algorithms were first proposed for a different objective function that also needed to be maximised in [10] and were inspired in the work from [7].

Chromosomes are a direct codification of schedules. If there are n jobs and m machines, each individual will be represented by a $n \times m$ matrix, where element (i, j) represents the completion time for the task in job J_i requiring resource M_j. Therefore, each row is the schedule of a job's tasks over the corresponding resources. Each chromosome in the *initial population* for the GA can be generated with fuzzy G&T algorithm, choosing a task at random from the conflict set B. To prevent premature convergence, it is advisable that the initial population be diverse enough. Hence, a new individual will only be incorporated to the population if similarity to other members of the population is less than a given threshold σ. Let $Pr_I(\theta)$ be the set of tasks preceding θ in its machine according to the ordering induced by individual I and let $Su_I(\theta)$ be the set of tasks following θ in its machine w.r.t. the same ordering. Then, the *similarity* between two individuals I_1 and I_2 is defined using phenotype distance as follows:

$$Sim(I_1, I_2) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} (|Pr_{I_1 \cap I_2}(\theta_{ij})| + |Su_{I_1 \cap I_2}(\theta_{ij})|)}{n \cdot m \cdot (m-1)} \tag{12}$$

where $|Pr_{I_1 \cap I_2}(\theta_{ij})|$ denotes the cardinal of $Pr_{I_1}(\theta_{ij}) \cap Pr_{I_2}(\theta_{ij})$ and $|Su_{I_1 \cap I_2}(\theta_{ij})|$ denotes de cardinal of $Su_{I_1}(\theta_{ij}) \cap Su_{I_2}(\theta_{ij})$.

The value of the *fitness function* for a chromosome is simply the value of the objective function for the corresponding schedule.

The *crossover operator*, applied with probability p_m, consists in performing the fuzzy G&T algorithm and solve non-determinism situations using the information from the parents. Every time the conflict set B has more than one element, the selected task is that with earliest completion time in the parents, according to the ranking algorithm. The *mutation operator* is embedded in the crossover operator, so that, with a given probability p_m, the task from the conflict set is selected at random.

```
1: Generate initial population divided in k groups P_1, ..., P_k containing K individuals
   each;
2: while terminating condition T_1 is not satisfied do
3:    for i = 1; i ≤ k; i++ do
4:       repeat
5:          select 2 parents at random from P_i;
6:          obtain 3 children by crossover and mutation;
7:          select the best of 3 children and the best of remaining children and parents
            for the new population NP_i;
8:       until a new population NP_i is complete
9:       Replace the worst individual in NP_i with the best of P_i.
10: Merge P_1, ..., P_k into a single population P;
11: while Terminating condition T_2 is not satisfied do
12:    Obtain a new population from P following the scheme above;
```

Algorithm 3. Genetic Algorithm for FJSSP

The *general scheme* of the GA, in Algorithm 3, is designed to avoid premature convergence to local optima by using a niche-based system. The population is initially divided in k sub-populations, containing K individuals each. This is more feasible, from a computational point of view, than generating a single initial population. Each sub-population evolves separately, until a certain convergence is obtained (in practice, for I_{min} generations). At this stage, these sub-populations are merged into a single population of N individuals, which will again evolve until some terminating condition holds (in practice, when a total of I_{max} generations is reached).

4 Experimental Results

Unfortunately, benchmark examples of FJSSP in the literature are scarce, clearly a problem for any thorough experimentation, where a sufficiently large and diverse set of problems is needed. For this reason, we propose a novel heuristic method to generate new problems. It will later be used to provide a sample of problems to test the different objective functions introduced in Section 2.

4.1 Generation of New Problems

In order to define a problem of size $n \times m$, where n is the number of jobs and m is the number of resources, the following must be defined: capacity constraints, assigning a machine to each task of every job, uncertain durations, in the form of TFNs, and flexible due-dates. Here, we have generated new problems (ten problems of size 10×10 and ten problems of size 20×5) using the following heuristic method. For a problem of size $n \times m$, *capacity constraints* are defined by a matrix R of size $n \times m$ where row i is a random permutation of $(1, \ldots, m)$ representing the machine assignments in job J_i. Regarding the *duration* of a given task, its most typical value a^2 is obtained at random from the interval

Table 1. Average results obtained with different objective functions. Parameter values are: $p_m = 0.03$, $p_c = 0.9$, $\sigma = 0.8$ and $N/I_{min}/I_{max}$ are 200/100/200, except for 6×6, where they are 100/50/100.

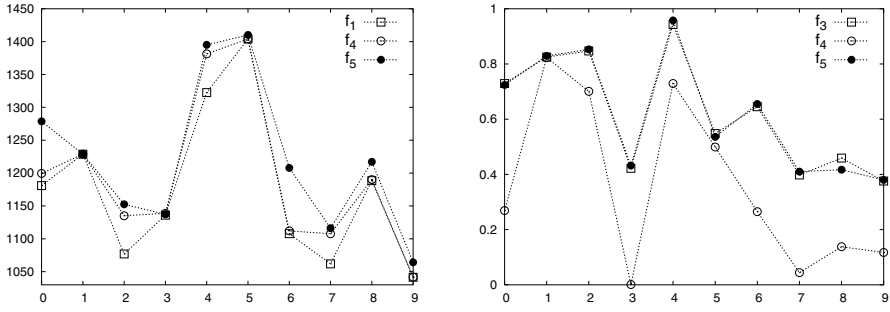
Problems	objective function	AI_{av}	AI_{min}	$C_1(C_{max})$
S&K	f_1	0.678	0.097	98.700
	f_2	0.888	0.545	115.960
	f_3	0.856	0.700	119.133
	f_4	0.871	0.525	105.192
	f_5	0.854	0.676	110.485
	f	0.811	0.444	106.346
10×10	f_1	0.948	0.662	870.591
	f_2	1.000	0.997	966.032
	f_3	1.000	0.999	964.852
	f_4	0.994	0.948	883.685
	f_5	0.999	0.994	903.654
	f	0.996	0.966	936.173
20×5	f_1	0.737	0.016	1174.879
	f_2	0.953	0.505	1251.991
	f_3	0.877	0.619	1237.380
	f_4	0.934	0.359	1193.927
	f_5	0.874	0.619	1220.791
	f	0.695	0.082	1273.926

$[1, 99]$ and a^1 and a^3 are random values from $[int(\frac{2}{3}a^2), a^2]$ and $[a^2, int(\frac{4}{3}a^2)]$ respectively, where $int(x)$ denotes the closest integer to a given real number x.

Due-date values are the most difficult to define. If they are too strict, the problem will have no solution and if they are too lenient, due-date constraints will always be satisfied, which is equivalent to having no constraints at all. For a given job J_i, let $\iota_i = \sum_{j=1}^{m} a_{i,j}^2$ be the sum of most typical durations across all its tasks. Also, for a given task $\theta_{i,j}$ let $\rho_{i,j}$ be the sum of most typical durations of all other tasks requiring the same machine as $\theta_{i,j}$, $\rho_{i,j} = \sum_{\theta_{i,j} \neq \theta : M(\theta) = M(\theta_{i,j})} a^2(\theta)$, where $M(\theta)$ denotes the machine required by task θ and $a^2(\theta)$ denotes its most typical duration. Finally, let $\rho_i = \max_{j=1,\ldots,m} \rho_{i,j}$ be the maximum of such values across all tasks in job J_i. Then, the earlier due-date d^1 is taken as a random value from $[d_m, d_M]$, where $d_m = \iota_i + 0.5\rho_i$ and $d_M = \iota_i + \rho_i$. The later due-date d^2 is a random value from $[d^1, int(1.1d^1)]$.

4.2 Results of the GA with Different Objective Functions

In addition to the 20 new problems, we consider the problems proposed in [7], three of size 6×6 and three of size 10×10, denoted S&K. For all 26 problems, we have run the GA using the five objective functions proposed in Section 2 and the objective function based on fuzzy decision making proposed in [10] (denoted f hereafter). For each problem and objective function, the GA from Section 3 is executed 20 times with the parameter values used in [10] ($p_m/p_c/\sigma/N/I_{min}/I_{max}$



a) Cmax (lower is better) b) AImin (higher is better)

Fig. 1. Results for problems of size 20×5 using different objective functions

equal to 0.03/0.9/0.8/200/100/200). For the obtained schedule, we measure AI_{av}, AI_{min} and $C_1(C_{max})$. Table 1 is a summary of the obtained results, with average values across all problems in a family. Notice that for problems of size 10×10, due dates that are in general easy to satisfy, whilst for problems of size 20×5 due dates are quite strict in some jobs. Let us now see if the results support the arguments used to define the different objective functions in Section 2. As expected when the objective functions were introduced, the results indicate that, when only the productivity goal of minimising C_{max} is considered, f_1 should be used. Notice that, in accordance with what has just been said, the lowest values of C_{max} are obtained with schedules for which at least a due-date restriction is not satisfied at all.

More surprising are the results with respect to due-date satisfaction. At first, we may feel tempted to conclude that, if the goal is to respect delivery dates in average, then f_2 should be used and, if the goal is to respect all delivery dates, then f_3 should be used. This would certainly correspond to the motivation for defining f_2 and f_3. However, a more careful look shows that if we use f_4 and f_5 instead, due dates are satisfied to almost the same degree and there is the added benefit of reducing the makespan and improving in productivity. Therefore, if the goal is to satisfy due dates, it seems preferable to use the more complex objective functions f_4 or f_5, instead of just using f_2 or f_3. Notice as well that using AI_{av} as objective function does not always provide high values for AI_{min}. Finally, if the goal consists in both maximising due-date satisfaction and minimising makespan, the results suggest that the objective function should be f_5. For all 26 problems, AI_{av} values are similar with both f_4 and f_5, whilst AI_{min} does improve considerably when f_5 is used. Regarding $C_1(C_{max})$, f_4 does obtain better schedules, but the difference does not compensate the loss of due-date satisfaction in AI_{min}. This is further illustrated by Figure 1. In any case, the objective function f, defined in [10] to simultaneously maximise $\frac{1}{C_1(C_{max})}$, AI_{av} and AI_{min}, obtains worse results than either f_4 or f_5. We may conclude then that overall it is preferable to use the simpler objective function f_5, proposed herein.

5 Conclusions and Future Work

We have considered the FJSSP, a version of JSSP that tries to model the imprecise nature of data in real-world problems, using fuzzy sets to represent uncertain processing times and considering flexible due-dates. Different objective functions have been proposed, depending on whether the aim is to optimise productivity or respect delivery dates, and their behaviour has been compared based on the results obtained using a GA. We have seen that, overall, it is preferable to use the function denoted f_5. This also obtains better results than the function from [10] and has the further advantage that its definition involves no parameters.

References

1. Brucker, P.: Scheduling Algorithms. 4th edn. Springer (2004)
2. Dubois, D., Fargier, H., Fortemps, P.: Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge. European Journal of Operational Research **147** (2003) 231–252
3. Słowiński, R., Hapke, M., eds.: Scheduling Under Fuzziness. Volume 37 of Studies in Fuzziness and Soft Computing. Physica-Verlag (2000)
4. Bierwirth, C., Mattfeld, D.C.: Production scheduling and rescheduling with genetic algorithms. Evolutionary Computation **7** (1999) 1–17
5. Fortemps, P.: Jobshop scheduling with imprecise durations: a fuzzy approach. IEEE Transactions of Fuzzy Systems **7** (1997) 557–569
6. Bortolan, G., Degani, R.: A review of some methods for ranking fuzzy subsets. In Dubois, D., Prade, H., Yager, R., eds.: Readings in Fuzzy Sets for Intelligence Systems. Morgan Kaufmann (1993) 149–158
7. Sakawa, M., Kubota, R.: Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms. European Journal of Operational Research **120** (2000) 393–407
8. Celano, G., Costa, A., Fichera, S.: An evolutionary algorithm for pure fuzzy flowshop scheduling problems. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **11** (2003) 655–669
9. Dubois, D., Prade, H.: A review of fuzzy set aggregation connectives. Information Sciences **36** (1985) 85–121
10. González Rodríguez, I., Vela, C.R., Puente, J.: An evolutionary approach to designing and solving fuzzy job-shop problems. Lecture Notes in Computer Science **3562** (2005) 74–83
11. Giffler, B., Thomson, G.L.: Algorithms for solving production scheduling problems. Operations Research **8** (1960) 487–503
12. Varela, R., Vela, C.R., Puente, J., Gómez, A.: A knowledge-based evolutionary strategy for scheduling problems with bottlenecks. European Journal of Operational Research **145** (2003) 57–71

Scheduling with Memetic Algorithms over the Spaces of Semi-active and Active Schedules

Miguel A. González, Camino R. Vela, and Ramiro Varela*

Artificial Intelligence Center, University of Oviedo (Spain)
raist@telecable.es, camino@aic.uniovi.es, ramiro@uniovi.es

Abstract. The Job Shop Scheduling Problem is a paradigm of Constraint Satisfaction Problems that has interested to researchers over the last decades. In this paper we confront this problem by means of a Genetic Algorithm that is hybridized with a local search method. The Genetic Algorithm searches over the space of active schedules, whereas the local search does it over the space of semi-active ones. We report results from an experimental study over a set of selected problem instances showing that this combination of search spaces is better than restricting both algorithms to search over the same space. Furthermore we compare with the well-known Genetic Algorithms proposed by D. Mattfeld and the Branch and Bound procedure proposed by P. Brucker and obtain competitive results.

1 Introduction

The Job Shop Scheduling Problem (JSSP) is a Constraint Satisfaction Problem (CSP) that has interested to many researchers over the last years. Consequently we can found in the literature a great number of approaches based on different meta-heuristics, constraint satisfaction techniques or operational research algorithms [7]. In particular Genetic Algorithms (GAs) and Branch and Bound (BB) algorithms are two of the most important approaches. For example the BB algorithm proposed by P. Brucker et al. in [3] and [4] is one of the most efficient exact approaches proposed so far and it is able to solve the majority of instances up to a size of about 15 jobs and 15 machines. Regarding GAs, the approaches proposed by Ch. Bierwirth and D. Mattfeld in [1], [2], [8] or the approach proposed by T. Yamada and R. Nakano in [15] are among the most efficient ones.

In this paper we propose a GA that combines judiciously a number of ideas proposed in the literature to obtain an efficient approach. To codify chromosomes we have chosen the permutation with repetition schema proposed in [1]. These chromosomes are then evaluated with a variant of the Giffler and Thomson algorithm proposed in [6]. Also we have considered a number of local search methods that rely on the concepts of critical path and critical block. The main principles of these models were proposed by R. J. M. Vaessens et al. [11], P. J. M. Van Lardoven et al. [12], M. Dell' Amico and M. Trubian [5], and E Taillard [10]. These models were afterwards used and developed by other researchers such as

* Authors supported by MCYT-FEDER Grant TIC2003-04153

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 370–379, 2006.
© Springer-Verlag Berlin Heidelberg 2006

for example D. Mattfeld [8] or E. Nowicki and C. Smutnicki [9] in combination with Genetic Algorithms.

The rest of the paper is organized as follows. In section 2 the JSSP is defined. In section 3 we describe the basic components of the GA used for solving the JSSP. Then Section 4 outlines the Local Search strategies, termed N_2, N_3 and N_4 in the literature, that we have exploited to improve the GA. Section 5 describes the experimental study, and finally in Section 6 we give the main conclusion of the paper.

2 Problem Formulation

In this paper we consider the Job Shop Scheduling Problem, also denoted JSSP. This problem requires scheduling a set of N jobs $\{J_0, \ldots, J_{N-1}\}$ on a set of M physical resources or machines $\{R_0, \ldots, R_{M-1}\}$. Each job J_i consists of a set of tasks or operations $\{\theta_{i0}, \ldots, \theta_{iM-1}\}$ to be sequentially scheduled. Each task θ_{il} has a single resource requirement, a fixed processing time or duration $p\theta_{il}$ and a start time $st\theta_{il}$ whose value should be determined.

The JSS has two binary constraints: *precedence constraints* and *capacity constraints*. Precedence constraints are defined by the sequential routings of the tasks within a job and translate into linear inequalities of the type: $st\theta_{il} + p\theta_{il} \leq st\theta_{i(l+1)}$ (i.e. θ_{il} before $\theta_{i(l+1)}$). Capacity constraints restrict the use of each resource to only one task at a time and translate into disjunctive constraints of the form: $st\theta_{il} + p\theta_{il} \leq st\theta_{jk} \vee st\theta_{jk} + p\theta_{jk} \leq st\theta_{il}$. The most widely used objective is to obtain a feasible schedule such that the completion time, i.e. the *makespan* denoted C_{max}, is minimized.

A problem solution may be represented by an acyclic graph showing job routings on the machines. Figure 1 shows a feasible solution to a problem with 3 jobs and 3 machines. Arcs are labelled with the processing time of the operation at the outcoming node. The makespan is the cost of the longest or critical path between the dummy nodes *start* and *end*. A subset of consecutive operations on this path requiring the same machine is a *critical block*.

Every operation v has at most two direct predecessors denoted PM_v and PJ_v and two successors denoted SM_v and SJ_v in the solution graph. PM_v is the predecessor of v in the machine sequence and PJ_v is the predecessor of v in its job sequence. Analogous, SJ_v and SM_v are the job successor and the machine successor respectively. When v is the first operation in its job $PM_v = PJ_v = start$, and analogous if v is the last one $SM_v = SJ_v = end$. Naturally, direct predecessors are not defined for operation *start* and also direct successors are not defined for operation *end*.

Given a solution graph, the earliest starting time that can be assigned to an operation v is the *head* of v and it is denoted r_v. Then the earliest completion time of an operation v is denoted C_v and is calculated as $r_v + p_v$. Hence, the head of the operation v is actually the cost of the longest path from node *start* to node v and is determined by

$$r_v = max(r_{PM_v} + p_{PM_v}, r_{PJ_v} + p_{PJ_v}) = max(C_{PM_v}, C_{PJ_v})$$

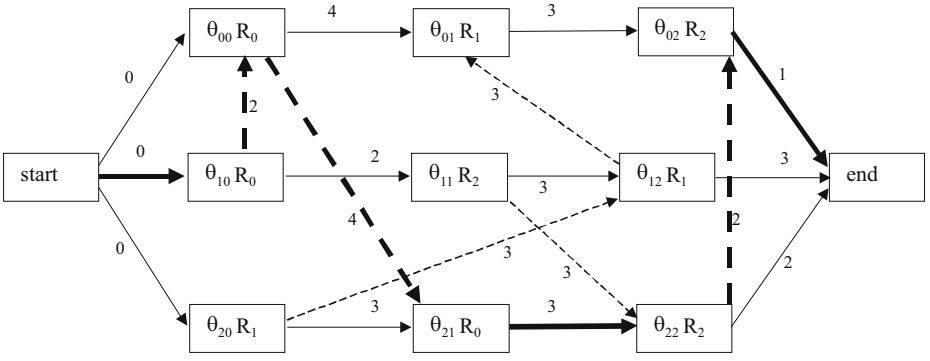


Fig. 1. A feasible schedule to a problem with 3 jobs and 3 machines. The bold face arcs shown a critical path whose length, that is the makespan, is 12.

r_{start} and p_{start} being 0. Analogous, the *tail* of an operation v in the solution graph is the cost of the longest path from node v to node *end* and is defined by

$$q_v = max(p_{SM_v} + q_{SM_v}, p_{SJ_v} + q_{SJ_v})$$

also q_{end} and p_{end} being 0. The tail of operation *start* is

$$q_{start} = max_{0 \leq i < N}(q_{\theta_{i0}} + p_{\theta_{i0}})$$

and the head of operation *end*

$$r_{end} = max_{0 \leq i < N}(r_{\theta_{iM-1}} + p_{\theta_{iM-1}})$$

No buffer time exist for any critical operation v, hence $C_{max} = r_v + p_v + q_v$. These equations are relevant for the feasibility test and makespan estimation considered in Section 4.

3 A Genetic Algorithm for the JSSP

The JSSP is a paradigm of constraint satisfaction problems and has been confronted by many heuristic techniques. In particular Genetic Algorithms [1], [2], [6], [8], [13] are a promising approach due to its ability to be combined with other techniques such as tabu search and simulated annealing. Moreover GAs allow to exploit any kind of heuristic knowledge from the problem domain. In doing so, GAs are actually competitive with the most efficient methods for JSSP.

As mentioned above, in this paper we consider a conventional GA for tackling the JSSP and make an experimental study of two local search methods. The selected approach to codify chromosomes is the permutation with repetition proposed by C. Bierwirth in [1]: a chromosome is a permutation of the set of

operations, each one being represented by its job number. This way a job number appears within a chromosome as many times as the number of job operations. For example the chromosome *(1 0 0 2 1 2 0 1 2)* actually represents the permutation of operations $(\theta_{10}, \theta_{00}, \theta_{01}, \theta_{20}, \theta_{11}, \theta_{21}, \theta_{02}, \theta_{12}, \theta_{22})$. This permutation should be understood as expressing partial schedules for every set of operations requiring the same machine. This codification presents a number of interesting characteristics: it is easy to evaluate with different algorithms and allows for efficient genetic operators. In [14] this codification is compared with other permutation based codifications and demonstrated to be the best one in average over the set of 12 selected problem instances considered in the experimental study of this work.

The way these partial schedules are translated to an actual schedule depends on the decoding algorithm. Here we consider a variant of the well-known $G\&T$ algorithm proposed by Giffler and Thomson in [6]. The $G\&T$ algorithm is an active schedule builder.

Definition 1 (*Active schedule*)**.** *A schedule is active if for any operation to start earlier, at least another one must be delayed.*

Active schedules are good in average and the space of active schedules contains at least an optimal one. For these reasons it is worth to restrict the search to this space. The $G\&T$ algorithm can be modified in order to reduce even more the search space by means of a parameter $\delta \in [0, 1]$ (see algorithm 1). When $\delta < 1$ the search space gets narrowed so that optimal schedules might no longer be included. At the extreme $\delta = 0$ the search is constrained to non-delay schedules.

Definition 2 (*Non–delay schedule*)**.** *A schedule is non-delay or dense if it is never the case that a resource is idle when a requiring operation is available.*

Experience demonstrates that as long as parameter δ decreases, the mean value of solutions within the search space tends to improve. Also of interest are often semi-active schedules.

Definition 3 (*Semi–active schedule*)**.** *A schedule is semi-active if for any operation to start earlier, the relative ordering of at least two operations must be swapped.*

It is easy to devise a decoding algorithm over the space of semi-active schedules with lower complexity than the $G\&T$ algorithm, but at the same the semi-active schedule is usually worse than the active schedule for a given chromosome.

The rest of the components of the GA are rather conventional. In the selection phase all chromosomes of the population are grouped into pairs. Then each pair is mated and the resulting offsprings mutated according to crossover and mutation probabilities Pc and Pm. We have chosen the job order crossover JOX described in [1]. Given two parents JOX selects a random subset of jobs and copies their genes to the offspring at the same positions as in the first parent, then the remaining genes are taken from the second parent so as they maintain their relative ordering. The mutation operator simply swaps two consecutive

1: Let $A = \{\theta_{j0}, 0 \leq j < n\}$; /*$n$ is the number of jobs*/
2: **while** $A \neq \emptyset$ **do**
3: $\forall \theta \in A$ let $st\theta$ be the earliest starting time of θ if it is scheduled the next one;
4: Let $\theta_1 \in A$ such that $st\theta_1 + p\theta_1 \leq st\theta + p\theta, \forall \theta \in A$;
5: Let $M = MR(\theta_1)$; /*$MR(\theta)$ is the machine required by operation θ*/
6: Let $B = \{\theta \in A : MR(\theta) = M, st\theta < st\theta_1 + du\theta_1\}$;
7: Let $\theta_2 \in B$ such that $st\theta_2 \leq st\theta, \forall \theta \in B$; /*the earliest starting time of every operation in B, if it is selected next, is a value of the interval $[st\theta_2, st\theta_1 + du\theta_1]$*/
8: Reduce set B: $B = \{\theta \in B : st\theta \leq st\theta_2 + \delta((st\theta_1 + du\theta_1) - st\theta_2), \delta \in [0, 1]\}$;/*now the interval is reduced to $[st\theta_2, st\theta_2 + \delta((st\theta_1 + du\theta_1) - st\theta_2)]$*/
9: Select $\theta^\star \in B$ such that is the leftmost operation in the chromosome and schedule it at time $st\theta^\star$;
10: Let $A = A \setminus \{\theta^\star\} \cup \{SUC(\theta^\star)\}$; /*$SUC(\theta)$ is the next operation to θ in its job, if any exists*/

Algorithm 1. Chromosome decoding algorithm G&T hybrid

genes selected at random. Finally the acceptation of chromosomes to the next generation is done by tournament from each group of two parents and their two offsprings.

4 Local Search

Conventional GAs as the one described above often produce moderate results. However meaningful improvements can be obtained by means of hybridization with other methods. One of these techniques is local search, in this case the GA is called a Memetic Algorithm. Roughly speaking local search is usually implemented by defining a neighborhood for each point in the search space as the set of chromosomes reachable by a given transformation rule. Then a chromosome is replaced in the population by one of its neighbors, if any of them satisfies the acceptation criterion. The local search from a given point terminates after a number of iterations or when no neighbor satisfies the acceptation criterion.

In this paper we consider the local search strategies termed N_2, N_3 and N_4 by D. Mattfeld in [8]. All these strategies rely on the concepts of critical path and critical block. They consider every critical block of a critical path and make a number of moves on the operations of each block. After a move inside a block the feasibility must be tested. Since an exact procedure is computationally prohibitive, the feasibility is estimated by the approximate algorithm described in Section 4.1 that is proposed by Dell' Amico and Trubian in [5]. This estimation ensures feasibility at the expense of omitting a few feasible solutions. Transformation rules for N_2, N_3 and N_4 are as follows.

Definition 4 (N_2). *Let operation v be a member of a block b such that $b = (b'vb'')$. In a neighboring solution v is moved to the beginning of b or to the the end of b if feasibility is preserved, otherwise v is moved closest to the first or the last operation of b for which the feasibility is preserved.*

Definition 5 (N_3). *Let v and w be successive operations on a critical path, and let PM_v and SM_w be the machine predecessor of v and the machine successor of w respectively. If neither PM_v and SM_w belong to the same block as v and w, then the only permutation is (w, v), otherwise all possible permutations of PM_v, v, w and v, w, SM_w are considered as neighboring if v and w are also reversed.*

Due to neither $N_2 \subset N_3$ nor $N_3 \subset N_2$ it is reasonable to define the N_4 neighborhood as

Definition 6 (N_4). $N_4 = N_3 \cup N_2$.

The acceptation criterion is based on a makespan estimation which is done in constant time as it is described in Section 4.2, instead of calculating the exact makespan of each neighbor. The estimation provides a lower bound of the makespan. The selected neighbor is the one with the lowest makespan estimation whenever this value is lower than the makespan of the current chromosome.

4.1 Feasibility Checking

To test the feasibility of a solution resulting from a block move, a labeling algorithm to detect cycles in the solution graph can be used. This procedure is exact but is also computationally expensive. Therefore, it is more efficient an estimation for testing feasibility such as the algorithm proposed in [5] for N_2 given by the following lemma.

Lemma 1. *For a move of an operation v inside a block $b = (b', b'', v, b''')$ yielding the sequence (b', v, b'', b'''), a cycle in the resulting solution graph can exist if and only if there exist an operation w of b'' such that there is a path from SJ_w to PJ_v.*

Thus, it is sufficient to test the inequality

$$r_{SJ_w} + p_{SJ_w} > r_{PJ_v}$$

for all w of b'' to guarantee that the resulting schedule is feasible. Analogous, for a move of v inside a block $b = (b', v, b'', b''')$ yielding (b', b'', v, b''') it is also sufficient to test the inequality

$$r_{SJ_v} + p_{SJ_v} > r_{PJ_w}$$

for all w of b''.

In the case of N_3 the feasibility checking is even more simple. Due to the reversal of (v, w) always leads to a feasible schedule and the following lemma, that is proved in [5], it is sufficient to consider as the only neighbor the permutation with the smallest makespan estimation.

Lemma 2. *The estimated makespan of a (v, w) reversal is smaller or equal than any estimated makespan resulting from the reversal of two arcs if such a reversal leads to an infeasible solution*

4.2 Makespan Estimation

The makespan estimation is based on heads and tails. Let us consider firstly the situation where the arc *(v,w)* is reversed. Here is important to remark that either PM_v or SM_w are not critical, otherwise reversing the arc *(v,w)* does not produce any improvement. The values of r'_w, r'_v, q'_v, q'_w after reversing the arc *(v,w)* are

$$r'_w = max(r_{PM_v} + p_{PM_v}, r_{PJ_w} + p_{PJ_w})$$
$$r'_v = max(r'_w + p_w, r_{PJ_v} + p_{PJ_v})$$
$$q'_v = max(q_{SM_w} + p_{SM_w}, q_{SJ_v} + p_{SJ_v})$$
$$q'_w = max(q'_v + p_v, q_{SJ_w} + p_{SJ_w})$$

Therefore the makespan can be estimated by

$$C'_{max} = max(r'_w + p_w + q'_w, r'_v + p_v + q'_v)$$

which is a lower bound of the actual makespan after reversing *(v,w)*.

Now let us consider the general case of N_2 where not only successive operations are moved. Let $CB = (O_1, ..., O_i, ..., O_j, ..., On)$ a critical block of a machine m and suppose that operation O_j is moved before O_i. Therefore in the neighboring solution we have the sequence $(O_1, ..., O_{i-1}, O_j, O_i, ..., O_{j-1}, O_{j+1}..., On)$ on the machine m. The head of operations $(O_1, ..., O_{i-1})$ and the tail of operations of $(O_{j+1}..., On)$ do not change with respect to the original solution. However, the tail of operations $(O_1, ..., O_{i-1})$ and the head of operations $(O_{j+1}..., On)$ in general change. But in this case, neither the exact values nor lower bounds of these new tails and heads can be estimated without calculating the new heads and tails for the whole set of operations, which is very time consuming. On the contrary, exact values of new heads and tails of operations $(O_j, O_i, ..., O_{j-1})$ may be efficiently calculated as it is indicated below. So, these new values may be used for the makespan estimation of the neighboring solution. An analogous reasoning can be done if O_j is moved towards the end of the critical block CB.

Now, the set $(O_j, O_i, ..., O_{j-1})$ is denoted $(L_1, ..., L_l)$, and O_{i-1} and O_{j+1} are denoted $first$ and $last$ respectively. As r_{first} and q_{last} will not change from the original solution to a neighboring one, the estimation of the new makespan can be calculated as follows.

$$r'_{L_1} = max(r_{first} + p_{first}, r_{PJ_{L_1}} + p_{PJ_{L_1}})$$
$$r'_{L_2} = max(r'_{L_1} + p_{L_1}, r_{PJ_{L_2}} + p_{PJ_{L_2}})$$
$$\dots$$
$$r'_{L_l} = max(r'_{L_{l-1}} + p_{L_{l-1}}, r_{PJ_{L_l}} + p_{PJ_{L_l}})$$
$$q'_{L_l} = max(q_{last} + p_{last}, q_{SJ_{L_l}} + p_{SJ_{L_l}})$$
$$q'_{L_{l-1}} = max(q'_{L_l} + p_{L_l}, q_{SJ_{L_{l-1}}} + p_{SJ_{L_{l-1}}})$$
$$\dots$$

$$q'_{L_1} = max(q_{L_2} + p_{L_2}, q_{SJ_{L_1}} + p_{SJ_{L_1}})$$

$$C'_{max} = r'_{L_1} + p_{L_1} + q'_{L_1}$$

$$C'_{max} = max(C'_{max}, r'_{L_2} + p_{L_2} + q'_{L_2})$$

$$\ldots$$

$$C'_{max} = max(C'_{max}, r'_{L_l} + p_{L_l} + q'_{L_l})$$

This method is also suitable for N_3. In this case the subsequence $(L_1, ..., L_l)$ is given by the first or last three operations of the critical block.

5 Experimental Study

In this study we have considered the set of 12 selected problems with sizes ranging from 10×10 to 20×15 which were used in other works such as [8]. These problem instances are considered hard to solve due to their resistance to be solved by the BB procedure proposed by Brucker et al. [3], [4] with only one exception (FT10). This procedure solves to optimality almost every other instance of these sizes. We have experimented with different combinations of neighborhoods and compare with the genetic algorithms GA1 and GA3 proposed by Mattfeld in [8]. GA1 is similar to our version but uses semi-active decoding instead, and GA3 is GA1 with two additional refinements: a structured population mechanism that restricts mating to the neighbors placed on a grid, and a diffusion model that allows to control population diversity. In all cases we parameterize our GA so that the number of chromosomes evaluated is similar to the experiments reported in [8] with GA1 and GA3. When comparing with the BB procedure, we run both algorithms, GA and BB, on the same machine during similar time.

Table 1 shows results from the comparison of our GA with both algorithms GA1 and GA3, that were executed on a SUN 10/41 running Solaris Operating System. Our GA was run on a Pentium IV at 1.7 Ghz running Windows OS, so that it evaluates the same number of chromosomes (about 10000) and the same number of trials (50) were done. In order to do that we have run the GA with the following parameters: 100 chromosomes in the population, 140 generations, $p_c = 0.7$, $p_m = 0.1$ and $\delta = 1$ in $G\&T$ algorithm. The neighborhood is N_3 as in GA3. In average the running time of GA3 is 6 times the running time of GA if the same number of chromosomes is evaluated. The whole set of experiments was made only one time. Regarding mean values of the 50 solutions, GA is better than GA1 in all 12 cases, and also is better than GA3 in 9 cases and quite similar in 3. However for the best of the 50 solutions GA3 is better in 6 cases, while GA is better in 3. The best values reached for GA1 are not reported in [8].

Table 2 shows results from the comparison of GA with BB. In this case the neighborhood N_4 was used. As we can observe, in every case GA reached much better mean solutions than BB, with only one exception (ft10) where BB reached the optimal solution. In these experiments we have used populations of 200 chromosomes and also a number of 200 generations.

Table 1. GA with N_3 vs. GA3 and GA1

Problems	Avg. GA1	Best GA3	Avg. GA3	Best GA	Avg. GA
ABZ7(665)	687.9	668	682.9	670	678.74
ABZ8(670)	699.6	684	696.2	683	692.16
ABZ9(665)	716.2	702	712.6	694	707.6
FT10(930)	959.1	930	943.7	930	934.82
FT20(1165)	1181.9	1165	1180.3	1165	1176.58
LA21(1046)	1070.0	1047	1059.4	1053	1055.86
LA24(935)	955.9	938	945.3	938	945.98
LA25(977)	990 .0	977	986.6	978	984.34
LA27(1235)	1265.7	1236	1261.6	1248	1261.26
LA29(1153)	1212.1	1180	1199.9	1167	1193.16
LA38(1196)	1235.2	1201	1222.5	1207	1220.62
LA40(1222)	1258.0	1228	1243.7	1233	1243.62

Table 2. GA with N_4 vs. BB

Problems	Best GA	Avg. GA	CPU-GA	BB	CPU-BB
ABZ7(665)	669	673.6	139.98	726	160
ABZ8(670)	674	683.83	161.50	767	162
ABZ9(665)	688	695.37	158.37	821	160
FT10(930)	930	932.13	17.45	930	22
FT20(1165)	1165	1171.93	46.63	1179	60
LA21(1046)	1052	1054.17	32.29	1117	60
LA24(935)	938	942.50	30.91	1009	60
LA25(977)	977	981.37	37.60	996	60
LA27(1235)	1240	1252.47	86.89	1349	90
LA29(1153)	1164	1175.07	82.86	1242	90
LA38(1196)	1202	1210.10	55.36	1295	60
LA40(1222)	1230	1239.70	60.82	1283	90

6 Concluding Remarks

We have proposed a GA that combines decoding into the space of active schedules and local search in the space of semi-active schedules. The reported experimental results demonstrated that GA is better than GA1, which exploits semi-active scheduling for both decoding and local search. Furthermore the GA is quite competitive (even better in average) with GA3, which is one of the most outstanding methods in solving JSS problems. We have also experimented with decoding with a semi-active schedule builder. In this case, if the same parameters were used, the GA reached better solutions, but the computation time were much larger as well. This is due to the local search (either N_3 or N_4) has more chances for improving. However, in the same amount of time, the GA with active scheduling reaches better results that the GA with semi-active scheduling. Moreover we have compared GA with the BB procedure proposed by Brucker et

al. This algorithm is the best exact approach proposed so far and is able to solve to optimality many instances up to a size of 15×15. In this case GA clearly outperforms BB in solving the selected instances. We have done experiments (not reported here) with much larger problem instances and the results of GA were even much better. Hence the proposed GA is quite competitive with the most efficient methods for solving JSS problems.

References

1. Bierwirth, C.: A Generalized Permutation Approach to Jobshop Scheduling with Genetic Algorithms. OR Spectrum **17** (1995) 87–92
2. Bierwirth, C., Mattfeld, D.C.: Production scheduling and rescheduling with genetic algoritms. Evolutionary Computation **7** (1999) 1–17
3. Brucker, P., Jurisch, B., Sievers, B.: A branch and bound algorithm for the job-shop scheduling problem Discrete Applied Mathematics **49** (1994) 107–127
4. Brucker, P.: Scheduling Algorithms. 4th edn. Springer (2004)
5. Dell' Amico, M., Trubian, M.: Applying Tabu Search to the Job-shop Scheduling Problem. Annals of Operational Research **41** (1993) 231–252
6. Giffler, B., Thomson, G.L.: Algorithms for solving production scheduling problems. Operations Research **8** (1960) 487–503
7. Jain, A. S. and Meeran, S.: Deterministic job-shop scheduling: Past, present and future. uropean Journal of Operational Research **113** (1999) 390–434
8. Mattfeld, D. C. : Evolutionary Search and the Job Shop. Investigations on Genetic Algorithms for Production Scheduling. Springer-Verlag (1995)
9. Nowicki, E., Smutnicki, C.: A Fast Taboo Search Algorithm for the Job Shop Scheduling Problem. Management Science, **42(6)** (1996) 797–813
10. Taillard, E. D.: Parallel Taboo Search Techniques for the Job Shop Scheduling Problem. ORSA Jorunal on Computing, **6** (1993) 108–117
11. Vaessens, R. J. M, Aarts, E. H. L., Lenstra, J. K.: A Local Search Template. In Mnner and Manderick (eds.)(1992) 65-74.
12. Van Laarhoven, P. J. M., Aarts, E. H. L., Lenstra, J. K.: Job Shop Scheduling by Simulated Annealing. ORSA Journal on Computing, **40** (1992) 113–125
13. Varela, R., Vela, C.R., Puente, J., Gómez, A.: A knowledge-based evolutionary strategy for scheduling problems with bottlenecks. European Journal of Operational Research **145** (2003) 57–71
14. Varela, R., Serrano, D., Sierra, M.: New Codification Schemas for Scheduling with Genetic Algorithms. LNCS 3562, Springer-Verlag **145** (2003) 11–20
15. Yamada, T. and R. Nakano.: Scheduling by Genetic Local Search with multi-step crossover. Fourth Int. Conf. On Parallel Problem Solving from Nature (PPSN IV), Berlin, Germany (1996) 960-969.

Chaos Detection with Lyapunov Exponents in Dynamical System Generated by Evolutionary Process

Iwona Karcz-Dulęba

Inst. of Computer Engineering, Control and Robotics, Wroclaw Univ. of Technology, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
iwona.duleba@pwr.wroc.pl

Abstract. The model of phenotypic evolution is considered where a population is ruled by proportional selection and normally distributed mutation. Expected values of the population state generate a discrete dynamical system. The system displays various asymptotic behavior depending on a fitness function and a mutation parameter. Stable fixed points, period-doubling bifurcations and chaos are observed. Lyapunov exponents are used to detect chaos in the system for some fitness functions.

1 Introduction

Optimization methods based on natural evolution rules (evolutionary algorithms) are very popular recently. Despite of usefulness, their theoretical foundations are frequently very weak. The methods are indeterministic, often based on heuristics, so proving their properties is very cumbersome. Many approaches to the algorithms' analysis can be found in the literature. Vose [11] proposed dynamical system models of genetic algorithms and showed the expected behavior of the algorithms for infinite populations. For phenotypic evolution the dynamical system model was developed in [1,3]. The model was successfully applied to study very small (two-elements) populations evolving in one-dimensional real-valued infinite search space [4,5,6,7,8]. The reproduction process is ruled by the proportional selection and normally distributed mutation. Usually, populations are regarded as assembles of individuals in the space of individual traits. In this model, populations are considered in a space where every point describes a state of the whole population. When regarding populations in the space of population states, expected values of the states generated a dynamical system. An asymptotic behavior of the system was studied for some fitness functions. Fixed points of the system can be stable or unstable, periodic orbits and chaotic behavior were observed in simulations as well. In the reported research Lyapunov exponents are applied to detect chaos in the system.

Although there are many approaches to detect chaos (time series examination, an auto-correlation function, phase portraits, the Fourier series), the exponents seem to be the most informative as they provide not only qualitative but also quantitative insight into chaotic phenomena.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 380–389, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Discrete Dynamical System

The model of phenotypic asexual evolution with the m-element population evolving in n-dimensional continuous real search space was introduced in [2]. Each individual $\boldsymbol{x}$ is described by the n-dimensional vector of traits and quality index $q(\boldsymbol{x})$. Reproduction is composed of two phases: proportional selection followed by normally distributed mutation (determined by the standard deviation of mutation parameter σ). The population is considered in a space of traits $T = R^n$, where each point represents an individual.

Alternatively, the population can be regarded in *the space of population states* S. In this space every point describes the whole population, thus $dim(S) = m \cdot n$. As evolution of the population should not depend on ordering of individuals within the population, the equivalence relation U is defined to make states independent on permutations of individuals. The space S equipped with the equivalence relation is reduced to the factor space S_U:

$$U: \quad S \to S_U = R^{n \cdot m}/U \in R^{m \cdot n}. \tag{1}$$

We concentrate on the simplest case of evolution when two-element populations ($m = 2$) evolve in one-dimensional search space ($n = 1$). The space of population states is easy to visualize in this case. The equivalence relation sets individuals x_1 and x_2 in decreasing order, thus the state space $S_U \in R^2$ is identified with the half-plane $S_U = \{(x_1, x_2) : x_1 \geq x_2\}$, bounded by the line $x_1 = x_2$ called the identity axis. A rotation of the coordinate frame X_1X_2 facilitates an analysis of the process. Counterclockwise rotation around the axis perpendicular to the plane X_1X_2 (and passing through the origin of the frame) by the angle $\pi/4$ transforms coordinates as follows $w = (x_1 - x_2)/\sqrt{2}$, $z = (x_1 + x_2)/\sqrt{2}$. The rotation converts space S_U into right half-plane, i.e. $\boldsymbol{s} = (R^+ \ni w, z \in R)$. The coordinate w is a measure of populations diversity and describes a distance of the population state from the identity axis. The coordinate z situates a state along the identity axis.

In papers [3]-[8] we argued that instead of modeling evolution of the population it is instructive and much easier to trace its average (expected) behavior preserving its qualitative and quantitative features. Moreover, expected values of position coordinates (w, z) of the population state $E(w|\boldsymbol{s})$, $E(z|\boldsymbol{s})$ in the $(i+1)$st generation can be calculated analytically based on the current state $\boldsymbol{s}$ and parameters of the evolution (the standard deviation of mutation and parameters of the fitness function used). In the space S_U the expected values generate two dimensional discrete dynamical system

$$\boldsymbol{s}_{i+1} = \boldsymbol{F}(\boldsymbol{s}_i) = \begin{pmatrix} F_1(\boldsymbol{s}_i) = E_{i+1}(w|\boldsymbol{s}_i) \\ F_2(\boldsymbol{s}_i) = E_{i+1}(z|\boldsymbol{s}_i) \end{pmatrix}, \qquad \boldsymbol{s}_i = \begin{pmatrix} w_i \\ z_i \end{pmatrix}, \tag{2}$$

described by the equations

$$\begin{cases} w_{i+1} = & \sqrt{\frac{2}{\pi}}\sigma + (1 - \Psi_i{}^2) \cdot \sigma \cdot \theta(\frac{w_i}{\sigma}) \\ z_{i+1} = & z_i + \Psi_i \cdot w_i \end{cases} \tag{3}$$

where

$$q_1 = q(\frac{w+z}{\sqrt{2}}),\ \ q_2 = q(\frac{z-w}{\sqrt{2}}),\ \ \Psi(w,z) = \frac{q_1 - q_2}{q_1 + q_2},\ \ \Psi_i = \Psi(w_i, z_i),$$

$$\phi_0(\xi) = \frac{1}{\sqrt{2\pi}}(\exp\left(-\xi^2/2\right) - 1),\ \ \Phi_0(\xi) = \frac{1}{\sqrt{2\pi}} \int_0^{\xi} \exp(-t^2/2)dt,$$

$$\theta(\xi) = \phi_0(\xi) + \xi \cdot \Phi_0(\xi).$$

3 Asymptotic Behavior of the Dynamical System

The asymptotic behavior of dynamical system (3) was studied for a large class of uni- and multimodal, symmetrical and asymmetrical fitness functions [4,5,6,7,8]. Fixed points of the system (3), derived from conditions $\boldsymbol{F}(\boldsymbol{s}) = \boldsymbol{s}$, are characterized by equations

$$w^s \cong 0.97 \cdot \sigma, \tag{4}$$

$$\Psi(w^s, z^s) = 0. \tag{5}$$

The w-coordinate of fixed points (4) depends only on the standard deviation of mutation. The z-coordinate (5) depends on a fitness function and it satisfies the condition $q((z^s + w^s)/\sqrt{2}) = q((z^s - w^s)/\sqrt{2})$. The number of fixed points is determined by the fitness modality and by the standard deviation of mutation. For uni-modal fitness functions, dynamical system (3) has one fixed point. For symmetrical functions, the fixed point is located on the symmetry axis. The asymmetry in fitness function influences only the value of the z-coordinate of the fixed point. For a fitness function with k optima the system has odd number of fixed points, no more than $2k + 1$. The points are located in the vicinity of optima and saddles of the fitness. The number of fixed points decreases as the value of σ is increased.

The linear approximation matrix for dynamical system (3) at fixed points (w^s, z^s) is diagonal and its eigenvalues are equal to

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \Phi_0(w^s) \\ w^s \cdot \frac{\partial \Psi(w,z)}{\partial z}|_{(w^s,z^s)} + 1 \end{pmatrix}. \tag{6}$$

Because $|\lambda_1| < 1$, the fixed point stability depends on the second eigenvalue, which is determined by the fitness function (via the function Ψ) and the standard deviation of mutation (via w^s). Fixed points located near saddles of the fitness are always unstable [8]. Optima fixed points are usually stable for small values of the parameter σ and lose their stability as σ is increased. For large σ the fixed point becomes unstable and a period-doubling (pitchfork) bifurcation occurs, giving rise to a stable orbit of period 2. For asymmetrical fitness functions series of period-doubling bifurcations leading to chaos were discovered (Fig. 1.1b,1c). The effect of period-doubling bifurcations depends on degree of asymmetry and manifests itself clearly for functions with distinct asymmetry.

4 Chaos Detection with Lyapunov Exponents

The period-doubling bifurcations and chaos were revealed in numerical experiments. In the simulations the system was iterated many times until stable/unstable behavior was observed. It is not a very accurate method of detecting chaos in a dynamical system. More reliable method of discovering chaos is to compute Lyapunov exponents [9,10]. Lyapunov exponents are used to verify sensitivity of trajectories on varying slightly initial conditions. In fact the maximal, dominant Lyapunov exponent is of primary importance as it determines the worst behavior of the system. Although Lyapunov exponents are not the only tool used to check for chaos (auto-correlating parts of a trajectory, checking spectrum of energy in the signal, and verifying behavior of the Poincare map around an orbit are the other features to check [10]), they have a good theoretical and analytical background. Note that the dominant Lyapunov exponent exceeding zero does not necessary mean that chaos is detected. However, in typical situations it is really the case. The dominant Lyapunov exponent is computed from the formula

$$\lambda = \frac{1}{N} \ln \mu_{\max}(D\boldsymbol{F}^N(\boldsymbol{s}_0)) \tag{7}$$

where $\mu_{\max}(A)$ is the maximal absolute value among eigenvalues of the matrix A and $D\boldsymbol{F}^N$ is the accumulated, in N steps, the Jacobi matrix of the map $\boldsymbol{F}$

$$D\boldsymbol{F}^N(\boldsymbol{s}_0) = D\boldsymbol{F}|_{\boldsymbol{F}^{N-1}(\boldsymbol{s}_0)} \cdot D\boldsymbol{F}|_{\boldsymbol{F}^{N-2}(\boldsymbol{s}_0)} \cdot \ldots \cdot D\boldsymbol{F}(\boldsymbol{s}_0) = \prod_{i=1}^{N} D\boldsymbol{F}(F^{i-1}(\boldsymbol{s}_0)), \tag{8}$$

where $\boldsymbol{s}_0$ is an initial state, $\boldsymbol{F}^0(\boldsymbol{s}) \equiv \boldsymbol{s}$ and

$$D\boldsymbol{F} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{s}} = \begin{pmatrix} \frac{\partial \boldsymbol{F}_1}{\partial w} & \frac{\partial \boldsymbol{F}_1}{\partial z} \\ \frac{\partial \boldsymbol{F}_2}{\partial w} & \frac{\partial \boldsymbol{F}_2}{\partial z} \end{pmatrix}. \tag{9}$$

The Lyapunov exponent λ, is useful to differ various types of orbits. For $\lambda < 0$ a fixed point or a periodic orbit is stable. Such systems exhibit asymptotic stability, the more negative the exponent, the wider stability margin. For $\lambda = 0$ a fixed point is neutral. The system is in some sort of steady state mode. Such systems display Lyapunov stability. For $\lambda > 0$ a fixed point is likely unstable and chaotic.

Lyapunov exponents were computed for system (3) and two uni-modal and one bimodal fitness functions. The uni-modal functions are represented by the family of tent functions given by

$$q(x) = \begin{cases} x/A + 1 & \text{for } x \in [-A, 0), \quad A > 0 \\ -x/B + 1 & \text{for } x \in [0, B], \quad B \geq A \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

and Gaussian functions

$$q(x) = \begin{cases} \exp(-a_1 x^2) & \text{for } x \leq 0 \\ \exp(-a_2 x^2) & \text{for } x > 0. \end{cases} \tag{11}$$

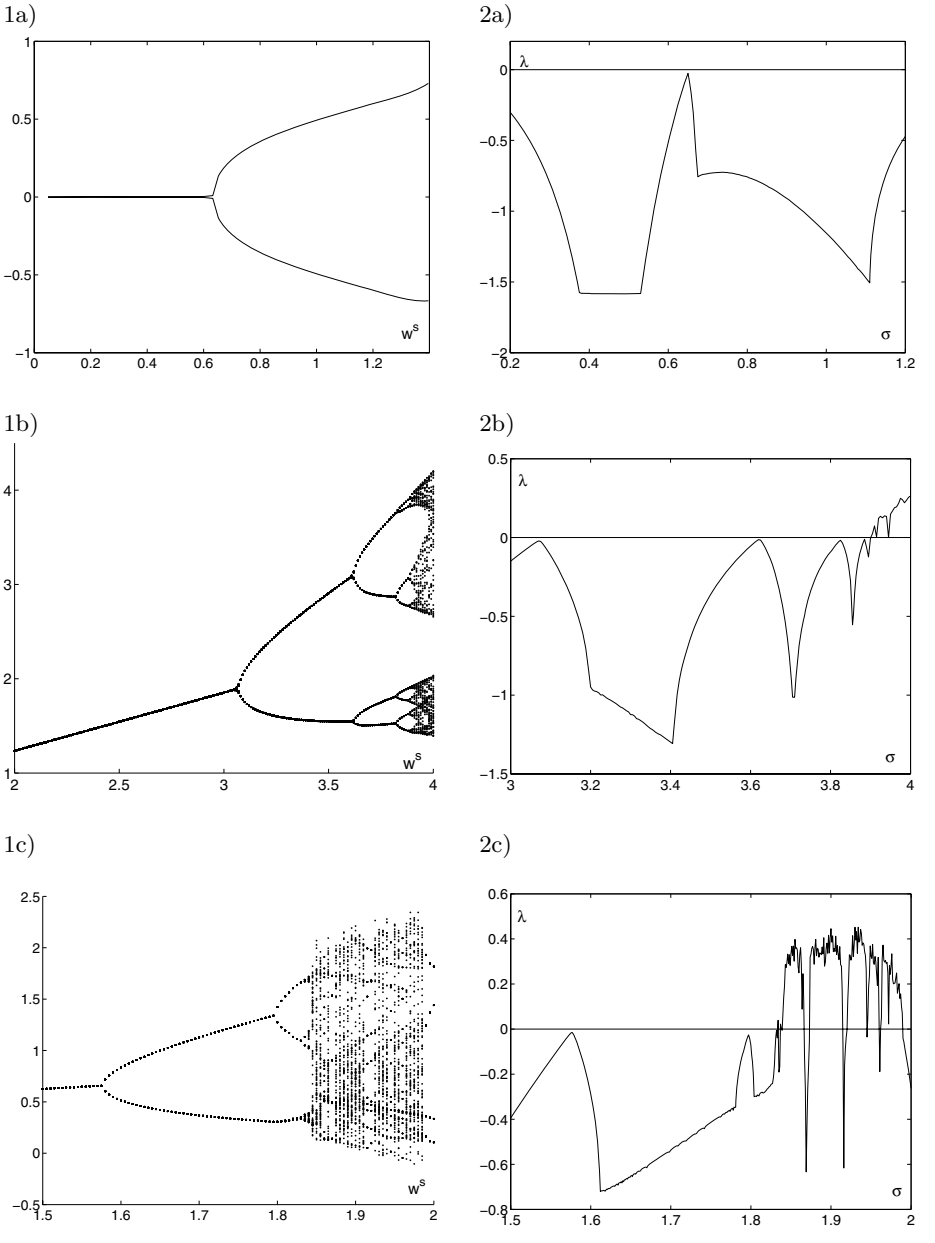


Fig. 1. Bifurcation diagram (1) and the dominant Lyapunov exponent (2) as the function of σ for: a) the symmetrical Gaussian function, $a_1 = a_2 = 5$; b) the asymmetrical Gaussian function, $a_1 = 1$, $a_2 = 0.05$; c) the asymmetrical tent function, $A = 1$, $B = 2.5$

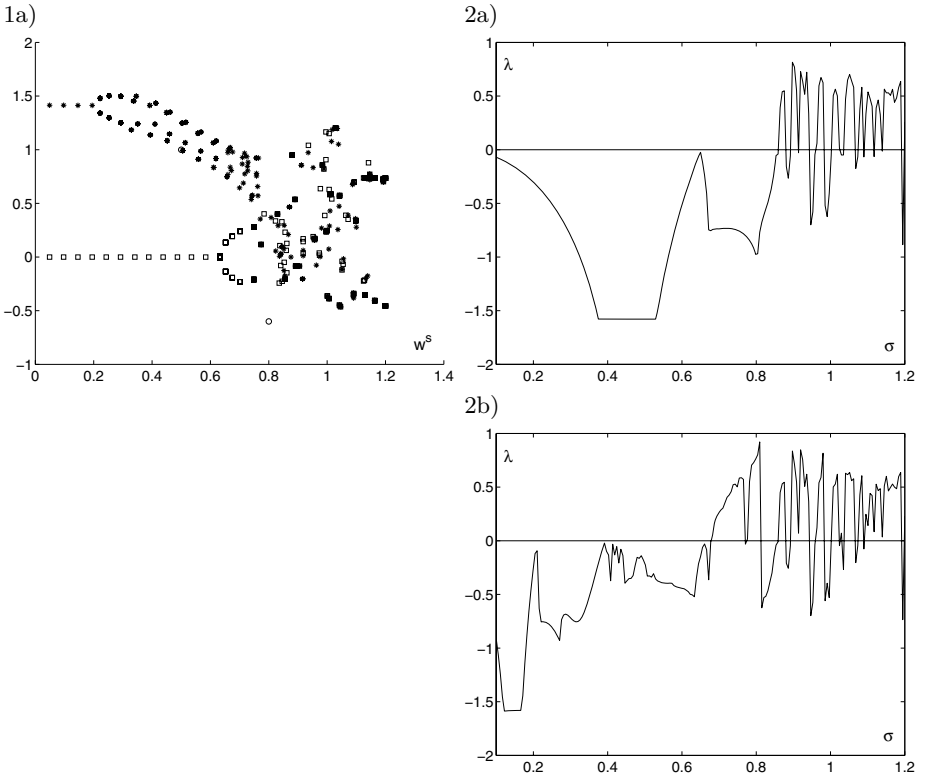


Fig. 2. Bifurcation diagram (1) and the dominant Lyapunov exponent (2) as the function of σ for the bimodal Gaussian function (12) ($a_1 = 5$, $a_2 = 50$, $h = 2$), and two initial states a) $(0.8, -0.6)$, b) $(0.5, 1)$. In the panel 1a) squares/stars come from the first/second initial state, marked with circles.

Depending on values of parameters (A, B or a_1, a_2) the fitness functions can be symmetrical or asymmetrical. The only fixed point of system (3) becomes unstable for large values of σ and the orbit of period 2 appeared (Fig. 1.1a). The period-doubling bifurcations and chaotic behavior is discovered for the functions with distinct asymmetry ($B \geq 1.5A$, $a_2 < 0.1a_1$) (Fig. 1.1b,1c) [6].

The sum of two Gaussian functions exemplifies bimodal fitness

$$q(x) = \exp(-a_1 x^2) + h \cdot \exp(-a_2(x-1)^2). \tag{12}$$

For $h > 1$ the second Gaussian peak sets the global optimum. For equi-width peaks ($a_1 = a_2$), the dynamical system has got two stable optima fixed points and one unstable saddle fixed point for small values of σ. As the standard deviation of mutation increases, the local optimum fixed point vanishes and only global ones remains. For large σ the periodic orbit is observed. When $a_2 > a_1$, the attraction region of the global optimum is smaller than the region of the local

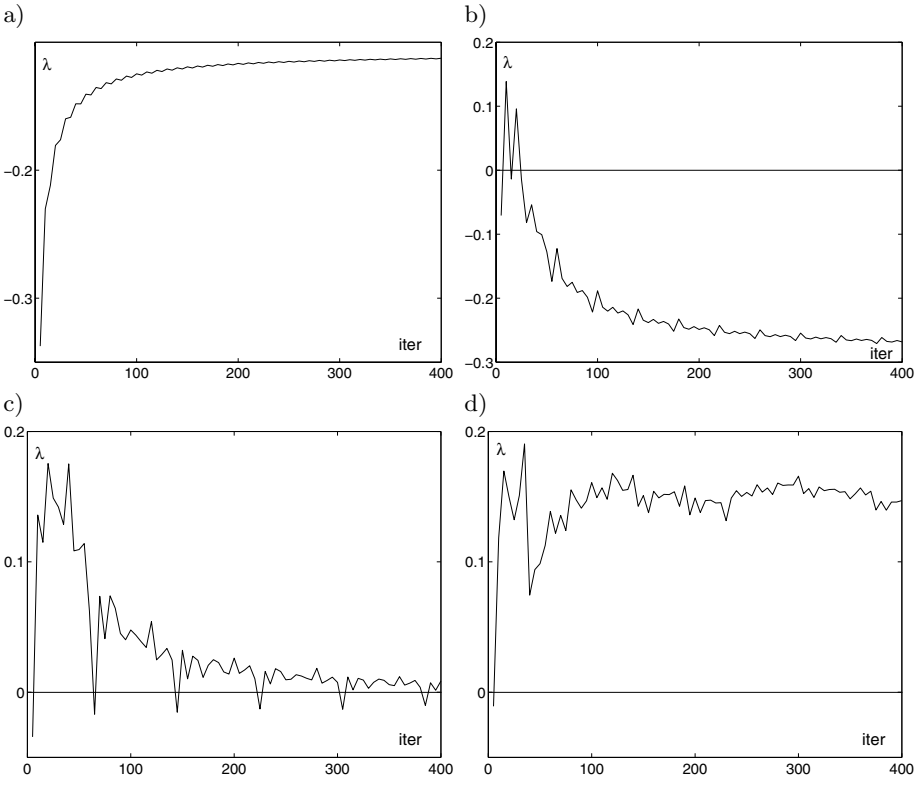


Fig. 3. The dominant Lyapunov exponent λ as the function of the number of iterations ($iter = N$), initial state $(0.6, 1.0)$, fitness (11) with $a_1 = 1$, $a_2 = 0.05$, and varied mutation parameter: a) $\sigma = 3.1$, b) $\sigma = 3.85$, c) $\sigma = 3.9$, d) $\sigma = 3.95$

optimum, as the global peak is narrower. The asymmetry in hills width results in chaotic behavior of dynamical system (3) (Fig. 2.1a) [8].

For the system (3), the bifurcation diagram and the dominant exponent as the function of the standard deviation of mutation are provided in Figs. 1 (unimodal fitness (10), (11)) and in Figs. 2 (bimodal fitness (12)). For bimodal fitness, diagrams of the Lyapunov exponents are presented (Fig. 2.2a,2b) for two initial states taken from the region of attraction of the local, $\boldsymbol{s}_0 = (0.8, -0.6)$, and the global optimum, $\boldsymbol{s}_0 = (0.5, 1.0)$. 200 iterations were performed to draw each point on the Lyapunov exponent characteristics. Comparing left and right panels in Figs. 1 and 2 one can observe that bifurcations rising periodic orbits are easy to detect when the Lyapunov exponent characteristics touch the horizontal axis ($\lambda = 0$). Chaotic behavior, clearly visualized on the bifurcation diagrams, manifests itself when the line $\lambda = 0$ is crossed upwards. Although the Lyapunov exponents are not fast to compute, they can be computed much faster than data required to draw bifurcation diagrams.

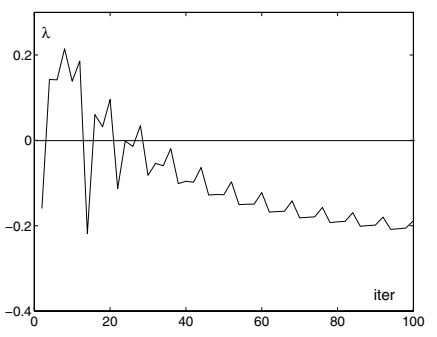


Fig. 4. The dominant Lyapunov exponent λ as the function of the number of iterations ($iter = N$), initial state $(0.6, 1.0)$, fitness (11) with parameters $a_1 = 1, a_2 = 0.05, \sigma = 3.85$

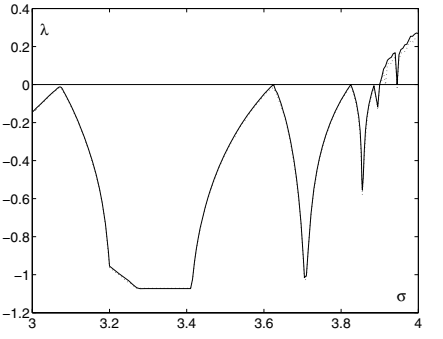


Fig. 5. The dominant Lyapunov exponent as the function of mutation parameter σ. 10 initial states taken randomly from the rectangle $w \in [0.1, 1]$, $z \in [-1, 1]$ for the fitness (11) with $a_1 = 1, a_2 = 0.05$. 100 iterations performed. Solid/dotted line - maximal/minimal value of the dominant Lyapunov exponent over all initial states.

It is instructive to analyze the impact of various parameters on the dominant Lyapunov exponent. Simulations were performed for uni-modal Gaussian function (11) ($a_1 = 1$, $a_2 = 0.05$). In Fig. 3 the dominant Lyapunov exponent is presented as the function of the numbers of iterations (N) to compute λ for different values of standard deviation of mutation σ. The values correspond to: the stable fixed point $\sigma = 3.1$, orbits of different periods $\sigma = 3.85$, $\sigma = 3.9$ and chaos $\sigma = 3.95$. For small numbers of iterations the dominant Lyapunov exponent varies significantly. A few hundreds of iterations are usually necessary to draw reliable conclusions. However, to determine orbits of large periods more iterations are required. In the cases characteristics $\lambda(N)$ very slowly and with small amplitude jumps tends to cross the line $\lambda = 0$. The more detailed diagram for small number iterations ($\sigma = 3.85$) is presented in Fig 4 which displays also periodic phenomena. For most of σ values, the exponent λ stabilizes itself after about 200 iterations. The influence of initial states on λ is depicted in Fig 5. 10 initial states were randomly chosen from the rectangle $w \in [0.1, 1]$, $z \in [-1, 1]$ and for each the dominant exponents was computed. The difference between minimal (dotted line) and maximal (solid line) values of the dominant Lyapunov exponent over all initial states displayed in Fig 5 reveals that initial states practically do not impact λ. Theoretically the dominant Lyapunov exponent should be computed for a large number of iterations. In this case the transient interval (and initial conditions) should not impact the dominant exponent. In practical situations, the number of iterations is restricted. In Fig 6 plots $\lambda(\sigma)$ are presented for two values of the number of iterations $N = 200, 400$. In order to check whether the transit interval is important, the characteristics are presented in

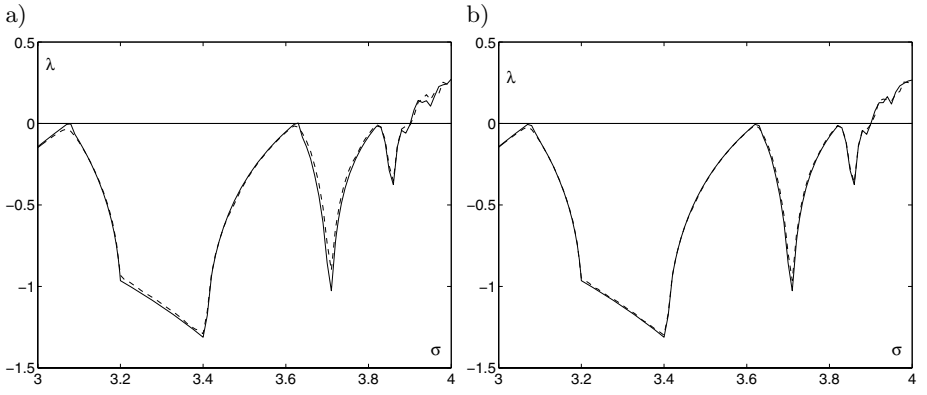


Fig. 6. The dominant Lyapunov exponent as the function of mutation parameter σ; the initial state $(0.5, 1.0)$, fitness (11) with parameters $a_1 = 1$, $a_2 = 0.05$, and the number of iterations to compute the exponent a) 200 b) 400, solid/dotted line - with/without omitting first 200 iterations

two variants with and without considering the first 200 iterations. It appears that the transient interval does not impact the dominant Lyapunov exponent significantly as well as omitting the very first few hundreds of iterations.

5 Conclusions

Criteria used to detect chaos in dynamical systems are not decisive and checking any of them is not sufficient to prove chaotic behavior of the systems. In this research Lyapunov exponents were used to detect chaos for the discrete dynamical system generated by phenotypic evolutionary process. Presented simulations confirmed that for asymmetrical fitness functions and large values of standard deviation of mutation parametr σ, chaos appeared easily. Also bifurcation values of σ were easy to detect. Generally, if the number of iterations was sufficient enough (in the simulations a few hundreds), the accuracy of computing the dominant Lyapunov exponent did not depend on parameters (initial states, the number of iterations to compute the exponent, omitting some first iterations).

References

1. Chorążyczewski, A., Galar, R., Karcz-Dulęba, I.: Considering Phenotypic Evolution in the Space of Population States. *Proc. 5th Conf. Neural Networks and Soft Computing*, (L.Rutkowski, R.Tadeusiewicz, Eds.), Zakopane, Poland, (2000) 615-620
2. Galar, R.: Handicapped individua in evolutionary processes. *Biol.Cybern.*, **51**, (1985) 1-9
3. Galar, R., Karcz-Dulęba, I.: The evolution of two: An Example of Space of States Approach. *Proc. 3rd Annual Conf. on Evolutionary Programming*, (A.V. Sebald, L.J. Fogel, Eds), San Diego CA, World Scientific, (1994) 261–268

4. Karcz-Dulęba, I.: Evolution of Two-element Population in the Space of Population States: Equilibrium States for Asymmetrical Fitness Functions. in: *Evolutionary Algorithms and Global Optimization*, (J.Arabas Ed.), Warsaw University of Technology Press, Warsaw, (2002) 35–46
5. Karcz-Dulęba, I.: Asymptotic Behavior of Discrete Dynamical System Generated by Simple Evolutionary Process, *Journ. Applied Math. & Comp. Sc.*, **14**(1) (2004) 79–90
6. Karcz-Dulęba, I.: Period-doubling bifurcations in discrete dynamical system generated by evolutionary process. *Proc. 7 Nat. Conf. Evol. Comp. and Global Opt.*, Kazimierz Dolny, Poland, (2004) 83–88
7. Karcz-Dulęba, I.: Bifurcations and chaos in phenotypic evolution for unimodal fitness functions, in: *Control Systems*, Greblicki, Smutnicki Eds., WKiL, 41-50, 2005.
8. Karcz-Dulęba, I.: Dynamics of two-element populations in the space of population states, *IEEE Evol. Comp.*, **10**(2) (2006) 199–209
9. Kudrewicz, J.: Fractals and chaos, WNT, Warsaw, (1993) in Polish
10. Schuster, H.G.: Deterministic chaos. VCH Verlagsgesellschaft (1988)
11. Vose, M.D.: The Simple Genetic Algorithm. Foundations and Theory, MIT Press (1999)

Improving Evolutionary Multi-objective Optimization Using Genders

Zdzislaw Kowalczuk and Tomasz Bialaszewski

Faculty of Electronics, Telecommunication and Computer Science,
Gdansk University of Technology, 80-952 Gdansk, Poland
kova@pg.gda.pl, bialas@eti.pg.gda.pl

Abstract. In solving highly dimensional multi-objective optimization (EMO) problems by evolutionary computations the concept of Pareto-domination appears to be not effective. The paper discusses a new approach to EMO by introducing a concept of genetic genders for the purpose of making distinction between different groups of objectives. This approach is also able to keep diversity among the Pareto-optimal solutions produced.

1 Introduction

Evolutionary computation EC [4], [5], [6], [20], [19], based on emulation of the evolution of biological systems, has turned up to be one of the most successful means of solving optimization problems [2], [6], [9], [13], [17] including those characterized by the possible discontinuity or multi-modality of partial objective functions.

ECs are most valued in solving difficult multi-objective optimization problems, where several objectives globally optimized at the same time [5], [17], [19], [20], [24]. Such tasks are though difficult to perform, as the notion of optimality is not obvious. In assessing the merit of solutions [5], [14], [20], the concept of optimality in the Pareto sense is usually applied. Its non-uniqueness can be straightforwardly utilized during EC runs. At the end, however, this idea of optimality does not give any hints as to the choice of a single solution from amongst the final Pareto-optimal solutions. The problem is even more complicated when the objectives are defined in a highly dimensional vector space.

In this paper we present a genetic-gender approach GGA [11], [12], [13] to solving multi-objective optimization (EMO) problems by evolutionary search. The concept is based on the information about a degree of membership to a given gender [11], [12], which is attributed to each solution being evaluated. This information is appropriately utilized in the process of parental crossover, in which only individuals of different genders are allowed to create their offspring.

As compared to other methods our approach is fairly simple, sticks to the very basics of the GA/EC methodology, more profoundly utilizes hints from the nature, fulfills the requirements of technical designs, and is open to other amendments and developments proposed in the literature.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 390–399, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Multi-objective Optimization Problem

The issue of simultaneous optimization of several objective functions is most critical in solving theoretical and practical tasks [2], [14], [17], where the designer has to make a choice from amongst equivalent 'optimal' solutions [16].

Consider a general multi-objective optimization problem formulated as a multi-profit maximization task without constrains:

$$\max_{\mathbf{x}} \mathbf{f}(\mathbf{x}), \tag{1}$$

$$\mathbf{x} = \left[x_1 \; x_2 \; \ldots \; x_n \right]^T \in \Re^n, \quad n \in \aleph, \tag{2}$$

$$\mathbf{f}(\mathbf{x}) = \left[f_1(\mathbf{x}) \; f_2(\mathbf{x}) \; \ldots \; f_m(\mathbf{x}) \right]^T \in \Re^m, \quad m \in \aleph. \tag{3}$$

To integrate those objectives, it is necessary to define the relations (or weights) between the partial objectives considered. For the purpose of solving such optimality problems, various methods have been proposed, including weighted profits [20], distance functions [20], sequential inequalities [23], or ranking with reference to Pareto-optimality [4], [5], [10], [11], [12], [13], [14], [17], [19], [22].

The substance of the first three methods lies in direct integration of many objectives into one submitted to optimization by using an arbitrary choice of the weighting vector, demand vector or limit values for the objective functions. Such choices are not always straightforward. They also definitely restrict and simplify the multi-objective optimization problem. In contrast to the above, the ranking method using the notion of Pareto-optimality avoids the arbitrary weighting of the objectives. Instead, a useful classification of the solutions is applied that takes into account particular objectives more objectively. Namely, with maximization tasks in mind, the solutions can be classified as dominated and non-dominated (Pareto-optimal). According to such an assessment, each individual is assigned a scalar rank representing its degree of domination. Thus, in practice, in assessing the merit of its solutions the concept of optimality in the Pareto sense and the related ranking method [5], [14], [16], [20] are usually most suitable.

Global Optimality. The Pareto-based ranking does not, however, give any directions as to the choice of a single solution from amongst the Pareto-optimal solutions found. In order to utilize that freedom, a development of the ranking methodology was proposed [10], [14], where the profit vector of each solution is mapped to a scalar global optimality level GOL, which equals a minimal co-ordinate of the vector. This approach permits a practical elimination of the issue of the ambiguity of final Pareto-solutions. What is important, problematic P-optimal solutions of high dimensional spaces which maximize only some criteria can then be easily ruled out.

3 The Genetic-Gender Approach

While considering multi-objective optimization problems one always has to be aware of the issue of dimensionality. It is well known that when the space of the objectives has a higher dimension many individuals fall within the category of being P-optimal, *i.e.*mutually equivalent in the Pareto sense. The same rank implies that they are indistinguishable from the P-optimality viewpoint. An outer effect of such a state of the Pareto assessment discussed is a low number of Pareto-fronts that obstructs distinction, estimation and ordering of solutions. This, in particular, means that during the process of selecting individuals to new generations the Pareto-based ranking is ineffective leading to a stochastic search with no 'conscious' progress.

On the other hand, when one confines the scope of optimality by reducing the dimension of the analyzed objectives space, the ability of the Pareto optimality method to differentiate between different individuals is facilitated [11], [12].

Another motivation is the fact that, practically, there have been only direct estimates of the fitness functions applied in the reproduction process of the GA algorithms. This means using solely the concept of one 'unisex' parental pool, unlike within the species met in the earth nature.

3.1 Recognition of the Genetic Gender

In nature the gender division of a species appears to differentiate individuals with reference to reproductive functions. According to this division, our concept of an artificial genetic gender (GG) consists in dividing the objective functions into several subsets, each of which has an attributed genetic gender Xj ($j = 1, 2, ..., s$) and portrays a partial-scope optimality value (a certain utility interpretation for the designer).

In this way, one gender set (Xj) can be constituted by objectives of a 'similar' character that are in a kind of internal/secondary rivalry (in terms of an equal meaning to the user from his point of view). Such an assortment can effectively discharge the designer from the task of the final isolation of a single solution from amongst all the Pareto-suboptimal ones obtained in the course of multi-objective optimization.

On the other hand, different gender sets can express various groups of 'interests' that are difficult to be judged by the user in advance. In general, this division can be employed to represent an external/primary rivalry, which is not simple to be resolved (in such cases the best method can be to use the notion of Pareto-optimality).

Thus in our consequent approach, we propose using the mechanism of the gender allotment during the whole computational evolution for the purpose of creating several parental pools of different genders and generating new offspring by mating only apparently dissimilar individuals.

The vector of the profit functions (3) can therefore be divided into s-subvectors

$$\mathbf{f}(\mathbf{x}) = \left[\mathbf{f}_1(\mathbf{x})\ \mathbf{f}_2(\mathbf{x})\ \ldots\ \mathbf{f}_s(\mathbf{x})\right]^T \in \Re^m, \tag{4}$$

$$\mathbf{f}_j(\mathbf{x})^T \in \Re^{m_i}, \quad m = \sum_{i=1}^{s} m_i. \tag{5}$$

Each j-th subvector ($j = 1, 2, ..., s$) specifies the GG set of individuals labeled by Xj. Within each of these sets, P-suboptimality-based ranking of individuals is applied. In effect, each of the individuals is allotted a vector of ranks

$$\mathbf{r}(\mathbf{x}) = \left[r_1(\mathbf{x})\ r_2(\mathbf{x})\ \ldots\ r_s(\mathbf{x})\right]^T \in \Re^s. \tag{6}$$

According to the proposed GG approach (GGA), a genetic gender l_i is assigned to each individual $\mathbf{x}_i$ in the population by computing the following:

$$r_{j_{\max}} = \max_{i=1,2,\ldots,N} \{r_j(\mathbf{x}_i)\}, \tag{7}$$

$$\varphi_i^j = \frac{r_j(\mathbf{x}_i)}{r_{j_{\max}}}, \tag{8}$$

$$l_i = \arg \max_{j=1,2,\ldots,s} \varphi_i^j, \tag{9}$$

$$\varphi_i = \max_{j=1,2,\ldots,s} \varphi_i^j. \tag{10}$$

where φ_i is the obtained highest degree of suboptimality, meaning a (fuzzy) measure of the memberships of the i-th individual to the l_i-th variant of the genetic gender, while the symbol $r_{j_{\max}}$ denotes the maximum rank from amongst all individuals with respect to the j-th subcriterion (Xj).

The method of selecting parental pools is carried out according to the stochastic -remainder method [5], [14], [17] based on the highest degrees of the membership to the gender set considered. The population of each gender-set is monitored in terms of an assumed minimal number of members ($N/(3s)$, for instance). The lacking positions can be filled up with individuals from the lowest Pareto front of another gender set which have been waived in the course of the GG selection process.

It is thus clear that introducing the GG approach we alleviate the issue of dimensionality by atomizing the scope of optimality and restricting the dimension of the spaces being considered. We can easily anticipate that this manipulation will produce a greater number of Pareto fronts leading to a measurable diversity of the generated subpopulations and to an improved effectiveness of the genetic search into the 'directions' of both the partial and full scope objectives.

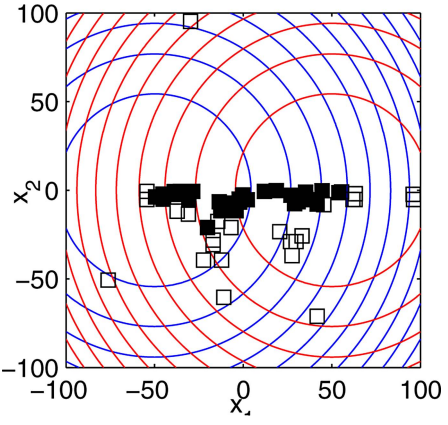


Fig. 1. GA search: Optimal solutions found in the space of the sought objectives

4 Illustrative Example

Let us now consider an abstract example of a two-dimensional vector of criteria:

$$\min_x \mathbf{f}(\mathbf{x}) = \min_x \left[f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \right], \tag{11}$$

$$f_1(x_1, x_2) = (x_1 + 50)^2 + x_2^2, \tag{12}$$

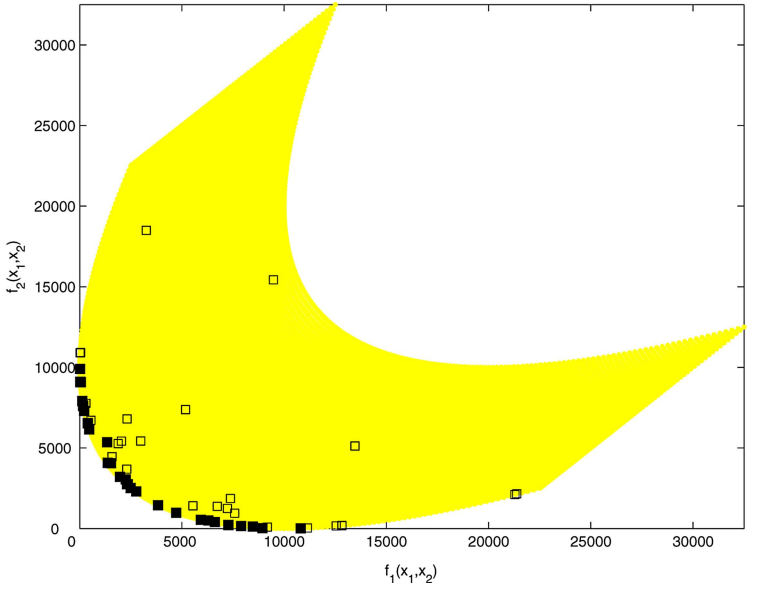


Fig. 2. Full-scope GA search: Optimal solutions against the co-domain of available two-dimensional objectives

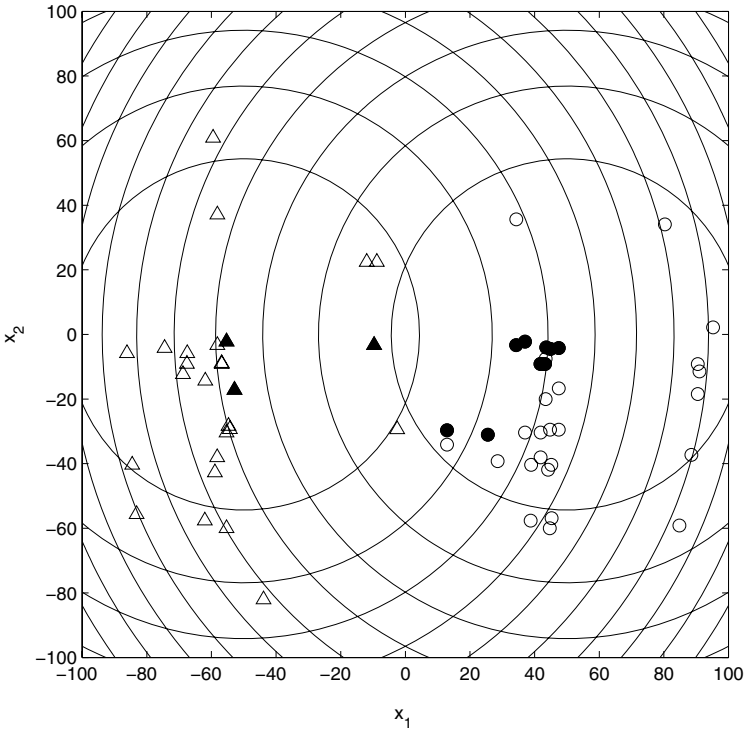


Fig. 3. Gender GGA search: Optimal solutions in their domain (the searched space)

$$f_2(x_1, x_2) = (x_1 - 50)^2 + x_2^2. \tag{13}$$

and $\mathbf{x} \in \Re^2$ is the optimized vector. The structure of this criterion allows a natural division into two singular GG sets, representing the two objectives of (11).

The results of a classical evolutionary quest for solutions minimizing the full-scope two-objective function are illustrated in Figs. 1 and 2. Fig. 1 portrays the objective functions projected on the two-dimensional searched space in the form of circles of equal values, with the non-dominated and dominated solutions marked by the full squares and the empty squares, respectively. Fig. 2 presents the co-domain of the transformation $\mathbf{f}(\mathbf{x})$ defined by (12), or, equivalently, the set of all vector costs obtained by scanning the domain $\Re^2$[-100,100] of all admissible solutions. Within this area, the 31 P-optimal individuals finally selected are marked by the full squares, while the dominated individuals are represented by the empty squares.

The same experiment has been performed for the GG modification of the EC, and the GGA results are shown in Figs. 3 and 4, where the Pareto-optimal individuals of different genders are marked with the seven full circles and the four full triangles.

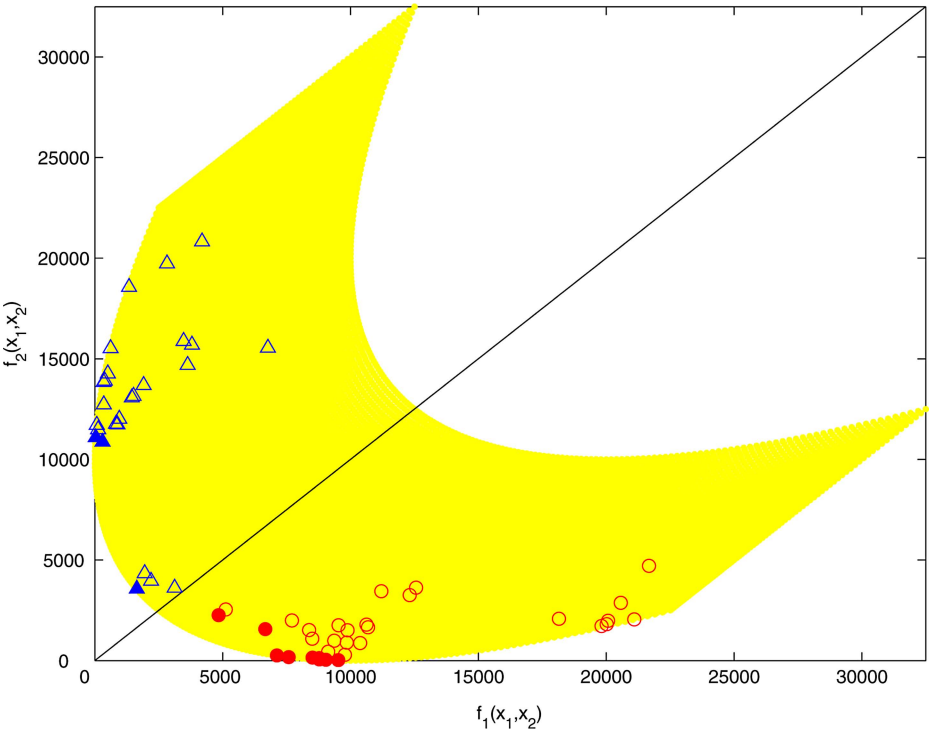


Fig. 4. Gender GGA search: Achievements of the optimal solutions (in their co-domain)

The individuals of the first and the second gender obtained after 100 steps of the GGA optimization process are shown in Fig. 3 by means of the circles and the triangles, respectively, on the background of the equal-cost circles projected on the two-dimensional parameter space.

As can be seen from the above results, the GGA method of restricting the crossover possibilities of the genetic algorithm proved useful in coping with the questions of convergence and diversity.

5 Conclusions

The proposed GGA method of solving EMO problems is based on the recognition of genetic genders. Information about a degree of membership to a given gender set is extracted in the Pareto-suboptimal process of ranking the fitness functions of the analyzed solutions. This information is exploited in the crossover process of mating, in which only individuals of different genders are allowed to create their offspring.

An instructive feature of the proposed multi-optimization approach is the way of utilizing the Pareto-optimization results. Namely, within each gender set

the Pareto-optimization is used as an effective tool of suboptimal judgment of the 'internal' single-gender rivals for the purpose of their uniform estimation and selection (in a greater number) to the new parental subset (and to the next generation) in each iteration cycle of evolutionary computations (or genetic algorithms, GA). It is worth emphasizing that despite of relying on this limited perspective the very notion of the set-fitting P-suboptimality is entirely clear and practically adequate.

The method can be interpreted in terms of: (a) a new mechanism of pre-selecting both the transient and the final individuals, and (b) a mutual inter-gender support in the genetic search. There are also several practical improvements in the performance of ECs gained by GGA (as opposed to the classical Gas), which appear in terms of: sensitivities to initial conditions and high dimensionality, the number of Pareto fronts, solution diversity, convergence, optimality, and the final selection.

The diversity, for instance, represents an attractive genetic 'search power', based on both the internal and external rivalry of individuals, in a more rational way than the popular niching mechanism [5], [20], [19], [17], [14], [7], [3], [8], [1] (see also the literature in [15]).

A major success of the gender approach can be attributed to the fact that it properly deals with a greater number of objectives by reducing the dimensionality of the Pareto-analyzed spaces. Observe that in the full scope optimization case, due to a high dimension of the objectives space, the number of Pareto fronts is strongly limited [13]. This means that many solutions are estimated as equivalent from the Pareto-optimality viewpoint (*i.e.* they have the same rank). As a result, the process of selecting individuals is not effective and the evolutionary search is overly stochastic with no indications and progress in particular directions represented by the stated criteria.

By introducing the GGA approach we solve the above issue by means of restricting the dimensions of the objectives spaces and bringing about a greater number of Pareto fronts within each gender population analyzed in subspaces of restricted dimensions (*i.e.* solely in the space of the assigned gender objectives). This leads to diversity among the individuals of the GGA subpopulations that can be easily estimated and used in effectively pushing the evolutionary exploration into the defined/desired directions on the basis of the achievable distinctive ordering.

As compared to other methods our approach is fairly simple (in both terms of conception and computation), sticks to the very basics of the GA/EC methodology, more profoundly utilizes hints from the nature, fulfills the requirements of technical design and ultimate decision making, and – what is more – is completely open to other amendments and developments proposed in the literature (see for instance [21], [5], [22], [7], [3], [4], [8], [1], [17], [10], [14], as well as the bibliography in [15]).

Our approach is also entirely different from other propositions. Although showing several instrumental (implemental) consequences, basically, the GGA method has a conceptual nature consisting in the objective space decomposition

of the initial problem (and the effective reduction of an originally highly-dimensional problem). Nevertheless, it is interesting to see that the VEGA algorithm [21] and others [18] have few limited similarities. Namely, their partial parental pools can be assigned genders from a maximal m-element set, though it does not make use of the Pareto optimality conception, its genders are not exclusive (one individual can have a number of genders), and the crossover mechanism has generally no gender restrictions.

For the purpose of making the final evaluation of the obtained outcomes, the solutions can be ordered with the use of the global optimality index (GOL), which gives a scalar measure of each solution relating to a common attainable maximal value of all relative partial quality indices.

References

1. Coello, C.C.A.: A short tutorial on evolutionary multiobjective optimization. Proc. 1st Intern. Conference on Evolutionary Multi-Criterion Optimization. *Lecture Notes in Computer Science*, No.1993. Springer-Verlag, Berlin (2001) 21–40
2. Cotta, C., Schaefer, R. (eds.): Special Issue on Evolutionary Computation. *International Journal of Applied Mathematics and Computer Science* **14**(3) (2004) 279–440
3. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multi-objective optimization: Formulation, discussion and modification. In: [4] (1993) 416–423
4. Forrest, S. (ed.): Proc. 5th Intern. Conf. on Genetic Algorithms. Morgan Kaufmann, San Mateo, CA (1993)
5. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA (1989)
6. Grefenstette, J.J. (ed.): Proc. Intern. Conf. on Genetic Algorithms and their Applications. Lawrence Erlbaum Associates, Pittsburgh, PA (1985)
7. Hajela, P., Lin C.-Y.: Genetic search strategies in multicriterion optimal design. *Structural Optimization* bf 4 (1992) 99–107
8. Horn, J., Nafpliotis, N., Goldberg, D.E.: A niched Pareto genetic algorithm for multiobjective optimization, Proc. 1st IEEE Conf. on Evolutionary Computation, IEEE World Congress on Computational Computation, Vol. 1. Piscataway, NJ (1994) 82–87
9. Korbicz, J., Koscielny, J.M., Kowalczuk, Z., Cholewa, W. (eds.): *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Springer-Verlag, Berlin, New York (2004)
10. Kowalczuk, Z., Bialaszewski, T.: Pareto-optimal observers for ship propulsion systems by evolutionary algorithms. Proc. IFAC Symp. on Safeprocess, Vol. 2. Budapest, Hungary (2000) 914–919
11. Kowalczuk, Z., Bialaszewski, T.: Evolutionary multi-objective optimization with genetic sex recognition. Proc. 7th IEEE Intern. Conf. on Methods and Models in Automation and Robotics, Vol. 1. Miedzyzdroje, Poland (2001) 143–148
12. Kowalczuk, Z., Bialaszewski, T.: Performance and robustness design of control systems via genetic gender multi-objective optimization. Proc. 15th IFAC World Congress (CD-ROM). Barcelona, Spain (2002)
13. Kowalczuk, Z., Bialaszewski, T.: Multi-gender genetic optimization of diagnostic observers. Proc. IFAC Workshop on Control Applications of Optimization. Visegrad, Hungary (2003) 15–20

14. Kowalczuk, Z., Bialaszewski, T.: Genetic algorithms in multi-objective optimization of detection observers. In: [9] (2004) 511–556
15. Kowalczuk, Z., Bialaszewski, T.: Niching mechanisms in evolutionary computations. *Int. Journal of Applied Mathematics and Computer Science*, **16**(1) (2006)
16. Kowalczuk, Z., Bialaszewski, T.: Improving evolutionary multi-objective optimization by niching. Proc. 8th Int. Conf. on AI and SC. Zakopane, Poland (2006)
17. Kowalczuk, Z., Suchomski, P., Bialaszewski, T.: Evolutionary multi-objective Pareto optimization of diagnostic state observers. *Int. Journal of Applied Mathematics and Computer Science* **9**(3) (1999) 689–709
18. Lis, J., Eiben, A.E.: A multi-sexual genetic algorithm for multiobjective optimization. Proc. IEEE Int. Conference on Evolutionary Computation. (1997) 59-64.
19. Man, K.S., Tang, K.S., Kwong, S., Lang, W.A.H.: *Genetic Algorithms for Control and Signal Processing*. Springer-Verlag, London (1997)
20. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin (1996)
21. Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: [6] (1985) 93–100
22. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* **2**(3) (1994) 221–248
23. Zakian, V., Al-Naib, U.: Design of dynamical and control systems by the method of inequalities. *IEE Proceedings on Control Theory and Applications* **120**(11) (1973) 1421–1427
24. Viennet, R., Fontiex, C., Marc, I.: Multicriteria optimisation using a genetic algorithm for determining a Pareto set. *International Journal of Systems Science* **27**(2) (1996) 255–260

Evolutionary Learning of Linear Trees with Embedded Feature Selection

Marek Krętowski and Marek Grześ

Faculty of Computer Science
Białystok Technical University
Wiejska 45a, 15-351 Białystok, Poland
{mkret, marekg}@ii.pb.bialystok.pl

Abstract. In the paper a new evolutionary algorithm for global induction of linear trees is presented. The learning process consists of searching for both a decision tree structure and hyper-plane weights in all non-terminal nodes. Specialized genetic operators are developed and applied according to the node quality and location. Feature selection aimed at simplification of the splitting hyper-planes is embedded into the algorithm and results in elimination of noisy and redundant features. The proposed approach is verified on both artificial and real-life data and the obtained results are promising.

1 Introduction

Decision trees are, besides decision rules, one the most popular forms of knowledge representation in data mining systems [8] and clones of the classical induction algorithms are included in almost all exploratory tools. The popularity of the decision tree approach can be explained by their ease of application, fast operation and what may be the most important, their effectiveness. Furthermore, the hierarchical structure of a tree classifier, where appropriate tests from consecutive nodes are sequentially applied, closely resembles a human way of decision making which makes decision trees natural and easy to understand even for the not experienced analyst.

There are two main types of decision trees [17]: more common *univariate* trees, where tests in non-terminal nodes use single features, and *linear* (oblique) ones, where splits are based on the dividing hyper-planes. The most known example of the first group is *C4.5* [20] with its commercial version *C5.0*, whereas *LTree* and *OC1* [16] can be treated as good representatives of the second group. There exist also heterogeneous systems, like the well-known *CART* [3], where both forms of tests are permitted.

In univariate trees an inequality test is equivalent to partitioning the feature space with an axis-parallel hyper-plane. However in many real-life problems decision borders are not axis-parallel and the use of only simple tests may lead to over-complicated classifiers (so called "staircase effect"). In such a situation, a piece-wise linear solution offered by an oblique tree is much more natural and appropriate. The richer representation of linear trees gives one the opportunity

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 400–409, 2006.
© Springer-Verlag Berlin Heidelberg 2006

to find simpler and more accurate classifiers. However, it should be mentioned that induction algorithms for this type of decision tree are computationally more complex.

Nature-inspired techniques like evolutionary computation [15] are known to be especially useful in difficult optimization tasks and they are successfully applied to various data mining tasks [9]. As for decision tree learning there are two main approaches to the induction: top-down and global. The first one is based on a greedy recursive procedure of test searching and sub-node creation until the stop condition is met. In contrast to this classical method, the global algorithm searches for both the tree structure and tests at the moment. Evolutionary methods were applied for both induction types and both tree types. In the framework of univariate trees, most of the research was concentrated on global induction (e.g. [11,18,19,14]), whereas for linear trees mainly top-down methods were developed, where only splitting hyper-planes in internal nodes were evolutionary searched (e.g. [5,4,12]). In [2] genetic programming was applied to induce classification trees with limited oblique splits.

In this paper we focused on the global learning of linear decision trees with embedded feature selection. In real applications, data gathered in operational databases, which are used as a learning set, often contain irrelevant or redundant features. The overall performance of the decision tree can be improved by noisy feature elimination and test simplification. Furthermore, the ability to understand and properly interpret the classifier can be increased.

The rest of the paper is organized as follows. In the next section the proposed approach is described. Section 3 presents experimental verification with both artificial and real-life data. In the last section, the paper is concluded and future research directions are outlined.

2 Evolutionary Algorithm for Global Induction of Oblique Decision Trees

The structure of the proposed approach follows the typical evolutionary algorithm framework as described in [15]. The algorithm can be seen as a continuation and significant extension of the work presented in [13]. Due to lack of space, we gave most of our attention to new issues introduced in this work: a new scheme of applying genetic operators and embedding feature selection into the induction process.

2.1 Preliminaries

A learning set is composed of M N-dimensional feature vectors $\mathbf{x}^j = [x_1^j, ..., x_N^j]^T$ $(j = 1, ..., M)(\mathbf{x}^j \in R^N)$ belonging to one of K classes. The feature space can be divided into two regions by a hyper-plane:

$$H(\mathbf{w}, \theta) = \{\mathbf{x} : \langle \mathbf{w}, \mathbf{x} \rangle = \theta\}, \tag{1}$$

where $\mathbf{w} = [w_1, ..., w_N]$ ($\mathbf{w} \in R^N$) is a weight vector, θ is a threshold and $\langle \mathbf{w}, \mathbf{x} \rangle$ represents an inner product. A linear decision tree is a binary tree with splitting hyper-planes in internal nodes and class labels in leaves.

2.2 Genetic Operators

In the majority of evolutionary approaches there are two types of genetic operators applied to individuals: *mutation*, which affects single chromosome and *cross-over*, which enables exchange of genetic information among two chromosomes. In a typical setup, a mutation-like operator is applied with equal and relatively small probability to any gene of the chromosome. In [13] we followed this scheme and every node of the tree has the same chance of being modified. However, it seems that this approach is not really appropriate for the non-linear and hierarchical structure of the decision tree. It is evident that modification of the test in the root node is very crucial because it affects all descendant nodes, whereas mutation of a node near leaves has only a local impact. It was also observed that for small trees mutations were very rare and this significantly slowed down the induction process. Furthermore, mutations of certain nodes, like leaves with feature vectors from only one class, are not profitable at all and should be avoided.

A new complex mutation-like operator is introduced to avoid the aforementioned shortcomings. It is applied with a given probability (default 0.5) to a tree and it guarantees that at least one node will be mutated. First, the type of the node (a leaf or an internal node) is randomly chosen with equal probability. Then the ranked list of nodes is created and a mechanism analogous to ranking linear selection [15] is applied to decide which node will be affected. While concerning leaves, the number of feature vectors from other classes than the decision assigned to the leaf (i.e. number of objects misclassified in this node) is used to put them in order. Additionally, homogeneous leaves (with objects only from one class) are excluded from the list. As a result, leaves which are better in terms of the classification accuracy are mutated with lower probability. As for internal nodes, locations (the level) of the node in the tree is taken into account. It allows us to mutate with higher probability nodes, which are situated on the lower levels of the tree. In Fig. 1 an example of constructing the ranking lists of leaves and internal nodes is presented.

When one node is chosen, possibilities of applying different variants of the mutation are checked and one of them is randomly chosen according to its relevance to the node and to the given probability. Among considered possibilities are:

- changing the role of the node (i.e. pruning an internal node to a leaf or replacing a leaf by a sub-tree);
- the dipolar operator introduced in [12]. It starts with the random choosing of one dipole[1] from the set of not-divided mixed dipoles and divided pure

[1] A *mixed* dipole is a pair of feature vectors from different classes, while in *pure* dipole both objects belong to the same class.

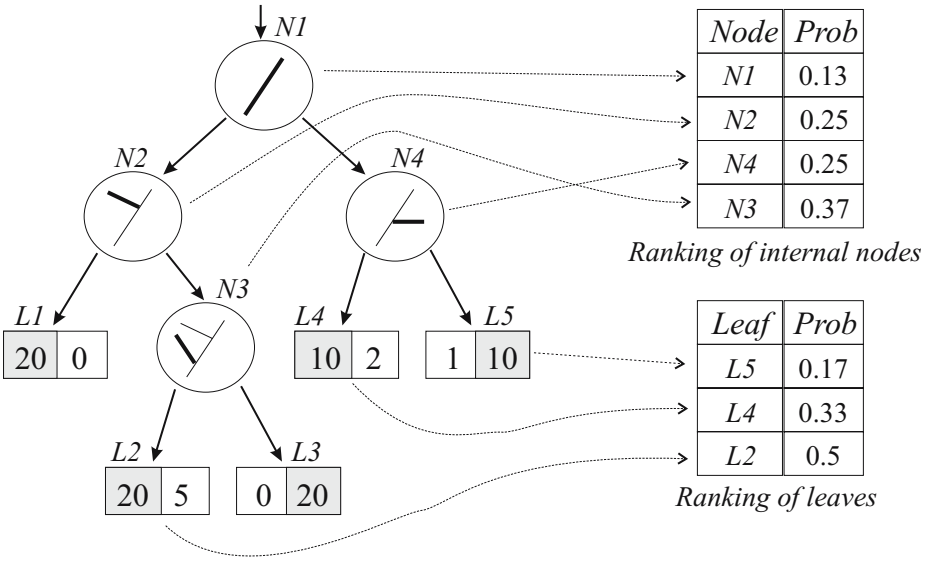


Fig. 1. The construction of two ranking lists of nodes (separate list for leaves and internal nodes) for the mutation operator

ones. If the mixed dipole is drawn, the hyper-plane is shifted to cut it. A new position is obtained by modification of only one randomly chosen weight (or threshold). In case of the pure dipole, the hyper-plane is shifted to avoid separation of objects from the same class and similarly as for mixed dipoles, only one coefficient is modified.

- the classical modification of a single weight of the hyper-plane; in relation to feature selection the probability of dropping a feature (i.e. assigning zero value to a weight) was increased because in the real value representation the number zero has extremely low probability;
- tests from the node and one of its sons are interchanged. This variant can be applied only if at least one son of the considered node is also a non-terminal node.
- one sub-tree can be replaced by another sub-tree from the same node;
- a test can be replaced by an entirely new test, which is chosen in a dipolar way [12]. One mixed dipole is randomly chosen and a hyper-plane is placed to split it. More precisely the hyper-plane is perpendicular to the segment connecting opposite ends of the dipole. The same method is applied for finding splitting hyper-planes during creation of an initial population.

In the presented system, there is also an operator analogous to the standard cross-over. One node is randomly chosen in each of two affected individuals and an exchange encompasses a sub-tree or is limited only to nodes (their hyper-planes). Additionally if both nodes are non-terminal ones, the typical one-point cross-over is applied on weight vectors and thresholds. In other cases nodes are

just substituted. It should be however noted, that according to the results [9] obtained in the framework of genetic programming, where a solution is also encoded in a tree-like structure, the context-insensitive cross-over operator has a rather destructive effect on the offspring. For that reason, the cross-over operator is applied with relatively low probability (default value is equal 0.2) in our system.

The application of a genetic operator to a tree makes its part rooted in the modified node invalid in relation to locations of the input feature vectors. For this reason renewed determination of the locations is necessary. This process can lead to a situation where certain nodes in the sub-tree are empty (or almost empty) and have to be removed. Additionally maximization of fitness is performed by pruning lower parts of the sub-tree on the condition it improves the value of the fitness.

The problem which is directly connected with feature selection is "underfitting" the training data [6], which often occurs near the leaves of the tree. The number of feature vectors used to search for a splitting hyperplane has to be significantly greater than the number of features used. In the presented system, the maximal number of features in a test is restricted using the number of available training objects (default value is 5 objects for each non-zero feature weight).

2.3 Fitness Function

A key issue for any decision tree induction algorithm is finding the appropriate balance between the re-classification accuracy and the generalization power related to the classifier complexity. In the classical top-down approach, the overfitting problem is mitigated by applying a post-pruning algorithm, but such a method has only a limited possibility to restructure the tree [7]. In contrast, the proposed evolutionary algorithm represents the global approach, where the search for an optimal tree structure is a built-in element of the process, thanks to a suitably defined fitness function. The fitness function, which is maximized, has the following form:

$$Fitness(T) = Q_{Reclass}(T) - \alpha \cdot (Comp(T) - 1.0), \tag{2}$$

where $Q_{Reclass}(T)$ is the re-classification quality, $Comp(T)$ is the complexity measure of the tree T and α is the relative importance of the complexity term (default value is 0.005) and a user supplied parameter. Subtracting 1.0 eliminates the penalty related to the complexity of the classifier when the tree is composed of only one leaf - the root node.

The complexity term $Comp(T)$, which is crucial for effective feature selection, should reflect both the tree size (the number of nodes) and the complexity of tests. This can be obtained by expressing $Comp(T)$ as a sum of test complexities in the internal nodes and the number of leaves (for any leaf we assume that

the complexity is equal 1.0). For the hyper-plane $H(\mathbf{w}, \theta)$ the test complexity $Comp(\mathbf{w})$ can be defined as follows:

$$Comp(\mathbf{w}) = (1 - \beta) + \beta \frac{n(\mathbf{w})}{N}, \tag{3}$$

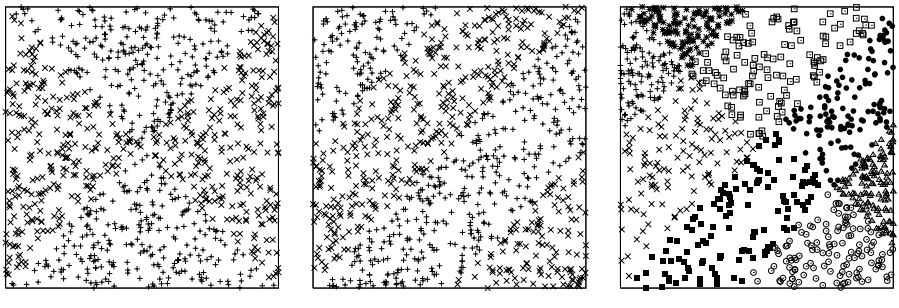
where $n(\mathbf{w})$ is the number of non-zero weights in the hyper-plane and β is a user supplied parameter designed for controlling the impact of test complexity on the tree size (default value = 0.5). When $n(\mathbf{w}) = N$ the test complexity is equal 1, which means that no feature is eliminated from the test. If this condition is met for all tests, $Comp(T)$ is equal to the number of nodes.

It should be noted that, when concerning a specific dataset, by tuning α and β parameters better results can be obtained in terms of accuracy or classifier complexity.

3 Experimental Results

Two groups of experiments are performed to validate the proposed approach (denoted in tables as *GDT*). For the purpose of comparison, results obtained by the well-known *OC1* system [16] are also presented. Both systems were run with default values of parameters. Presented in tables results correspond to averages of 10 runs and were obtained by using test sets (when available) or by 10-fold stratified cross-validation. The average number of leaves is given as a complexity measure of classifiers.

In the first group, artificial datasets with analytically defined decision borders are analyzed. Analogous experiments are described *e.g.* in [16], but original datasets are not available, hence similar configurations were generated by using a random number generator (see Fig. 2). All these datasets are two-dimensional, except *LS5* and *LS10* problems which are defined with 5 and 10 features correspondingly. The number of feature vectors in the learning sets is 1000.



a) b) c)

Fig. 2. Examples of 2-dimensional artificial datasets: a) *rotated chessboard*, b) *zebra1* and c) *zebra3*

Table 1. Results obtained for artificial datasets. Classification accuracy [%] is given as the quality measure and number of leaves as the tree size.

	without noise				50% noise				100% noise			
	GDT		OC1		GDT		OC1		GDT		OC1	
Dataset	Quality	Size	Quality	Size	Quality	Size	Quality	Size	Quality	Size	Quality	Size
chess2	98.4	4	99.3	6	97.3	4.0	89.9	16	96.5	4.0	90.7	22
zebra1	99.1	4.1	98.2	8	98.7	4.1	97.8	8	98.3	4.0	96.7	8
zebra3	97.4	9.0	95.1	15	93.5	9.7	96.3	8	93.9	8.4	90.4	8
LS2	99.8	2	99.7	2	99.9	2	99.9	2	99.8	2	98.8	6
LS5	99.5	2	98.7	2	98.7	2	99.4	4	99	2	98.4	2
LS10	98.1	2.0	96	4	97.3	2.0	93.5	4	97.3	2.0	93.5	4

In order to check the robustness of induction algorithms and especially the efficiency of the embedded feature selection, noisy features were added. They were generated randomly without taking into account class labels and obviously they are irrelevant to classification. Two levels of added noise are analyzed: the number of noisy features is equal to the half of the original amount in the features (denoted as "50% noise") and the number of additional features is equal to the number of features in the original dataset ("100% noise").

Results of experiments with artificial datasets are gathered in the Table 1. For all domains *GDT* performed very well, both in terms of classification accuracy and tree complexity. Compared to *OC1* it was able to find simpler trees with competitive accuracy. It is should be emphasized that *GDT* was also much more robust for added noise than its rival. It can be observed that the accuracy of our system is only slightly decreased, whereas *OC1* performed significantly worse in noisy scenarios.

One of the important innovations introduced in the paper is a new way of applying mutation-like operators (i.e. mutation is applied to the tree and not independently to nodes). It is rather difficult to analyze the impact of any single

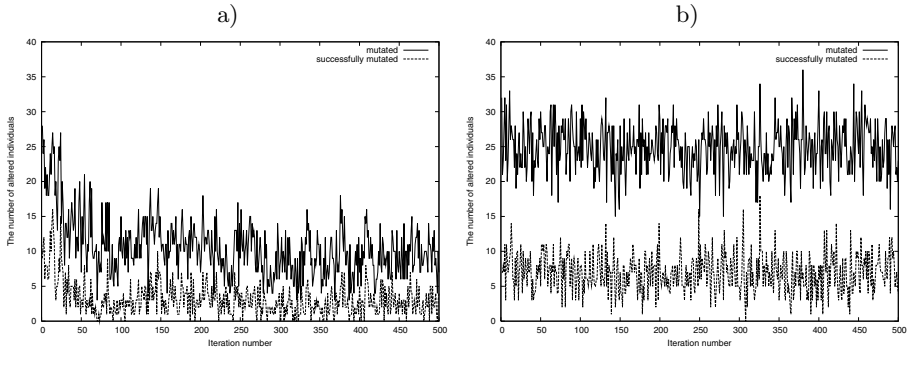


Fig. 3. The performance of two mutation strategies on *LS5* dataset: a) mutation of nodes (5%), b) mutation of the tree (50%)

Table 2. UCI datasets description. In brackets the number of feature vectors in the testing set is provided when such a set is available.

Dataset	Number of examples	Number of features	Number of classes
breast-w	683	9	2
bupa	345	6	2
iris	150	4	3
page-blocks	5473	10	5
pima	768	7	2
sat	4435(2000)	36	7
vehicle	846	18	4
waveform	600(3000)	21	3

operator modification only by looking at the final classification results and more detailed simulations are necessary. In Fig. 3 frequency and efficiency of two types of mutation are compared in one typical run of the algorithm. All other parameters except the mutation type are exactly the same. Performance of two mutation strategies is presented on *LS5* dataset because the optimal decision tree for this problem is composed of only one internal node. It can be easily observed that with the convergence of the search the mutation of nodes becomes less frequent and effective, whereas the mutation of the tree performs equally effectively and can finely fit the dividing hyper-plane position.

In the second series of experiments, a few datasets taken from UCI Machine Learning Repository [1] are analyzed to assess the performance of the proposed system in solving real-life problems. In order to avoid the problem of coding nominal features and treating missing values the datasets with only continuous-valued features and without missing values were chosen. Table 2 presents characteristics of investigated datasets. Obtained results are gathered in Table 3.

The proposed system performed well on almost all analyzed datasets, but its superiority over *OC1* is not so evident as for artificial datasets. The worst

Table 3. Results obtained for real-life datasets from UCI repository. Classification accuracy [%] is given as the quality measure and number of leaves as the tree size.

	GDT		OC1	
Dataset	Quality	Tree size	Quality	Tree size
breast-w	96.7	2.0	95.3	3.0
bupa	68.8	3.5	67.5	6.9
iris	97.0	3.0	96.7	3.0
page-blocks	95.2	3.0	97.0	12.0
pima	75.6	2.1	72.6	5.1
sat	83.7	6.3	85.4	45.0
vehicle	67.6	8.2	70.2	15.4
waveform	82.4	4.2	78.0	3.0

result was obtained by the *GDT* system with a default set of parameters for the *vehicle* dataset. It was verified that significantly better classification accuracy can be easily obtained just by relaxing the stoping condition. Finally, it was once more confirmed that the global approach finds more compact trees with at least comparable accuracy.

As can be expected, the induction times of EA-based system are longer that its top-down rival. However, even for *page-blocks*, which is the biggest dataset composed of more than 5000 feature vectors, the computation time is equal to about 58 min as measured on a standard PC (PIV 3GHz, 1GB RAM). It seems that such an amount of time is acceptable in most analytical applications.

4 Conclusion

In the paper a new evolutionary algorithm for global induction of linear decision trees is proposed. In contrast to the classical top-down approaches, both the structure of the classifier and all hyper-planes in internal nodes are searched during one run of the algorithm. The modified scheme of applying genetic operators leads to a more effective search process that is not sensitive to the tree size. The presented system is able to detect and eliminate noisy or irrelevant features from tests, thanks to feature selection embedded into the induction algorithm. Experimental validation of our approach shows that resulting trees are simpler and more compact with at least the same classification accuracy as existing counterparts.

The presented approach is constantly improved and currently we are working on introducing also univariate tests (tests with nominal outcomes and inequality tests for continuous-valued features) into our system. This should allow the algorithm to better adapt to the problem solved and to locally choose the most suitable test representation.

While investigating evolutionary algorithms there is always a strong motivation for speeding them up. Because they are well suited for parallel architecture we are contemplating re-implementing our system in a distributed environment. This is especially important in the context of modern data mining applications, where huge learning sets are analyzed.

Acknowledgments. This work was supported by the grant W/WI/5/05 from Białystok Technical University.

References

1. Blake, C., Keogh, E., Merz, C.: *UCI repository of machine learning databases*,[http://www.ics.uci.edu/˜mlearn/MLRepository.html]. Irvine, CA: University of California, Dept. of Computer Science (1998).
2. Bot, M., Langdon, W.: Application of genetic programming to induction of linear classification trees. In *EuroGP 2000*. Lecture Notes in Computer Science 1802, (2000) 247–258.

3. Breiman, L., Friedman, J., Olshen, R., Stone C.: *Classification and Regression Trees*. Wadsworth Int. Group (1984).
4. Cantu-Paz, E., Kamath, C.: Inducing oblique decision trees with evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* **7**(1) (2003) 54–68.
5. Chai, B., Huang, T., Zhuang, X., Zhao, Y., Sklansky, J.: Piecewise-linear classifiers using binary tree structure and genetic algorithm. *Pattern Recognition* **29**(11) (1996) 1905–1917.
6. Duda, O., Heart, P., Stork, D.: *Pattern Classification*. 2^{nd} edn. J. Wiley (2001).
7. Esposito, F., Malerba, D., Semeraro, G.: A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(5) (1997) 476-491.
8. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy R., (Eds.): *Advances in Knowledge Discovery and Data Mining*, AAAI Press, (1996).
9. Freitas A.: *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer (2002).
10. Gama, J., Brazdil, P.: Linear tree. *Inteligent Data Analysis* **3**(1) (1999) 1–22.
11. Koza, J.: Concept formation and decision tree induction using genetic programming paradigm, In *Proc. of PPSN 1.*, Lecture Notes in Computer Science 496 (1991) 124–128.
12. Kręt owski, M.: An evolutionary algorithm for oblique decision tree induction, In: *Proc. of ICAISC'04.*, Lecture Notes in Computer Science 3070, (2004) 432–437.
13. Kręt owski, M., Grześ, M.: Global induction of oblique decision trees: an evolutionary approach, In: *Proc. of IIPWM'05.*, Springer, Advances in Soft Computing, (2005) 309–318.
14. Kręt owski, M., Grześ, M.: Global learning of decision trees by an evolutionary algorithm, In: *Information Processing and Security Systems.*, Springer, (2005) 401–410.
15. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. 3^{rd} edn. Springer (1996).
16. Murthy, S., Kasif, S., Salzberg, S.: A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research* 2 (1994) 1–33.
17. Murthy, S.: Automatic construction of decision trees from data: A multidisciplinary survey. *Data Mining and Knowledge Discovery* 2 (1998) 345–389.
18. Nikolaev, N., Slavov, V.: Inductive genetic programming with decision trees. *Intelligent Data Analysis* 2 (1998) 31–44.
19. Papagelis, A., Kalles, D.: Breeding decision trees using evolutionary techniques. In: *Proc. of ICML'01.*, Morgan Kaufmann (2001) 393–400.
20. Quinlan, J.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann (1993).

Solving the Balanced Academic Curriculum Problem with an Hybridization of Genetic Algorithm and Constraint Propagation

T. Lambert[1,2], C. Castro[3], E. Monfroy[1,3,⋆], and F. Saubion[2]

[1] LINA, Université de Nantes, France
Firstname.Name@lina.univ-nantes.fr
[2] LERIA, Université d'Angers, France
Firstname.Name@univ-angers.fr
[3] Universidad Santa María, Valparaíso, Chile
Firstname.Name@inf.utfsm.cl

Abstract. In this paper, we are concerned with the design of a hybrid resolution framework including genetic algorithms and constraint propagation to solve the balanced academic curriculum problem. We develop a theoretical model in which hybrid resolution can be achieved as the computation of a fixpoint of elementary functions. These functions correspond to basic resolution techniques and their applications can easily be parameterized by different search strategies. This framework is used to solve a specific problem and we discuss the experimental results showing the interest of the of the model to design such hybridizations.

Keywords: CSP, genetic algorithms, constraint propagation, hybrid resolution.

1 Introduction

Constraint Satisfaction Problems (CSP) are usually defined by a set of variables associated to domains of possible values and by a set of constraints over these variables. They provide a modeling framework for many computer aided decision making practical applications (such as planning, scheduling, time tabling,...). CSP model is extended, for real world applications, in order to optimize a given objective function. The Balanced Academic Curriculum Problem (BACP) has been introduced in [6] and consists in planning the different courses of an academic curriculum on a given set of periods, satisfying some constraints to insure the most suitable organization for the students.

Solving such a problem consists in finding an assignment of values to the variables that satisfies the constraints and optimizes the given criterion. Many resolution algorithms have been proposed and can be classified in two main groups.

⋆ The author has been partially supported by the Chilean National Science Fund through the project FONDECYT N°1060373.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 410–419, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Complete methods aim at exploring the whole search space in order to find all the solutions or to detect that the CSP is not consistent. Concerning complete resolution techniques, we are here mainly concerned with constraint propagation with split of domains of variables (e.g., enumeration), i.e., one of the most famous techniques of Constraint Programming (CP) [3].

Incomplete methods mainly rely on the use of metaheuristics providing a more efficient exploration of interesting areas of the search space in order to find some solutions. Most commonly used approaches are based on evolutionary [12,10] and local search algorithms [1]. In this paper, we focus on the first ones and more especially on genetic algorithms (GA).

In order to improve the efficiency of the solving algorithms, combinations of resolution paradigms and techniques have been studied (e.g. [8] presents an overview of possible uses of local search [1] in constraint programming [3]).

The benefit of the hybridization GA+CP is well-known (see [4] , [5]). Most of these works are rather algorithmic approaches which define a kind of master-slave combination, (e.g., LS to guide the search in CP, or CP to reduce interesting area of the search space explored in LS) or ad-hoc realizations of systems for specific classes of problems.

Our purpose is twofold: on the one hand, we aim at solving efficiently BACP by combining a genetic algorithm with constraint programming techniques, and on the other hand, we propose a general modelling framework to precisely design such hybrid resolution process and to highlight their characteristics and properties. This framework allows one to design and manage new and finer solving strategies and extensions.

To this end, we use our uniform generic hybridization framework [15] which is based on K.R. Apt's chaotic iterations [2], a mathematical framework for iterations of a finite set of functions over "abstract" domains with partial ordering. In this framework, basic hybrid resolution processes (such as domain variable reductions, offspring generation, and enumeration) are considered and managed at the same level by a single mechanism. Hence, we may adjust the application rate of the different resolution process in order to model various search strategies. We integrated GA functions and optimization aspects in our constraint system [7] (which is based on our hybrid framework) and solved instances of the BACP. The results show the benefits of our framework and of hybridization as well.

The paper is organized as follows: after having raised the problem in Section 2, we present in Section 3 an overview of GA and CP, confronted with experimentations in Section 4 before concluding in Section 5.

2 The Balanced Academic Curriculum Problem (BACP)

The problem consists in organizing courses in order to balance the work load of students for each period of their curriculum. Each course is given a number of credits representing the amount of work necessary to successfully follow the course. The load of a period is the sum of the credits of each course of the period.

Some more constraints are added: there is a maximum and minimum load per period, and some precedence relationships are established among some courses.

- Parameters:
  m, n: number of courses and number of periods
  α_i: number of credits of course i : $\forall i = 1, \ldots, m$
  β, γ: minimum and maximum academic loads allowed per period
  δ, ϵ: minimum and maximum numbers of courses per period
- Variables:
  x_i: period of course i , $\forall i = 1, \ldots, m$,$x_i \in [1..n]$
  c_i: academic load of course i , $\forall i = 1, \ldots, m$
- Objective function: $Min\ c = max(\sum_{k=1}^{m} c_k \mid x_k = j, \forall j = 1, \ldots, n)$
- Constraints:
  Courses b has course a as prerequisite : $x_a < x_b$
  Each period j has a greater or equal and a less or equal academic load allowed: $\beta \leq \sum_{k=1}^{m} c_k \mid x_k = j \leq \gamma$
  Each period j has a greater or equal and a less or equal number of courses allowed: $\delta \leq \sum_{k=1}^{m} 1 \mid x_k = j \leq \epsilon$

Thus, a solution is a fair assignment of courses to periods: we translate it as the minimization of the highest load period. For a more detailed description, the reader can refer to [6].

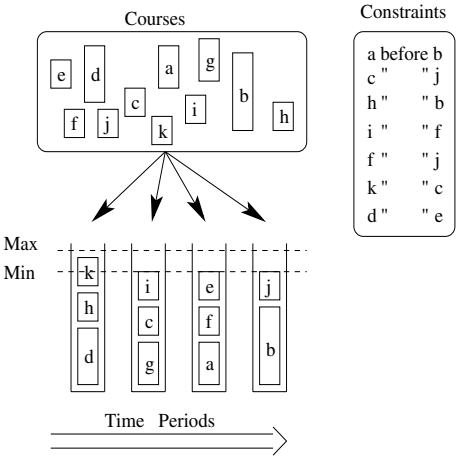


Fig. 1. Example of a courses distribution

As an example, for the 10 periods problem, 42 courses have to be assigned in the 10 periods. Every course has a cost affected from 1 to 5. The number of prerequisite constraints is 32 and for each period the number of courses has to be between 2 and 10, the charge (sum of load) between 10 and 24.

3 The Hybrid Algorithm

We first recall the resolution paradigms related to CSP, optimization, and GA and how they will be integrated in our hybrid algorithm.

3.1 Constraint Programming

A CSP is a tuple (X, D, C) where $X = \{x_1, \cdots, x_n\}$ is a set of variables taking their values in their respective domains $D = \{D_1, \cdots, D_n\}$. A constraint $c \in C$ is a relation $c \subseteq D_1 \times \cdots \times D_n$. In order to simplify notations, D will also denote the Cartesian product of D_i and C the union of its constraints. A tuple $d \in D$ is a solution of a CSP (X, D, C) if and only if $\forall c \in C, d \in c$.

Constraint propagation, one of the most famous techniques for solving CSP consists in iteratively reducing domains of variables by removing values that do not satisfy the constraints. However, reduction mechanisms use one or some of the constraints of the CSP. Thus, they enforce a local consistency property (such as arc-consistency) but not a global consistency of the CSP. These reductions must be interleaved with a splitting mechanism (such as enumeration) in order to obtain a complete solver, (i.e., solver which returns only solutions and does not loose any solution).

Constraint optimization problems, although similar to constraint solving, is comparatively harder because it only accepts solutions (i.e., values of variables) that minimize or maximize a given objective function while satisfying the constraints.

Prerequisite constraints are easily converted to binary constraint and abstracted to arc-consistency reduction functions on course domains. The constraints on periods: sum of loads and number of courses are represented as global constraints and used to prune the search tree by detecting inconsistencies. Thus, for the 8-period problem we have the following global constraints: $load(i, 10, 24), i = 1, .., 8$ standing for the allowed range (10,24) of the sum of load for the period i, and $courses(i, 2, 10), i = 1, .., 8$ for the allowed range (2,10) of the sum of courses for the period i.

During the search, in a given CSP, the global constraint $period(i, \delta, \epsilon)$ computes the number of domains within the value i. If less than δ occurrences of the period i are present in the different domains of courses, then the current CSP is locally inconsistent. The global constraints $load(i, \beta, \gamma)$ counts the charge range for a given period i in the current CSP.

3.2 Genetic Algorithms

Evolutionary algorithms are mainly based on the notion of adaptation of a population of individuals to a criterion using evolution operators like crossover [10]. Based on the principle of natural selection, *Genetic Algorithms* [12,13] have been quite successfully applied to optimization problems such as scheduling or transportation problems.

The key principle of this approach states that, species evolve through adaptations to a changing environment and that the gained knowledge is embedded in the structure of the population and its members, encoded in their chromosomes. If individuals are considered as potential solutions to a given problem, applying a genetic algorithm consists in generating better and better individuals w.r.t. the problem by selecting, crossing, and mutating them. This approach reveals very useful for problems with huge search spaces. We had to adapt some basic techniques and slightly modify some definitions to fit our context.

In the context of GA, for the resolution of a given CSP (X, D, C), the search space can be usually defined with the set of tuples $D = D_1 \times \cdots \times D_n$. We consider populations g of size i, $g \subseteq D$ such as $|g| = i$. An element $s \in g$ is an individual and represents a potential solution to the problem. Our genetic algorithm aims at generating from a population k, a new population k+1 of 60 individuals selected among 100 issued from k. Each time GA is called by the main algorithm, the following different cases can occur:

- The population $k+1$ has less than 100 individuals: an individual is selected randomly; then, either it is coupled with another parent to create 2 children in the population $k+1$, either it is submitted to mutation, or it is not change in the population $k+1$.
- Population $k+1$ has 100 individuals: a selection of the 60 best ones is made according to the evaluation function.

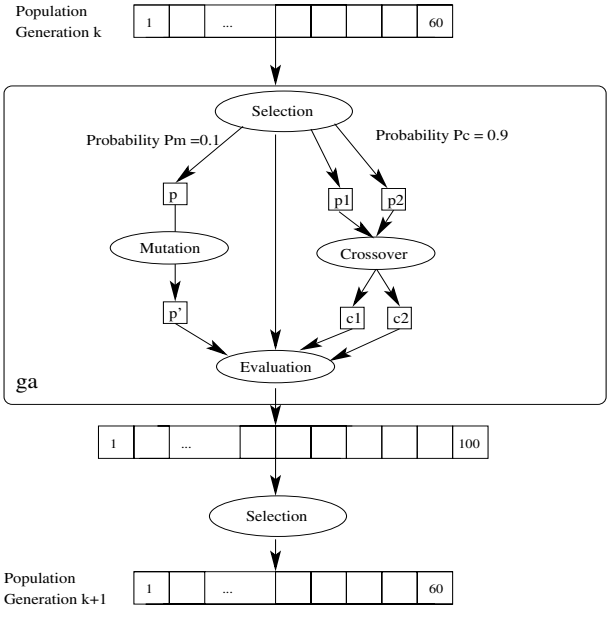


Fig. 2. Genetic Algorithm

3.3 Designing the Hybrid Algorithm

K.R. Apt proposed in [2] a general theoretical framework for modelling such reduction operators. In this context, domain reduction corresponds to the computation of a fixpoint of a set of functions over a partially ordered set. These functions, called *reduction functions*, abstract the notion of constraint. The model is extended with splitting operators and GA in order to model the different hybrid solving methods as the computation of a fixpoint of a set of functions. This model is also now extended to optimization problems such as BACP. The computation of the least common fixpoint of a set of functions F is achieved by the following algorithm:

GI: Generic Hybrid Algorithm

$d := \bot$;
$G := F$;
While $G \neq \emptyset$ do
    choose $g \in G$;
    $G := G - \{g\}$;
    $G := G \cup update(G, g, d)$;
    $d := g(d)$;
endwhile

where G is the current set of functions still to be applied ($G \subseteq F$), $d \in D$ a partially ordered set (the domains in case of CSP), and for all G, g, d the set of functions $update(G, g, d)$ from F is such that:

- A : $\{f \in F - G \mid f(d) = d \wedge f(g(d)) \neq g(d)\} \subseteq update(G, g, d)$.
- B : $g(d) = d$ implies that $update(G, g, d) = \emptyset$.
- C : $g(g(d)) \neq g(d)$ implies that $g \in update(G, g, d)$

This abstract framework corresponds to the hybridization of resolution techniques since we want to use a evolutionary optimization techniques combined with constraint propagation techniques.

Basically, we define three families of functions corresponding to domain reduction functions, split and the generation of a new population by a one step genetic algorithm. These functions are then alternatively chosen according to a given strategy and applied until a fixpoint is reached on the structure (this fixpoint characterizes either an optimal solution or a maximum number of allowed iterations). Therefore, hybridization and strategies are easily usable in this generic context as it will be shown in next section.

4 Experimental Results

In [14], we presented the constraint based solving system we built for hybridization of local search and constraints propagation. We have re-used our system and integrated the GA module (i.e., *ga* functions) in it. We have also added the notion of optimization to the notion of solution we had.

All tests are performed on a cluster with 22 processors used sequentially running at 2.2 GHz with 1 Go of RAM each.

4.1 Selected Problems

We consider the bacp8, bacp10 and bacp12 problems issued from the CSPlib [9] and latest data of these three curricula to form a new problem where some courses are shared by the different curricula. In order to give a point of reference, we present the results of [6] using the linear programming solver lp_solve for the 8-periods and 10-periods problems (Figure 3) and using our hybrid solver (Figure 4). If lp_solve is able to find the optimal solution for the first one, it is not the case for the second one.

Sol quality	bacp 8	Sol quality	bacp 10
24	137.08	33	9.11
23	218.23	32	25.38
21	218.43	30	25.65
20	712.84	29	1433.18
19	1441.98	27	1433.48
18	1453.73	26	1626.49
17 [1]	1459.73	24	1626.84

Fig. 3. Results in seconds using lp_solve

Sol quality	bacp 8	bacp 10	bacp 12
24	0.47	4.71	2.34
23	0.54	4.67	2.40
22	0.61	3.68	2.48
21	0.61	4.36	2.76
20	0.69	4.63	3.20
19	0.83	4.95	4.25
18	1.20	5.13	35.20
17	15.05 [1]	5.60	
16		6.39	
15		8.53	
14		34.84 [1]	

Fig. 4. Results using GA+CP

4.2 Strategies

We control the rates of each family of functions (reduction, split and genetic) by giving as strategy a tuple $(\%_{dr}, \%_{sp}, \%_{ga})$ of application rates. These values corresponds indeed to a probability of application of a function of each family but, in practice, we measure in Fig 5 the real rate of application (i.e., we only count the functions which are chosen according to the strategy and which have a real impact on the resolution).

4.3 Analysis

The most interesting in such an hybridization is the completeness of the association GA-CP, and the roles played by GA and CP in the search process (see Figures 5) : GA optimizes the solutions in a search space progressively being locally consistent (and thus smaller and smaller) using constraints propagation

[1] Optimum found.

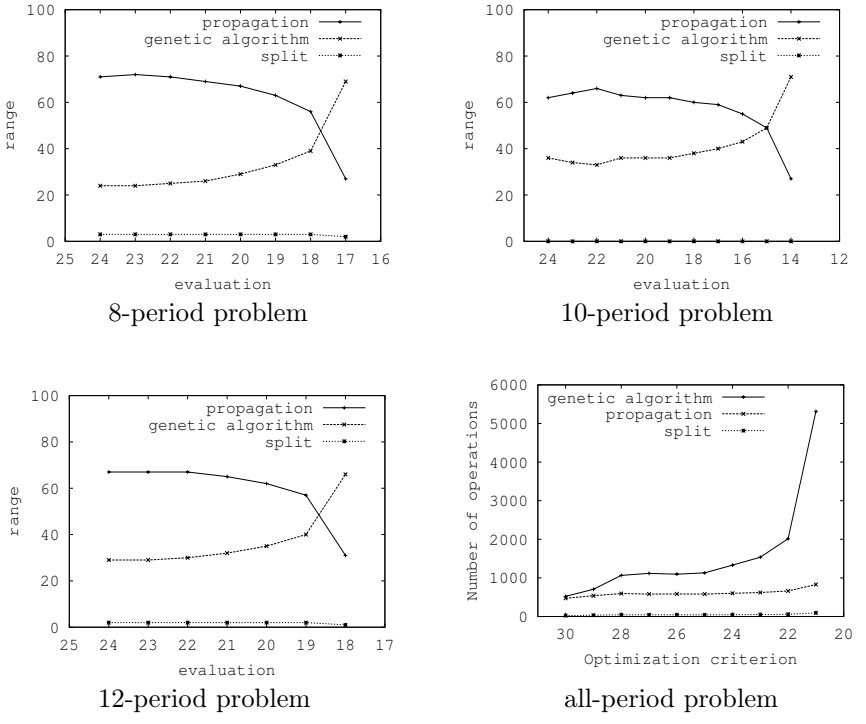


Fig. 5. Evolution of CP vs GA during the optimization process

and split. To evaluate the benefits of each of the components we have measured for CP: the number of effective reductions that are performed and the number of split and for GA: the fact that the next generation is globally better than the previous one.

Concerning the single problems (8, 10, 12), at the beginning, CP represents 70% of the effort: constraint propagation narrows the search space. On the contrary, GA represents about 30%. During this period, not enough local consistency is enforced by constraint propagation, and GA only finds solutions (satisfying all constraints) with a cost greater than 21. Then, at the beginning of the second half of the search process, in terms of costs, CP and GA converge: most of the sub CSP have reach the local consistency and tests over constraints do not improve domain reduction. At the end, GA performs 70% of the search effort to find the optimal solution.

Concerning strategies using GA **and** CP **alone**. In this implementation, CP is unable to find a feasible solution in 10 minutes cpu time. GA is able to find alone the optimal value but is 10 times slower w.r.t. the hybrid resolution $GA + CP$. Therefore, we have not included these results in the tables.

In the graph for the all-period problem, CP and GA start searching with the same efficiency but while CP seems to be stable, most of the operations are performed by the genetic process to obtain better solutions. This could be

explained by the fact that, in this problem, constraints are not strong enough w.r.t. the number of variables and the size of the generated search space. But, in our hybrid resolution system, GA appears as a powerful method even if most of the constraint operators have not reached their fixpoints.

5 Perspectives and Conclusion

Most of hybrid approaches are ad-hoc algorithms based on a master-slave combination: they favor the development of systems whose efficiency is strongly related to a given class of CSPs. In this paper, we have used a more suitable general framework to model hybrid optimization solving algorithms.

The results over the BACP show the benefits of our framework and of hybridization. They also allow us to identify the interaction between the different resolution mechanisms. Such studies could be used to tune general purpose hybrid solvers in the future.

A future extension will consist in providing "tools" to help designing finer strategies in the GI algorithm in our particularly suitable uniform framework. To this end, we plan to extend works of [11] where strategies are built using some composition operators in the GI algorithm. Moreover, this will also open possibilities of concurrent and parallel application of reduction functions inside the model.

References

1. E. Aarts and J. K. Lenstra. *Local Search in Combinatorial Optimization*. John Wiley & Sons, Inc., 1997.
2. K. R. Apt. From chaotic iteration to constraint propagation. In *Proceedings of ICALP'97*, pages 36–55. Springer-Verlag, 1997.
3. K. R. Apt. *Principles of Constraint Programming*. Cambridge Univ. Press, 2003.
4. N. Barnier and P. Brisset. Combine and conquer: Genetic algorithm and cp for optimization. In *Proc. of CP'98*, page 463, LNCS 1520, Springer, 1998.
5. E. K. Burke, D. Elliman, and R. F. Weare. A hybrid genetic algorithm for highly constrained timetabling problems. In *Proc. of ICGA*, pages 605–610, Morgan Kaufmann, 1995.
6. C. Castro and S. Manzano. Variable and value ordering when solving balanced academic curriculum problems. In *Proceedings of 6th Workshop of the ERCIM WG on Constraints. CoRR cs.PL/0110007*, 2001.
7. E. Monfroy and F. Saubion and T. Lambert. Hybrid CSP Solving. Proceeding of 5th International Workshop Frontiers of Combining Systems (FroCoS). LNCS 3717, Springer, 2005
8. F. Focacci, F. Laburthe, and A. Lodi. Local search and constraint programming. In *Handbook of Metaheuristics*, Kluwer Academic, 2002.
9. I. Gent, T. Walsh, and B. Selman. http://www.4c.ucc.ie/ tw/csplib/, funded by the UK Network of Constraints.
10. D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.

11. L. Granvilliers and E. Monfroy. Implementing constraint propagation by composition of reductions. In *Proc. of ICLP'2003*, *LNCS 2916*, pages 300–314. Springer, 2003.
12. J. H. Holland. *Adaptation in Natural and Artificial Systems*. 1975.
13. K. A. D. Jong. *An analysis of the behavior of a class of genetic adaptive systems.* Phd thesis, University of Michigan, 1975.
14. T. Lambert, E. Monfroy, and F. Saubion. Solving strategies using a hybridization model for local search and constraint propagation. In *Proceedings of ACM SAC'2005*. ACM, 2005.
15. E. Monfroy, F. Saubion, and T. Lambert. On hybridization of local search and constraint propagation. *Proc. of ICLP'04*, *LNCS 3132*, pp 299–313. Springer, 2004.

A Graph-Based Genetic Algorithm for the Multiple Sequence Alignment Problem

Heitor S. Lopes* and Guilherme L. Moritz

Bioinformatics Laboratory, CPGEI
Federal Technological University of Paraná
Av. 7 de setembro, 3165 80230-901 Curitiba (PR), Brazil
hslopes@pesquisador.cnpq.br, moritz@cpgei.cefetpr.br

Abstract. We developed a new approach for the multiple sequence alignment problem based on Genetic Algorithms (GA). A new method to represent an alignment is proposed as a multidimensional oriented graph, which dramatically decreases the storage complexity. Details of the proposed GA are explained, including new structure-preserving genetic operators. A sensitivity analysis was done for adjusting running parameters of the GA. Performance of the proposed system was evaluated using a benchmark of hand-aligned sequences (Balibase). Overall, the results obtained are comparable or better to those obtained by a well-known software (Clustal). These results are very promising and suggest more efforts for further developments.

1 Introduction

In biological systems, proteins are the most abundant and functionally diverse molecules and almost all vital processes depend on these macromolecules, which are composed by amino acids chains. The common 20 different types of amino acids can be combined in a linear sequence having the necessary information for the generation of a unique tri-dimensional structure. The comparison of two protein sequences (or a group of them) is known as alignment. It consists in the systematic comparison of the amino acids compounding the sequences throughout their whole extension (or only definite regions), and then computing a similarity score. From the computational viewpoint, the multiple sequence alignment (MSA) of proteins or DNA is a very difficult task and it was proved to be NP-complete [12]. However, alignment is the most important tool for discovering and representing similarities between sequences, and can unravel the evolutionary history, critical preserved motifs, and details of the tertiary structure or important clues about function. Therefore, MSA is a recurrent issue of extensive research in Bioinformatics and Computer Science, aiming at finding more efficient algorithms, as well as speeding-up existing ones [5,6].

* This work was partially supported by the Brazilian National Research Council – CNPQ, under grants 305720/04-0, 475049/03-9 and 402018/03-6.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 420–429, 2006.
© Springer-Verlag Berlin Heidelberg 2006

There are known computational algorithms that allow finding suitable alignments among sequences (local/global alignment; pairwise/multiple alignment). The main difference between them is the quality of the alignment. Frequently, algorithms that give good alignments are computationally expensive or even unfeasible for a large number of sequences. The optimum MSA can be obtained with multidimensional dynamic programming [9], but the time complexity is $O(2^N.L^N)$, and the memory complexity is $O(L^N)$, where N is the number of sequences and L the average length of the sequences. Therefore, this approach is unfeasible for large problems, and then, many heuristic methods have been proposed for this case, including Genetic Algorithms – GA (see, for instance, [10]). GA's are efficient heuristic methods for dealing with large search spaces and general optimization problems, especially when conventional methods fail. The objective of this work is to present a simple and efficient GA for multiple sequence alignment.

2 A Graph-Based Genetic Algorithm for MSA

The Needleman-Wunsch algorithm [9] requests a $n \times n$ matrix, where n is the number of sequences to be aligned. Once the matrix is constructed, a backtracking is done, starting from cell (n, n) and ending in cell $(1, 1)$. For each point of this path, it is necessary to discover which is the neighbor cell that was responsible for the generation of the current cell. If we consider this path in the matrix as a directed graph, a MSA problem can be modelled as a problem of finding the shortest path in a directed graph embedded in a n-dimensional space. In this case, the backtracking is substituted by a forward tracking, provided it is kept in each cell the information of which is its generating cell. The graph-based approach requires a vector with the dimension no larger than the sum of the length of all sequences. Therefore, the great advantage of this approach is the dramatic reduction of the memory complexity to solve the problem, estimated in $O(N.L)$.

The edges of the graph have weights corresponding to the evolutionary distances between between pairs of amino acids, given by: a substitution matrix (BLOSUM62) [7], a gap opening penalty and a gap extension penalty [1]. Figure 1a) shows the representation of the graph for the particular case of two proteins to be aligned. The way a graph is encoded in the GA is a key point in this work and detailed below.

The use of GA for real-world problems encompasses de definition of two basic issues: how problem's variables are encoded, and how candidate solutions are evaluated by a fitness function. These issues are approached in the next sections, as well as the generation of the initial population and the special genetic operators created for this problem. Regarding the selection method of the GA, we used the well-known stochastic tournament of size k. Elitism is also used in conjunction with the selection method. Two forms of elitism are defined. The first elitism selects the best individual of the current population and uses current the gap penalties (generation dependent). The other form of elitism selects the

5 best individuals of current generation, considering the SP when applying the final gap penalties. By default, both forms of elitism are enabled.

2.1 Chromosome Encoding

Let $P = p_1, p_2, \ldots, p_n$ be the set of n sequences to be aligned. Suppose that these sequences are in a n-dimensional matrix, in such a way that each dimension corresponds to a sequence. Sequences may have different lengths, and $\lambda(p_i)$ is the length of the i-th sequence. Each individual represents a path over the matrix, that is, a multidimensional directed graph. The maximum possible coordinates of the graph, corresponding to the dimensions of the matrix, are the length of the sequences. A valid path (graph) always begin in the cell corresponding to the first amino acid of all sequences, and ends in the cell corresponding to the last amino acid of all sequences.

To represent the weighted graph, we used a single chromosome of variable length. This chromosome is composed by m genes, where $\max_{i=1..n}[\lambda(p_i)] \leq m \leq [\sum_{i=1}^{n} \lambda(p_i) - 1]$. Each gene, in turn, is composed by a string of bits representing segments of the path in the graph. Recall that the graph will be translated into a MSA. To construct the alignment using the graph, we start with the first coordinate, and follow the steps of the graph until the last coordinate. For each step, a single column is defined in the MSA. The amino acid corresponding to the dimension to which the current step is advanced, is added to the column (of the alignment) that is under construction. For the remaining dimensions, gaps are added to the alignment.

Once a valid path starts in $\{0, 0, \ldots, 0\}$ and finishes in $\{\lambda(p_1), \lambda(p_2), \ldots \lambda(p_n)\}$, a chromosome will be composed of a set of genes that represent such path. Figure 1a shows an example of a matrix for the particular case of two sequences to be aligned, and a possible path in the matrix. Figure 1b shows the corresponding encoding of possible moves in a bi-dimensional matrix. Figure

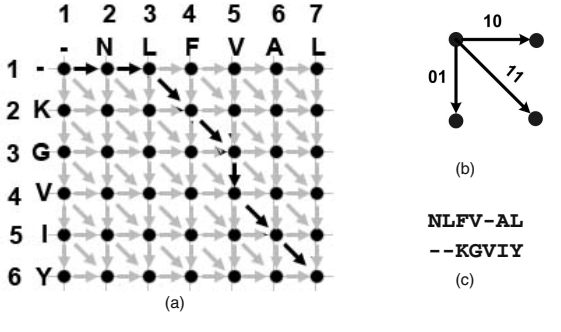


Fig. 1. (a) Example of the alignment matrix for two sequences. (b) Detail of the encoding, showing the possible moves in a 2-dimensional matrix. (c) Corresponding alignment of the two sequences represented by the graph in the matrix.

1c shows the final alignment represented by the path. For this example, the individual of the GA that encodes the path shown in the matrix corresponds to the binary vector (10 − 10 − 11 − 11 − 01 − 11 − 11). Only for the sake of clarity, genes are separated by hyphens. The number of bits per gene, that is, the dimension of the matrix, depends on the number of sequences to be aligned.

2.2 Fitness Function

To evaluate a candidate solution, a chromosome is decoded into a MSA. Next, two quality measures are computed, based on the classical sum-of-pairs (SP), defined in Equation 1:

$$SP = \sum_{k=1}^{m} \sum_{x=1}^{n-1} \sum_{y=x+1}^{n} D(aa_{x,k}, aa_{y,k}) \tag{1}$$

where: m = length of the alignment, n = number of sequences of the alignment, aa_{ij} = amino acid located in line i, column j of the alignment, $D(x, y)$ = evolutionary distance between amino acids x and y, according to a given substitution matrix.

The first quality measure is used in the selection procedure of the GA, and takes into account progressive gap penalties. Considering that substitution matrices can have negative values, the computed SP can be negative. Normalization is then necessary to make the fitness function always positive. To do so, we add the SP value computed for the worst individual of the first generation to the fitness of every other generated individual, in any generation. If, anytime a given individual have a negative value for its fitness, this value is set to zero. The second quality measure is used only for the elitism procedure of the GA, used in conjunction with the selection, and corresponds to SP, but uses fixed gap penalties.

2.3 Initial Population and Genetic Operators

Considering the complexity of the MSA problem and how individuals evolve throughout generations for this problem, the way the initial population is generated is of great concern to achieve good results. Hence, in the proposed GA, there are four different methods for generating the initial population, as follows:

- Progressive alignment: Sequences to be aligned are randomly selected for a pre-alignment using the dynamic programming method. Initially, sequences are aligned and, them, profiles are, in the same way that was proposed by [4] and also used in the profile alignment of Clustal [8]. First, two sequences are optimally aligned. Then, a third one is added to the other two, and so on, until all sequences are added. This procedure has a very high computational cost and, by default, it is not enabled.
- Pairwise alignment: Pairs of sequences are randomly selected and aligned using dynamic programming. After, these pairs of aligned sequences are simply

added to the MSA without any additional work. The added sequences are juxtaposed to the existing ones, simply by adding columns of gaps to make lengths coincide.
- Shuffling: Sequences are juxtaposed after a variable random number of gaps be inserted in the left side.
- Random generation: This is the default method. Individuals are randomly constructed, assuring that the corresponding graph meet the constraint of ending at position $\{\lambda(p_1), \lambda(p_2), \ldots \lambda(p_n)\}$, where the last element is the length of the n-th sequence (the n-th dimension of the graph).

Differently from other approaches in the literature where a large number of special genetic operators are defined [10], in this work we defined only three genetic operators for the problem: crossover, random mutation and dynamic mutation. Since chromosomes represent a path in a graph, our genetic operators were designed to preserve the integrity and validity of the path.

Crossover, or recombination, is applied to two chromosomes, in the way illustrated in Figure 2. First, a random gene is selected in both parents (dot point in Figure 2a,b, corresponding to coordinates (4,3)). All genes, from the first until the selected one are copied from parent-1 the offspring. If the remaining part of the chromosome was filled up with the equivalent part of parent-2, in most cases an invalid individual would be generated due to de discontinuity in the path. Hence, the following procedure is done. Parent-2 is scanned backwards, from the final coordinate to the first, aiming at finding the last gene of the chromosome (called U) whose coordinates in all dimensions are larger than the coordinate of the last gene of parent-1. Once found this gene, it is copied to the offspring. In the example of Figure 2b, it corresponds to gene 7, between coordinates (4,7) and (5,7). This gene is the first to be copied from parent-2 to the offspring. Now, the same procedure of random generation of the initial population is used, just to create some genes to fill up the gaps between the initial and final segments inherited from parent-1 and parent-2, respectively (as shown in Figure 2c).

The random mutation operator aims at improving the genetic diversity of the population and works as follows. Two genes and one dimension of the matrix are randomly selected. The bits corresponding to that dimension (a gene in the chromosome) are inverted. To guarantee that the path in the matrix is still

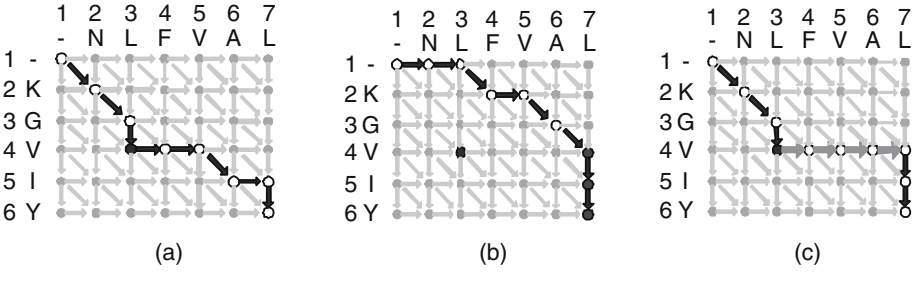


Fig. 2. Example of the crossover operator: (a) parent-1, (b)parent-2, (c) offspring

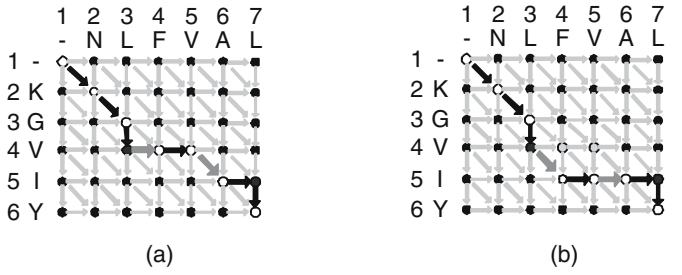


Fig. 3. Example of the random mutation operator: (a) parent, (b) offspring

valid after this operation, it is necessary that the two bits are complimentary. Otherwise, the second point selected is shifted to the right until it satisfies the continuity constraint. An example of the application of this operator is shown in Figure 3. In this example, genes 4 and 6 were random selected (Figure 3a). The dimension selected was the line (not the column). Hence, the bits of the selected dimension of the two genes satisfy the continuity constraint, and can be inverted so as to create a new individual shown in Figure 3b.

The objective of the dynamic mutation operator is the same as the previous one and it is illustrated in Figure 4 for the alignment of three proteins (P1, P2 and P3) with 22 amino acids each. First, a segment of the individual, comprising a number of genes, is randomly selected. In the figure this segment is the central region. Then, the columns of the alignment, corresponding to the selected segment, are found. To these columns, we apply a progressive dynamic alignment, using the same strategy used to initialize individuals, as mentioned in the beginning of this section. Therefore, besides the probability of application of this operator, it is also necessary to define the width of the segment to which the operator is applied to. This parameter is called WIDDM and for the example given, its value is 7. The lower lines of Figure 4 represent the protein sequences after the application of the dynamic mutation operator.

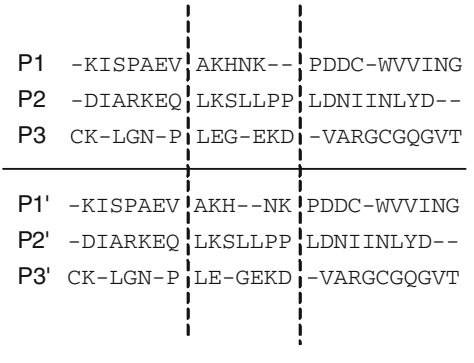


Fig. 4. Example of the dynamic mutation operator

2.4 Progressive Gap Penalties

Empirically, we observed that, case the gap penalties were too high, the GA could not evolve good individuals. This is a consequence of the fact that all individuals of the first generations have a large number of gaps. Therefore, despite of the genetic diversity of the population, most fitness values are too low to create a useful selective pressure. To circumvent this problem, we implemented a progressive gap penalty strategy. The effective value of a gap (either for opening a gap or extending a gap) is proportional to the generation number. By default, it starts with -4 for gap opening and -1 for gap extension, and is incremented until reaching -10 and -3, respectively, when reaching mature generations. As a consequence of this procedure, the same individual will have a different fitness, depending on the current generation it is considered, except for the case when it has no gaps. Therefore, there is a difference between the objective function (absolute value) and the fitness function (relative value).

3 Computational Experiments

3.1 Parameters Adjustment

Several preliminary tests were done to find suitable values for the control parameters of the GA. All tests were done using a PC computer with AMD Athlon XP 2.4 MHz and 512 Mbytes of RAM.

The sequences used in the experiments are from BaliBase 1.0 [11], a benchmark specially developed for the comparison of MSA algorithms. Alignments in BaliBase were manually refined, and regions that can be aligned without ambiguity are explicitly marked. A software is also provided with the database to compare results using specific metrics. BaliBase is composed by 142 reference alignments, with more than 1,000 sequences. These alignments are divided into four sets. These sets, in turn, are divided according to the length of the sequences (short, medium, long) and the identity between sequences (<25%, 20-40%, >35%).

For the adjustment of parameters, we used alignment $1ajsA$, from subgroup $Test-1$ of $Ref-1$. All experiments run 10 times, and results shown are the average value. For the adjustment of parameters, the probability of using the dynamic mutation operator was set to zero ($PROBDM = 0$). The remaining parameters were set as follows: number of individuals ($POP = 100$), number of generations ($GEN = 500$), initial gap open penalty $= -4$, initial gap extension penalty $= -1$, final gap open penalty $= -10$, final gap open extension $= -3$. Regarding to the generation of the initial population, the percent of individuals generated according each proposed method was: 20% by shuffling, 10% by pairwise alignment, 0% by progressive alignment and 70% by random generation.

A total of 15 experiments were done with all combinations of probability for crossover and random mutation in the ranges 0.6, 0.8, 1.0 and 0.05, 0.2, 0.4, 0.6, 1.0, respectively. There were no significant differences in the processing time for different values of the parameters tested. Therefore, the processing time was not

taken into account in the selection of the following best set values: 0.6 for the probability of crossover and 0.6 for the probability of random mutation.

Using the values of parameters previously defined, we varied the tourney size (K), number of generations (GEN) and population size (POP) in the ranges 3, 5, 8, 100, 200 and 500, 800, respectively. For this case, and the following, the processing time is strongly influenced by the specific value of parameters. Hence, the selection of default values for parameters must consider not only the quality of the alignment, given by the sum-of-pairs (SP), but also the processing time ($TIME$).This takes to a multiobjective optimization problem, for which the analysis of the Pareto front was necessary [3].

We observed no significant change in results when the tourney size was varied. Therefore, we set $K = 3$ individuals henceforth. It was also observed that better results were obtained using a larger population compared with a smaller population. However, it is not possible to conclude anything about the number of generations, since experiments suggested that more generations are necessary to stabilize the system. Consequently, a new set of tests were done, keeping $K = 3$ and expanding GEN to 800, 1300, 2000 and POP to 200, 300, 500. These experiments showed that beyond 1300 generations the gain of SP is minimal, but at the expense of a large increment in $TIME$. We also observed that the increment in POP has an important consequence in the value of SP, but reflects linearly in $TIME$. Hence, we adopted as default values a combination that gives a good performance in a fair processing time, that is: $GEN = 1300$ and $POP = 300$.

The dynamic mutation operator, is computationally expensive. This is why it was not used in the previous experiments. Using the default parameters defined so far, we tested the influence of $PROBDM$ and $WIDDM$ in the behavior of the GA, according to both SP and $TIME$. We tested $PROBDM$ and $WIDDM$ in the ranges 0.1, 0.3, 0.4 and 10, 50, respectively. The analysis of these results in the Pareto plane shown in Figure 5 suggested that $PROBMD$ has a small influence in SP, but has a direct relationship with $TIME$. Keeping fixed $PROBDM = 0.1$, further tests were done with $WIDMD$ in the range 20, 30, 80, 100. The default value of $WIDMD$ was set to 30, because it reflects the best compromise between SP and $TIME$.

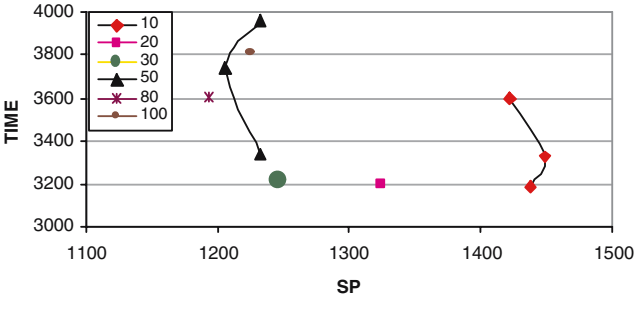


Fig. 5. Results of the variation of $PROBDM$ and $WIDDM$ in the Pareto plane

3.2 Performance Assessment

Using the default parameters previously defined, the proposed GA was tested with nine unseen instances of $Ref-1$ of BaliBase, including one reference alignment for each length and identity. For each instance, shown in Table 1, 10 runs were done and the average value of SP was compared with Clustal [8], a well-known software for MSA. However, to do such comparison, the alignment performed by Clustal was submitted to the same algorithm for computing SP, in order to assure a fair comparison.

The experiments shown in Table 1 cover all combinations of length of the sequences (short, medium, long) and identity between sequences (<25%, 20-40%, >35%). In this table, the leftmost column refers to the specific alignment of Reference-1 of the database, and "Rel.Dif." means how good is the result obtained by the GA relative to that of Clustal. A positive difference means that the GA performed better than Clustal, and a negative difference means the opposite. Recall that this comparison is relative only to SP. In fact, Clustal does not use sum-of-pairs as its internal quality measure. In this table, it can be observed that our GA performed better for short sequences than for the medium or the long ones. Regarding the identity of sequences, our GA gives better performances than Clustal for sequences with 20-40% or >35% of identity.

Table 1. SP obtained by the proposed GA and Clustal, for several instances of Balibase

Ref-1	Type			Identity			GA	Clustal	Rel.Dif.
	short	medium	long	<25%	20-40%	>35%			
1Abo	✓			✓			-212.5	-289.0	26.47%
1hpi	✓				✓		499.3	515.0	-3.05%
1fkj	✓					✓	1789.0	1725.0	3.71%
1sbp		✓		✓			-1537.5	1229.0	-25.10%
1ad2		✓			✓		1380.8	1276.0	-8.21%
1amk		✓				✓	5531.4	5846.0	-5.38%
1ajsA			✓	✓			687.6	1569.0	-56.17%
1ac5			✓		✓		-1246.9	-1258.0	0.88%
1ad3			✓			✓	5585.4	5652.0	-1.18%

4 Conclusions

We have presented an efficient genetic algorithm especially devised for the problem of multiple sequence alignment. The graph-based approach for modelling a possible alignment proposed in this work allows a dramatic reduction of the estimated memory complexity for solving the problem, when compared with the traditional multidimensional Needleman-Wunsch method [9]. We devised a simple fitness function, based on the sum-of-pairs, and three genetic operators to modify existing solutions. Other GA approaches in the literature have proposed much more complex operators and fitness function. A careful sensitivity analysis was done (detailed in section 3.1), so as to find near-optimal values for the

control parameters of the GA, trying to establish a fair compromise between the quality measure, and the processing time.

A number of tests were done using a benchmark database of alignments and our GA was compared with Clustal, a well-established MSA software. When analyzing results, two factors should be taken into account. First, we used a substitution matrix (BLOSUM62), which is a standard for ungapped matching and, possibly, it is not the most suited for the type of sequences we used here. Second, sum-of-pairs, although widely used and simple to compute, is recognized to present some problems for multiple sequence alignment, rather than pairwise alignment. Therefore, it is suggested to use a weighting scheme altogether (see [2]). In fact, a more suitable quality measure to compare multiple alignments should be devised. This is a well-known fact in Bioinformatics.

Notwithstanding, even considering these issues and the stochastic nature of GA, overall results can be considered quite satisfactory. The methodology is promising and future work will include refinements in the algorithm, moving towards distributed processing [6], as well as more experiments with other benchmarks.

References

1. Altschul, S.F.: Gap costs for multiple sequence alignment. J. Theor. Biol. **138** (1989) 297-309.
2. Altschul, S.F., Carrol, R.J. and Lipman, D.J.: Weights for data related by a tree. J. Mol. Biol. **207** (1989) 647-653.
3. Deb, K.: Multi-objective Optimization using Evolutionary Algorithms. Wiley Interscience, New York (2001).
4. Feng, D.-F. and Doolitle, R.F.: Progressive sequence alignment as a prerequisite to correct phylogenetic trees. J. Mol. Evol. **25** (1987) 351-360.
5. Leach, A.R. Molecular Modelling: Principles and Applications, 2_{nd} ed. Prentice-Hall, Dorset (2001).
6. Lopes, H.S. and Moritz, G.L.: A distributed approach for multiple sequence alignment using a parallel virtual machine. In: Proc. 27th Annual International Conference of IEEE EMBS, Shanghai, China, 2005.
7. Henikoff, S. and Henikoff, J.G.: Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. USA **89** (1992) 10915-10919.
8. Higgins, D.G. and Sharp, P.M.: CLUSTAL: a package for performing multiple sequence alignments on a microcomputer. Gene **73** (1988) 237-244.
9. Needleman, S.B. and Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. **48** (1970) 443-453.
10. Notredame,C. and Higgins, D.G.: SAGA: sequence alignment by genetic algorithm. Nucl. Acid. Res. **24** (1996) 1515-1524.
11. Thompson, J., Plewniak, F. and Poch, O.: BAliBASE: A benchmark alignments database for the evaluation of multiple sequence alignment program. Bioinformatics **15** (1999) 87-88.
12. Wang, L. and Jiang, T.: On the complexity of multiple sequence alignment. J. Comp. Biol. **1** (1994) 337-348.

Improved Multi-Objective Diversity Control Oriented Genetic Algorithm

Theera Piroonratana[1] and Nachol Chaiyaratana[2]

[1] Department of Production Engineering,
King Mongkut's Institute of Technology North Bangkok,
1518 Piboolsongkram Road, Bangsue, Bangkok 10800, Thailand
theerapi@yahoo.com
[2] Research and Development Center for Intelligent Systems,
King Mongkut's Institute of Technology North Bangkok,
1518 Piboolsongkram Road, Bangsue, Bangkok 10800, Thailand
nchl@kmitnb.ac.th

Abstract. This paper presents an improved multi-objective diversity control oriented genetic algorithm (MODCGA-II). The improvement includes the introduction of an objective-domain diversity control operator, which is chromosome representation independent, and a solution archive. The performance comparison between the MODCGA-II, a non-dominated sorting genetic algorithm II (NSGA-II) and an improved strength Pareto evolutionary algorithm (SPEA-II) is carried out where different two-objective benchmark problems with specific multi-objective characteristics are utilised. The results indicate that the MODCGA-II solutions are better than the solutions generated by the NSGA-II and SPEA-II in terms of the closeness to the true Pareto optimal solutions and the uniformity of solution distribution along the Pareto front.

1 Introduction

It is undeniable that a major factor that contributes to the success of genetic algorithms in the field of optimisation is the parallel search mechanism embedded in the algorithm itself. However, this does not prevent the occurrence of premature convergence in the situation when the similarity among individuals in the population becomes too high. As a result, the prevention of a premature convergence must also be considered during the genetic algorithm design. One of the direct approaches for achieving the necessary prevention is to maintain diversity within the population [1].

Various strategies can be used to maintain or increase the population diversity. Nonetheless, a modification on the selection operation has received much attention. For instance, Mori et al. [2] has introduced a notion of thermodynamical genetic algorithm where the survival of individuals is regulated by means of monitoring the free energy within the population. The modification on the selection operation can also be done in the cross-generational sense [3,4,5]. Whitley [3] has proposed a GENITOR system where offspring generated by standard operators are chosen for replacing parents based upon the ranks of the individuals.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 430–439, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In contrast to Whitley [3], Eshelman [4] recommends the application of mating restriction while Shimodaira [5] suggests the use of variable-rate mutation as a means to create offspring. Then a cross-generational survival selection is carried out using a standard fitness-based selection technique in both cases.

In addition to the early works described above, another genetic algorithm has been specifically developed by Shimodaira [6] to handle the issue of population diversity; this algorithm is called a diversity control oriented genetic algorithm or DCGA. Similar to most genetic algorithms, offspring in the DCGA are generated using standard crossover and mutation operators. However, during the cross-generational survival selection, duplicated individuals in the merged population containing both parent and offspring individuals are first eliminated. Then, the remaining individuals are selected based on either the associate fitness or the consideration on both the fitness and the genomic similarity between the interested individual and the elite individual. The performance of the DCGA has been benchmarked using various test problems [7].

With a minor modification, the DCGA can also be used in multi-objective optimisation. One possible approach for achieving this is to integrate the DCGA with other genetic algorithms that are specifically designed for multi-objective optimisation such as a multi-objective genetic algorithm or MOGA [8]. Such approach has been investigated by Sangkawelert and Chaiyaratana [9] where the inclusion of cross-generational survival selection with the multi-objective genetic algorithm is equivalent to the use of elitism, which is proven to be crucial to the success of various multi-objective algorithms including a non-dominated sorting genetic algorithm II or NSGA-II [10] and an improved strength Pareto evolutionary algorithm or SPEA-II [11]. In addition, the similarity measurement between the non-elite individual and the elite individual required by the diversity control operator is still carried out in the genotypic space. The resulting combined algorithm, which can be uniquely referred to as a multi-objective diversity control oriented genetic algorithm or MODCGA has been successfully tested using a two-objective benchmark suite [12]. Although some insights into the behaviour of the MODCGA have been gained through the benchmark trial by Sangkawelert and Chaiyaratana [9], further studies can be made and are required. In particular, the initial study of the MODCGA is conducted with a similarity measurement between two individuals being carried out in genotypic space. However, in multi-objective optimisation the trade-off surface, which is the direct result from the spread of solutions, is generally defined in objective space. This means that diversity control can also be achieved by considering the similarity between objective vectors of the individuals.

The organisation of this paper is as follows. In section 2, the explanation of the original DCGA and how it can be modified to cope with multi-objective optimisation problems is given. In section 3, the multi-objective benchmark problems and performance evaluation criteria are explained. Next, the multi-objective benchmarking results of the improved MODCGA or MODCGA-II are illustrated and discussed in section 4. Finally, the conclusions are drawn in section 5.

2 DCGA and Its Extension

The original DCGA developed by Shimodaira [6] can only be used to solve single-objective optimisation problems. However, the algorithm can be easily combined with other genetic algorithms. The extension of the DCGA for use in multi-objective optimisation that will be used throughout this paper involves the integration between the DCGA and MOGA. However, in contrast to the previous work by Sangkawelert and Chaiyaratana [9] where the similarity measurement between individuals is conducted in genotypic space, in this work the measurement will be carried out in objective space. Detailed explanation of the DCGA, MOGA and algorithm integration is given as follows.

2.1 Diversity Control Oriented Genetic Algorithm

The diversity control oriented genetic algorithm (DCGA) was first introduced by Shimodaira [6]. Similar to other single-objective steady-state genetic algorithms, the parent population and the offspring population are merged together during the DCGA run where the appropriated individuals are extracted from the merged population. However, instead of selecting the highly fit individuals from the population straightaway, the extraction process in the DCGA starts with the elimination of duplicated individuals in the merged population. The remaining individuals are then sorted according to their fitness values in descending order. Following that the best individual from the remaining individuals is determined and kept for passing onto the next generation. Then either a cross-generational deterministic survival selection (CDSS) method or a cross-generational probabilistic survival selection (CPSS) method is applied in the top-down fashion to the remaining non-elite individuals in the sorted array. In the case of the CDSS, the remaining non-elite individuals with high fitness value will have a higher chance of being selected since they reside in the top part of the array and hence have a higher selection priority than individuals with low fitness values. In contrast, a survival probability value is assigned to each non-elite individual according to its similarity to the best individual in the case of the CPSS. This survival probability (p_s) is given by

$$p_s = \{(1 - c)d_h/L + c\}^{\alpha} \tag{1}$$

where d_h is the Hamming distance between the interested individual and the best individual, L is the binary chromosome length, c is the shape coefficient and α is the exponent coefficient. With this form of survival probability assignment, if the genomic structure of the individual interested is very close to that of the best individual, the survival probability assigned to this individual will be close to zero. On the other hand, if the chromosome structure of this individual is quite different from that of the best individual, its survival probability will be close to one. Each individual will then be selected according to the assigned survival probability where the survival selection of the sorted non-elite individuals is still carried out in the top-down manner. With the use of sorted

individual array, the selection of non-elite individuals will depend entirely on the assigned fitness in the CDSS scheme. On the other hand, a decision either to select or not to select an individual according to the CPSS scheme depends on both the assigned fitness and the survival probability. Basically, the individual that have a high chance of being selected must possess high fitness and have a genomic structure that is quite different from that of the best individual. This also means that both a highly fit individual that is quite resemble to the best individual and a mediocre individual that is different from the best individual would not have a high chance of being picked. If the total number of all selected individuals including the pre-selected elite individual does not reach the required population size after the survival selection loop, randomly generated individuals will be added to the individual array until the required number is met. A comprehensive description of the DCGA and its benchmarking performance in various continuous test problems can be found in Shimodaira [7].

2.2 Multi-Objective Genetic Algorithm

The multi-objective genetic algorithm (MOGA) was first introduced by Fonseca and Fleming [8]. The MOGA functions by seeking to optimise the components of a vector-valued objective function. Unlike single-objective optimisation, the solution to a multi-objective optimisation problem is a family of points known as the Pareto optimal set. Each point in the set is optimal in the sense that no improvement can be achieved in one component of the objective vector that does not lead to degradation in at least one of the remaining components. Given a set of possible solutions, a candidate solution is said to be Pareto optimal if there are no other solutions in the solution set that can dominate the candidate solution. In other words, the candidate solution would be a non-dominated solution. Assuming, without loss of generality, a minimisation problem, an m-dimensional cost vector $\mathbf{u}$ is said to be dominating another m-dimensional cost vector $\mathbf{v}$ if, and only if, $\mathbf{u}$ is partially less than $\mathbf{v}$ ($\mathbf{u}\, p < \mathbf{v}$), i.e.

$$\mathbf{u}\, p < \mathbf{v} \leftrightarrow \forall i = 1, \ldots, m : u_i \leq v_i \wedge \exists i = 1, \ldots, m : u_i < v_i. \tag{2}$$

By identifying the number of solutions in the solution set that dominate the solution of interest, a rank value can be assigned to the solution. In other words, the rank of a candidate solution is given by the number of solutions in the solution set that dominate the candidate solution. After a rank has been assigned to each solution, a fitness value can then be interpolated onto the solution where a genetic algorithm can subsequently be applied in the optimisation procedure. Since the aim of a search by the MOGA is to locate Pareto optimal solutions, in essence the multi-objective optimisation problem has also been treated as a multi-modal problem. Hence, the use of additional genetic operators including the fitness sharing and mating restriction procedures is also required. However, in addition to the usual application of the fitness sharing and mating restriction procedures in the decision variable space, they can also be carried out in the objective space. A comprehensive description of the MOGA, which covers other

advanced topics including goal attainment and priority assignment strategies, can be found in Fonseca and Fleming [8].

2.3 Genetic Algorithm Integration

By combining the MOGA and the DCGA together, the resulting algorithm can be referred to as a multi-objective diversity control oriented genetic algorithm or MODCGA. Similar to the MOGA, the rank of each individual will be obtained after comparing it with the remaining individuals. However, the comparison will be made among individuals in the merged population, which is the result from combining parent and offspring populations together. Since the best individuals in the MOGA are the non-dominated individuals, in the case where the CPSS method is used there will be more than one survival probability value that can be assigned to each dominated individual. In this study, the lowest value in the probability value set is chosen for each dominated individual. After the survival selection routine is completed and the fitness values have been interpolated onto the individuals, the standard genetic operations can then be applied to the population in the usual way. In the early work by Sangkawelert and Chaiyaratana [9], a similarity measurement between dominated and non-dominated individuals, which leads to the survival probability assignment, is carried out in genotypic space. In this work, the similarity measurement will be conducted in objective space instead; two advantages are gained through this modification. Firstly, since the aim of multi-objective optimisation is to obtain multiple solutions at which together produce a trade-off objective surface that represents a Pareto front, diversity control in objective space would directly enforce this aim. Secondly, a diversity control operator that is designed for use in objective space would be independent of the chromosome encoding scheme utilised. Recent investigation into multi-objective optimisation using genetic algorithms usually involves problems with large number of decision variables [10,11,12]. The use of a binary representation would lead to an excessively long chromosome and hence degrades the algorithm performance. As a result, real-value chromosome encoding is generally employed instead. With the modification described above, in the case of CPSS scheme the survival probability as given in equation (1) will change to

$$p_s = \{(1 - c)d/d_{max} + c\}^{\alpha} \tag{3}$$

where d is the distance between the interested individual and a non-dominated individual in objective space and d_{max} is the maximum distance between two individuals in the population. In order to distinguish between the early work by Sangkawelert and Chaiyaratana [9] and the present work, the MODCGA where the similarity measurement is done in objective space will be referred throughout this paper as the MODCGA-II. In this investigation, the fitness sharing strategy utilised in the MODCGA-II is similar to the one described in Fonseca and Fleming [8] where the fitness sharing is carried out in objective space.

In addition to the modification on the diversity control operation, the use of a preserved non-dominated solution archive is included in the MODCGA-II. Basi-

cally, the parent individuals will be picked from a population which includes both individuals obtained after the diversity control and that from the archive. Each time that a new population is created after the diversity control operation, non-dominated solutions within the archive will be updated. If the solution that survives the diversity control operation is neither dominated by any solutions in the archive nor a duplicate of a solution in the archive, then this solution will be added to the archive. At the same time, if the solution that survives the diversity control operation dominates any existing solution in the archive, the dominated solution will be expunged from the archive. In order to maintain the diversity within the preserved non-dominated solution archive, k-nearest neighbour clustering technique [11] is used to regulate the size of the archive.

3 Multi-objective Problems and Performance Criteria

The MODCGA-II will be benchmarked using six optimisation test cases developed by Deb et al. [13]. The problems DTLZ1–DTLZ6 are scalable minimisation problems with n decision variables and m objectives. In this paper, two-objective problems with 11 decision variables are investigated. DTLZ1 has a linear Pareto front and contains multiple local fronts. DTLZ2 has a spherical Pareto front. DTLZ3 and DTLZ4 also have spherical Pareto fronts where DTLZ3 contains multiple local fronts while the DTLZ4 solutions are non-uniformly distributed in the search space. DTLZ5 has a curve Pareto front. DTLZ6 also has a curve Pareto front but the problem contains multiple local fronts.

Zitzler et al. [12] suggest that to assess the optimality of non-dominated solutions identified by a multi-objective optimisation algorithm, these solutions should be compared among themselves and with the true Pareto optimal solutions. Two corresponding measurement criteria are considered: the average distance between the non-dominated solutions to the Pareto optimal solutions (M_1) and the distribution of the non-dominated solutions (M_2). These criteria are calculated from the objective vectors of the solutions obtained. A low M_1 value implies that the solutions are close to the true Pareto optimal solutions. In addition, when two solution sets have similar M_1 indices, the set with a higher M_2 value would have a better distribution.

4 Results and Discussions

In this section, the results from using the MODCGA-II to solve test problems DTLZ1–DTLZ6 will be presented. The results will be benchmarked against that obtained from the non-dominated sorting genetic algorithm II or NSGA-II [10] and the improved strength Pareto evolutionary algorithm or SPEA-II [11] where the executable codes for the implementation of both algorithms are obtained directly from A Platform and Programming Language Independent Interface for Search Algorithms (PISA) web site (`http://www.tik.ee.ethz.ch/pisa`). Both CDSS and CPSS techniques are utilised in the implementation of the MODCGA-II. The diversity control study will be conducted with other genetic parameters

Table 1. Parameter setting for the MODCGA-II, NSGA-II and SPEA-II

Parameter	Value and Setting
Chromosome coding	Real-value representation
Fitness sharing	Triangular sharing function (MODCGA-II only)
Fitness assignment	Linear fitness interpolation (MODCGA-II only)
Selection method	Stochastic universal sampling (MODCGA-II) or tournament selection (NSGA-II and SPEA-II)
Crossover method	SBX recombination with probability = 1.0 [14]
Mutation method	Variable-wise polynomial mutation with probability = 1/number of decision variables [14]
Population size	100
Archive size	100 (MODCGA-II and SPEA-II only)
Number of generations	300 (MODCGA-II) or 600 (NSGA-II and SPEA-II)
Number of repeated runs	30

remain fixed throughout the trial. The parameter setting for the MODCGA-II, NSGA-II and SPEA-II that is used in all problems is displayed in Table 1.

Five values of the shape coefficient (c)—0.0, 0.25, 0.50, 0.75 and 1.00—and six values of the exponent coefficient (α)—0.00, 0.20, 0.40, 0.60, 0.80 and 1.00—are used to create 30 different diversity control settings for the MODCGA-II. From equation (3), the settings of $c = 1.00$ and $\alpha = 0.00$ are for the implementation of the CDSS technique since the survival probability of each dominated individual is equal to one. For each setting, the MODCGA-II runs for the DTLZ1–DTLZ6 problems with two objectives are repeated 30 times. The M_1 and M_2 performance indices from each run are subsequently obtained and the average values of the two indices calculated from all problems are displayed in the form of contour plots in Fig. 1. The M_2 index is calculated using the neighbourhood parameter $\sigma = 0.488$ and the M_2 index has been normalised by the maximum attainable number of non-dominated individuals from a single run. From Fig. 1, it is noticeable that a significant performance variation can be detected in the benchmark problems. The region where the M_1 index has a small value coincides with the area where the M_2 index is small. At the same time the region where the M_1 index is high is also in the vicinity of the area where the M_2 index has a large value. In a successful multi-objective search, the M_1 index should be as small as possible. Although a large M_2 index usually signifies a good solution distribution, the interpretation of the M_2 result must always be done while taken the M_1 index into consideration. This is because in the case where the solutions are further away from the true Pareto optimal solutions, the obtained value of the M_2 index is generally high since each solution would also be far apart from one another. In other words, the M_2 index has a lesser priority than the M_1 index and should be considered only when the obtained values of the M_1 index from two different algorithms or algorithm settings are close to one another. Using the above argument, multiple settings of the c and α values in Fig. 1 can be used to achieve low M_1 indices. In the current investigation, the setting where

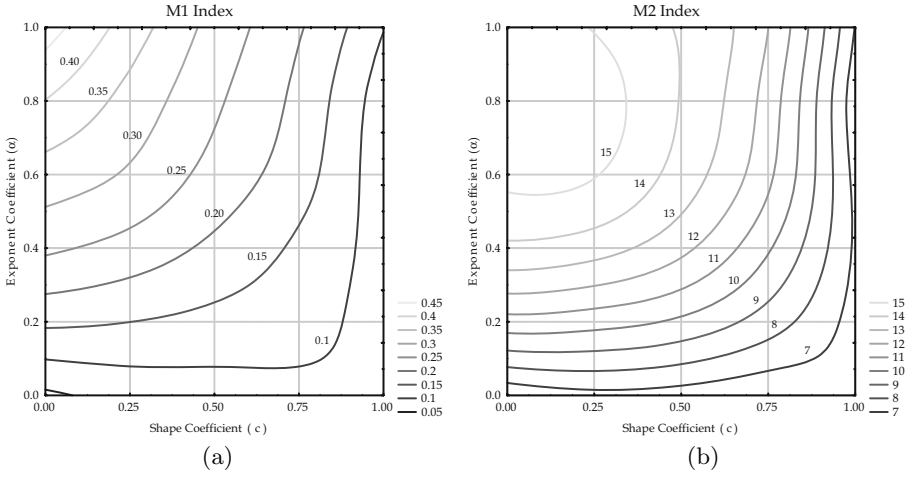


Fig. 1. Average values of M_1 and M_2 indices from all multi-objective problems for each diversity control setting (a) M_1 index (b) M_2 index

$c = 0.75$ and $\alpha = 0.2$ is chosen as the candidate setting that represents the diversity control that leads to a low M_1 value.

The search performance of the MODCGA-II with $c = 0.75$ and $\alpha = 0.2$ will be compared with that from the NSGA-II and SPEA-II. As stated in Table 1, each algorithm run will be repeated 30 times where the M_1 and normalised M_2 indices are subsequently calculated for each repeated run. After all repeated runs are completed, the individuals from all runs are merged together where the final non-dominated individuals are then extracted. The performance of the MODCGA-II, NSGA-II and SPEA-II in terms of the average and standard deviation of the M_1 and normalised M_2 indices on the two-objective DTLZ1–DTLZ6 problems is summarised in Table 2. In Table 2, the neighbourhood parameter (σ) for the calculation of M_2 indices for all test problems is also set to 0.488; the parameter is set using the extent of the true Pareto front in the objective space as the guideline.

In terms of the average distance from the non-dominated solutions identified to the true Pareto front or the M_1 criterion, the MODCGA-II posses the highest performance in all six test problems. Nonetheless, the MODCGA-II is unable to identify the true Pareto optimal solutions in the DTLZ3 and DTLZ6 problems. These two problems are difficult to solve since they contains multiple local Pareto fronts. Although the DTLZ1 problem also contains numerous local Pareto fronts, the majority of results from all 30 MODCGA-II runs indicate that the MODCGA-II is capable solving this problem. This means that the shape of Pareto front in two-objective problems can also affect the algorithm performance since the DTLZ1 problem has a linear Pareto front while the DTLZ3 and DTLZ6 problems have spherical and curve Pareto fronts, respectively. The M_1 index also reveals that the performance of NSGA-II and SPEA-II are very similar in all six problems. Since the M_1 indices from both algorithms are quite

Table 2. Summary of the MODCGA-II, NSGA-II and SPEA-II performances on the two-objective DTLZ1–DTLZ6 problems

Problem	Index	MODCGA-II		NSGA-II		SPEA-II	
		Average	S.D.	Average	S.D.	Average	S.D.
DTLZ1	M_1	3.1157	4.7837	11.9186	5.1490	12.9616	5.2649
	M_2	0.4326	0.2942	0.6391	0.0474	0.7810	0.0547
DTLZ2	M_1	0.0030	0.0008	0.0148	0.0088	0.0190	0.0096
	M_2	0.5039	0.0439	0.5672	0.0310	0.5053	0.0494
DTLZ3	M_1	22.2335	18.1880	78.6069	24.8055	88.4823	22.4487
	M_2	0.5642	0.3941	0.6119	0.0814	0.7463	0.0745
DTLZ4	M_1	0.0023	0.0018	0.0238	0.0138	0.0252	0.0104
	M_2	0.3353	0.2111	0.2871	0.2353	0.3457	0.2417
DTLZ5	M_1	0.0030	0.0006	0.0148	0.0088	0.0175	0.0079
	M_2	0.4972	0.0495	0.5672	0.0310	0.5026	0.0501
DTLZ6	M_1	1.0199	0.3685	6.4295	0.3509	6.4986	0.3355
	M_2	0.8044	0.0603	0.7157	0.0493	0.8946	0.0166

close, a further inspection on the M_2 indices can be easily made. Again, the M_2 indices from the NSGA-II and SPEA-II are also very close to one another. This leads to the conclusion that the capability of both the NSGA-II and SPEA-II is similar.

5 Conclusions

In this paper, an improved multi-objective diversity control oriented genetic algorithm or MODCGA-II is presented. The proposed algorithm differs from the MODCGA described in Sangkawelert and Chaiyaratana [9] in the sense that the MODCGA-II performs diversity control via similarity measurement in objective space and the use of a preserved non-dominated solution archive is also included. Six scalable benchmark problems described in Deb et al. [13] are utilised. In addition, the criteria used to assess the algorithm performance include the closeness of non-dominated solutions to the true Pareto front and the distribution of the solutions across the front [12]. The analysis indicates that the MODCGA-II can produce non-dominated solutions that are better than that generated by the NSGA-II [10] and the SPEA-II [11] when the number of objectives in the benchmark problems is limited to two.

Acknowledgements

This work was supported by a Research Career Development Grant from the Thailand Research Fund (Grant Number: RSA4880001).

References

1. Mauldin, M.L.: Maintaining diversity in genetic search. In: Proceedings of the National Conference on Artificial Intelligence, Austin, TX (1984) 247–250
2. Mori, N., Yoshida, J., Tamaki, H., Kita, H., Nishikawa, Y.: A thermodynamical selection rule for the genetic algorithm. In: Proceedings of the Second IEEE International Conference on Evolutionary Computation, Perth, WA (1995) 188–192
3. Whitley, D.: The GENITOR algorithm and selection pressure: Why rank-based allocation of reproduction trials is best. In: Proceedings of the Third International Conference on Genetic Algorithms, Fairfax, VA (1989) 116–121
4. Eshelman, L.J.: The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In: Rawlins, G.J.E. (ed.): Foundations of Genetic Algorithms, Vol. 1. Morgan Kaufmann, San Mateo, CA (1991) 265–283
5. Shimodaira, H.: A new genetic algorithm using large mutation rates and population-elitist selection (GALME). In: Proceedings of the Eighth IEEE International Conference on Tools with Artificial Intelligence, Toulouse, France (1996) 25–32
6. Shimodaira, H.: DCGA: A diversity control oriented genetic algorithm. In: Proceedings of the Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, Glasgow, UK (1997) 444–449
7. Shimodaira, H.: A diversity-control-oriented genetic algorithm (DCGA): Performance in function optimization. In: Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, Korea (2001) 44–51
8. Fonseca, C.M., Fleming, P.J.: Multiobjective optimization and multiple constraint handling with evolutionary algorithms–Part 1: A unified formulation. IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans **28**(1) (1998) 26–37
9. Sangkawelert, N., Chaiyaratana, N.: Diversity control in a multi-objective genetic algorithm. In: Proceedings of the 2003 Congress on Evolutionary Computation, Canberra, Australia (2003) 2704–2711
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2) (2002) 182–197
11. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, K., Tsahalis, D., Periaux, J., Papailiou, K., Fogarty, T. (eds.): Evolutionary Methods for Design, Optimisation and Control. International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain (2002) 95–100
12. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation **8**(2) (2000) 173–195
13. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multi-objective optimization. In: Abraham, A., Jain, L.C., Goldberg, R. (eds.): Evolutionary Multiobjective Optimization: Theoretical Advances and Applications. Springer, London, UK (2005) 105–145
14. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms. Wiley, Chichester, UK (2001)

Directional Distributions and Their Application to Evolutionary Algorithms

Przemysław Prętki and Andrzej Obuchowicz

Institute of Control and Computation Engineering
University of Zielona Góra,
ul. Podgórna 50, 65–246 Zielona Góra, Poland
P.Pretki@issi.uz.zgora.pl
A.Obuchowicz@issi.uz.zgora.pl

Abstract. In this paper, a concept of directional mutations for phenotypic evolutionary algorithms is presented. The proposed approach allows, in a very convenient way, to adapt the probability measure underlying the mutation operator during evolutionary process. Moreover, the paper provides some guidance, along with suitable theorems, which makes it possible to get a deeper understanding of the ineffectiveness of isotropic mutations for large-scale problems.

1 Introduction

Many stochastic optimization algorithms apply isotropic random vectors to produce new candidate solutions [5,8]. Spherically symmetric distributions guarantee that there is no preferable direction in the search space, which is a desired property especially at the beginning of the optimization process. Moreover, this also means that the effectiveness of an optimization technique does not depend on a reference frame. This, in the evolutionary algorithms (EAs) framework was intensively studied by Obuchowicz [8]. An isotropic sampling strategy may be effective for the low-dimensional problems, but its efficiency may drastically decreases for large-scale problems. Thus, many evolutionary algorithms are supplied with auxiliary heuristics which try to neutralize a negative influence of the dimensionality problem [5,8,11]. An intuitive idea of dealing with the problem is to adjust a probability measure on the basis of information gained during the optimization process. In this paper, a special class of the so-called directional distributions is introduced. This new class gives an access to constructing new techniques for adaptation probability measures in the mutation operator.

The paper is organized as follows. In the first section the reason why isotropic mutations are doomed to failure for high-dimensional problems is clearly explained. In the second section the concept of directional distributions with rotational symmetry is introduced. The third section contains the definition and the most important characteristics of directional distributions. In forth section some well-known techniques for estimating the most preferable direction of mutation

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 440–449, 2006.
© Springer-Verlag Berlin Heidelberg 2006

are briefly presented. The fifth section contains comparative studies regarding effectiveness of different heuristics based on the directional distribution. Finally, the last section concludes the paper.

2 Isotropic Mutation and High-Dimensional Problems

Evolutionary algorithms dedicated to continuous optimization problems very often favour an isotropic distribution in the mutation operator. Even though this class of multivariate distributions guaranty a *fair* searching way, it may be extremely ineffective for high-dimensional problems. In this section it is looked forward a source of the above-mentioned phenomenon.

It can be proved [4] that every spherically symmetric random vector $\boldsymbol{Z}$ can be decomposed into

$$\boldsymbol{Z} = r\boldsymbol{u}^{(n)}, \tag{1}$$

where $\boldsymbol{u}^{(n)}$ is uniformly distributed on the n-dimensional unit sphere, and $r > 0$ stands for the so-called generating variate. Moreover, the relationship between r and $\boldsymbol{Z}$ is one-to-one, what implies, that the number of spherical distributions is essentially equal to the number of nonnegative random variable [4]. It is worth noticing that the random variable $\boldsymbol{u}^{(n)}$ possesses the greatest possible entropy among all distributions defined on the unit sphere.

The impair of the effectiveness of EAs with spherically symmetric mutations can be explained with the help of the following theorem:

Theorem 1. *Let us consider any spherically distributed random vector $\boldsymbol{Z}$, and a new candidate solution formed in the following way $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{Z}$. Moreover, let $\boldsymbol{\mu}_t$ denotes the most profitable direction in the search space at point $\boldsymbol{x}_t$. Then, the probability that the candidate solution $\boldsymbol{x}_{t+1}$, will lie on the direction perpendicular to $\boldsymbol{\mu}_t$, tends to one when the dimension of the search space increases to the infinity.*

Proof. Let $\boldsymbol{\mu} \in \mathbb{R}^n$ be an unit vector, and $\boldsymbol{u}^{(n)}$ denote a random vector uniformly distributed on the n-dimensional unit sphere. Then the distribution of the random variate $t = \boldsymbol{\mu}^T \boldsymbol{u}^{(n)}$ has a density given as [4]:

$$f_n(t) = \frac{1}{\beta(\frac{n-1}{2}, \frac{1}{2})}(1-t^2)^{\frac{n-3}{2}}, \tag{2}$$

where $\beta(\cdot,\cdot)$ stands for the Beta function. Let us consider the following limit for $t \in (-1,0) \cup (0,1)$

$$\lim_{n\to\infty} f_n(t) = \pi^{-1/2} \lim_{n\to\infty} \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})}(1-t^2)^{\frac{n-3}{2}}. \tag{3}$$

Substituting $k = 2(n+1)$, and $b = (1-t^2) \in (0,1)$, (3) can be written as:

$$\lim_{n\to\infty} f_n(t) = \lim_{k\to\infty} \frac{\Gamma(k+1)}{\Gamma(k+1/2)} b^{k-1/2}. \tag{4}$$

Next, let us notice that

$$a_k = 0 \leq \frac{\Gamma(k+1)}{\Gamma(k+1/2)} b^{k-1/2} < (k+1/2)^{1/2} b^{k-1/2} \leq (k+1/2) b^{k-1/2} = c_k.$$

The limit of the sequence c_k can be obtained with help of L'Hospital's rule:

$$\lim_{k\to\infty} c_k = \lim_{k\to\infty} \frac{k+1/2}{b^{1/2-k}} = \lim_{k\to\infty} \frac{1/2}{-\ln(b) b^{1/2-k}} = -\frac{b^{-1/2}}{2\ln(b)} \lim_{k\to\infty} b^k = 0$$

Since both sequences a_k and c_k converge to 0, thus the sequence d_k also approaches 0. Therefore, the density function (2) tends to 0 for every $t \in [-1, 0) \cup (0, 1]$ and goes to infinity for $t = 0$. Thus, it is obvious that, in the limit, the whole probability mass is focused in the point $t = 0$, which means that the functions' sequence $\{f_n\}$ approximates the Dirac Delta Distribution. Now, if we assume that the vector $\boldsymbol{\mu}$ stands for the most preferable direction in the search space (e.g. for smooth function it can be the gradient of the objective function) then the probability of generating a random vector other than perpendicular to $\boldsymbol{\mu}$ is equal to zero, if the dimension of the search space tends to infinity. What completes the proof. □

3 Directional Distribution

The problem revealed in Theorem 1 can be neutralized with help of a technique which is based on the so-called directional distributions. In the literature, several classes of such distributions can be found [7], while for the need of evolutionary computations, the class of the so-called rotationally symmetric distributions $\mathcal{M}$ seems to be very attractive. The class $\mathcal{M}$ is usually parameterized by a pair $\{\boldsymbol{\mu}, \kappa\}$, where $\boldsymbol{\mu}$ is the mean direction, and κ stands for the concentration parameter. Therefore, the mutation operator can be perceived as a two stage process: first the direction of mutation is chosen according to $\mathcal{M}(\boldsymbol{\mu}, \kappa)$, and then the phenotype of an individual is changed in this direction by adding an one dimensional generate variable (in fact, heavy-tailed symmetric α-stable distribution $S_\alpha S(\gamma)$ is utilized [14] i.e.:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + r\boldsymbol{d}_k, \tag{5}$$

where $r \stackrel{d}{=} \chi_{\alpha,\gamma} = |S_\alpha S(\gamma)|$, and $\boldsymbol{d}_k \stackrel{d}{=} \mathcal{M}(\boldsymbol{\mu}, \kappa)$. At this stage, the choice of the symmetric α-Stable distribution at this stage is not an accidental one. In recent years, the class of $S_\alpha S(\gamma)$ distributions, has received an increasing interest of the evolutionary computation community [6,9,15]. The family of Symmetric Stable Distribution $S\alpha S(\gamma)$ is characterized by two parameters: stable index α, which define the shape of its p.d.f., and scale γ. Bearing in mind that evolutionary algorithms are not convergent to the optimal solution, but to the some area around it [3], the class of $S_\alpha S(\gamma)$ distribution allows to establish a well-balanced compromise between two mutually exclusive properties: an accuracy in locating potential solution and an ability of escaping from the local optima - two the most challenging problems of stochastic optimization.

Despite the fact that in literature one can find several well-known classes of distributions with rotational symmetry, each of them has its own disadvantage. For example, the most popular class, the so-called von Mises-Fisher distribution (vMF) with the density given by [2]:

$$f_n(\boldsymbol{x};\kappa,\boldsymbol{\mu}) = \frac{\kappa^{n/2-1}}{(2\pi)^{n/2} I_{\frac{n}{2}-1}(\kappa)} \exp(\kappa \boldsymbol{\mu}^T \boldsymbol{x}), \tag{6}$$

possesses the marginal density of $t = \boldsymbol{\mu}^T \boldsymbol{X}$, given by:

$$f_n(t;\kappa) = \frac{(\frac{\kappa}{2})^{n/2-1}}{\Gamma(\frac{n-1}{2})\Gamma(n/2) I_{\frac{n-1}{2}}(\kappa)} \exp(\kappa t)(1-t^2)^{\frac{n-3}{2}}, \tag{7}$$

where $\|\boldsymbol{\mu}\|_2 = 1$, $I_n(\cdot)$ stands for the modified Bessel function of the first kind. As one can observe, even though information collected during the optimization process allows to detect *the best* direction in the search space, it appears that vMF will not prefer this direction at all. Thus, the following question arises: is it possible to obtain a random vector distributed in the similar way as that of vMF, but having an arbitrarily chosen marginal density $t = \boldsymbol{\mu}^T \boldsymbol{X}$? Looking for the suitable subclass of directional distributions one can make use of the fact that every rotationally symmetric random variable $\boldsymbol{X}$ can be uniquely determined by its tangent normal decomposition [7]:

$$\boldsymbol{X} = t\boldsymbol{\theta} + \sqrt{1-t^2}\boldsymbol{\xi}, \tag{8}$$

where t is invariant under rotation about $\boldsymbol{\theta}$. Moreover $\boldsymbol{\xi}$ and t are independent and $\boldsymbol{\xi}$ is uniform on a surface of the unit ball $S^{(n-2)}$. The above fact can be used for creating a random variable with an arbitrarily chosen marginal distribution. To see this, let us substitute $\boldsymbol{\theta} = [0, 0, \ldots, 1]^T \in \mathbb{R}^n$, and then consider an orthogonal transformation $\boldsymbol{Q}\boldsymbol{X}$ that satisfies $\boldsymbol{Q}\boldsymbol{\theta} = \boldsymbol{\mu}$. It is easy to check that the random vector $\boldsymbol{X}$, obtained int this way, has the marginal distribution $\boldsymbol{X}^T\boldsymbol{\theta} \stackrel{d}{=} t$. Thus, by selecting a proper distribution for t, one can control the degree of probability mass concentration around the mean direction $\boldsymbol{\mu}$. In fact, the attention is focused on the marginal distribution of the form: $t = 2X - 1$, where X is Beta distributed random variable $\beta(a, b)$. Therefore, the density of t is given by:

$$f(t|a,b) = \frac{2^{1-a-b}}{\beta(a,b)}(1-t)^{b-1}(1+t)^{a-1}, \tag{9}$$

where a and b are parameters of the Beta distribution. The concentration parameter κ can be provided by using parameters of density (9) with: $a = \frac{n-1}{2}$ and $b = \kappa\frac{n-1}{2}$. The expectation and variance of the random variable $T = \boldsymbol{\theta}^T\boldsymbol{X}$ are as follows:

$$\mathrm{E}[T] = \frac{1-\kappa}{1+\kappa}, \qquad \mathrm{Var}(T) = \frac{8\kappa}{n(1+k)^2(1-\kappa^2)}. \tag{10}$$

The direction distribution simulation process is summarized in the Tab. 1.

Table 1. Algorihtm to simulating directional distribution $\mathcal{M}(\boldsymbol{\mu}, \kappa)$

Input data

$\boldsymbol{\mu} \in \mathbb{R}^n$ – mean direction

$\kappa \in (0, 1]$ – concentration parameter

Output data

Y – pseudo-random vector of $\mathcal{M}(\boldsymbol{\mu}, \kappa)$ distribution

Algorithm

$t = 2\beta(\frac{n-1}{2}, \frac{\kappa(n-1)}{2}) - 1$, where $\beta(a, b)$ gives random number from Beta distribution

$X \leftarrow \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{n-1})$

$Z \leftarrow X/\|X\|_2$

$Y \leftarrow [\sqrt{1-t^2}Z^T, t]^T$

$Y \leftarrow [\boldsymbol{I}_n - \boldsymbol{v}\boldsymbol{v}^T]\, Y$ where $\boldsymbol{v} = \frac{[0,0,\ldots,1]^T - \boldsymbol{\mu}}{\|[0,0,\ldots,1]^T - \boldsymbol{\mu}\|_2}$

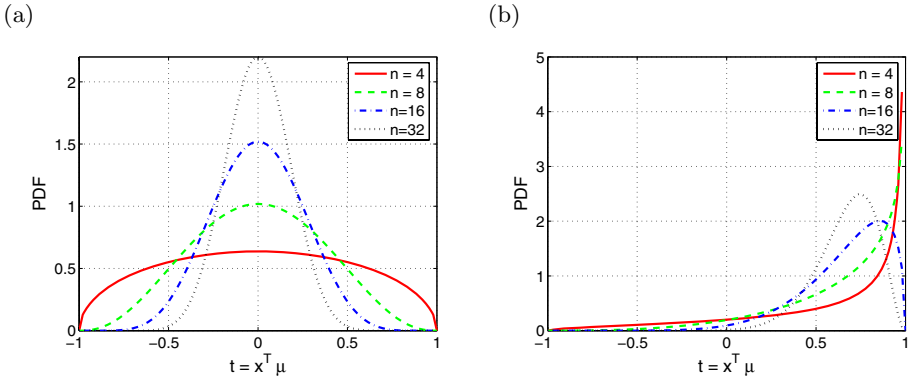


Fig. 1. p.d.f. of random variable $t = \boldsymbol{X}^T\boldsymbol{\mu}$ for isotropic mutation **(a)**, and for directional mutation with mean direction $\boldsymbol{\mu}$ and concentration parameters $\kappa = 0.1$ **(b)**

4 Adaptation of the Probabilistic Measure in Mutation Operator

The EAs belongs to the class of the so-called meta-heuristic stochastic optimization algorithms, which are based on a very simple rules adopted from Darwinian evolution theory: better individuals are preferable during evolution process, what means that they posses greater chance to survive and generate descendants. In the continuous evolutionary algorithms, new candidate solutions $\boldsymbol{x}_{k+1}$, are usually generated, by adding some realization of random variable $\boldsymbol{Z}$ to the before selected parent:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{Z}. \tag{11}$$

It is worth to stress that the effectiveness of any continues EAs depends, to a large extent, on the choice of a suitable distribution for $\boldsymbol{Z}$. The knowledge, needed for choosing the optimal distribution in the mutation operator, in practical applications, is neither complete nor sufficient. In order to avoid the problem, a probabilistic measure should be learnt during an optimization process. Most approaches presented in the literature utilize the multivariate Normal distribution [12]. Thus, their heuristics are mainly focused at the adaptation of a covariance matrix. Despite the fact that heuristics used to adapt Gaussian distribution during the evolution process allow to obtain algorithms with an effectiveness invariant to linear transformation of the solution space, it is worth to focus attention on a serious drawback related to the Normal distribution. Namely, even if one can imagine the best heuristic, that are able to point to the most preferable direction in the search space, the symmetry of Gaussian distribution will favour the worst direction, which is equally well preferable.The concept of directional distributions presented in the previous section allows to remove the above-mentioned problem. Naturally, the effectiveness of EAs with mutation based on a class $\mathcal{M}(\boldsymbol{\mu}, \kappa)$ will depend at least on two factors: the correctness of establishing the mean direction of mutation $\boldsymbol{\mu}$, and the value of the concentration parameter κ, which controls the dispersion around the mean direction. In fact, concentration parameter κ, defined in the Section 3, allows to obtain on one side an isotropic distribution on the sphere ($\kappa = 1$), and on the other side, a degenerate distribution at the mean direction ($\kappa = 0$). The idea of forcing mutation direction boils down to utilizing a traditional way of creating a new individual (5). Since directional distributions, are parameterized by a pair $\{\boldsymbol{\mu}, \kappa\}$, then one must determine the strategy of adjusting their values. In this paper, the attention is restricted to the parameter $\boldsymbol{\mu}$ only. In the literature, several techniques doing this can be found:

- **Heuristic No. 1** – the most promising direction is given by the formula:

$$\boldsymbol{\mu}^t = -\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|_2}, \quad \text{where} \quad \boldsymbol{z} = [\boldsymbol{P}_t^T \boldsymbol{P}_t]^{-1} \boldsymbol{P}_t^T \boldsymbol{\Phi}_t,$$

  where $\boldsymbol{P}_t \in R^{\eta \times n}$ and $\boldsymbol{\Phi}_t \in R^{\eta}$ stands for a matrix of phenotypes and a vector of individual fitness respectively, e.g.:

$$\boldsymbol{P}_t = [\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \ldots, \boldsymbol{x}_\eta^t]^T \qquad \boldsymbol{\Phi}_t = [\phi(\boldsymbol{x}_1^t), \phi(\boldsymbol{x}_2^t), \ldots, \phi(\boldsymbol{x}_\eta^t)]^T$$

  An obvious drawback of the method lies in the fact that the inversion of the matrix $\boldsymbol{P}_t^T \boldsymbol{P}_t$ requires at least as many individuals as the dimension of the search space.
- **Heuristic No. 2** – the method, firstly proposed by Salomon [13], for the class of algorithms known as Evolutionary Gradient Search:

$$\boldsymbol{\mu}^t = \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|_2}, \quad \text{where} \quad \boldsymbol{z} = \sum_{k=1}^{\eta} \frac{\phi(\boldsymbol{x}_k^{t-1}) - \phi(\boldsymbol{x}_k^t)}{\phi(\boldsymbol{x}_k^{t-1})} \frac{\boldsymbol{x}_k^{t-1} - \boldsymbol{x}_k^t}{\|\boldsymbol{x}_k^{t-1} - \boldsymbol{x}_k^t\|_2}$$

– **Heuristic No. 3** – the approach proposed by Obuchowicz [8] for the class of evolutionary algorithms known as *Evolutionary Search with Soft Selection and Forced Direction of Mutation* (ESSS-FDM):

$$\boldsymbol{\mu}^t = \frac{\langle \boldsymbol{x}^t \rangle - \langle \boldsymbol{x}^{t-1} \rangle}{\| \langle \boldsymbol{x}^t \rangle - \langle \boldsymbol{x}^{t-1} \rangle \|}, \quad \text{where} \quad \langle \boldsymbol{x}^t \rangle = \frac{1}{\eta} \sum_{k=1}^{\eta} \boldsymbol{x}_k^t$$

4.1 Evolutionary Search with Soft Selection

Evolutionary algorithms used in simulation experiments in this work are based on the ESSS algorithm (*Evolutionary Search with Soft Selection*), which is based on a probably the simplest selection-mutation model of the Darwinian's evolution [5]. To stress that original algorithm is modified by applying directional mutation (5) the abbreviation $ESSS_\alpha - DM$ is used. The evolution is a motion

Table 2. Outline of the $ESSS_\alpha - DM$ algorithm

Input data

η – population size;
$t_{\max}$ – maximum number of iterations (epochs);
γ, α, κ – parameters of mutation: scale, stable index and concentration;
$\phi : \mathbb{R}^n \to \mathbb{R}$ – fitness function;
$\boldsymbol{x}_0^0$ – initial point.

Algorithm

1. Initialize

$$P(0) = \big(\boldsymbol{x}_1^0, \boldsymbol{x}_2^0, \ldots, \boldsymbol{x}_\eta^0\big), \quad \boldsymbol{x}_k^0 = \boldsymbol{x}_0^0 + \boldsymbol{Z},$$
$$\text{where} \quad \boldsymbol{Z} \sim \mathcal{N}(0, \gamma \boldsymbol{I}_n), \quad k = 1, 2, \ldots, \eta$$

2. Repeat

(a) *Estimation*

$$\Phi\big(P(t)\big) = \big(q_1^t, q_2^t, \ldots, q_\eta^t\big), \quad \text{where} \quad q_k^t = \phi\big(\boldsymbol{x}_k^t\big), \quad k = 1, 2, \ldots, \eta.$$

(b) *Proportional selection*

$$P(t) \longrightarrow P(t)' = \big(\boldsymbol{x}_{h_1}^t, \boldsymbol{x}_{h_2}^t, \ldots, \boldsymbol{x}_{h_\eta}^t\big) \ ,$$

(c) *Estimation of the most promising direction of mutation*

$$\boldsymbol{\mu}(t) \longleftarrow H\big(P'(t), P'(t-1)\big),$$

(d) *Mutation*

$$P(t)' \longrightarrow P(t+1);$$
$$\boldsymbol{x}_k^{t+1} = \boldsymbol{x}_{h_k}^t + \chi_{\alpha,\gamma} \boldsymbol{u}^{(n)}, \ \boldsymbol{u}^{(n)} \sim \mathcal{M}(\boldsymbol{\mu}(t), \kappa), \ k = 1, 2, \ldots, \eta.$$

Until $t > t_{\max}$.

of individuals in the phenotype space, called also the adaptation landscape. This motion is caused by the selection and mutation process. Selection leads to the concentration of individuals around the best ones, but mutation introduces the diversity of phenes and disperses the population in the landscape.

The above-described assumptions can be formalized by the algorithm presented in Tab. 2.

5 Simulation Experiments

The aim of first experiment was to minimize an objective function of the form $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{x}$. Four versions of $ESSS_\alpha - DM$ algorithm were investigated, with stable index $\alpha = 1.5$. Initial population $\eta = 45$ was placed at the point $\boldsymbol{x}^0 = [20, 20, \ldots, 20]^T \in \mathbb{R}^{40}$. During the optimization process, two quantities were observed: the value of the best individual in the population, and the cosine of the angles between an exact gradient vector and the direction of a particular mutation e.g. $T = \boldsymbol{d}^T\boldsymbol{\mu}^*$, where $\boldsymbol{d} = \frac{\boldsymbol{x}^t - \boldsymbol{x}^{t-1}}{\|\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\|_2}$. Both results are presented in the Fig. 2.

For the spherical model, heuristics presented in Section 4, lead to the faster convergence of $ESSS_\alpha - DM$ algorithm. In comparison with the isotropic mutation, the number of useless population steps was minimized, which is clearly visible in the Fig. 2 (b) - histograms related to different version of $ESSS_\alpha - DM$ algorithm are shifted towards 1, which means that more mutations lay closer to the steepest descent direction.

The objective of the second experiment, was to check the effectiveness of the proposed heuristics in multimodal landscapes. The two standard benchmark functions [1] were chosen: the Ackley Function and Rastringin Function. In both cases the initial population was placed at $\boldsymbol{x}^0 = [20, 20, \ldots, 20]^T \in \mathbb{R}^{20}$, and the fitness value of the best individual was observed. It must to be stressed, that

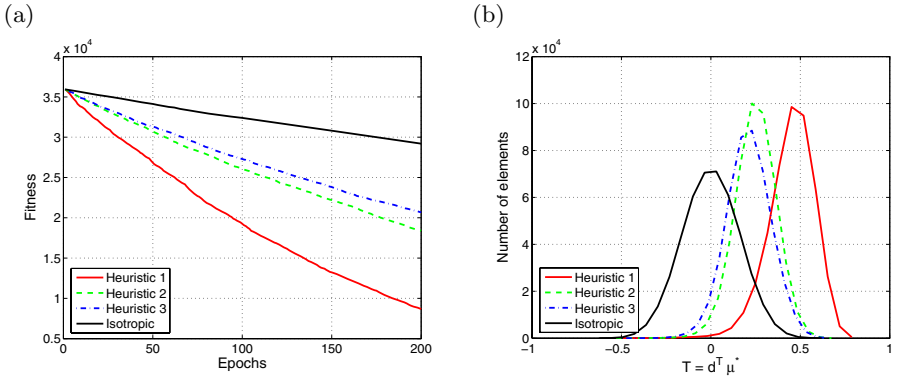


Fig. 2. The best fitness value vs. epochs - **(a)** (results averaged over 50 runs), **(b)** - histogram of the cosine angles between the exact gradient direction and preferable direction by the mutation operator for $ESSS_\alpha - DM$ algorithms with different heuristics

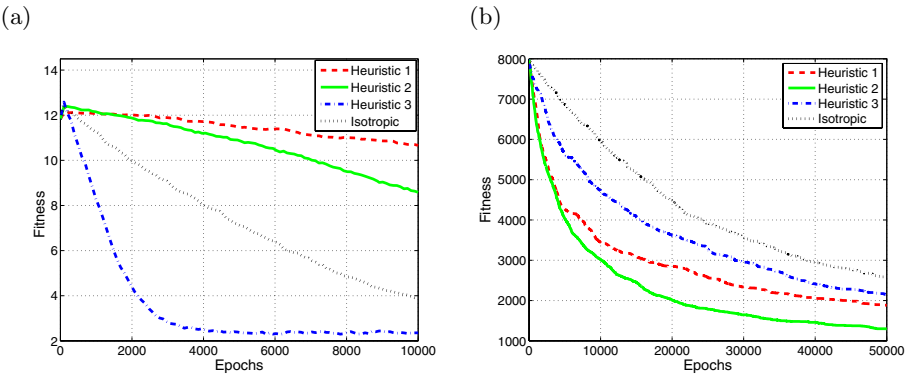


Fig. 3. The best fitness value vs. epochs (results averaged over 50 runs). **(a)** - Ackley Function , **(b)** - Rastringin Function.

parameters of mutation $\alpha = 1.5$, $\gamma = 0.1$, $\kappa = 0.1$ were chosen in such way as to assure slow convergence of the algorithm.

In the case of the Ackley function (Fig. 3), heuristics No. 1 and 2 make the evolutionary algorithm more susceptible to being trapped in local solutions. In this respect heuristic No. 3 guaranteed the fastest convergence, and appeared to be much more efficient than the isotropic mutation with the same stable index. The results obtained for the Rastringin Function confirm advantages of the proposed directional mutation. In this case a completely different situation is observed. In Fig. 3 one can see that all of the proposed approaches led to the faster convergence than the mutation with spherical symmetry.

6 Conclusion

In this paper, the general concept for the adaptation of a probabilistic measure in a mutation operator of phenotypic EAs is introduced. The proposed approach is based on the directional distributions which are parameterized by the mean vector and concentration parameter. Three techniques for adjusting the most promising direction of mutation are introduced and their efficiency is investigated via numerical simulations. The proposed mutation improves the effectiveness of evolutionary algorithms by making use of an information collected during every mutation procedure. Moreover, it also provides an access to construction of more sophisticated techniques which aim at improving effectiveness of evolutionary algorithms for high-dimensional problems.

Acknowledgment

The work was partially supported by Ministry of Education an Science in Poland under the grant 4T11A01425 *Modelling and identification of non-linear dynamic systems in robust diagnostics.*

References

1. E. Alba, F. Luna, A. J. Nebro: Advances in parallel heterogenous genetic algortihms for continous optimization, Int. J. Appl. Math. Comput. Sci., 2004, vol. 14, No. 3,pp. 317-333.
2. I. S. Dhillon and S. Sra: Modeling Data using Directional Distributions. Technical Report $TR-03-06$, 2003.
3. I. Karcz-Dulęba : Asymptotic behaviour of a discrete dynamical system generated by a simple evolutionary process, Int. J. Appl. Math. Comput. Sci., Vol. 14, No. 1, pp. 7990, 2004.
4. K.-T. Fang, S. Kotz, and K.-W. Ng: Symmetric Multivariate and Related Distributions. Chapman and Hall, London, 1990.
5. R. Galar, Evolutionary search with soft selection, Biological Cybernetics, Vol.60, 1989, pp.357–364.
6. M. Gutowski: Lévy flights as an underlying mechanism for a global optimization algorithm. Proc. 5th Conf. Evolutionary Algorithms and Global Optimization, Jastrzębia Gora, Poland, Warsaw University of Technology Press, 2001, pp. 79–86.
7. K.V. Mardia, and P. Jupp: Directional Statistics. John Willey and Sons Ltd., 2nd edition, 2000.
8. A. Obuchowicz: Evolutionary Algorithms in Global Optimization and Dynamic System Diagnosis. Lubuskie Scientific Society, Zielona Góra 2003.
9. A. Obuchowicz,P.Prętki: Phenotypic evolution with mutation based on symmetric α-stable distribution. Int. J. Appl. Math. Comput. Sci., 2004, vol. 14, No. 3,pp. 289-316.
10. Pelikan, M., D. E. Goldberg, and E. Cantu-Paz: BOA: The Bayesian Optimization Algorithm. In: W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith (eds.): Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, Vol. I. Orlando, FL, pp. 525532, Morgan Kaufmann Publishers, San Fransisco, CA.
11. G. Rudolph: On Correlated Mutations in Evolution Strategies. Parallel Problem Solving from Nature 2. Proc. 2nd Int. Conf. on Parallel Problem Solving from Nature, Brussels 1992.
12. S. Kern, S. Muller, D. Buche, N. Hansen, and P. Koumoutsakos: Learning probability distributions in continuous evolutionary algorithms, in Workshop on Fundamentals in Evolutionary Algorithms, Thirtieth International Colloquium on Automata, Languages and Programming, Eindhoven, 2003.
13. R. Salomon: Evolutionary Algorithms and Gradient Search: Similarities and Differences. IEEE Transactions on Evolutionary Computation, Vol. 2, No. 2, pp. 45–55, 1998.
14. G. Samorodnitsky and M.S. Taqqu, *Stable Non-Gaussian Random Processes*, Chapman & Hall, New York, 1994
15. Ch.Y. Lee, and X. Yao : Evolutionary Programming Using Mutations Based on the Lévy Probability Distribution. IEEE Transactions on Evolutionary Computation, Vol. 8, No. 1, pp. 1-13, 2004.

Adaptive Inertia Weight Particle Swarm Optimization

Zheng Qin[1,2], Fan Yu[1], Zhewen Shi[1], and Yu Wang[2]

[1] Department of Computer Science and Technology, Xian JiaoTong University, Xian 710049, P.R. China
[2] Department of Computer Science and Technology, Tsinghua University, Beijing 100084

Abstract. Adaptive inertia weight is proposed to rationally balance the global exploration and local exploitation abilities for particle swarm optimization. The resulting algorithm is called adaptive inertia weight particle swarm optimization algorithm (AIW-PSO) where a simple and effective measure, individual search ability (ISA), is defined to indicate whether each particle lacks global exploration or local exploitation abilities in each dimension. A transform function is employed to dynamically calculate the values of inertia weight according to ISA. In each iteration during the run, every particle can choose appropriate inertia weight along every dimension of search space according to its own situation. By this fine strategy of dynamically adjusting inertia weight, the performance of PSO algorithm could be improved. In order to demonstrate the effectiveness of AIW-PSO, comprehensive experiments were conducted on three well-known benchmark functions with 10, 20, and 30 dimensions. AIW-PSO was compared with linearly decreasing inertia weight PSO, fuzzy adaptive inertia weight PSO and random number inertia weight PSO. Experimental results show that AIW-PSO achieves good performance and outperforms other algorithms.

1 Introduction

Particle Swarm Optimization (PSO), a new evolutionary computation technique inspired by social behavior simulation, has achieved promising performance on nonlinear function optimization. Since the original version PSO is first introduced by Kennedy and Eberhart in 1995 [1], many interesting variations of PSO have emerged [2-5]. Among them, linearly decreasing inertia weight PSO (LDW-PSO) [5] has been widely regarded as the standard version PSO algorithm. Many engineering optimization application are based on LDW-PSO [6-10]. The concept inertia factor is very important and useful in PSO community. In 1998, Angeline [11] found that the original version has a poor ability to search at a fine grain because it lacks velocity control mechanism. In order to overcome this disadvantage, a linearly decreasing inertia factor was first introduced by Eberhart and Shi in 1998 [5]. At the beginning of the process, a larger inertia factor is used for global exploration. During the search, inertia factor is becoming smaller

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 450–459, 2006.
© Springer-Verlag Berlin Heidelberg 2006

and smaller for local exploitation. By doing this, the inertia factor balances the global wide-rang exploration and the local nearby exploitation abilities of the swarm, which leads to significant improvement in the performance of PSO. In 2001 Shi [12] used a fuzzy system to dynamically adapt the inertia weight in order to further improve the performance. In 2003 Zhang [13] investigated the effect of random inertia weight in PSO. As reported in their experimental results, the resulting algorithm achieved better performance than LDW-PSO.

Depending on the concept inertia weight, above methods achieve promising results, but they still did not adequately exploit the effect of it. The inertia factor has potential to aid in further performance improvement. In this paper, we define a measure, Individual Search Ability (ISA), to indicate each particle lacks whether global exploration or local exploitation abilities. Depending on the ISA, the inertia weight of each particle in each dimension is calculated with the defined transform function so as to enhance the corresponding weak search abilities. Thus, along every dimension, every particle has different values of inertia weight in every iteration, which can promote to choose the appropriate inertia weight. Then rational combination of global and local search abilities is achieved for every particle. The proposed method is called adaptive inertia weight particle swarm optimization algorithm (AIW-PSO).

The rest of this paper is organized as follows. In Section 2, AIW-PSO is proposed, and then the mechanism of individual activity and transform function are discussed in detail. In Section 3, benchmark functions and experimental setup are described. Subsequently, results are presented. Finally, Section 4 gives conclusions.

2 Adaptive Inertia Weight Particle Swarm Optimization

A swarm consists of N particles moving around in a D-dimensional search space. The i-th particle at the t-th iteration has a position $X_i^{(t)} = (x_{i1}, x_{i2}, \ldots, x_{iD})$, a velocity $V_i^{(t)} = (v_{i1}, v_{i2}, \ldots, v_{iD})$, the best solution achieved so far ($pbest$) by itself $P_i^{(t)} = (p_{i1}, p_{i2}, \ldots, p_{iD})$. The best solution achieved so far by the whole swarm ($gbest$) is represented by $P_g^{(t)} = (p_{g1}, p_{g2}, \ldots, p_{gD})$. The position of the i-th particle at the next iteration will be calculated according to the following equations:

$$V_{id}^{(t+1)} = w * V_{id}^{(t)} + c_1 * rand() * (P_{id}^{(t)} - X_{id}^{(t)}) + c_2 * rand() * (P_{gd}^{(t)} - X_{id}^{(t)}) \tag{1}$$

$$X_{id}^{(t+1)} = X_{id}^{(t)} + V_{id}^{(t+1)} \tag{2}$$

where c_1 and c_2 are two positive constants, called cognitive learning rate and social learning rate respectively; $rand()$ is a random function in the range $[0, 1]$; w is inertia factor. In addition, the velocities of the particles are confined within $[Vmin, Vmax]^D$. If an element of velocities exceeds the threshold $Vmin$ or $Vmax$, it is set equal to the corresponding threshold.

The inertia weight is critical for the performance of PSO, which balances global exploration and local exploitation abilities of the swarm. A large inertia weight facilitates exploration, but it makes the particle long time to converge. Conversely, a small inertia weight makes the particle fast converge, but it sometimes leads to local optimum. During the search every particle dynamically changes its position, so every particle locates in a complex environment and faces different situation. Therefore, each particle along every dimension may have different trade off between global and local search abilities. In this paper, inertia weight is dynamically adapted for every particle along every dimension. A measure, Individual Search Ability (ISA), which characterizes the faced situation for every particle is defined. Basing on this measure, the particle could decide to whether to increase or decrease the values of inertia weight. The fine strategy of dynamically adjusting inertia weight could lead to improvement in performance of PSO.

Definition 1. *Given $\varepsilon > 0$, the Individual Search Ability of the i-th particle along the jth dimension, ISA_{ij}, is defined:*

$$ISA_{ij} = \frac{|x_{ij} - p_{ij}|}{|p_{ij} - p_{gj}| + \varepsilon} \tag{3}$$

where x_{ij} is the position of the i-th particle in the j-th dimension; p_{ij} is the own best solution, while p_{gj} is the current global best solution. |...| denotes the absolute value and ε is a positive constant close enough to zero.

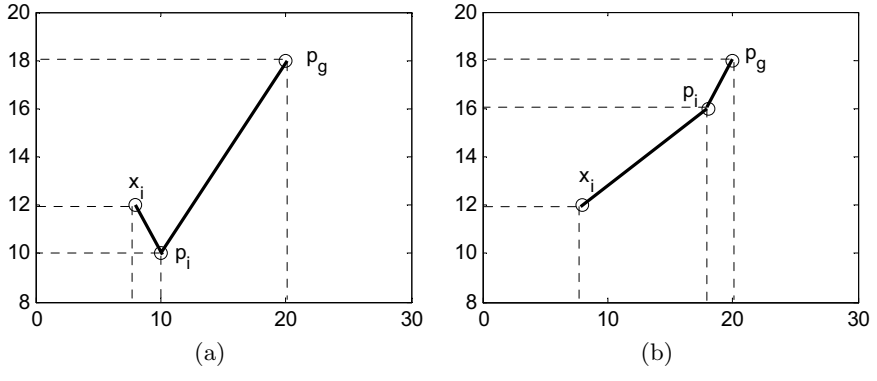


Fig. 1. The relation between x_i, p_i, and p_g in 2-dimensional optimization problem: (a) p_i is far from p_g (b) p_i is close to p_g

As illustrated in Figure 1 (a), the particle i moves around p_i , and p_i is far from p_g, It is possible that p_i is the local optimum and the particle i is trapped. According to formula (3), ISA gets a small value at this scenario. The global exploration should be enhanced in order to escape from the local optimum. On the other hand, as illustrated in Figure 1 (b) p_i is close to p_g, which means is

a good position, but the particle i is far from it. It is possible that the particle i has too much strong global exploration ability but lacks local exploitation ability, which leads to a large value of ISA. As it only utilizes the information of the current iteration, ISA is expected to be a simple and effective measure to indicate individual search ability. A large ISA means strong global exploration ability, inertia weight should be decreased. While a small ISA means that the inertia weight should be increased. Then, the value of inertia weight for every particle along every dimension, $w_{ij}(i = 1, \cdots, N; j = 1, \cdots, D)$, is dynamically calculated with the following transform function.

Definition 2. *The transfer function is*

$$w_{ij} = 1 - \alpha(\frac{1}{1 + e^{-ISAij}}) \tag{4}$$

where α is a positive constant in the range (0,1].

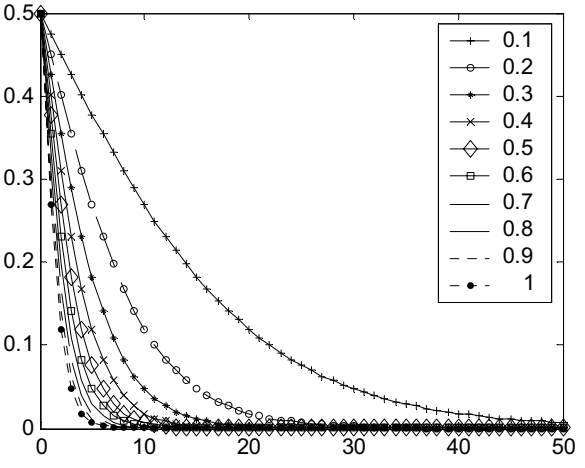


Fig. 2. w vs ISA with different α

Figure 2 shows the change of inertia weight w with respect to ISA with different α varied from 0.1 to 1. The parameter α controls the decreasing speed of inertia weight. In order to observe the impact of α on the performance of PSO, parameter α is varied from 0.1 to 1 with step size 0.1. AIW-PSOs with different values of α, 20 particles, 2000 maximum iterations, were conducted on three 30-dimensional benchmark functions. For each experimental setting, 100 runs of the algorithm were performed. Table 1 listed the mean best fitness values averaged over 100 runs. It is clear that the values in range [0.1, 0.4] for α can all lead to acceptable performance. In present paper, α is set to 0.3.

Table 1. The mean best fitness values with different values of α

α	f_1	f_2	f_3
0.1	170.1461	31.2124	0.0148
0.2	208.9064	24.6422	0.0137
0.3	241.9252	22.0608	0.0113
0.4	179.2728	23.9913	0.0144
0.5	511.6168	27.6693	0.0181
0.6	738.7308	28.7811	0.0174
0.7	768.6959	31.3867	0.0305
0.8	1855.2588	37.2883	0.0416
0.9	1261.3114	38.1208	0.0766
1	1944.9399	45.9375	0.1877

3 Experiments

Three well-known benchmark functions (all minimization) were used in our experiments.

The first function is the Rosenbrock function(Figure 3):

$$f_1(x) = \sum_{i=1}^{n-1} \left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right) \qquad (-100 \le x_i \le 100) \tag{5}$$

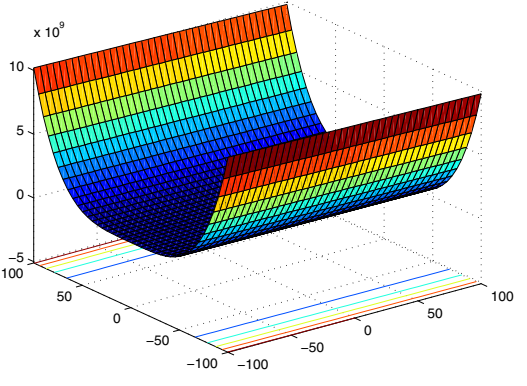
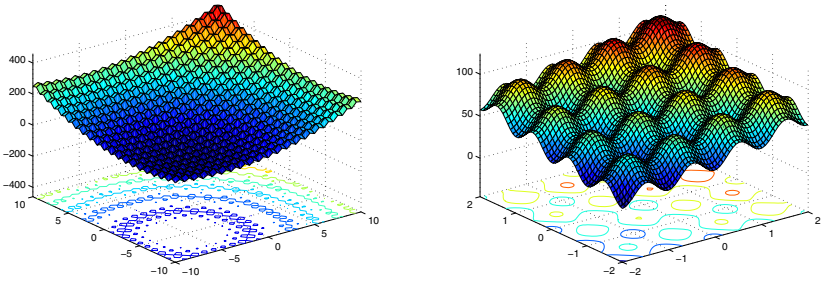


Fig. 3. The Rosenbrock function

The second function is the generalized Rastrigrin function(Figure 4):

$$f_2(x) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right) \qquad (-10 \le x_i \le 10) \tag{6}$$

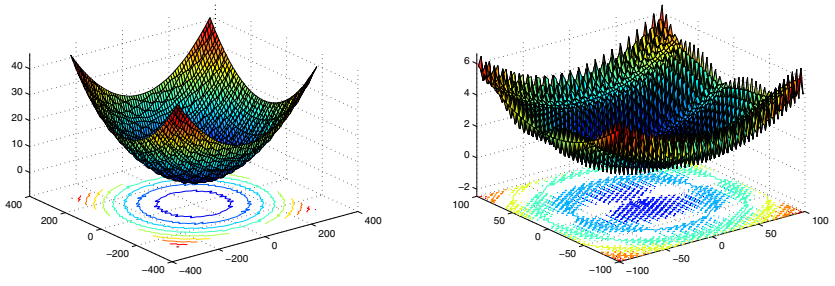


(a) (b)

Fig. 4. The generalized Rastrigrin function

The third function is the generalized Griewank function(Figure 5):

$$f_3(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1 \qquad (-600 \le x_i \le 600) \tag{7}$$



(a) (b)

Fig. 5. The Griewank function

Figures 3, 4 and 5 show the surface landscapes of three benchmark functions: Rosenbrock, generalized Rastrigrin, and Griewank when the dimension is set to 2 respectively. Rosenbrock is a unimodal function,which has a global minimum of 0 at the point $(1, \cdots, 1)$. Compared with multimodal function, the unimodal function is easy to optimize, but the optimum solution is difficult to achieve when problem dimension goes high. As shown in Figure 4 and 5, generalized Rastrigrin and Griewank are multimodal functions which have many local minima. The optimum solution of generalized Rastrigrin is $(0, \cdots, 0)$ with the global minimum 0; Griewank also has a global minimum of 0 when the vector is $(0, \cdots, 0)$.

In our experiments, AIW-PSO were compared with linearly decreasing inertia weight PSO (LDW-PSO) [5] , fuzzy adaptive inertia weight PSO (FUW-PSO)

Table 2. Search space and initialization range

Function	Search Space $[X_{min}, X_{max}]$	Initialization Range
$f_1(x)$	$-100 \leq x_i \leq 100$	$15 \leq x_i \leq 30$
$f_2(x)$	$-10 \leq x_i \leq 10$	$2.56 \leq x_i \leq 5.12$
$f_3(x)$	$-600 \leq x_i \leq 600$	$300 \leq x_i \leq 600$

Table 3. The best fitness values for the Rosenbrock function $f_1(x)$

Population Size	Dim	Max Iteration	AIW PSO	RNW PSO	FUW PSO	LDW PSO
20	10	1000	48.6378	65.28474	66.0141	96.1715
	20	1500	115.1627	147.52372	108.2865	214.6764
	30	2000	218.9012	409.2443	183.8037	316.4468
40	10	1000	24.5149	41.32016	48.7652	70.2139
	20	1500	60.0686	95.48422	63.8841	180.9671
	30	2000	128.7677	253.81490	175.0093	299.7061
80	10	1000	19.2232	20.77741	15.8165	36.2945
	20	1500	52.8523	82.75467	46.0000	87.2802
	30	2000	149.4491	156.00258	124.4184	205.5596

[12], and random number inertia weight PSO (RNW-PSO) [13]. For the purpose of comparison, parameters were assigned as same as that in literature [5] [12] [13]. For each function, three dimensions were tested: 10, 20 and 30, the maximum numbers of generations were set to 1000, 1500 and 2000 correspondingly. In order to investigate the scalability of PSO algorithms, three population sizes 20, 40 and 80 were used for each function with different dimensions. V_{min} is set equal to X_{min}; V_{max} equal to X_{max}. The learning rates were $c_1 = c_2 = 2$. The parameter of AIW-PSO, α, is set to 0.3. For each experimental setting, 100 runs of the algorithm were performed. The search space and initialization range for each test function were listed in Table 2. In order to give right indications of relative performance, an asymmetric initialization was adopted according to literature [11].

Tables 3, 4 and 5 respectively listed the mean fitness values of the best solutions achieved by four algorithms on Rosenbrock, Rastrigrin, and Griewank functions with each experimental setting. For fair comparisons, the results of RNW-PSO, FUW-PSO, and LDW-PSO were directly taken from their original publications. The results of AIW-PSO were averaged over 100 trails with each experimental setting.

In these Tables, AIW-PSO algorithm exhibits good performance. It outperforms RNW-PSO and LDW-PSO on all benchmark problems. AIW-PSO is little inferior to FUW-PSO on Rosenbrock function with some dimension, while

Table 4. The best fitness values for the generalized Rastrigrin function $f_2(x)$

Population Size	Dim	Max Iteration	AIW PSO	RNW PSO	FUW PSO	LDW PSO
20	10	1000	3.7415	5.04258	4.9552	5.5572
	20	1500	11.1323	20.3111	23.2733	22.8892
	30	2000	22.1155	42.5813	48.4756	47.2941
40	10	1000	1.9900	3.2255	3.2834	3.5623
	20	1500	7.2145	13.8481	15.0445	16.3504
	30	2000	17.5765	32.1564	35.2015	38.5250
80	10	1000	1.0051	1.8593	2.3282	2.5379
	20	1500	5.0615	9.9500	10.8610	13.4263
	30	2000	13.1237	25.4412	22.5239	29.3063

Table 5. The best fitness values for the Griewank function $f_3(x)$

Population Size	Dim	Max Iteration	AIW PSO	RNW PSO	FUW PSO	LDW PSO
20	10	1000	0.0734	0.0962	0.9162	0.0919
	20	1500	0.0252	0.0300	0.0273	0.0303
	30	2000	0.0120	0.0167	0.0216	0.0182
40	10	1000	0.0671	0.0870	0.0757	0.0862
	20	1500	0.0266	0.0342	0.0312	0.0286
	30	2000	0.0146	0.0168	0.0122	0.0127
80	10	1000	0.0106	0.0715	0.0683	0.0760
	20	1500	0.0258	0.0283	0.0260	0.0288
	30	2000	0.0106	0.0159	0.0149	0.0128

AIW-PSO is superior to FUW-PSO on Rastrigrin, and Griewank functions with all dimensions.

The performance improvement of AIW-PSO results from the two aspects. On the one hand, the variety of inertia weight is maintained in every iteration. During the search, the particles face different situation, so they get different the values of ISA and then inertia weight. At the early iteration, part particles with large inertia weight have strong global search abilities and locate the promising search areas, while part particles with small inertia weight can help to accelerate convergence. Near the end of the run part particles with small inertia weight is responsible to fine search, while part particles with large inertia weight can promote escaping from the local optima. On the other hand, the particles can rationally adjust inertia weight according to particle's own situation. Every particle first calculates ISA to know its situation, and then chooses appropriate inertia weight along every dimension. When the particle achieves a small ISA,

a large value of inertia weight will be set. Meanwhile, if the ISA is large, the value of inertia weight will be small. Thus, the particles can duly achieve corresponding search ability at different situation so that they can capture better solutions.

4 Conclusions

In this paper, we proposed AIW-PSO algorithm where every particle dynamically adjusts inertia weight along every dimension of search space according to its faced situation. Individual Search Ability (ISA) was proposed to identify the situations for particles. ISA can indicate each particle lacks whether global exploration or local exploitation abilities in a situation. The calculation of ISA involves only three parameters: the position of the particle, pbest and gbest, so ISA is a simple and effective measure. Then a transform function was defined to translate ISA into inertia weight. Because the particles face different situation and complex environment, they achieve different suitable values of inertia weight. The variety of inertia weight results in the whole swarm good balance of global and local search ability. Moreover, the particle can rationally adjust inertia weight according to its situation indicated by ISA. Therefore, the particle can duly achieve corresponding search ability at different situation. By this fine strategy of dynamically adjusting inertia weight, the performance of PSO algorithm was improved. Experimental results show that AIW-PSO achieves better solutions than LDW-PSO, FUW-PSO and RNW-PSO on three well-known benchmark functions.

Acknowledgements

This work was supported by the National Grand Fundamental Research 973 Program of China (No.2004CB719401) and the National High-tech Research and Development Plan of China (No. 2003AA412020).

References

1. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceeding of IEEE International Conference on Neural Networks (ICNN'95). Volume 4., Perth, Western Australia, IEEE (1995) 1942–1947
2. van den Bergh, F., Engelbrecht, A.P.: A cooperative approach to particle swarm optimization. IEEE Transactions On Evolutionary Computation **8** (2004) 225–239
3. Peram, T., Veeramachaneni, K., Mohan, C.K.: Fitness-distance-ratio based particle swarm optimization. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium. (2003) 174–181
4. Silva, A., Neves, A., Costa, E.: An empirical comparison of particle swarm and predator prey optimisation. Lecture Notes in Computer Science **2464** (2002) 103–110

5. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, Singapore (1998) 69–73
6. Zwe-Lee, G.: Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE Transactions on Power Systems **18** (2003) 1187–1195
7. Zwe-Lee, G.: A particle swarm optimization approach for optimum design of pid controller in avr system. IEEE Transactions on Energy Conversion **19** (2004) 384–391
8. Jinho, P., Kiyong, C., Allstot, D.: Parasitic-aware rf circuit design and optimization. IEEE Transactions on Circuits and Systems **51** (2004) 1953–1965
9. Baskar, S., Zheng, R.T., Alphones, A., Ngo, N.Q., Suganthan, P.N.: Particle swarm optimization for the design of low-dispersion fiber bragg gratings. IEEE Photonics Technology Letters **17** (2005) 615–617
10. Abido, M.A.: Optimal design of power-system stabilizers using particle swarm optimization. IEEE Transactions on Energy Conversion **17** (2002) 406–413
11. Angeline, P.J.: Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. Lecture Notes in Computer Science **1447** (1998) 601–610
12. Shi, Y., Eberhart, R.C.: Fuzzy adaptive particle swarm optimization. In: Proceedings of the 2001 Congress on Evolutionary Computation CEC2001, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, IEEE Press (2001) 101–106
13. Zhang, L., Yu, H., Hu, S.: A new approach to improve particle swarm optimization. In: Genetic and Evolutionary Computation. (2003) 134–139

Estimation of the Evolution Speed for the Quasispecies Model: Arbitrary Alphabet Case

Vladimir Red'ko[1] and Yuri Tsoy[2]

[1] Institute of Optical Neural Technologies, Russian Academy of Sciences, Moscow, 119333, Russia
vgredko@gmail.com
[2] Computer Engineering Department, Tomsk Polytechnic University, Tomsk, 634034, Russia
qai@mail.ru

Abstract. The efficiency of the evolutionary search in M. Eigen's quasispecies model for the case of an arbitrary alphabet (the arbitrary number of possible string symbols) is estimated. Simple analytical formulas for the evolution rate and the total number of fitness function calculations are obtained. Analytical estimations are proved by computer simulations. It is shown that for the case of unimodal fitness function of λ-ary strings of length N, the optimal string can be found during $(\lambda-1)N$ generations under condition that the total number of fitness function calculations is of the order of $[(\lambda-1)N]^2$.

1 Introduction

Evolutionary algorithms and methods [1-5] were intensively investigated and applied during the last two decades. In this connection it is important to analyze quantitatively the questions: How efficient is the evolution as an optimization technique? What is speed of evolutionary processes? How many evolving organisms should be processed before "the optimal organism" is found for the given genome size N?

A number of researchers analyzed efficiency of evolutionary algorithms in different contexts [6-10]. Obtained results reveal many significant aspects of the evolution models; nevertheless, there is no clear answer on the questions stated above.

In this paper we analyze the efficiency of evolutionary search in the quasispecies model that was proposed and investigated by Manfred Eigen [11,12]. Some estimations of the evolution rate in this model were made by Hans Kuhn with coworkers [13]. The quasispecies model can be considered as simple canonical model of evolution with well defined scheme. This model is equivalent to the genetic algorithm [1,2] without crossover.

We obtain rough analytical estimations for the rate of evolution and the total number of evolution participants (the number of fitness estimations) and prove analytical estimations by computer simulations. We pay a special attention to stochastic character of evolution and to neutral selection that are due to finite population size [14].

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 460–469, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In the previous work [15] we analyzed the special case of the quasispecies model, in which the strings consist of two symbols. In this paper we investigate the more general case: the string symbols are taken from an arbitrary alphabet.

The paper is organized as follows. Section 2 describes the scheme of the quasispecies model. In Section 3 we outline general features of evolution in the quasispecies model qualitatively. Section 4 characterizes the role of neutral selection. Analytical estimations of the rate and efficiency of the evolution algorithm are made in Section 5, the results of computer simulations are described in Section 6. Section 7 includes discussion and conclusion.

2 The Scheme of the Quasispecies Model

The main assumptions of the quasispecies model [11,12,15-17] are as follows.

1. We consider evolution of a population of model "organisms" $\{\mathbf{S}_k\}$, where each organism $\mathbf{S}_k$ is defined by a string of λ-ary symbols S_{ki} , $i = 1, ..., N$; $k = 1, ..., n$; N is the length of the strings and n is the population size. We assume that symbols are taken from the predetermined alphabet and that the number of letters in the alphabet is λ. For example, $\lambda = 4$ in case of DNA or RNA. The string $\mathbf{S}_k$ can be considered as a model DNA of the k^{th} organism.
2. We assume that fitness function $f(\mathbf{S})$ is unimodal, that is there is the optimal string $\mathbf{S}_m$, that has maximum fitness value, and fitness of any other string $\mathbf{S}$

Table 1. The scheme of evolution

Step 0. The formation of the initial random population $\{\mathbf{S}_k(0)\}$. For each $k = 1, 2, ..., n$ and each $i = 1, 2, ..., N$, a symbol S_{ki} is chosen at random from the considered alphabet.
Step 1. Selection
Substep 1.1. Fitness calculation. For the population $\{\mathbf{S}_k(t)\}$ (t is the number of the generation), $f(\mathbf{S}_k)$ is evaluated for each $k = 1, 2, ..., n$.
Substep 1.2. Formation of the new population $\{\mathbf{S}_k(t+1)\}$. n strings are selected from $\{\mathbf{S}_k(t)\}$ into the next generation $\{\mathbf{S}_k(t+1)\}$ by means of the "roulette-wheel sampling" [2]. Namely, selection of a string into the new population takes place exactly n times; every time, the probability that the k^{th} string is selected for the next generation $\{\mathbf{S}_k(t+1)\}$ is proportional to its fitness $f(\mathbf{S}_k)$.
Step 2. Mutations. For each $k = 1, 2, ..., n$ and for each $i = 1, 2, ..., N$, the symbol $S_{ki}(t+1)$ is replaced with probability P_m by arbitrary symbol from the considered alphabet; P_m is the mutation intensity.
Step 3. Organization of the sequence of generations. Steps 1,2 are repeated for $t = 1, 2, ...$

decreases exponentially as the Hamming distance $\rho(\mathbf{S}, \mathbf{S}_m)$ between $\mathbf{S}$ and $\mathbf{S}_m$ (the number of noncoinciding symbols at the respective positions of these strings) increases:

$$f(\mathbf{S}) = \exp[-\beta\rho(\mathbf{S}, \mathbf{S}_m)], \quad (1)$$

where β is the parameter of selection intensity.

3. The evolution process consists of a number of generations. In each generation, selection of the organisms into the next generation (in accordance with their fitness) and mutations (replacements of symbols S_{ki} by random symbols from the considered alphabet) take place.
4. The string length N and the population size n in a certain evolution process are invariable and large: $N, n \gg 1$.

The formal scheme of the evolution process for the considered model is presented in Table 1.

It should be noted that the described model is characterized in full extent by the following parameters: $N, n, \beta, P_m, \lambda$.

3 Qualitative Picture of Evolution

In previous studies [15-17] and in this research, series of computer simulations have been done for the considered model. Computer simulations demonstrate that if the mutation intensity is sufficiently small ($\beta \geq P_m N, 1 \geq P_m N$), then the evolution can be characterized as follows (an example calculation is shown in Fig. 1):

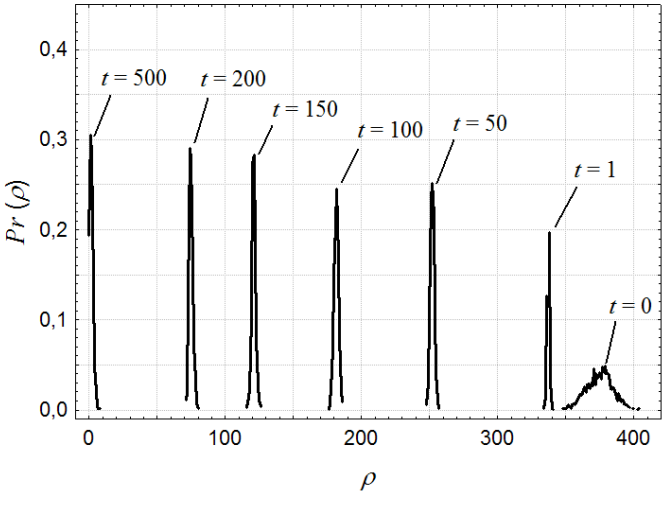


Fig. 1. The evolution of distribution $Pr(\rho)$ of Hamming distance ρ from the optimal string in the population. $Pr(\rho)$ is the fraction of organisms with certain value ρ in the population, t is the number of the generation. $\lambda = 4, N = 500, n = 500, \beta = 1, Pm = 1/N = 0.002$. The distribution for $t = 500$ corresponds to quasispecies.

- initial distribution $Pr_0(\rho)$ with respect to ρ in the population (for $t = 0$) is close to the normal distribution with mean $<\rho>= (\lambda-1)N/\lambda$ and variance $(\lambda-1)N\lambda^{-2}$ [16]; $<\rho>$ is the average (over the population) Hamming distance to the optimal string $\mathbf{S}_m$;
- the dynamics of the distribution $Pr(\rho)$ can be characterized by two stages, rapid and slow;
- during the first (rapid) stage, organisms from the left side of the initial distribution $Pr_0(\rho)$ are selected and the distribution $Pr(\rho)$ contracts;
- during the second (slow) stage, new organisms with smaller values of ρ can appear in the population only through mutations, and the distribution $Pr(\rho)$ drifts to small values of ρ with low rate;
- the final distribution characterizes the quasispecies, the quasispecies is the string distribution in a neighborhood of the optimal string $\mathbf{S}_m$;
- at small selection and mutation intensities ($1 \gg \beta \geq P_m N$), the quasispecies distribution $Pr(\rho)$ is close to the Poisson distribution with mean $(\lambda-1)P_m N/(\lambda\beta)$ [16].

4 The Role of Neutral Selection

In what follows, we assume that $\lambda^N >> n$, i.e., that the length N of each string $\mathbf{S}$ is sufficiently large. The number of strings corresponding to certain species in the population is not large, and many species are absent at all. For this reason, fluctuations of the number of species are essential, and the evolution processes under consideration have stochastic character. In particular, the neutral selection, i.e., the selection independent of fitness [14], must be taken into account.

To demonstrate the neutral selection impact explicitly, let's consider the following pure neutral evolutionary game:

There is a population of black and white balls; the total number of the balls in the population is equal to n. The evolution consists of a sequence of consequent generations. Each generation consists of two steps. At the first step all balls are doubled: each black ball has two black descendants, and each white ball has two white descendants. At the second step, precisely one half of the balls are randomly removed from the population, independently of their colors.

We say that the population is in l-state, if the numbers of black and white balls at a considered generation are equal to l and $n-l$, respectively. The evolution can be characterized by the probabilities P_{lm}, where P_{lm} is the probability of transition from l-state to m-state during one generation. Using straightforward combinatorial consideration, we can calculate values of P_{lm}:

$$P_{lm} = \begin{cases} \binom{2l}{m} \times \binom{2n-2l}{n-m} \Big/ \binom{2n}{n}, & \text{if } 2l-n \leq m \leq 2l; \\ 0, & \text{if } m > 2l \text{ or } m < 2l-n. \end{cases} \tag{2}$$

where $\binom{a}{b} = \dfrac{a!}{(a-b)!b!}$.

The matrix P_{lm} determines the random Markovian process that can be considered as a simple stochastic genetic process [18]. Using general methods of analysis of such processes [18], we can deduce that:

- the considered process always converges to one of the two absorbing states, namely, to 0-state (all balls are white), or to n-state (all balls are black);
- for large n , the characteristic number of generations T_n , required for the process to converge to either absorbing state, is equal to $2n$:

$$T_n = 2n. \tag{3}$$

Thus, although this evolution process is purely neutral (black and white balls have equal chances to survive), nevertheless only one species is selected. The value of T_n characterizes neutral selection rate, $T_n \sim n$.

5 Analytical Estimations

In this section we estimate the efficiency of the algorithm described above. We assume that the population size n is sufficiently large, i.e.:

$$T_n \geq T, \qquad [1-(1-P_c)^N]^n \ll 1, \tag{4}$$

where T_n is the characteristic time of neutral selection ($T_n \sim n$), T is the characteristic time (number of generations) of the convergence of the entire evolution process, P_c is the probability that a symbol S_{ki} is changed by mutations, $P_c = P_m(\lambda-1)/\lambda$. The first inequality in (4) means that the influence of the neutral selection is sufficiently small. The second inequality corresponds to ignorance of the mutation losses of already found "good organisms" in the population.

Let us estimate the characteristic convergence time T of the evolution process. The value of T is determined by the second (slow) stage of the evolution. At this stage, new strings with smaller values of ρ arise because of the mutations and are fixed in the population through selection. Let us estimate the characteristic time t_{-1} during which $<\rho>$ decreases by 1. This time can be approximated by the expression

$$t_{-1} \sim t_m + t_s, \tag{5}$$

where $t_m \sim (NP_m/\lambda)^{-1}$ is the characteristic mutation time during which the strings mutate in proper direction (namely, towards the optimal sequence) and $t_s \sim \beta^{-1}$ is the characteristic selection time during which the strings with $\rho = (<\rho> -1)$ replace the strings with $\rho =<\rho>$ in the population in the course of the selection. Setting $T \sim t_{-1}N$, we obtain

$$T \sim \lambda P_m^{-1} + N\beta^{-1}. \tag{6}$$

The total number of strings involved in the evolution is $n_{total} = nT$. Let us estimate the value of n_{total} for given N by choosing the parameters β, P_m, n so as to minimize n_{total}. We assume that the intensity of the selection is sufficiently

large, i.e., $\beta \geq P_m N/\lambda$; this allows us to ignore the second term in (6). Setting $P_c \sim N^{-1}$ and taking into account that $P_c = P_m(\lambda - 1)/\lambda$, we calculate P_m as $P_m = \lambda N^{-1}/(\lambda - 1)$. For this value of P_m, on the one hand, new strings appear in the population via mutations sufficiently quickly and, on the other hand, it is possible to ignore the mutation losses (the second inequality in (4) holds). Thus, we have $T \sim (\lambda - 1)N$. We also assume that the first inequality in (4) is valid at the utmost limit, i.e., $n \sim T_n \sim T \sim (\lambda - 1)N$; in other words, we assume that the population has minimal admissible size for which the loss of felicitous strings caused by neutral selection is inessential. With respect to all the assumptions we have:

$$T \sim (\lambda - 1)N, \qquad n_{total} \sim [(\lambda - 1)N]^2. \tag{7}$$

6 Results of Computer Simulations

In order to verify the analytical estimations (7), we investigated the dependence of the evolutionary search effectiveness on the string size N by computer simulations. The parameters were set to values corresponding to the conditions for obtaining the analytical estimates, namely: $P_m = \lambda N^{-1}/(\lambda - 1), \beta = 1, n = (\lambda - 1)N$. We made computations for $\lambda = 2$ and $\lambda = 4$.

The computations were organized as follows. The time dependencies $<\rho>(t)$ of the average (over the population) distance to the optimum were obtained for

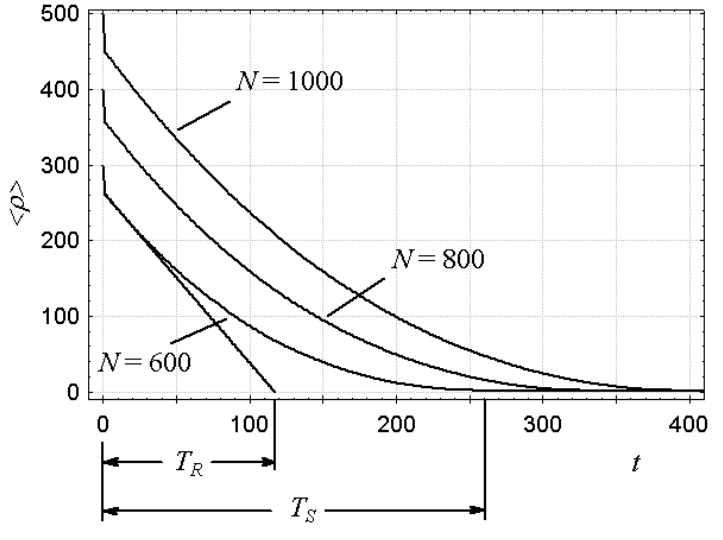


Fig. 2. The dependencies $<\rho>(t)$ for various values of N; $\lambda = 2$. Scheme of estimation of T_R and T_S values for $N = 600$ is shown. The dependencies are averaged over 50 independent runs.

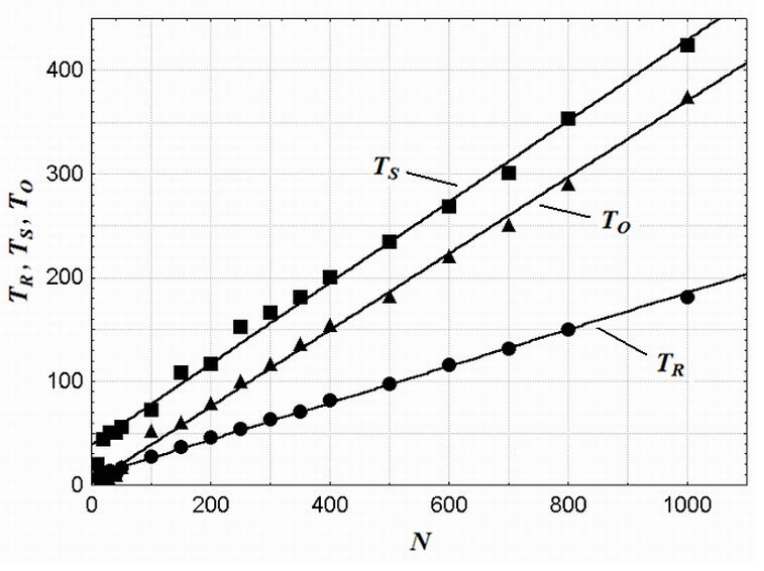


Fig. 3. The dependencies of relaxation time T_R , stabilization time T_S and time to find optimal solution T_O on the string length N for $\lambda = 2$. The dependencies are averaged over 50 independent runs.

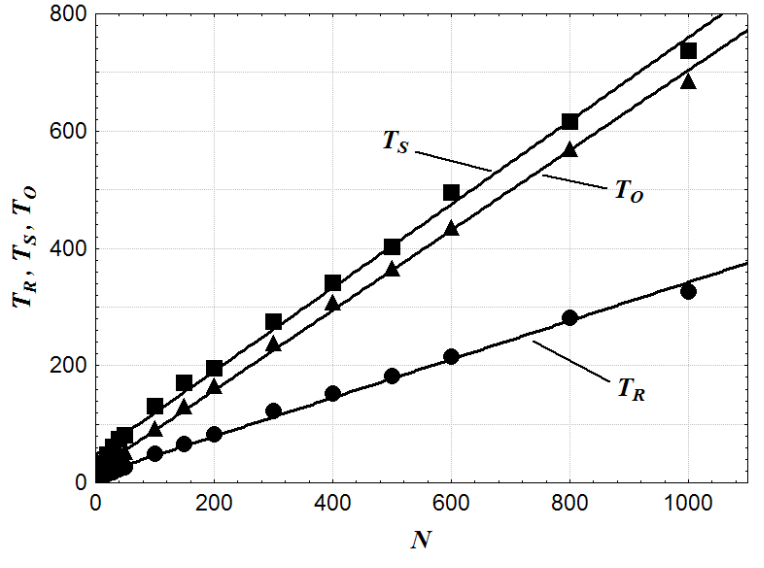


Fig. 4. The dependencies $T_R(N), T_S(N)$ and $T_O(N)$ for $\lambda = 4$. The dependencies are averaged over 50 independent runs.

various N (see Fig. 2). These dependencies were used to estimate the characteristic time T of convergence of the evolution in two ways:

1. The characteristic relaxation time T_R in the dependencies $<\rho>(t)$ was calculated from the initial slopes of these dependencies.
2. The time T_S of reaching the stationary value $<\rho>(t)$ attained at large t was evaluated.

Obtained dependencies $T_R(N)$ and $T_S(N)$ are presented in Fig. 3 (for $\lambda = 2$) and Fig. 4 (for $\lambda = 4$). In addition, the time T_O of the first appearance of the optimal string $\mathbf{S}_m$ in the population was determined. The resulting dependencies $T_O(N)$ are also shown in Figs. 3,4.

We see that, for sufficiently large N, all these dependencies are approximately linear: $T_R(N) = k_R N + T_{R0}, T_S(N) = k_S N + T_{S0}, T_O(N) = k_O N + T_{O0}$, where for $\lambda = 2 : k_R = 0.1772, k_S = 0.3903, k_O = 0.3685, T_{R0} = 8.2709, T_{S0} = 38.7356$, and $T_{O0} = 2.1288$; for $\lambda = 4 : k_R = 0.3283, k_S = 0.7113, k_O = 0.6826, T_{R0} = 13.7219, T_{S0} = 48.3387$, and $T_{O0} = 21.3483$.

These dependencies are in good agreement with estimates (7).

7 Discussion and Conclusion

Let us compare the evolution method of optimization of fitness function (1) under consideration with the two simplest methods, sequential search and random search. For simplicity we consider here the case of binary strings $\lambda = 2$.

We organize the sequential search as follows. We start from an arbitrary string $\mathbf{S}$ which symbols are $S_i = 1$ or -1. Then, for each i $(i = 1, 2, ..., N)$, we change the sign of the i^{th} symbol ($S_i \rightarrow$ -S_i). If the fitness $f(\mathbf{S})$ at this change increases, then we accept the new value of the symbol; otherwise, we return the old value S_i. As a result, after N tests, we obtain an optimal string $\mathbf{S}_m$. Thus, for the sequential search, the total number of strings, which should be processed before the optimal string $\mathbf{S}_m$ is found, is equal to $N : n_{total} = N$.

To find an optimal string by random search, the number of strings to be tested is of the order of $2^N : n_{total} \sim 2^N$.

The estimates obtained are given in Table. 2.

These estimates demonstrate that the evolution process as an optimization algorithm is "suboptimal": it does not ensure the maximal speed of search (for particular problems, more efficient algorithms are possible; in the case under consideration, such an algorithm is sequential search); nevertheless, it is much more efficient than random search. Since the evolution method of search is simple and universal, it can be qualified as a good heuristic optimization method for a large class of problems.

Note that, although the estimates were obtained for unimodal fitness function (1), similar estimates can be made for the spin-glass evolution model, in which the number of local maxima of the fitness function exponentially increases with the string dimension N [17].

Thus, efficiency of the evolutionary search in quasispecies model is estimated. It is shown that for the case of unimodal fitness function of λ-ary strings of

Table 2. Comparison of the search methods efficiency for the case $\lambda = 2$

Search method	n_{total}	n_{total} for $N = 1000$
Sequential	N	1000
Evolutionary	$\sim N^2$	$\sim 10^6$
Random	$\sim 2^N$	$\sim 100^{300}$

length N, the optimal string can be found during $(\lambda - 1)N$ generations under condition that the total number of fitness function calculations is of the order of $[(\lambda-1)N]^2$. These analytical estimations are confirmed by computer simulations.

Acknowledgements

This work was partially supported by the Russian Foundation for Basic Research (project no. 04-07-90038) and the program "Intelligent Computer Systems" of the Russian Academy of Sciences (project no. 2-45). The authors would like to thank the anonymous reviewers for many useful comments on an earlier version of this paper.

References

1. Holland, J.H.: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, MI. (1975), MIT Press, Boston (1992)
2. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley (1989)
3. Fogel, D.B.: Evolutionary Programming: an Introduction and Some Current Directions. Statistics and Computing, 4 (1994) 113-130
4. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press, Boston (1992)
5. Schwefel, H.-P.: Evolution and Optimum Seeking. Wiley Inc., New York (1995)
6. De Jong, K.: An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Doctoral dissertation, University of Michigan, Ann Arbor (1975) (University Microfilms No. 76-9381)
7. Goldberg, D. E., Deb, K., and Clark, J.H.: Genetic Algorithms, Noise and the Sizing of Populations. Complex Systems, 6 (1992) 333-362
8. Deb, K. Agrawal, S.: Understanding Interactions among Genetic Algorithm Parameters. In: Foundations of Genetic Algorithms 5. Morgan Kauffman, San Francisco (1999) 265-286
9. Muhlenbein, H.: How Genetic Algorithms Really Work I: Mutation and Hill-Climbing. In: Parallel Problem Solving from Nature, 2. Elsevier Science Publishing, Amsterdam (1992) 15-25
10. Jansen, T., Wegener, I.: On the Analysis of Evolutionary Algorithms - A Proof that Crossover Can Really Help. Technical Report CI-51/98, University of Dortmund, Dortmund (1998)
11. Eigen, M.: Selforganization of Matter and the Evolution of Biological Macromolecules. Springer-Verlag, Berlin Heidelberg New York (1971)

12. Eigen, M., Schuster, P.: The Hypercycle: A Principle of Natural Self-Organization. Springer-Verlag, Berlin Heidelberg New York (1979)
13. Fosterling, H.D., Kuhn H., Tews, K.H.: Computermodell zur Bildung Selbstorganisierender Systeme. Angewandte Chemie. 84 (1972) 862-865
14. Kimura, M.: The Neutral Theory of Molecular Evolution. Cambridge University Press (1983)
15. Red'ko V.G. Tsoy, Yu. R.: Estimation of the Efficiency of Evolution Algorithms. Doklady Mathematics (Doklady Akademii nauk). 72 (2005) 810-813
16. Red'ko, V.G.: Estimation of Evolution Rate in Eigen's and Kuhn's Models. Biofizika. 31 (1986) 511-516 (in Russian)
17. Red'ko, V. G. Spin Glasses and Evolution. Biofizika, 35(5) (1990) 831-834. (in Russian). Short version of this paper is published in Red'ko, V.G.: Spin-glass Model of Evolution. In: Heylighen, F., Joslyn, C., Turchin, V. (eds.): Principia Cybernetica Web (Principia Cybernetica, Brussels). (1998) URL: http://pespmc1.vub.ac.be/SPINGL.html
18. Karlin, S.: A First Course in Stochastic Processes. Academic Press, New York London (1968)

Partitioning of VLSI Circuits on Subcircuits with Minimal Number of Connections Using Evolutionary Algorithm*

Adam Słowik and Michał Białko

Department of Electronics and Computer Science, Technical University of Koszalin, ul. Śniadeckich 2, 75-453 Koszalin, Poland
aslowik@ie.tu.koszalin.pl

Abstract. In this paper, we present an evolutionary algorithm application to partitioning VLSI circuits on subcircuits with minimal number of connections between them. The algorithm is characterized by a multi-layer chromosome structure. Due to this structure, the partition of circuits is possible without applying a repair mechanism in the algorithm. The test circuits chosen from literature and created randomly are partitioned using proposed method. Results obtained by this method are compared with results obtained using a traditional Kernighan-Lin algorithm.

1 Introduction

With the quick development of VLSI integrated circuits, it is possible to place larger and larger numbers of electronic elements in a single integrated circuit. The problem of the circuit parititioning on subcircuits is still important, especially in early phases of layout design of integrated circuits [1]. The VLSI integrated circuits are partitioned into subcircuits in order to fulfill constraints connected with: a) the number of external connections of the integrated circuit, b) signal delays in the circuit, c) electrio-thermal interactions between elements, d) the size of the area for the circuits, and e) the testability of integrated circuits. The first constraint is essential, because the large number of external connections dramatically increases the overall expense of producing the circuit [2]. The reduction of the number of connections between subcircuits is a primary objective of partitioning algorithms. It is important from a few reasons. First, electric signal are delayed during transmission through external connections; second, external connections occupy large place on circuit printed boards; and third, external connections decrease reliability of the system [2]. The problem of the circuit partitioning into k subcircuits with minimal number of connections between them is NP-complete [3] thus, the typical approach to partitioning problems is an application of heuristic algorithms which allow to find suboptimal solutions. The

* This work was supported by the Polish Ministry of Scientific Research and Information Technology (MNiI) under Grant No. 3 T11B 025 29.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 470–478, 2006.
© Springer-Verlag Berlin Heidelberg 2006

most popular heuristics is Kernighan-Lin algorithm [4] and its modification proposed by Fiduccia-Mattheyses [5]. Recently, evolutionary algorithms are applied to partitioning problems of electronic circuits. This comes out of the fact that evolutionary algorithms are the very effective tools for solving difficult combinatorial problems. A few partitioning algorithms based on evolutionary algorithms are described in literature, including Raman [6], Vemuri [7], and Kozieł [1]. In the Kozieł paper [1], he has presented an algorithm for multiobjective partitioning of VLSI circuits based on evolutionary algorithm. This algorithm was based on single-layer chromosome representation of each individual (potential solution). However, this approach caused that in the case when the circuit was to be partitioned into k equal-number subcircuits, then a typical crossover operator created solutions that did not satisfy the constraints. In this article, we introduce the evolutionary algorithm with multi-layer chromosomes (similar to that applied in [10]) to partitioning of electronic circuits. Here, the circuit partitioning into k equal-number subcircuits is possible without applying additional repair algorithms, because each created solution is acceptable. Suitable genetic operators were proposed to this data structure. This algorithm (described in section 3) is named MLCEA-PP (*Multi Layer Chromosome Evolutionary Algorithm for Partitioning Problem*). Results obtained by this algorithm were compared with the results obtained using Kernighan-Lin algorithm.

2 Problem of VLSI Circuits Partitioning

Each electronic circuit can be modelled using the *G(V, E)* graph, where V represents a set of graph nodes, that is the set of electronic elements, and E corresponds to a set of graph edges, that is the set of connections between electronic elements in the circuit. In Fig. 1 an example of a digital circuit composed of 8 gates, and 9 connections between them, is presented.

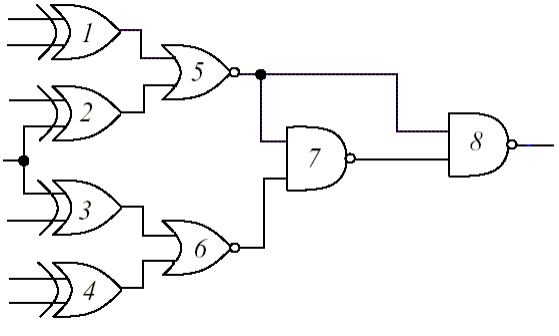


Fig. 1. Example of the digital circuit

In Fig. 2 the graph corresponding to the circuit shown in Fig. 1 is presented. This graph has 8 nodes, and 9 edges between them.

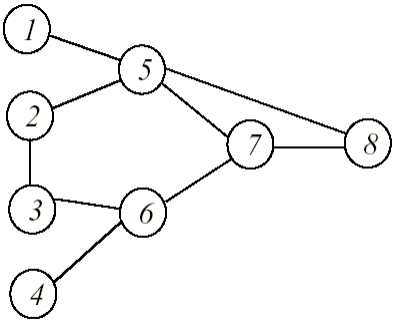


Fig. 2. Graph corresponding to the digital circuit of Fig. 1

The described problem corresponds to such partition of the set V of circuit nodes into k subsets (subcircuits) to optimize given objective function including possible constraints. It usually depends on minimization of the connection number between particular subcircuits (subsets). In Fig. 3 two examples of the circuit partitioning into two equal-number subcircuits (k=2) are presented.

Additionally, the graphs corresponding to the circuits of Fig. 3 are shown in Fig. 4.

The circuit (graph) partitioning presented in Fig. 3a (Fig. 4a) possesses four connections between subcircuits (subgraphs), however the partitioning shown in Fig. 3b (Fig. 4b) has only two connections between subcircuits (subgraphs). So, we can say that partitioning presented in Fig. 3b (Fig. 4b) is better. Generally an objective function in circuit partitioning problem is the sum of products of external graph edge, and weight value assigned to it. In the case when all weight values of the graph edges are equal to one, then the objective function is the number of the connections between particular subcircuits (subgraphs).

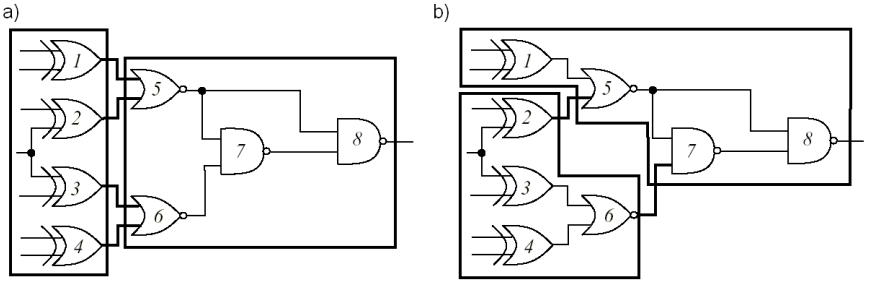


Fig. 3. Two examples of circuit partitioning into two equal-number subcircuits

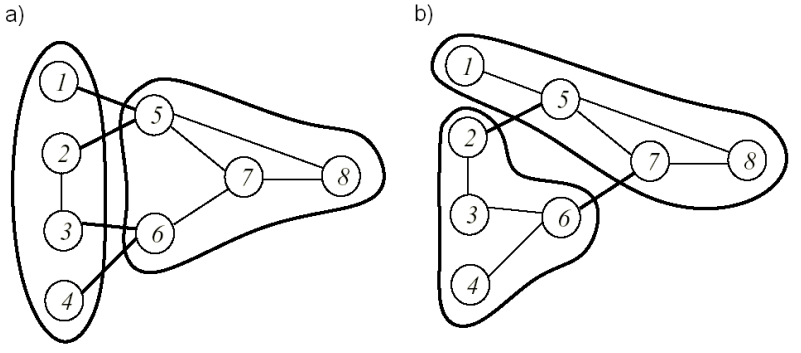


Fig. 4. Graphs partitioning corresponding to the circuits partitioning into two equal-number subcircuits from Fig. 3

3 MLCEA-PP Algorithm

3.1 Representation of Individuals

Each individual is represented by a multiple-layer chromosome which structure is shown in Fig. 5.

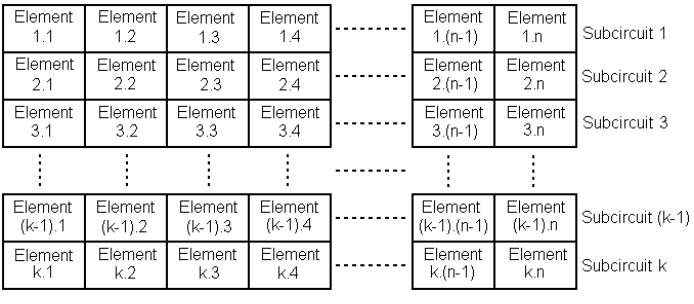


Fig. 5. Structure of the chromosome

The particular chromosome consists of k layers where each layer represents a single subcircuit (subset). With this representation, the circuit partitioning into k equal-number subcircuits each including n elements is possible without applying additional repair algorithms. Each solution is acceptable solution, because the constraints is "build-in" the proposed data structure. It is necessary to point out that the introduced structure is not limited only and exclusively to the circuit partitioning into k equal-number subcircuits. Introducing so called dummy nodes, that are not connected with any other node, enables a graph partitioning into k subgraphs with different number of elements. In Fig. 6, an example circuit partitioned into three subcircuits is presented.

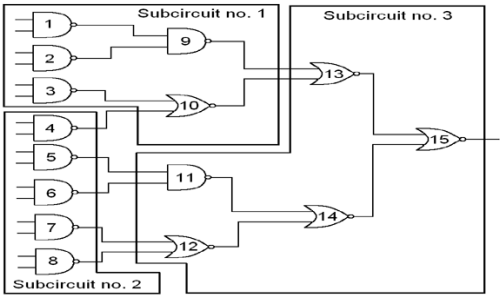


Fig. 6. Example circuit partitioned into three subcircuits

In Fig. 7 the graph corresponding to the circuit form Fig. 6 is shown.

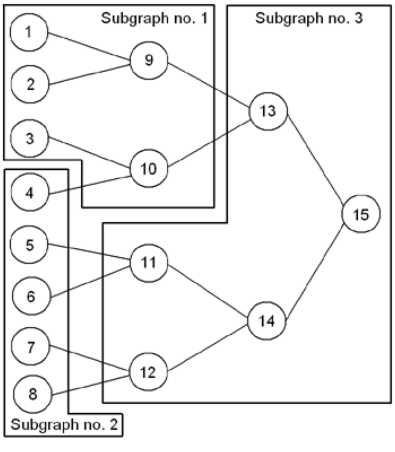


Fig. 7. Graph corresponding to the circuit from Fig. 6

In Fig. 8 the multi-layer chromosome representing the partitioned circuit (graph) from the Fig. 6 (Fig. 7) is presented.

1	2	3	9	10	Subcircuit no. 1 (Subgraph no. 1)
4	5	6	7	8	Subcircuit no. 2 (Subgraph no. 2)
11	12	13	14	15	Subcircuit no. 3 (Subgraph no. 3)

Fig. 8. Multi-layer chromosome representing the partitioned circuit

3.2 Used Genetic Operators

In the described algorithm, we use the well known PMX [11] crossover; however, we have modified it to make it suitable for operation on the multi-layer chromosome. This crossover operator has one important advantage: the child individuals that are obtained using it always represent acceptable solutions and additional repair algorithms are not required. In Fig. 9 the PMX [11] crossover procedure is presented.

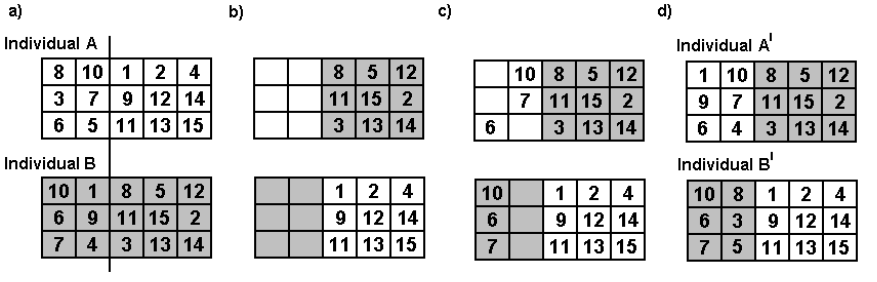


Fig. 9. PMX crossover operation

The crossover consists of three steps. In the first step, after choosing parent chromosomes and determining the point of the crossover for them (in the Fig. 9a crossing is after 2 gene) the code sequences after the cross-point are exchanged. As a result of this operation a pair of individuals presented in Fig. 9b is obtained. In the second step, we insert "not-colliding" genes from the parental individual into the blank places, (not-colliding genes, that is not existing still in the newly created child individuals). In the case of the individual A such genes are: "10", "7", and "6"; while for individual B the genes "10", "6", and "7" are inserted. After this operation the second step of crossing is finished, what is shown in Fig. 9c. In the third step, remaining vacancies are filled in the chromosomes according to the following rule. In the first gene of the first layer of the created individual A' the gene "8" would be inserted from parental individual A. However, it exists already in the newly created individual; therefore we exchange the genes according to the following rule: we are looking which gene from the created individual B' corresponds to the gene "8" from the created individual A'. The responding gene is "1" which does not exist in the individual A', therefore it is inserted to it. Following this way the genes: "9", and "5" are inserted to the individual A', and genes: "8", "3", "5" to individual B'. This completes the crossover, and as the effect of this operation is the pair of child individuals A', and B' presented in Fig. 9d. Besides the crossover, two mutation operators are introduced in the algorithm. After selection of a given gene for the mutation, the kind of mutation is determined randomly with equal probability. The first mutation operator choices randomly two genes from the parental individual which are exchanged during the mutation. In the relation to the electronic circuit (graph) this mutation causes exchange of one element between two subcircuits (subgraphs). This mutation operation is shown in Fig. 10.

Fig. 10. First mutation operation

The second mutation operation causes a circulation of the column (chosen randomly). In reference to the electronic circuit (graph) it causes the exchange of all subcircuits (nodes in subgraphs). The second mutation operation is shown in Fig. 11.

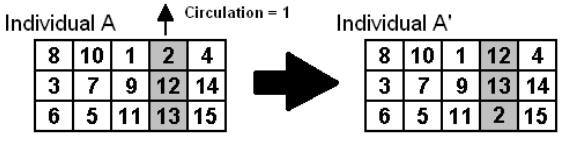


Fig. 11. Second mutation operation

3.3 Objective Function

The following objective function is used in the algorithm:

$$fc = \frac{NC}{NOC} \tag{1}$$

where: NC-number of all connections in the graph (the constant parameter for a given graph), NOC-number of all connections between created subgraphs (external connections). During operation of the algorithm the objective function is maximized. It is necessary to notice, that during the maximization of the objective function, the minimization of the number of external connections occurs (i.e. connections between created subgraphs, what is the main objective of this problem).

3.4 Description of the Algorithm

The following evolutionary algorithm was applied to circuit partitioning into k subcircuits with the minimal number of connections between them. At the start the initial population was created. Then, we check whether the algorithm converged (lack of change of the best solution). If the algorithm converged, the result represented by the best individual in the population is printed out and algorithm is stopped. If the algorithm did not converge, the following operations are used: crossover, mutation, calculation of value of the objective function for particular individuals, and the fan selection [8]. Then the algorithm convergence is checked again and the whole process is repeated.

4 Description of Experiments

During performed experiments the parameters of evolutionary algorithm were: population size=100, crossover probability=0.5, mutation probability=0.05, parameter of fan selection a=0.7. Results obtained by MLCEA-PP algorithm were compared with results obtained using the Kernighan-Lin (K-L) algorithm (Tab. 1). The circuits were partitioned into k equal-number subcircuits. A few test circuits were chosen to the experiments from [9] (they are marked with the symbol "*" in the Tab. 1); also randomly created graphs with different number of nodes and connections between were tested. It is assumed that each connection is equally important (i.e., in the graph all edges weights have value "1"). The experiments were repeated 10 times, and results obtained as an average values from 10 starts are shown in Tab. 1. The algorithm has converged and stopped after 100 generations for 10, 12, 24, 32 nodes graphs, after 400 generations for 48 nodes graphs, and after 1000 generations for 96 nodes graphs. In Tab. 1 the symbols used are as follows: NN-number of nodes, NC-number of connections (edges), k-the number of subcircuits (subsets), NOC-number of connections between subcircuits (external connections).

Table 1. Results of graph partitioning for algorithms: MLCEA-PP (M-PP), and Kernighan-Lin (K-L)

			M-PP	K-L					M-PP	K-L					M-PP	K-L
NN	NC	k	NOC	NOC		NN	NC	k	NOC	NOC		NN	NC	k	NOC	NOC
10*	21*	2	5	6.2		24	96	3	45.5	48.8		48	384	4	236	245
10*	21*	2	7	7		24	96	4	53.8	56.5		48	384	6	272.9	281.6
32*	54*	2	7.4	7.4		24	96	6	64.8	68.2		48	384	8	292.7	300.7
32*	54*	4	12.9	18.4		24	96	8	73	74.9		48	768	2	339.8	343
12	24	2	8.11	8.9		48	96	2	19.7	22.7		48	768	3	465	472.4
12	24	3	11	12.1		48	96	3	30.8	34.1		48	768	4	533	539.8
12	24	4	13.2	14.2		48	96	4	37.8	42.3		48	768	6	606.5	613.7
12	48	2	22	22		48	96	6	45.8	51.5		48	768	8	649	652.4
12	48	3	31	31.1		48	96	8	51.2	56.6		96	192	2	35.5	36
12	48	4	36	36.1		48	192	2	55.9	58.4		96	192	3	58.5	62.7
24	48	2	12.6	13.4		48	192	3	83.8	89.4		96	192	4	73.5	78.5
24	48	3	18.5	19.9		48	192	4	97.7	105.6		96	192	6	95.1	93.3
24	48	4	21.6	24.6		48	192	6	116.1	122.4		96	768	2	270.2	271.1
24	48	6	27.6	29.3		48	192	8	128	134		96	768	3	390.1	397.7
24	48	8	31.5	32.6		48	384	2	146.2	149.8		96	768	4	459.5	459.4
24	96	2	30.3	32.7		48	384	3	202.5	210.6		96	768	6	506	530.7

5 Conclusions

Results obtained for the algorithm MLCEA-PP are comparable or better than results obtained using the K-L algorithm. It is necessary to add, that in relation

to the algorithm of paper [1], in the presented algorithm MLCEA-PP any repair mechanisms were not used. Due to application of multi-layer chromosome, the crossover operator does not create unacceptable solutions or infringe the constraints.

References

1. Kozieł S., "Evolutionary algorithms and their applications to the optimisation and modelling of the analogue electronic circuits", Ph. D. Thesis, Technical University of Gdańsk, Department of Electronics, Telecommunications, and Computer Science, 1999,
2. Sait S. M., Youssef H., "VLSI physical design automation. Theory and practise", IEEE Press, New York, 1995,
3. Chen Y. P., Wang T. Ch., Wong D. F., "A Graph partitioning problem for multiple-chip design", Proceedings ISCAS 1993, pp. 1778-1781, 1993,
4. Kernighan B. W., Lin S., "An efficient heuristic procedure for partitioning graphs", Bell System Technical Journal, 49(2), pp. 291-307, 1970,
5. Fiduccia C. M., Mattheyses R. M., "A linear time heuristic for improving network partitions", Proceedings of the ACM/IEEE Design Automation Conference, pp. 175-181,
6. Raman S., Patnaik L. M., "Performance-Driven MCM Partitioning Through an Adaptive Genetic Algorithm", IEEE Transaction on VLSI Systems, Vol. 4, No. 4, December 1996,
7. Vemuri R., "Genetic algorithms for MCM partitioning", Electronic Letters, Vol. 30, No. 16, pp. 1270-1272, 1994,
8. Słowik A., Białko M., "Modified Version of Roulette Selection for Evolution Algorithm - The Fan Selection", Proceedings of 7th International Conference on Artifficial Intelligence and Soft Computing, ICAISC 2004, LNAI Vol. 3070/2004, pp. 474-479, Springer-Verlag,
9. Rutkowski J., Zieliński Ł., "Using evolutionary techniques for chosen optimalization problems related to analog circuits design", In Proceedings of ECCTD 2003 Conference,
10. Słowik A., Białko M., "Design and Optimization of Combinational Digital Circuits Using Modified Evolutionary Algorithm", Proceedings of 7^{th} International Conference on Artifficial Intelligence and Soft Computing, ICAISC 2004, LNAI Vol. 3070/2004, pp. 468-473, Springer-Verlag,
11. Goldberg D. E., Lingle R., "Alleles, Loci, and the TSP", Proceedings of the First International Conference on Genetic Algorithms, Lawrence Erlbaum Associates, Hillsdale, NJ, 1985, pp. 154-159

Genetic Approach to Modeling of a Dispatcher in Discrete Transport Systems

Tomasz Walkowiak and Jacek Mazurkiewicz

Institute of Computer Engineering, Control and Robotics, Wroclaw University of Technology, ul. Janiszewskiego 11/17, 50-372 Wroclaw, Poland
Tomasz.Walkowiak@pwr.wroc.pl, Jacek.Mazurkiewicz@pwr.wroc.pl

Abstract. The essential problem in transport system functional and economic analysis is a technique of modeling the human dispatcher presented in real systems. It is very hard to find an "intelligent" algorithm of dispatcher - an algorithm which is giving results which significantly differs from pure random algorithms. In a paper we propose an evolutionary approach to this problem. A set of heuristic rules for dispatcher is searched by genetic algorithm. The fitness function is defined by the economic measure. The discrete transport system is modeled using Monte-Carlo simulation. A more accurate model of dispatcher, found by genetic algorithm, allows to obtain more realistic results of functional and economic analysis. The proposed, novelty approach can serve for practical solving of essential decision problems related to an organization and parameters of transport systems.

1 Introduction

Modern transportation systems often have a complex network of connections [10]. From the reliability and functional point of view [8] the systems are characterized by a very complex structure of different elements. The transportation system are often driven by a dispatcher - an person which allocates vehicles to the tasks introduced into system. The dispatcher ought to take into account different features which describe the actual situation of all elements of transport systems. The modeling of transport systems with dispatcher is not a trivial challenge. The most effective method is to use a time event simulation with Monte Carlo analysis [3]. It allows to calculate any point wise parameters. We can also estimate different distributions of time being in a state or in a set of states and combine reliability measures with functional features and economic parameters. We have proposed the formal model of Discrete Transport System (DTS) [4] to analyze reliability and economic aspects of complex systems. Our previous works [12], [11] showed that it is very hard to find an "intelligent" algorithm of dispatcher - an algorithm which is giving results which significantly differs from pure random algorithms. In many papers, i.e. [2], it was shown that the evolutionary approach to challenging science or engineering problem gives very promising results. Therefore, in this paper we propose a novel approach for modeling a dispatcher in DTS by a usage of the genetic algorithm. Applying them requires a proper definition of a fitness function. Economic measures [11]

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 479–488, 2006.
© Springer-Verlag Berlin Heidelberg 2006

are very important in decision problems related to an organization of transport systems [9]. Therefore, we propose to define the fitness function by an economic measure of the DTS.

2 Discrete Transport System (DTS)

The model can be described as following quintuple [4], [11], [12]:

$$DTS = \langle N, R, V, T, M \rangle \tag{1}$$

where: N - set of nodes, R - set of routes, V - set of vehicles, T - set of tasks, M - set of maintenance crews.

Commodities: The single kind of a commodity is transported in the system. There are no parameters describing commodities.

Nodes: Nodes are inputs and outputs of the system. DTS nodes are described by no parameters.

Routes: Each route is described by following parameters: initial node u, terminal node v and length l which is the basis for journey time calculation if the vehicle velocity (when hauling the commodity and when empty):

$$R = \langle u, v, l \rangle \quad u, v \in N \quad l > 0 \tag{2}$$

Each route is a connection of two nodes in the directed graph which describes the topology of the network. The length of the route can be not equal to the length of the "opposite direction" route, because these routes are placed in the set independently. Each node is connected to all other nodes, if initial and terminal node is the same node the length of the route equals to zero. The DTS contains a map - 2-dimensional matrix (number of nodes $\times$ number of nodes) which describes all routes.

Vehicles: Each vehicle is described by following functional and reliability parameters: mean speed of a journey, capacity, reliability function and time of vehicle maintenance, initial node where the vehicle is available when the DTS observation begins. The temporary state of each vehicle is characterized by following data: vehicle state, distance traveled from the begin of the route, route where the vehicle is attached to, capacity of the commodity, capacity of already allocated commodity, task - currently executed by vehicle. The vehicle running to the end of the route is hauling only a single kind of a commodity. The vehicle hauling a commodity is always fully loaded or taking the last part of the commodity if it is less than its capacity.

Tasks: Each task can be described by: route for a commodity, amount of commodity to transfer, maximum number of vehicles which can be assigned to a given task, time when task is introducing into system, time when task begins, time when task ought to be finished. The temporary state of task is characterized by following data: remaining amount of commodity to transfer, amount of allocated commodity to vehicles, but not loaded yet, amount of commodity on the routes, number of vehicles temporary focused on the task.

Maintenance Crews: Maintenance crews are identical and unrecognized. The crews are not combined to any node, are not combined to any route, they operate in the whole system and are described only by the number of them. The temporary state of maintenance crew is characterized by: number of crews which are not involved into maintenance procedures, queue of vehicle waiting for the maintenance.

3 Simulator

A simulator, performing a Monte-Carlo simulation [3], is composed of four basic elements: input data file, system description language analyzer, event-driven simulator, output data file. The system description language analyzer creates, based on data taken from the input file objects which represent system in the memory. The event-driven simulator repeats N-times the following loop [5]:

1. state of a DTS initialization,
2. event state initialization, set time $t = 0$,
3. repeat until t less than T:
   - take first event from event list,
   - set time equals time of event,
   - realize the event.

The event is a basis for a simulation process. It is described by the following data: time of event occurring, type of event - vehicle failure for example, part of the DTS where event has its influence. The events are placed in the ordered list. Time of event occurring is the key for the list order. We have following events in the DTS: vehicle reached the node, vehicle is failing, vehicle is repaired, task is starting, end of simulation. [11]

After the end of a single simulation loop the following data are stored for each task: percentage of task realization, time of task end - if task is not realized in 100% this time equals to time of simulation, required time of task end - defined in input data. The set of data related to each vehicle stored after the end of single simulation loop envelops: number of repairs, number of unloads, number of loads, distance traveled hauling the commodity, distance traveled in empty state. There is also a special "null task" to aggregate data uncombined to any other tasks - failure of vehicle waiting in the node for example. Data stored in output file can be used for different measures calculations. In this paper we proposed to analyze the profit function - which is presented in the next chapter.

4 Economic Measures Calculation

The economic analysis is realized from vehicle owner's view-point [7]. Such discussion is the most justifiable because in real world the owner of vehicles is responsible for different agreements related to transportation tasks and he/she has to dispose his/her means in most sensible way. The economic quality of discrete transport system is described by "profit function" *P(T)* estimated in given time-period T as follow [5]:

$$P(T) = f(DTS, T) = RV(T) - C(T) - L(T), \tag{3}$$

where: *RV(T)* - revenue in time-period T, *C(T)* - system operating cost in time-period T, *L(T)* - loss in time-period T.

The system operating cost *C(T)* is modeled as follows [13]:

$$C(T) = \sum_{i=1}^{N_v} OC_i(T) + \sum_{i=1}^{N_v} VC_i \cdot T + MC \cdot T, \tag{4}$$

where: N_v - number of vehicles T, $OC_i(T)$ - vehicle i operating cost in time-period T, VC_i- vehicle i cost calculated based on leasing agreement, MC - cost of necessary maintenance agreement related to the whole system.

The vehicle operating cost $OC_i(T)$ in time-period T is calculated as a sum of partial products: cost of each kilometer traveled by single vehicle with and without the commodity, cost of each load and each unload of single vehicle, cost of each repair of single vehicle, cost of each hour of driver's work.

Loss *L(T)* is connected with penalties payed if tasks are not realized within required time slot. It is calculated in a simple linear model as a multiplication of a the penalty constant by delays of task realization (if any).

5 DTS Functioning Description

5.1 System Simulation

At the beginning DTS places vehicles to their initial nodes. There are no failed vehicles, all maintenance crews are available, the task list is empty. Tasks are entered to DTS during whole operation time, all of them can be also entered at the beginning moment. The available (empty) vehicles not allocated to any task are waiting in these nodes which were terminal in case of previously realized task. When the task is allocated, the empty vehicle runs from that node to initial node defined by allocated task - all possible routes are available. Then the vehicle is loaded up to its capacity or to exhaustion of commodity declared in the task and the vehicle runs to the terminal node of the task. If there are no available vehicles when the task appears, the task is placed into the queue. The task can also be placed into the queue if vehicles are available, but a dispatcher algorithm (see below for the dispatcher description) did not allocate vehicle to task realization. The vehicle can fail in any time - hauling the commodity or in an empty state. The failed vehicle is waiting for the maintenance crew - if the crew is not available at the moment of the failure. When the maintenance is finished the vehicle continues the task, if it has not attached to any task the dispatcher algorithm is called [5].

5.2 Dispatcher Algorithm

The dispatcher [11] is an algorithm which allocates vehicles to the tasks. Vehicles are attached to the already started tasks. The allocation decision is only taking

when: a new task is starting, a vehicle is unloading in a node - it is possible to reattach it to any task, a vehicle running for allocated commodity fails, maintenance procedure related to empty vehicle ends - it means that new vehicle is available in the node.

$$Dispatcher : DTS \rightarrow V \times T \tag{5}$$

Dispatcher algorithms consist of two combined decision functions which allocate vehicle or vehicles to a single task. Task allocation means that vehicle changes its state, receives information about allocated task and also we can observe task change: the amount of commodity to transfer decreases, the commodity is allocated to vehicle, but vehicle is still empty. Then the vehicle is traveling to initial node of a proper task (all routes ought to be available) or it does not change the node if the node where vehicle is present is initial node of the task. The vehicle is loaded in the initial node - its state changes from empty to full and starts the journey to terminal node.

The usage of two decision functions differs by a single argument describing which resource of DTS is released. The decision A is taken when any vehicle is available. The function called in two cases: when a vehicle reached the end of a route and was unloaded or when a vehicle is repaired and is not allocated to any task. The decision B is taken when any task is available. It happens in two cases: when a task is starting or when a vehicle fails during travel to initial node - already allocated commodity is available again as a part of task and this part of commodity ought to be allocated once more to another vehicle.

We have not described trivial situation - when there are no available vehicles and decision function has nothing to do. We describe below three different dispatchers.

5.3 Stochastic Dispatcher

In case of the decision A the task taken from the task list is randomly attached to a vehicle. We assume that the attached task is already started but not finished yet. Whereas in case of the decision B the vehicles with no attached task taken from available vehicle list are randomly attached one of available (already started) tasks. This function works until the number of vehicles is exhausted or task ends - there is no commodity to be taken. This simple algorithm works in a way that vehicles are attached to tasks by groups - we can say that every task "sucks" vehicles and therefore the other tasks have to wait for their turn.

5.4 Stochastic (Delta) Dispatcher with Multiple Choice in Bordered Time Horizon

Both decisions A and B are taken in the same way. All parallel tasks are grouped - tasks which start in bordered time horizon Δ and vehicles which in the bordered time horizon Δ are available it means reach terminal nodes of previous tasks.

The grouped tasks are operated by single dispatcher function call. Single vehicle and single task is randomly chosen. The vehicle is removed from list. If the task after drawing ends - there is no more commodity to be taken - the task is also removed from list. The procedure is repeated for the next pair: vehicle and task. The process ends when there are no more vehicles or no more tasks.

The process of vehicles attaching to tasks is finished, if the currently hauling commodity vehicle had been attached then this decision is cleared. Such solution ought to prevent the situation when the vehicle with next task attached failed before the final of the previous journey.

5.5 Global FIFO Dispatcher

FIFO dispatcher takes global decision how to distribute vehicle fleet to currently proceeded tasks. When the decision is taken next journeys of the vehicle - with following parts of commodity are deterministic and next dispatcher decisions are need-less. In case of decision A the FIFO dispatcher attachers following available vehicles to task until the vehicles run out or the commodity within single task exhausted. In case of decision B, if vehicle reaches the terminal node next part of commodity within the attached task is loaded to the vehicle, if the commodity within attached task exhausted next global dispatcher decision ought to be taken - appearing available vehicles are attached to next task until the vehicles run out or commodity exhausted.

5.6 Dispatcher Algorithm Selection

All mentioned dispatcher algorithms has been implemented and tested in different exemplar DTS systems [11], [12]. However, in almost cases, most of the results (i.e economic measures) differs only a bit from the most simple stochastic dispatcher with attached vehicles to running tasks in a purely random way. Therefore, there is a need to develop a new dispatcher algorithm.

6 Evolutionary Dispatcher

As it was mentioned in the introduction the main idea in this paper is to develop the dispatcher algorithm by means of the genetic algorithm. One could imagine a large set of heuristic rules for the dispatcher. However, it is hard to decide a-priori which rules should be used. Therefore, we decided to describe a set of rules designed by human expert in IF *condition* THEN *statement* form. Then we applied the classical genetic algorithm to select the best set of rules. The chromosome consists of 8 IF-THEN rules encoded in a binary form (1 bit for information if a rule is on or off, 3 bits for eight *condition*s and next 3 bits for different *statement* rules). The fitness was defined as an average profit (3) gained in an exemplar DTS (described in the next section). For each set of rules analyzed by genetic algorithm the Monte-Carlo simulation of DTS was performed and a profit function was calculated for each loop of simulator.

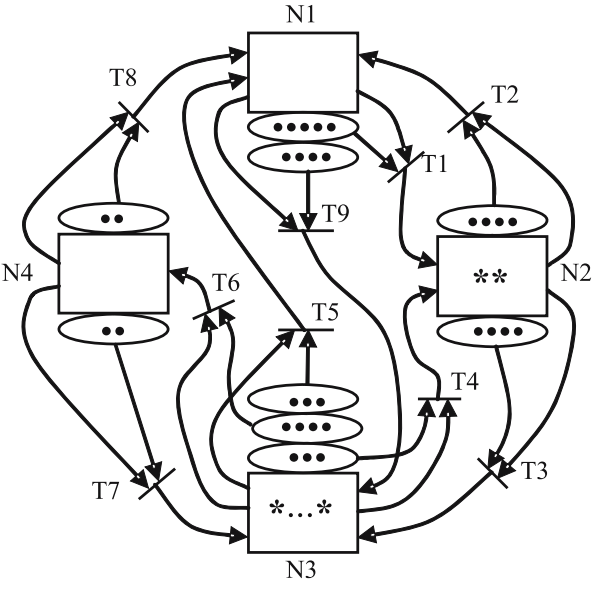


Fig. 1. Exemplar Discrete Transport System - routes and tasks

7 Case Study

Performance of the proposed evolutionary approach in a compare to the stochastic dispatcher was tested on an exemplar DTS illustrated in Fig. 1. The system is composed of four nodes numbered from $N1$ to $N4$. The nodes are begins or ends of 12 routes with defined lengths described in Tab. 1. The system is equipped by

Table 1. Routes defined in exemplar DTS

Route	Initial node	Terminal node	Length
$R1$	$N1$	$N2$	100
$R2$	$N2$	$N1$	100
$R3$	$N2$	$N3$	200
$R4$	$N3$	$N2$	200
$R5$	$N3$	$N1$	300
$R6$	$N3$	$N4$	300
$R7$	$N4$	$N3$	300
$R8$	$N4$	$N1$	400
$R9$	$N1$	$N3$	300
$R10$	$N1$	$N4$	400
$R11$	$N4$	$N2$	400
$R12$	$N2$	$N4$	400

Table 2. Tasks defined in exemplar DTS

Tasl	Start time	End time	Route	Volume
$T1$	10	30	$R1$	50
$T2$	5	20	$R2$	40
$T3$	10	30	$R3$	40
$T4$	2	60	$R4$	30
$T5$	20	55	$R5$	30
$T6$	10	30	$R6$	40
$T7$	5	45	$R7$	20
$T8$	7	70	$R8$	20
$T9$	20	65	$R9$	40

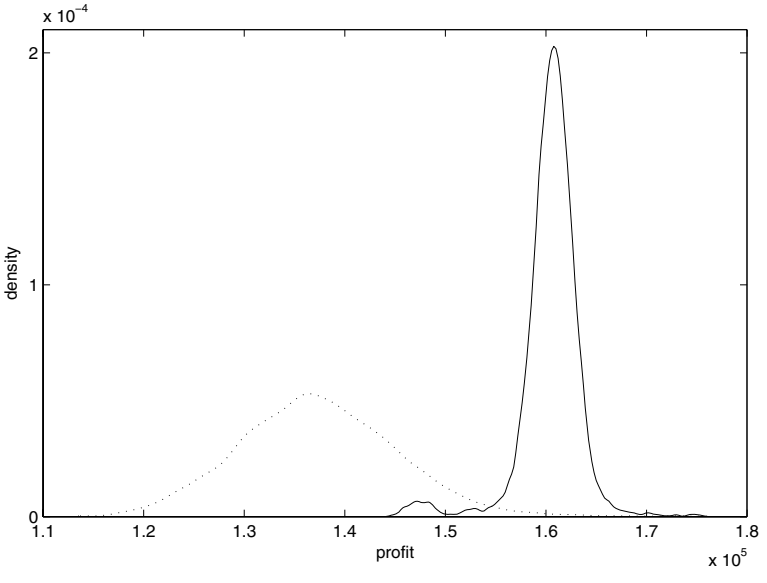


Fig. 2. Probability distribution of the profit function the for stochastic dispatcher (dotted line) and the evolutionary dispatcher (solid line)

nine tasks - and 10 identical vehicles presented . Vehicles are described by stars, the volumes of tasks are described by dots. At the beginning of simulation there are two vehicles in $N2$ and five vehicles in $N3$. We use for illustration Petri net like methodology. The single journey related to single task can be started (the transition is fired) if a pair: star and dot is available in the initial node. It means there is a vehicle available and a commodity - single dot describes amount of commodity which equals to vehicle capacity. This way dots are step by step eliminated - this process corresponds to task realization, the position of stars changes - it means

that vehicles are available in different nodes. Tasks parameters are presented in Tab. 2. All vehicles travel with the same velocity: speed hauling commodity equals to 45 units, speed in empty state equals to 80 units.

8 Results

The achieved results, the distribution of the profit function (3), for the stochastic dispatcher (dotted line) and the dispatcher "designed" by genetic algorithm (solid line) is presented in Fig. 2. It could be noticed, that the evolutionary dispatcher is giving a higher profit, in average more then 16%. Whearas the manualy designed dispatchers were better in an average over the stochastic dispatcher not more than 5%. We think, that the results are promising but a larger number of tests is needed. The evolutionary dispatcher was trained (i.e. dispatcher rules were selected) and tested on the same exemplar DTS. It is worth to investigate how dispatcher rules selected by genetic search on one exemplar DTS performs in different DTS. Moreover, it is worth to develop a more complicated set of rules analyzed by genetic algorithm as well as to use genetic programming [6].

9 Conclusion

Summarizing, we have proposed a new method of transport system modeling for economic analysis. It is based on evolutionary modeling of dispatcher rules to achieve a dispatcher which gives better economic results. We hope that the evolutionary algorithm is able to "design" a dispatcher which is able to mimic the behavior of the real dispatcher - human being. The presented approach could be used as a foundation for a new methodology of the functionality and economic analysis of transport systems with dispatcher, which is much closer to the practice experience.

Acknowledgment. Work reported in this paper was sponsored by a grant No. 5 T12C 021 25, (years: 2003-2006) from the Polish Committee for Scientific Research (KBN).

References

1. Barlow, R., Proschan, F.: Mathematical Theory of Reliability. Society for Industrial and Applied Mathematics, Philadelphia (1996)
2. Corne, D.W., Fogel, G. (eds.): Evolutionary Computation in Bioinformatics. Morgan Kaufman Publishers (2003)
3. Fishman, G.: Monte Carlo: Concepts, Algorithms, and Applications. Springer-Verlag (1996)
4. Jarnicki, J., Mazurkiewicz, J., Zamojski, W.: Model of Discrete Transport System (in Polish). XXX Winter School of Reliability, Poland, Szczyrk (2002) 149–157
5. Kaplon, K., Mazurkiewicz, J., Walkowiak, T.: Economic Analysis of Discrete Transport Systems. Risk Decision and Policy **8** (2003) 179–190

6. Koza, J.: Genetic Programming: On The Programming of Computers by Means of Natural Selection. MIT Press (1992)
7. Mazurkiewicz, J., Walkowiak, T.: Fuzzy Economic Analysis of Simulated Discrete Transport System. Artificial Intelligence and Soft Computing - ICAISC 2004, Springer-Verlag, LNAI 3070 (2004) 1161–1167
8. Podofillini, L., Zio, E., Marella, M.: A Multi-State Monte Carlo Simulation Model of a Railway Network System. Advances in Safety and Reliability - ESREL 2005, Taylor & Francis Group (2005) 1567–1575
9. Pozsgai, P., Bertsche, B.: Modeling and Simulation of the Operational Availability and Costs of Complex Systems - a Case Study. Advances in Safety and Reliability - ESREL 2005, Taylor & Francis Group (2005) 1597–1605.
10. Sanso, B., Milot, L.: Performability of a Congested Urban-Transportation Network when Accident Information is Available, Transportation Science **1, 33** (1999)
11. Walkowiak, T., Mazurkiewicz, J.: Reliability and Functional Analysis of Discrete Transport System with Dispatcher. Advances in Safety and Reliability - ESREL 2005, Taylor & Francis Group (2005) 2017–2023
12. Walkowiak, T., Mazurkiewicz, J.: Simulation Based Management and Risk Analysis of Discrete Transport Systems. IEEE TEHOSS 2005 Conference, Poland (2005) 431–436
13. Walkowiak T., Mazurkiewicz J.: Fuzzy approach to economic analysis of dispatcher driven discrete transport systems. DepCoS-RELCOMEX'06 International Conference, IEEE Press, Poland, Szklarska Poreba (2006) (accepted for publication)

Interactive Analysis of Preference-Ordered Data Using Dominance-Based Rough Set Approach

Jerzy Błaszczyński[1], Krzysztof Dembczyński[1], and Roman Słowiński[1,2]

[1] Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland
{jblaszczynski, kdembczynski, rslowinski}@cs.put.poznan.pl
[2] Institute for Systems Research, Polish Academy of Sciences, 01-447 Warsaw, Poland

Abstract. We present a method of interactive analysis of preference ordered data that is based on Dominance-based Rough Set Approach (DRSA). The presented here methodology is conceptually similar to multi-dimensional reports (pivot tables) applied in On-Line Analytical Processing (OLAP). However, it allows to identify patterns in data that remain undiscovered by traditional approaches to multi-dimensional reporting. The main difference consists in use of specific dimensions and measures defined within DRSA. The method permits to find a set of reports that ensures specified properties of analyzed data and is optimal with respect to a given criterion. An example of reports generated for a well-known breast cancer data set is included.

1 Introduction

The method presented in this paper originates from On-Line Satisfaction-Analysis (OLSA) that was introduced to analyze customer satisfaction data. Some of basic elements of this method were already described in [1]. It has also some common elements with the work by Michalski [8] that extended the concept of a Karnaugh-like map to visualize non-Boolean functions. Similar problem was also addressed by Kohavi and Sommerfield [7]. They present decision tables in a spreadsheet-like manner. Kohavi and Sommerfeld pointed that business users of data mining tools *found the projections of multi-dimensional cubes into spreadsheets very intuitive.*

Our method, based on Dominance-based Rough Set Approach (DRSA) introduced by Greco et al. [3,4,5,6], differs from those approaches because it is applicable to preference-ordered data and it can be used to discover inconsistencies coming from violation of the dominance principle. Depending on the character of decision problem described by ordered data, the order can either be interpreted as a preference order or a general order. In the first case, it is meaningful to consider semantic correlation between preference order on condition and decision side of object description, e.g. "the more the car is speedy, the more it is preferred". In the second case, the general order makes meaningful just a monotonic relationship between the order on condition and decision side of object description, e.g. "the greater the clump thickness, the greater the

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 489–498, 2006.
© Springer-Verlag Berlin Heidelberg 2006

malignity of the tumor". In this paper, we will use terms related to preference order, however, the second meaning would be equally as good. We believe that the method that takes into account orders occuring in the data allows finding more general dependencies.

One of the first steps of the method is transformation of the original decision table containing input data into a pivot table that is referred to as *decision cross-table*. Moreover, we have defined an optimal set of decision cross-tables that preserves quality of approximation of the original decision table. This set is also optimal with respect to complexity of included decision cross-tables. Such a set is obtained by solving a 0-1 mathematical programming problem.

Let us notice that the approach by Kohavi and Sommerfield and our method presented here has a lot in common with On-Line Analytical Processing (OLAP). OLAP allows pivoting data around different dimensions and drilling down and rolling up the multi-dimensional cube. However, classical aggregations functions are here substituted by functions aggregating a decision value with respect to the chosen dimensions.

The structure of the paper is as follows. In section 2, a brief reminder of DRSA and basic definitions of Interactive Analysis of Preference-Ordered Data are presented. In section 3, a problem of finding an optimal set of decision cross-tables is defined. Section 4 shows results obtained by the presented method applied to well-known Wisconsin breast cancer data. The last section concludes the paper and gives outlines of future research directions.

2 Interactive Analysis of Preference-Ordered Data

In this section we propose a methodology called Interactive Analysis of Preference-Ordered Data (IAPOD). It extends the dimensional modeling of data in a specific way for dealing with ordered data.

A decision table $S = \langle U, C, D \rangle$ is 3-tuple, where U is a set of objects described by set C of condition criteria (i.e., attributes with a preference-ordered domains) and D is a set of decision criteria (for simplicity, it is assumed that $D = \{d\}$). Domain of a criterion $c_i \in C$, $i = 1, \ldots, n$, is defined as $V_{c_i} = \{c_i^1, c_i^2, \ldots, c_i^m\}$. For simplicity, it is assumed without loss of generality that domains of criteria are numerically coded with an increasing order of preference. Value of an object $x \in U$ on criterion c_i is denoted by $c_i(x)$. Domain of a decision criterion d is defined analogously to domains of condition criteria. Value of an object $x \in U$on decision criterion d is denoted by $d(x)$. Let us notice that d induces a partition of U into a finite number of classes $Cl(D) = \{Cl_t, t \in V_d\}$. It is assumed that the classes are preference-ordered according to an increasing order of class indices (i.e., for all $r, s \in T$, such that $r > s$, the objects from Cl_r are strictly preferred to the objects from Cl_s).

Given a decision table S, one may construct a decision cross-table $DCT(S)$, shortly DCT. It is defined as 4-tuple $DCT = \langle Dim, Con, Dec, Mes \rangle$, where Dim is a subset of condition criteria C that are referred to as *dimensions*. Dim consists of $\{c_1, c_2, \ldots, c_k\} \subseteq C$ condition criteria. It is said that the decision

cross-table DCT is of a degree k, where k is a number of dimensions. Con is a set of elementary conditions that constrains set U, the resulting set of objects is then denoted U^{Con}. Dec is a subset of decision criteria D (here, $Dec = \{d\}$). Mes is a set of measures that are defined later in the paper.

DCT can be presented as multi-criteria report spanned over condition criteria. In such a report, columns and rows correspond to dimension's values or combinations of dimension's values. For each of those combinations, a specific measure aggregating decision value is computed and set into a cell inside a multi-criteria report. Such a report corresponds in many ways to planar diagrams defined by Michalski [8], classical cross-tables, cross-tables defined by Kohavi and Sommerfield [7] and OLAP reports. However, it is different by the fact that it is used to present multi-criteria space. The measures that we use to represent DCT are divided into two groups. The first group consists of typical measures and statistics, such as number of objects described by a given combination of dimension's values (i.e., count). The second group of measures is defined in the framework of DRSA. We refer to [3,4,5,6] for a detailed description of DRSA. Here we introduce only notions that are necessary for further presentation.

We will consider a function called *generalized decision* [2] defined for object $x \in U$ as:

$$\delta_P(x) = \langle l_P(x), u_P(x) \rangle, \tag{1}$$

where

$$l_P(x) = \min\{d(y) : y \in U, yD_Px\},$$
$$u_P(x) = \max\{d(y) : y \in U, xD_Py\}.$$

In the above, D_P is a dominance relation defined as $xD_Py \Leftrightarrow c_i(x) \geq c_i(y), \forall c_i \in P$, $P \subseteq C$. In this way, we have defined an interval of decision values of object x taking into account all of inconsistencies coming from violation of the dominance principle. This principle requires that object x having not worse values of condition criteria than object y, has not worse decision value than y. This relates to assignment of object x to approximations of unions of decision classes in the following way. $l_P(x)$ represents the highest upward union of decision classes to which lower approximation object x belongs (i.e., pessimistic assignment of object x). $u_P(x)$ represents the lowest downward union of decision classes to which lower approximation object x belongs (i.e., optimistic assignment of object x). Lower dominance-based rough approximations are defined as [4]:

$$\underline{P}(Cl_t^{\geq}) = \{x \in U, \{y \in U : yD_Px\} \subseteq Cl_t^{\geq}\},$$
$$\underline{P}(Cl_t^{\leq}) = \{x \in U, \{y \in U : xD_Py\} \subseteq Cl_t^{\leq}\},$$

where $Cl_t^{\geq} = \{x \in U : d(x) \geq t\}$ and $Cl_t^{\leq} = \{x \in U : d(x) \leq t\}$ are called upward and downward union of decision classes, respectively. Correspondence between generalized decision and lower dominance-based rough approximations is the following:

$$l_P(x) = \max\{t : x \in \underline{P}(Cl_t^{\geq})\},$$
$$u_P(x) = \min\{t : x \in \underline{P}(Cl_t^{\leq})\}.$$

The process of finding dependencies in data in presented here context consists in specifying selections of objects from decision cross-tables. This selections are specified by complexes that are composed of selectors. Let us define a complex by $\Phi = \phi_1^{\propto} \wedge \phi_2^{\propto} \wedge \ldots \wedge \phi_l^{\propto}$, where $\phi_i^{\propto}$ is a selector defined as $c_i(x) \propto c_i^j$, $c_i^j \in V_{c_i}$; and $\propto$ is specified as $=, \geq, \leq$. Each selector $\phi_i^{\propto}$ is defined on dimension $c_i \in Dim$ and corresponds to rows or columns of the multi-criteria report. In other words, complex Φ is a set of selectors that allows to select a subset of objects. Objects covered by complex Φ are denoted by $[\Phi]$ and referred to as cover of a complex Φ.

Let us now define the above mentioned measures *Mes* more formally:

- *count*: $count(\Phi^{\propto}) = card([\Phi^{\propto}])$, counts the number of objects that are covered by the complex $\Phi^{\propto}$,
- *exact upward decision*: $d^{\geq}(\Phi^{=}) = min\{l_{Dim}(x) : x \in [\Phi^{=}]\}$,
- *exact downward decision*: $d^{\leq}(\Phi^{=}) = max\{u_{Dim}(x) : x \in [\Phi^{=}]\}$.

In the definition above $\Phi^{=} = \phi_1^{=} \wedge \phi_2^{=} \wedge \ldots \wedge \phi_l^{=}$. This implies a complex that selects only objects that have exactly the same description (i.e., have equal values on each criterion). The first measure *count* is a standard one that is usually defined for OLAP-like analysis. It is useful to differentiate facts that are frequent (i.e., complexes that cover high number of objects) from those that are not. However it says nothing about dependencies between condition and decision criteria and inconsistencies in analyzed data. The further two measures are defined in the DRSA context. These measures indicate the relation between values on condition criteria and the value on decision criterion. Comparison of those two measures is useful to detect inconsistencies between objects.

To show an example of DCT, let us consider the problem of Wisconsin breast cancer prediction (BCW) [10]. In this problem, we have to do with general order in data and monotonic relationship between conditions and decision, however, we keep the vocabulary adopted in this paper, i.e., the preference terms. This data set consists of objects that are assigned to two decision classes *benign* (denoted by value 2) and *malignant* (denoted by value 4). Objects are described by the following condition criteria: Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Mitoses.

An example of two dimensional report generated for those data is presented in Figure 1. In the case of this report exact upward decision measure is used. In cells, for which $[\Phi^{=}] = \emptyset$, the value is computed as follows. Let us assume that for each cell that is not empty, a DRSA decision rule is created, with a condition part corresponding to complex $\Phi^{\geq}$ and decision part in the form $d(x) \geq d_i^{\geq}(\Phi^{=})$. In other words, the decision rule is as follows: $\Phi^{\geq} \Rightarrow d(x) \geq d^{\geq}(\Phi^{=})$. The value of cells, for which $[\Phi^{=}] = \emptyset$, is then set to maximal decision suggested by decision rules covering those cells (i.e., a piece of multi-criteria space determined by complex $\Phi^{=}$). In the figures, these cells are marked with lighter colors. For example, one can find the following decision rule in the report presented in Figure 1:

$$Clump\ Thickness(x) \geq 7 \wedge Single\ Epithelial\ Cell\ Size(x) \geq 6 \Rightarrow d(x) \geq 4.$$

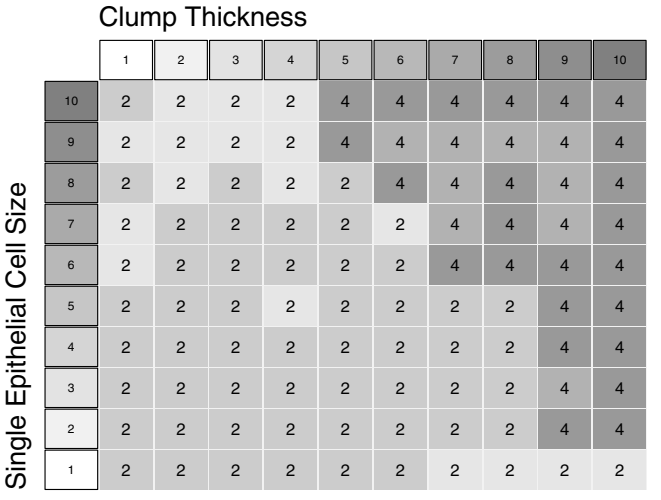


Fig. 1. Multi-criteria report including two criteria describing BCW problem (Clump Thickness, Single Epithelial Cell Size). The measure used is the exact upward decision $d^{\geq}(\Phi^{=})$.

Cells in the report in Figure 1 denoted by $d^{\geq}(\Phi^{=}) = 4$ represent objects for which one can be certain that their decision value is $d(x) = 4$. It is worth to observe that all cells denoted by $d^{\geq}(\Phi^{=}) = 2$ represent objects that have decision values $d(x) = 2$ as well as those that have $d(x) = 4$. To be certain which cells represent objects with decision value $d(x) = 2$, one have to generate a report using exact downward decision measure. Such a report is presented in Figure 2. In this report, all cells denoted by $d^{\leq}(\Phi^{=}) = 2$ represent those objects for which one can be certain that they have decision values $d(x) = 2$. On the other hand, all cells denoted by $d^{\leq}(\Phi^{=}) = 4$ represent on this report objects that have decision values $d(x) = 4$ as well as those that have $d(x) = 2$. For example, one can find the following decision rule in the report presented in Figure 2:

$$\textit{Clump Thickness}(x) \leq 1 \wedge \textit{Single Epithelial Cell Size}(x) \leq 3 \Rightarrow d(x) \leq 2.$$

Cells that have different values on these reports ($d^{\geq}(\Phi^{=}) = 2$ and $d^{\leq}(\Phi^{=}) = 4$) represent objects that are inconsistent with respect to criteria *Clump Thickness* and *Single Epithelial Cell Size*. It means that there are objects assigned to class 4 that have values on these criteria that are not better than values of objects corresponding to these cells assigned to class 2. Of course, the same occur in the opposed direction. There exist objects assigned to class 2 that have values on these criteria that are not worse than values of objects corresponding to these cells assigned to class 4. For example, objects for which $\textit{Clump Thickness}(x) = 5 \wedge \textit{Single Epithelial Cell Size}(x) = 5$ are inconsistent. However, objects that are inconsistent on one pair of downward and upward exact decision reports can become consistent with respect to other pair of reports (that correspond to

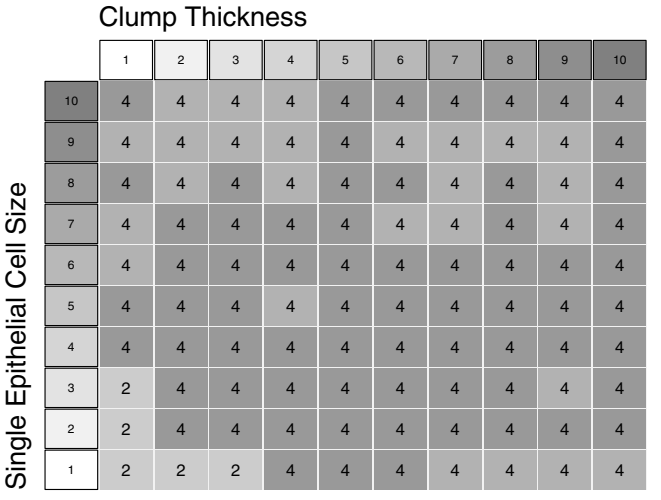


Fig. 2. Multi-criteria report including two criteria describing BCW problem (Clump Thickness, Single Epithelial Cell Size). The measure used is the exact downward decision $d^{\leq}(\Phi^{=})$.

different dimensions). In other words, those objects when present on such other pair of reports can have the same value of decision.

We can show that in case of BCW data set there exists monotonic dependence between condition criteria and decision class. It is reflected by gradual change of belonging to decision class with change of criterion value. This kind of dependency can by shown for all condition criteria in this data set.

3 Optimal Set of Decision Cross-Tables

Let us formulate the following problem. Find a set of decision cross-tables that is minimal with respect to complexity of these tables and maintains some properties of a original decision table. The complexity of cross-tables may be defined as a total number of cells computed as a sum of cells of each decision cross-table in the set. A property that is maintained by the set of decision cross-tables can be, for example, the same quality of approximation as for the original decision table. Let us remind, that the quality of approximation is defined as a ratio of number of objects from the original decision table that are consistent with respect to all criteria to number of all objects from this decision table. The obtained set is referred to as an *optimal set of decision cross-tables*.

The above formulation is in some sense similar to the problem of reducing decision tables and inducing decision rules. A set of decision cross-tables maintains the quality of approximation and minimizes the complexity of the set of decision cross-tables. Moreover, it is easy to identify decision rules in each of decision cross-tables.

The problem of finding optimal decision cross-tables can be expressed in terms of 0-1 mathematical programming. In fact, this formulation is not efficient and naïve, however, our objective is to define the problem and to give, at the beginning, its simplest solution. Let us assume that $P_1, P_2, \ldots, P_{2^{card(C)}-1}$ is a sequence of all subsets of the set of condition criteria C excluding an empty set. The solution is to minimize:

$$\min \rightarrow \sum_{j=1}^{2^{card(C)}-1} w_j \cdot x_j, \quad \text{subject to:} \quad \mathbf{Ax} \geq \mathbf{1}$$

where x_j is decision variable ($\mathbf{x}$ is a vector containing all decision variables) that indicates a decision cross-table based on a subset of condition criteria $P_j \subseteq C$, w_j is a cost of a decision cross-table, computed as $\prod_{c_l \in P_j} card(V_{c_l})$ and $\mathbf{A}$ is a matrix with $2^{card(C)} - 1$ (!) columns and $card(U)$ rows defined as:

$$a_{ij} = 1, \text{ if } l_{P_j}(x) = u_{P_j}(x), \text{ and } a_{ij} = 0, \text{ otherwise,}$$

where P_j is the j-th subset of condition criteria.

To constrain the complexity of the problem one may choose only decision cross-tables up to degree $k = card(P_j)$ (let $k = 3, 4, 5$). In such a case, however, the quality of approximation for a given set of decision cross-tables may be lower than for the original decision table. Note that, the quality of approximation for a decision table S is defined as:

$$\gamma(C) = \frac{card(\{x \in U, u_C(x) = l_C(x)\})}{card(U)}.$$

We define the quality of approximation for a set of decision cross-tables up to degree k:

$$\gamma(k) = \frac{\sum_{i=1}^{card(U)} \max_{j=1,\ldots,c}\{a_{ij}\}}{card(U)},$$

where P_j, $j = 1, \ldots, c$ is a sequence of subsets of condition criteria containing up to k criteria.

4 Example of Application

In this section we present results obtained by using the presented methodology to BCW data set. The analysis was performed for decision cross-tables up to degree $k = 4$. This constrains the defined mathematical programming problem to 255 decision variables. The quality of approximation for such decision cross-tables is $\gamma(4) = 0.974$ (quality of approximation for the original decision table is $\gamma(C) = 0.976$). The resulting set of decision cross-tables consists of six tables (that corresponds to twelve reports taking into account exact upward decision and exact downward decision measures). Those decision cross-tables include the following sets of criteria:

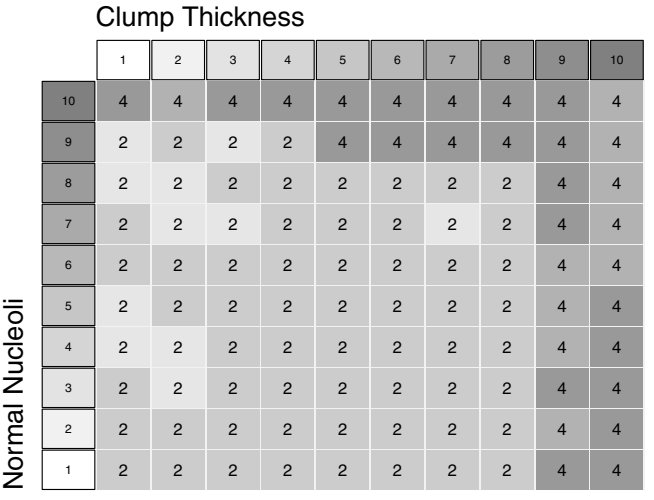


Fig. 3. Multi-criteria report including two criteria describing BCW problem (Clump Thickness, Normal Nucleoli). The measure used is the exact upward decision $d^{\geq}(\Phi^{=})$.

- Clump Thickness, Single Epithelial Cell Size (presented on Figure 1; exact upward decision measure is used, and presented on Figure 2; exact downward decision measure is used),
- Clump Thickness, Normal Nucleoli (presented on Figure 3; exact upward decision measure is used),
- Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin (not presented because of space limit),
- Bare Nuclei, Bland Chromatin, Normal Nucleoli (not presented because of space limit),
- Clump Thickness, Uniformity of Cell Shape, Marginal Adhesion, Bare Nuclei (presented on Figure 4; exact upward decision measure is used),
- Clump Thickness, Marginal Adhesion, Bare Nuclei, Mitoses (not presented because of space limit).

All reports presented on Figures 1-4 were created under R environment [9].

5 Conclusions

The presented results extend the original DRSA by adding a visualization technique. Interactive analysis of preference-ordered data allows to presents data in cross-tables similar to that known from OLAP. Moreover, there is a possibility to identify optimal set of decision cross-tables that shows important dependencies in analyzed data. Interpretation of those tables from decision rule perspective

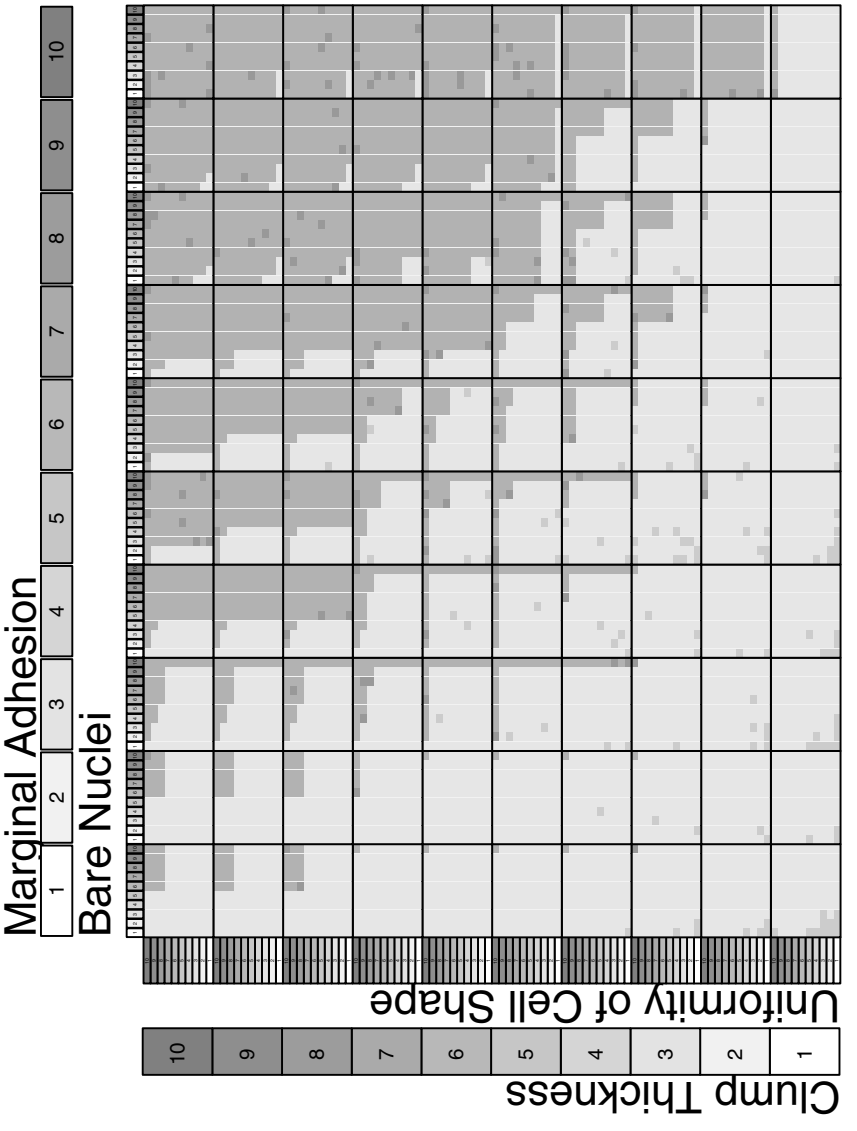


Fig. 4. Multi-criteria report including four criteria describing BCW problem (Clump Thickness, Uniformity of Cell Shape, Marginal Adhesion, Bare Nuclei). The measure used is the exact upward decision $d^{\geq}(\Phi^{=})$.

seems to be very natural. However, the problem of automatic discovery of optimal cross-decision tables is very demanding in the sense of computational complexity. That is why there is a need to construct a heuristic algorithm. It is included in our further research plans.

Acknowledgements. The authors wish to acknowledge financial support from the Ministry of Education and Science (grant no. 3T11F 02127).

References

1. Błaszczyński., Dembczyński, K., J., Słowiński, R.: On-Line Satisfaction Analysis using Dominance-based Rough Set Approach. In Rutkowska, D. et al. (eds.): Selected Problems of Computer Science (2005)
2. Dembczyński, K., Greco, S., Słowiński, R.: Second-order Rough Approximations in Multi-criteria Classification with Imprecise Evaluations and Assignments, LNAI, **3641** (2005) 54–63
3. Greco S., Matarazzo, B. and Słowiński, R.: A new rough set approach to evaluation of bankruptcy risk. In Zopounidis, C. (ed.): Operational Tools in the Management of Financial Risks. Kluwer, Dordrecht (1998) 121–136
4. Greco S., Matarazzo, B. and Słowiński, R.: Rough approximation of a preference relation by dominance relations. European Journal of Operational Research **117** (1999) 63–83
5. Greco S., Matarazzo, B. and Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research **129** (2001) 1–47
6. Greco S., Matarazzo, B. and Słowiński, R.: Rough sets methodology for sorting problems in presence of multiple attributes and criteria. European Journal of Operational Research **138** (2002) 247–259
7. Kohavi, R., Sommerfield, D.: Targeting Business Users with Decision Table Classifiers. Knowledge Discovery and Data Mining (1998) 249–253
8. Michalski, R.S.: A Planar Geometrical Model for Representing Multi-Dimensional Discrete Spaces and Multiple-Valued Logic Functions. ISG Report No. 897, Department of Computer Science, University of Illinois, Urbana, (1978)
9. R Development Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, `http://www.R-project.org`, Vienna, (2005)
10. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases, `http://www.ics.uci.edu/~mlearn/MLRepository.html`. Irvine, CA: University of California, Department of Information and Computer Science (1998)

Additive Preference Model with Piecewise Linear Components Resulting from Dominance-Based Rough Set Approximations

Krzysztof Dembczyński[1], Wojciech Kotłowski[1], and Roman Słowiński[1,2]

[1] Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland
{kdembczynski, wkotlowski, rslowinski}@cs.put.poznan.pl
[2] Institute for Systems Research, Polish Academy of Sciences, 01-447 Warsaw, Poland

Abstract. Dominance-based Rough Set Approach (DRSA) has been proposed for multi-criteria classification problems in order to handle inconsistencies in the input information with respect to the dominance principle. The end result of DRSA is a decision rule model of Decision Maker preferences. In this paper, we consider an additive function model resulting from dominance-based rough approximations. The presented approach is similar to UTA and UTADIS methods. However, we define a goal function of the optimization problem in a similar way as it is done in Support Vector Machines (SVM). The problem may also be defined as the one of searching for linear value functions in a transformed feature space obtained by exhaustive binarization of criteria.

1 Introduction

The rough set approach has often proved to be an interesting tool for solving a classification problem that consists in an assignment of objects from set A, described by *condition attributes*, to pre-defined *decision classes* Cl_t, where $t \in T$ and T is a finite set of numerically coded labels. In order to solve the problem (i.e., to classify all objects from A), a decision rule model is induced from a set of reference (training) objects $U \subset A$. The rough set analysis starts with computing lower and upper rough approximations of decision classes. The lower approximation of a decision class contains objects (from U) *certainly* belonging to the decision class without any inconsistency. The upper approximation of a decision class contains objects *possibly* belonging to the decision class that may cause inconsistencies. In the simplest case, the inconsistency is defined as a situation where two objects described by the same values of condition attributes (it is said that these objects are indiscernible) are assigned to different classes. In the next step, decision rules are induced from lower and upper rough approximations. These rules represent, respectively, certain and possible patterns explaining relationships between conditions and decisions. The model in the form of decision rules permits to classify all objects from A.

Greco. Matarazzo and Słowiński [5,6,13] have introduced a rough set approach (called Dominance-based Rough Set Approach — DRSA) for solving the problem

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 499–508, 2006.
© Springer-Verlag Berlin Heidelberg 2006

of multi-criteria classification. In this problem, it is additionally assumed that the domains of attributes (scales) are preference-ordered. The decision classes are also preference-ordered according to an increasing order of class labels, i.e. for all $r, s \in T$, such that $r > s$, the objects from Cl_r are strictly preferred to the objects from Cl_s. The condition attributes are often referred to as *condition criteria*. DRSA extends the classical approach by substituting the indiscernibility relation by a dominance relation, which permits taking into account the preference order. The inconsistency is defined in view of a dominance principle that requires that any object x, having not worse evaluations than any object y on the considered set of criteria, cannot be assigned to a worse class than y. Moreover, unlike in the classical rough set approach, there is no need in DRSA to make discretization of numerical attributes.

The preference model is a necessary component of a decision support system for multi-criteria classification. Construction of preference model requires some *preference information* from the Decision Maker (DM). Classically, these are substitution rates among criteria, importance weights, comparisons of lotteries, etc.. Acquisition of this preference information from the DM is not easy. In this situation, the preference model induced from decision examples provided by the DM has clear advantages over the classical approaches. DRSA, but also UTA [8] and UTADIS [9,15], follows the paradigm of inductive learning (in Multi-Criteria Decision Analysis referred to as a preference-disaggregation approach). It is very often underlined by Słowiński, Greco and Matarazzo (see, for example [13]) that a decision rule model has another advantage over other models, i.e. it is intelligible and speaks the language of the DM. However, in the case of many numerical criteria and decision classes, the set of decision rules may be huge and may loose its intelligibility. In such situations, an additive model composed of marginal value (utility) functions, like in UTA and UTADIS, may be helpful. The marginal value functions are usually presented graphically to the DM in order to support her/his intuition.

In the following, we present an extension of DRSA, where after computing rough approximations, additive value functions are constructed instead of a set of decision rules. The additive value function is composed of piecewise linear marginal value functions. Its construction is proceeded by solving a problem of mathematical programming similar to that formulated in UTA and UTADIS. The main difference is that we define a goal function of the optimization problem in a similar way as it is done in Support Vector Machines (SVM) [14]. However, the obtained additive value functions, for lower and upper rough approximations of decision classes, may not cover accurately all objects belonging to corresponding rough approximations. It is caused by a limited capacity of an additive model based on piecewise linear functions to represent preferences as proved in [7,12]. The problem may be also defined as the one of searching linear value functions in a transformed feature space obtained by exhaustive binarization of criteria.

The paper is organized as follows. In Section 2, DRSA involving piecewise linear marginal value functions is presented. Section 3 contains first experimental results of the methodology. The last section concludes the paper.

Table 1. Decision table: q_1 and q_2 indicate criteria, d class label. The last two columns present range of generalized decision function; objects x_2 and x_3 are inconsistent.

U	q_1	q_2	$d(x)$	$l(x)$	$u(x)$
x_1	0.25	0.3	-1	-1	-1
x_2	0.5	0.65	1	-1	1
x_3	0.75	0.7	-1	-1	1
x_4	1	0.6	1	1	1

2 Piecewise Linear Marginal Value Functions and Dominance-Based Rough Set Approach

Assume, we have a set of objects A described by n criteria. We assign to each object x a vector $\mathbf{x} = (q_1(x), \ldots, q_n(x))$, where i-th coordinate $q_i(x)$ is a value (evaluation) of object x on criterion q_i, $i = 1, \ldots, n$. For simplicity, it is assumed that domains of criteria are numerically coded with an increasing order of preference. The objective of multi-criteria classification problem is to build a preference model, according to which a class label $d(x)$ from a finite set T is assigned to every object from A. Here, for simplicity, it is assumed that $T = \{-1, 1\}$. It corresponds to that the objects from Cl_1 are strictly preferred to the objects from Cl_{-1}. We assume that the DM provides a preference information concerning a set of *reference objects* $U \subset A$, assigning to each object $x \in U$ a label $d(x) \in T$. Reference objects described by criteria and class labels are often presented in the *decision table*. An example of the decision table is presented in Table 1.

The criteria aggregation model (preference model) is assumed to be additive value function:

$$\Phi(x) = \sum_{i=1}^{n} w_i \phi_i(q_i(x)) \tag{1}$$

where $\phi_i(q_i(x))$, $i = 1, \ldots, n$, are non-decreasing marginal value functions, normalized between 0 and 1, w_i is a weight of $\phi_i(q_i(x))$. A similar aggregation model with was used in [8] within UTA method (for ranking problems) and UTADIS [9,15] (for multi-criteria classification problems), where marginal value functions were assumed to be piecewise linear. The use of this aggregation model for classification requires existence of threshold ϕ_0, such that $d(x) = 1$ if $\Phi(x) \geq \phi_0$ and $d(x) = -1$ otherwise (so $d(x) = \mathrm{sgn}(\Phi(x) - \phi_0)$). The error, which is the sum of differences $|\Phi(x) - \phi_0|$ of misclassified objects is minimized.

Assume however, that objects can be *inconsistent*. By inconsistency we mean violation of the *dominance principle*, requiring that any object x, having not worse evaluations than any object y on the considered set of criteria, cannot be assigned to a worse class than y. If such inconsistencies occur, the UTA method is not able to find any additive value function compatible with this information, whatever the complexity of the marginal functions (number of breakpoints) is, since none monotonic function can model this information. Within DRSA, such inconsistencies can be handled by using concepts of lower and upper approximations

of classes. It was shown [3] that it corresponds to generalized decision function δ for an object $x \in U$:

$$\delta(x) = \langle l(x), u(x) \rangle \tag{2}$$

where

$$l(x) = \min\{d(x) : yDx, y \in U\} \quad u(x) = \max\{d(x) : xDy, y \in U\} \tag{3}$$

where D is a dominance relation defined as $xDy \Leftrightarrow \forall_{i \in \{1,\ldots,n\}} q_i(x) \geq q_i(y)$. In other words, given the preference information, for object x there is determined a range of decision classes to which x may belong. This range results from taking into account inconsistencies caused by x. Remark that without inconsistencies, for all $x \in U$, $l(x) = u(x)$. Moreover, if we assign to each $x \in U$ a class index $l(x)$ (instead of $d(x)$), the decision table becomes consistent (similarly if we assign a class index $u(x)$ for all $x \in U$). Thus one can deal with inconsistent set U, by considering two consistent sets with two different labelings. The values of generelized decision function are also presented in Table 1. In terms of further classification, the response of such model is a range of classes, to which an object may belong.

For the two consistent decision tables it is possible to derive compatible value functions $\Phi^l(x)$, $\Phi^u(x)$ respectively, and corresponding marginal value functions $\phi_i^l(q_i(x))$ and $\phi_i^u(q_i(x))$ We assume that both $\phi_i^l(q_i(x))$ and $\phi_i^u(q_i(x))$ have piecewise linear form:

$$\phi_i(q_i(x)) = \sum_{r=1}^{k-1} c_i^r + \frac{c_i^k}{h_i^k - h_i^{k-1}}(q_i(x) - h_i^{k-1}), \quad \text{for } h_i^{k-1} \leq q_i(x) \leq h_i^k \tag{4}$$

where h_i^k is the location of the k-th brakepoint on the i-th criterion ($k = 1, \ldots, \kappa_i$, where κ_i is a number of brakepoints on i-th criterion), and c_i^k is an increment of marginal value function between brakepoints h_i^{k-1} and h_i^k, i.e. $\phi_i(h_i^k) - \phi_i(h_i^{k+1})$. Equation (4) states that function ϕ_i evaluated at $q_i(x)$ equals to the sum of increments on all intervals on the left of $q_i(x)$ and linearly approximated increment in the interval where $q_i(x)$ is located. The example is shown on Figure 1.

In the simplest form, the corresponding optimization problem can be formulated for lower bound of generalized decision ($\Phi_i^l(x)$) as follows:

$$\min\colon \sum_{j=1}^{m} \sigma_j^l \tag{5}$$

subject to constraints:

$$\Phi^l(x_j) \leq \phi_0^l + \sigma_j^l \qquad \forall x_j\colon l(x_j) = -1 \tag{6}$$

$$\Phi^l(x_j) \geq \phi_0^l - \sigma_j^l \qquad \forall x_j\colon l(x_j) = 1 \tag{7}$$

$$\sigma_j^l \geq 0 \qquad \forall x_j \tag{8}$$

$$\phi_i^l(z_i^*) = 1 \qquad \forall i \in \{1, \ldots, n\} \tag{9}$$

$$\phi_i^l(z_{i*}) = 0 \qquad \forall i \in \{1, \ldots, n\} \tag{10}$$

$$c_i^k \geq 0 \quad \forall i \in \{1, \ldots, n\}, k \in \{1, \ldots, \kappa_i\} \tag{11}$$

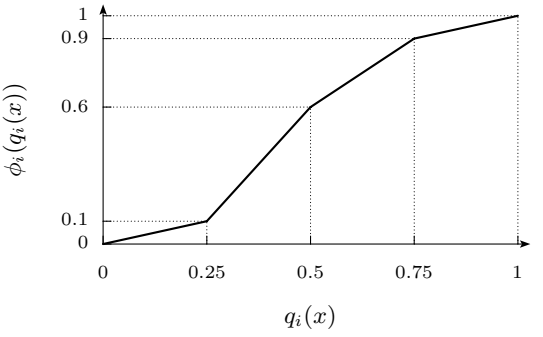


Fig. 1. Piecewise linear marginal value function ϕ_i defined by Equation 4. The increments are: $c_i^1 = 0.1$, $c_i^2 = 0.5$, $c_i^3 = 0.3$, $c_i^4 = 0.1$.

where m is the number of reference objects, σ_j are possible errors, z_i^* and z_{i*} are the highest and the lowest value on i-th criterion. Constraints (6) and (7) ensure correct separation, (9) and (10) control scaling and (11) preserves monotonicity. Analogous problem can be formulated for upper bound of generalized decision (function $\Phi^u(x)$). It is worth noting, that the method does not assure all errors become zero as the complexity of $\phi_i^l(q_i(x))$ and $\phi_i^u(q_i(x))$ grows, however, it avoids errors caused by inconsistencies. If all σ_i^l and σ_i^u become 0, the obtained model is concordant with DRSA in the sense that all objects belonging to lower or upper approximations will be reassigned by the obtained functions to these approximations.

It is worth introducing some measure of complexity of marginal functions and minimize it, to avoid building complex models. Notice, that as the function $\phi_i(q_i(x))$ is multiplied in (1) by weight w_i, we can introduce new coefficients $w_i^k = c_i^k w_i$ in order to keep the problem linear. Now control of the complexity is done by minimizing a regularization term:

$$||\mathbf{w}||^2 = \sum_{i=1}^{n} \sum_{k=1}^{\kappa_i} (w_i^k)^2 \tag{12}$$

instead of controlling the scale of functions in (9) and (10). Minimizing of term may lead to rescaling the utility function, so that all values of $\phi_i(q_i(x))$ will decrease down almost to zero. To avoid that, constraints are modified introducing the unit margin around threshold, in which no object may appear without error. Thus we rewrite equations (6) and (7) as:

$$(\Phi^l(x_j) - \phi_0^l) l(x_j) \geq 1 - \sigma_i^l \tag{13}$$

The objective of the optimization is now:

$$\min\colon ||\mathbf{w}^l||^2 + C \sum_{j=1}^{m} \sigma_j^l \tag{14}$$

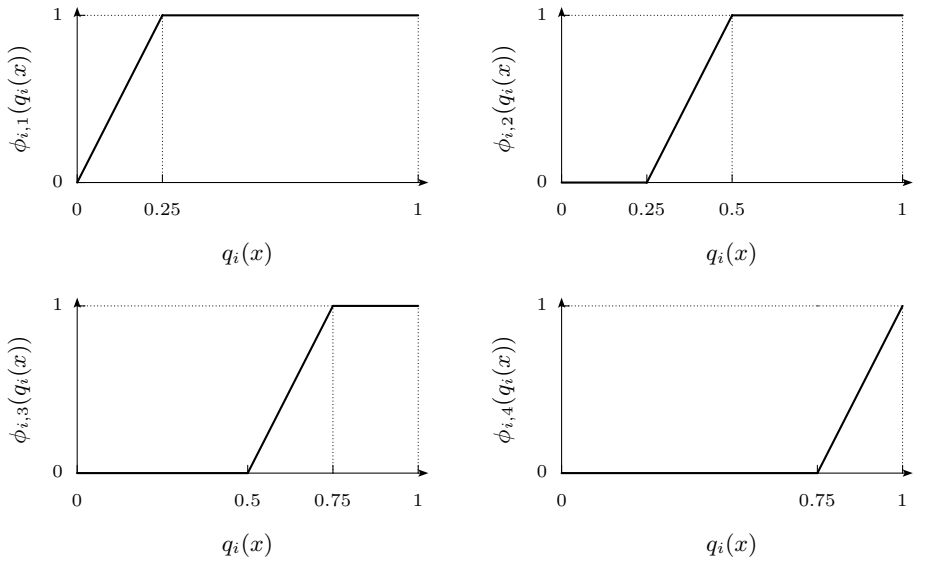


Fig. 2. Functions obtained by decomposing $\phi_i(q_i(x))$. Notice that $\phi_i(q_i(x) = 0.1\phi_{i,1}(q_i(x)) + 0.5\phi_{i,2}(q_i(x)) + 0.3\phi_{i,3}(q_i(x)) + 0.1\phi_{i,4}(q_i(x))$.

where C is the complexity constant. Analogous reasoning may be proceeded for $\Phi^u(x)$. Such problem resembles maximal margin classifier and Support Vector Machines [14]. We will try to bring it even more similar.

Let us first modify $\Phi(x)$ to be $\Phi(x) = \sum_{i=1}^{n} w_i\phi_i(q_i(x)) - \phi_0$. Now, we decompose each function $\phi_i(q_i(x))$ into κ_i functions $\phi_{i,k}(q_i(x))$ in the following way:

$$\phi_i(q_i(x)) = \sum_{k=1}^{\kappa_i} c_k^i \phi_{i,k}(q_i(x)). \tag{15}$$

An example of such decomposition is shown on Figure 2. One can treat the family of functions $\{\phi_{1,1}(q_1(x)), \ldots, \phi_{n,\kappa_n}(q_n(x))\}$ as a transformation of the space of criteria. Namely, there is a map $\mathcal{T}\colon A \to \mathbb{R}^s$ where $s = \sum_{i=1}^{n} \kappa_i$, such that $\mathcal{T}(\mathbf{x}) = (\phi_{1,1}(q_1(x)), \ldots, \phi_{n,\kappa_n}(q_n(x)))$. By substituting $w_i^k = w_i c_i^k$ and denoting $\mathbf{w} = (w_1^1, \ldots, w_n^{\kappa_n})$, the function (1) becomes:

$$\Phi(x) = \langle \mathbf{w}, \mathcal{T}(\mathbf{x}) \rangle - \phi_0 \tag{16}$$

where $\langle \cdot, \cdot \rangle$ is a canonical dot product in $\mathbb{R}^s$.

Finally, after reformulating the problem we obtain:

$$\min\colon ||\mathbf{w}^l|| + C \sum_{j=1}^{m} \sigma_j^l \tag{17}$$

subject to constraints:

$$(\langle \mathbf{w}^l, \mathcal{T}^l(\mathbf{x}) \rangle - \phi_0^l) l(x) \geq 1 - \sigma_i^l \qquad \forall x_i \in U \tag{18}$$

$$\sigma_i^l \geq 0 \qquad \forall i \in \{1, \ldots, m\} \tag{19}$$

$$w_i^k \geq 0 \qquad \forall i \in \{1, \ldots, n\}, k \in \{1, \ldots, \kappa_i\} \tag{20}$$

Notice that the regularization term is just squared Euclidean norm of the weight vector in new feature space.

Motivated by the above result, we introduce a *kernel function* $k\colon A \times A \to \mathbb{R}$, defined as:

$$k(x, y) = \sum_{i=1}^{n} k_i(x, y) \tag{21}$$

where

$$k_i(x, y) = \sum_{j=1}^{\kappa_i} \phi_{i,j}(q_i(x))\phi_{i,j}(q_i(y)). \tag{22}$$

Notice, that $k(x, y) = \langle \mathcal{T}(\mathbf{x}), \mathcal{T}(\mathbf{y}) \rangle$. Assume that we have a brakepoint in each evaluation point of all objects $x \in U$, the set of brakepoints for i-th criterion is $\{q_i(x_1), \ldots, q_i(x_m)\}$. Then the computing of the marginal kernel function $k_i(x, y)$ boils down to:

$$k_i(x, y) = \min\{rank_i(x), rank_i(y)\} \tag{23}$$

where $rank_i(x)$ is a position (in ranking) of value $q_i(x)$ on i-th criterion.

Thus, the problem may be formulated in a dual form. The most essential advantage of such approach is reduction in number of variables, irrespective to the number of brakepoints of marginal functions. As the complexity of marginal functions increases, the optimization problem remains the same and only the computation of the kernel function becomes harder. However there is a problem, how to ensure monotonicity of the resulting utility function. In the dual formulation the information about each criterion is lost, thus not all the weights may be non-negative.

Let us remark that the transformed criteria space obtained by image of mapping $\mathcal{T}(A)$ (i.e. $(\phi_{1,1}(q_1(x)), \ldots, \phi_{n,\kappa_n}(q_n(x))), x \in A$) may be also seen as a result of binarization of criteria. This type of binarization should be called an exhaustive one by analogy to other approaches well-known in rough set theory or in logical analysis of data (see for example, [1]).

The exhaustive binarization is proceeded by choosing cut points in each evaluation point of all objects $x \in U$. More precisely, the binarization of the i-th criterion is accomplished in a straightforward way by associating with each value v on this criterion, for which there exists an object x, such that $q_i(x) = v$, a boolean attribute q_{i_v} such that:

$$q_{i_v}(x) = \begin{cases} 1 \text{ if } q_i(x) \geq v \\ 0 \text{ if } q_i(x) < v \end{cases}. \tag{24}$$

Table 2 shows the exhaustive binarization of criteria from Table 1.

Moreover, let us remark that the binarized decision table contains almost the same information as dominance matrix introduced in [2]. The dominance matrix DM is defined as follows:

$$DM = \{dm(x, y) : x, y \in U\}, \text{ where } dm(x, y) = \{q_i \in Q : q_i(x) \geq q_i(y)\}. \tag{25}$$

Table 2. Decision table from Table 1 with binarized criteria

U	$q_{1\,0.25}$	$q_{1\,0.5}$	$q_{1\,0.75}$	$q_{1\,1}$	$q_{2\,0.3}$	$q_{2\,0.6}$	$q_{2\,0.65}$	$q_{2\,0.7}$	d
x_1	1	0	0	0	1	0	0	0	-1
x_2	1	1	0	0	1	1	1	0	1
x_3	1	1	1	0	1	1	1	1	-1
x_4	1	1	1	1	1	1	0	0	1

where $Q = \{q_i, i = 1, \ldots, n\}$. Dominance Matrix DM is usually implemented as 3-dimensional binary cube $\mathbf{C}$ defined as $c_{jki} = 1$, if $q_i \in dm(x_j, x_k)$, and $c_{jki} = 0$ otherwise, where $j, k = 1, ..., m$ and $i = 1, \ldots, n$. Such a structure is very useful in a procedure of generating exhaustive set of decision rules [2], because all computations may be quickly proceeded as bitwise operations. It is a counterpart of a discernibility matrix [11] well-known in classical rough set approach. It is easy to see that the following occurs:

$$c_{jki} = 1 \Leftrightarrow q_{i_{q_i(x_k)}}(x_j) = 1, \quad x_j, x_k \in U.$$

3 Experimental Results

We performed a computational experiment on Wisconsin breast cancer (BCW) data obtained from the UCI Machine Learning Repository [10]. This problem was chosen since it is known to have monotonic relationship between values on condition attributes and decision labels. Thus, all attributes can be interpreted as criteria, enabling DRSA. BCW consist of 699 instances described by 9 integer-valued attributes, from which 16 instances have missing values. Each instance is assigned to one of two classes (malignant and benign).

Several approaches have been compared with the methodology presented in the previous section that will be referred to as Piecewise Linear DRSA (PL-DRSA). These are k-Nearest Neighbours, linear Support Vector Machines, Logistic Regression, J48 Decision Trees and Naive Bayes. WEKA [4] software was used for the experiment. For all algorithms a criteria selection was performed, by using backward elimination. The number of criteria left and the leaving-one-out (loo) accuracy estimate are shown in Table 3.

Table 3. Experimental results for Wisconsin breast cancer data

Algorithm	Number of criteria	loo estimate
k-NN (k = 1)	6	96.8%
linear SVM	6	97.2%
J48	4	97.7%
Logistic Regr.	6	97.2%
Naive Bayes	6	97.1%
PL-DRSA	6	97.4%

Apparently, all the accuracy estimates are similar. PL-DRSA was conducted with setting breakpoints on each evaluation point of all objects on each criterion. Two models were created, one for lower bound of the decision range, second for the upper bound. The classification rule was the following: if $\Phi^l(x) - \Phi^u(x) \geq 0$ then assign x to class Cl_1 otherwise assign x to Cl_{-1}. The achieved accuracy was one of the best, competitive to other methods. However, marginal value functions constructed in PL-DRSA may be presented graphically and easily interpreted. Moreover, PL-DRSA shows the inconsistent data both in learning and classification stage.

4 Conclusions

Within DRSA framework, the decision rule model were always considered for multicriteria decision analysis. We presented an alternative method, related to additive aggregation model, similar to the one used in the UTA method. The described approach has several advantages. First, it is flexible and allows various shapes of separating function to be obtained. Marginal value functions may also be presented graphically and interpreted by DM. Finally, PL-DRSA can control the complexity of the additive function by fixing the number of breakpoints and minimizing the slopes in each breakpoint. DRSA plays important role in handling inconsistencies, which affect the data. Ignoring them may cause errors and, therefore generate wrong decision model. The method can be also interpreted in terms of criteria transformation (in a specific case, also in terms of binarization) and Support Vector Machines.

Acknowledgements. The authors wish to acknowledge financial support from the Ministry of Education and Science (grant no. 3T11F 02127).

References

1. Boros, E., Hammer, P. L., Ibaraki, T., Kogan, A., Mayoraz, E., Muchnik, I.: An Implementation of Logical Analysis of Data. IEEE Trans. on Knowledge and Data Engineering **12** (2000), 292–306
2. Dembczyński, K., Pindur, R., Susmaga R.: Generation of Exhaustive Set of Rules within Dominance-based Rough Set Approach. Electr. Notes Theor. Comput. Sci, 82 **4** (2003)
3. Dembczyński, K., Greco, S., Słowiński, R.: Second-order Rough Approximations in Multi-criteria Classification with Imprecise Evaluations and Assignments. LNAI **3641** (2005) 54–63
4. Frank, E., Witten, I. H.: Data Mining: Practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann, San Francisco, (2005)
5. Greco S., Matarazzo, B. and Słowiński, R.: Rough approximation of a preference relation by dominance relations, European Journal of Operational Research **117** (1999) 63–83
6. Greco S., Matarazzo, B. and Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research, **129** (2001) 1–47

7. Greco S., Matarazzo, B. and Słowiński, R.: Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules. European Journal of Operational Research, **158**, (2004) 271–292
8. Jacquet-Lagréze, E., Siskos, Y.: Assessing a set of additive utility functions for multicriteria decision making: The UTA method. European Journal of Operational Research, **10** (1982) 151–164
9. Jacquet-Lagréze, E.: An application of the UTA discriminant model for the evaluation of R&D projects, In Pardalos, P.M., Siskos, Y., Zopounidis, C., (eds.): Advances in multicriteria analysis. Kluwer Academic Publishers, Dordrecht (1995) 203–211
10. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases, `http://www.ics.uci.edu/~mlearn/MLRepository.html`. Irvine, CA: University of California, Department of Information and Computer Science (1998)
11. Skowron A., Rauszer C.: The discernibility matrices and functions in information systems. In Slowinski R. (ed.): Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory. Kluwer Academic Publishers, Dordrecht (1992) 331–362
12. Słowiński, R., Greco, S., Matarazzo, B.: Axiomatization of utility, outranking and decision-rule preference models for multiple-criteria classification problems under partial inconsistency with the dominance principle. Control & Cybernetics **31** (2002) 1005–1035
13. Słowiński, R., Greco S., Matarazzo, B., Rough Set Based Decision Support, Chapter 16, in Burke, E., Kendall, G. (eds.): Introductory Tutorials on Optimization, Search and Decision Support Methodologies. Springer-Verlag, Boston (2005)
14. Vapnik, V.: The Nature of Statistical Learning Theory, New York, Springer-Verlag (1995)
15. Zopounidis, C., Doumpos, M.: PREFDIS: a multicriteria decision support system for sorting decision problems, Computers & Operations Research **27** (2000) 779–797

Induction of Decision Rules Using Minimum Set of Descriptors

Andrzej Dominik and Zbigniew Walczak

Warsaw University of Technology, Institute of Radioelectronics
Nowowiejska 15/19, 00-665 Warsaw, Poland
A.Dominik@elka.pw.edu.pl, Z.Walczak@elka.pw.edu.pl

Abstract. In this paper we focus our attention on the classification problem. We use rough set theory and propose new methods for induction of decision rules. Our approach generalize the concept of a reduct in a dataset. We use minimal set of descriptors gained from decision table. A reduct of descriptors is a set of descriptors which allows us to distinguish between objects as well as the whole set of descriptors present in the dataset. Two types of descriptors are considered: attribute-value and attribute-object-value. We propose appropriate methodology for dealing with descriptors and inducing decision rules. We also present performed experiments on different datasets and compare them with results obtained by other algorithms for object classification based on rough sets.

1 Introduction

Classification problem has been deeply researched due to variety of its applications. Classification can be considered in different fields of science and industry and may be done using different technics: e.g. neural networks, decision rules, rough sets etc.

In this paper we mainly focus on methods based on rough sets theory [8]. We generalize the concept of reduct of attributes to the reduct of descriptors. We consider two types of descriptors: a pair of atribute-value (AV) and a tiriple atribute-object-value (AOV). A reduct of descriptors is a set of descriptors which allows us to distinguish between objects as well as the whole set of descriptors present in the dataset.

2 State of Art

Most of the rough set oriented methods for decision rules inducing [4] take advantage of concept of a reduct. Classic reduct is defined as subset of condition attributes which distinguishes all objects as well as whole set of condition attributes. Two special types of reducts are considered: relative reduct and local relative reduct.

In the first case [10], [9] a common way is to calculate one relative reduct or all relative reducts for training decision table. Relative reducts are later used for

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 509–517, 2006.
© Springer-Verlag Berlin Heidelberg 2006

inducing rules for objects from training decision table. Such rules are minimal with respect to number of condition attributes.

Similar approach is proposed for local relative reducts [10], [9]. These rules, on the other hand, are minimal with respect to number of attribute-value descriptors used in every rule.

3 Reduct of a Set of Descriptors

Let decision table be a triple $(\mathcal{U}, \mathcal{C}, \mathcal{D})$, where $\mathcal{U}$(universum) is a non-empty, finite set of objects, $\mathcal{C}$ is a non-empty finite set of condition attributes and $\mathcal{D}$ is a non-empty finite set of decision attributes. A set of all attributes is denoted by $\mathcal{A} = \mathcal{C} \cup \mathcal{D}$. A domain of an attribute $a \in \mathcal{A}$ is denoted by V_a and its value for an object $u \in \mathcal{U}$ is denoted by a(u). Further, we will consider tables with only one decision attribute $\mathcal{DT} = (\mathcal{U}, \mathcal{C}, \{d\})$ and a term attribute will refer to a condition attribute. Let descriptor attribute-value $s^{AV}_{a,u}$ and descriptor attribute-object-value $s^{AOV}_{a,u}$ be defined as follows:

$$s^{AV}_{a,u} = (a, a(u)) \qquad\qquad s^{AOV}_{a,u} = (a, u, a(u))$$

Thus $s^{AV}_{a,u}$ is a pair of atribute and value of this atribute for a given object in the dataset. The $s^{AOV}_{a,u}$ is a triple which contains also the identifier of the object. Each set of attribute-object-value descriptors X^{AOV} can be transformed into attribute-value set of descriptors X^{AV} by $\mathcal{T}$-transformation:

$$X^{AV} = \mathcal{T}(X^{AOV}) = \{s^{AV}_{a,u} \mid s^{AOV}_{a,u} \in X^{AOV}\}$$

Each set of attribute-value descriptors X^{AV} can be transformed into attribute-object-value set of descriptors X^{AOV} by $\mathcal{G}$-transformation:

$$X^{AOV} = \mathcal{G}(X^{AV}) = \{s^{AOV}_{a,u} \mid s^{AV}_{a,u} \in X^{AV}\}$$

Let X^{AV}, Y^{AV}, Z^{AV} be sets of attribute-value descriptors and X^{AOV}, Y^{AOV}, Z^{AOV} be sets of attribute-object-value descriptors such that: $X^{AOV} = \mathcal{G}(X^{AV})$ and $Y^{AOV} = \mathcal{G}(Y^{AV})$. The following facts are true:

$$X^{AOV} \subseteq Y^{AOV} \Leftrightarrow X^{AV} \subseteq Y^{AV}$$

$$Z^{AV} = \mathcal{T}(\mathcal{G}(Z^{AV}))$$

$$Z^{AOV} \subseteq \mathcal{G}(\mathcal{T}(Z^{AOV}))$$

Each object $u \in \mathcal{U}$ from a decision table can be treated as a set of appropriate descriptors: attribute-value descriptors ($\mathcal{O}^{AV}_u$) or attribute-object-value descriptors ($\mathcal{O}^{AOV}_u$):

$$\mathcal{O}^{AV}_u = \bigcup_{a \in \mathcal{C}} s^{AV}_{a,u} = \bigcup_{a \in \mathcal{C}} (a, a(u))$$

and

$$\mathcal{O}_u^{AOV} = \bigcup_{a \in \mathcal{C}} s_{a,u}^{AOV} = \bigcup_{a \in \mathcal{C}} (a, u, a(u))$$

Obviously $\mathcal{O}_u^{AV} = \mathcal{T}(\mathcal{O}_u^{AOV})$. Let $\mathcal{S}^{\mathcal{AV}}$ be a set of all attribute-value descriptors and $\mathcal{S}^{\mathcal{AOV}}$ be a set of all attribute-object-value descriptors for the given decision table:

$$\mathcal{S}^{\mathcal{AV}} = \bigcup_{u \in \mathcal{U}} \mathcal{O}_u^{AV}$$

and

$$\mathcal{S}^{\mathcal{AOV}} = \bigcup_{u \in \mathcal{U}} \mathcal{O}_u^{AOV}$$

Let $\mathcal{X}^{AV} \subseteq \mathcal{S}^{AV}$ and $\mathcal{X}_u^{AV} = \mathcal{O}_u^{AV} \cap \mathcal{X}^{AV}$. An indiscernibility relation $IND(\mathcal{X}^{AV})$ is defined as follows:

$$IND(\mathcal{X}^{AV}) = \{(u, v) \in \mathcal{U} \times \mathcal{U} \mid \mathcal{X}_u^{AV} \subseteq \mathcal{X}_v^{AV} \vee \mathcal{X}_u^{AV} \supseteq \mathcal{X}_v^{AV} \vee d(u) = d(v)\}$$

Let $\mathcal{X}^{AOV} \subseteq \mathcal{S}^{AOV}$ and $\mathcal{X}_u^{AOV} = \mathcal{O}_u^{AV} \cap \mathcal{T}(\mathcal{X}^{AOV})$. An indiscernibility relation $IND(\mathcal{X}^{AOV})$ is defined as follows:

$$IND(\mathcal{X}^{AOV}) = \{(u, v) \in \mathcal{U} \times \mathcal{U} \mid \mathcal{X}_u^{AOV} \subseteq \mathcal{X}_v^{AOV} \vee \mathcal{X}_u^{AOV} \supseteq \mathcal{X}_v^{AOV} \vee d(u) = d(v)\}$$

Note that sets: $\mathcal{X}_u^{AV}$ and $\mathcal{X}_u^{AOV}$ are not identical. Obviously $IND(\mathcal{S}^{AV}) = IND(\mathcal{S}^{AOV})$. A set of descriptors $\mathcal{R}$ (either attribute-value or attribute-object-value) is called reduct of set of descriptors $\mathcal{S}$ if it meets following conditions:

$$\begin{cases} \mathcal{R} \in \mathcal{S} \\ IND(\mathcal{S}) = IND(\mathcal{R}) \\ \neg \exists s \in \mathcal{R} : \ IND(\mathcal{S}) = IND(\mathcal{R} - \{s\}) \end{cases}$$

Reducts with minimal number of descriptors are called minimal reducts. Let $\mathcal{RED}$ be set of all reducts of $\mathcal{S}$ and $\mathcal{RED}_M$ be set of all reducts of minimal size.

4 Finding Reduct of Descriptors

Problem of finding minimal reduct (all reducts) of descriptors is very similar to problem of finding reduct of attributes. The only difference is definition of indiscernibility relation.

Problem of finding minimal reduct is $\mathcal{NP}$-hard. Most of the algorithms calculate all reducts and then select minimal one. The paper [7] proposes a method which reduces the number of needed computations. Classic exhaustive approach is based on indiscernibility matrix and indiscernibility function. This problem can be also expressed as integer (boolean) programming task [5].

Heuristics methods are widely used as well. The most popular approaches use either greedy search (i.e. algorithm proposed by Johnson [4]) or genetic and evolutionary algorithms [12] [11].

5 Classification Using Reduct of Descriptors

We use a reduct of descriptors as a tool for creating classifier based on decision rules. Each rule r consists of two parts: condition part k (set of attribute-value descriptors) and decision part d (category). Each object $u \in \mathcal{U}$ from decision table is transformed into decision rule $r(u)$ according to different formulas for different kind of minimal set of descriptors:

- attribute-value descriptors (AV): $r(u) : \mathcal{O}_u^{AV} \cap \mathcal{R}^{AV} \rightarrow d(u)$
- attribute-object-value descriptors (AOV): $r(u) : \mathcal{T}(\mathcal{O}_u^{AOV} \cap \mathcal{R}^{AOV}) \rightarrow d(u)$

Let us notice that if the decision table is deterministic, such rules correctly classify all objects (coverage and accuracy are equal to 1.0).

6 Example

Let us consider the decision table $\mathcal{DT} = (\mathcal{U}, \mathcal{C}, \{d\})$ presented below, where $\mathcal{U} = \{1, 2, 3, 4\}$ and $\mathcal{C} = \{a, b, c\}$.

No.	a	b	c	d	$\mathcal{O}^{AV}$	$\mathcal{O}^{AOV}$
1	1	0	1	1	{(a,1), (b,0), (c,1)}	{(a,1,1), (b,1,0), (c,1,1)}
2	1	1	2	1	{(a,1), (b,1), (c,2)}	{(a,2,1), (b,2,1), (c,2,2)}
3	0	0	1	1	{(a,0), (b,0), (c,1)}	{(a,3,0), (b,3,0), (c,3,1)}
4	0	2	2	0	{(a,0), (b,2), (c,2)}	{(a,4,0), (b,4,2), (c,4,2)}
5	0	1	0	0	{(a,0), (b,1), (c,0)}	{(a,5,0), (b,5,1), (c,5,0)}

$\mathcal{S}^{AV}$={ (a,0), (a,1), (b,0), (b,1), (b,2), (c,0), (c,1), (c,2) }

Let us calculate indiscernibility relation for a few sets of attribute-value descriptors:

$IND(\mathcal{S}^{AV})$ ={ {1,1}, {1,2}, {1,3}, {2,1}, {2,2}, {2,3}, {3,1}, {3,2}, {3,3}, {4,4}, {4,5}, {5,4}, {5,5} }

$IND(\{(a,0),(a,1)\})$ = { {1,1}, {1,2}, {1,3}, {2,1}, {2,2}, {2,3}, {3,1}, {3,2}, {3,3}, {3,4}, {3,5}, {4,3}, {4,4}, {4,5}, {5,3}, {5,4}, {5,5} }

$IND(\{(c,0),(c,1),(c,2)\})$ ={ {1,1}, {1,2}, {1,3}, {2,1}, {2,2}, {2,3}, {2,4}, {3,1}, {3,2}, {3,3}, {4,2}, {4,4}, {4,5}, {5,4}, {5,5} }

$IND(\{(a,0),(b,0),(b,1),(c,2)\})$ ={ {1,1}, {1,2}, {1,3}, {2,1}, {2,2}, {2,3}, {3,1}, {3,2}, {3,3}, {4,4}, {4,5}, {5,4}, {5,5} }=$IND(\mathcal{S}^{AV})$

Let us calculate set of all reducts ($\mathcal{RED}^{AV}$) and set of all minimal reducts ($\mathcal{RED}^{AV}$) of $\mathcal{S}^{AV}$:

$\mathcal{RED}^{AV}$ ={ {(a,0), (b,0), (b,1), (c,2)}, {(a,0), (b,1), (c,1), (c,2)}, {(a,1), (b,0), (b,2), (c,0)}, {(a,1), (b,2), (c,0), (c,1)}, {(a,0), (a,1), (b,0), (b,1), (b,2)}, {(a,0),

(a,1), (b,0), (c,0), (c,2)}, {(a,0), (a,1), (b,1), (b,2), (c,1)}, {(a,0), (a,1), (c,0), (c,1), (c,2)}, {(b,0), (b,1), (b,2), (c,0), (c,2)}, {(b,1), (b,2), (c,0), (c,1), (c,2)}}

$\mathcal{RED}_M^{AV}$ ={ {(a,0), (b,0), (b,1), (c,2)}, {(a,0), (b,1), (c,1), (c,2)}, {(a,1), (b,0), (b,2), (c,0)}, {(a,1), (b,2), (c,0), (c,1)}}

Let $\mathcal{R}_M^{AV} \in \mathcal{RED}_M^{AV}$. Table 1 contains sets of rules induced using different reducts of attribute-value descriptors.

Table 1. Rules induced using different reducts of attribute-value descriptors

$\mathcal{R}_M^{AV}$	Rules	Minimal Rules	Number of distinct descriptors
{(a,0), (b,0), (b,1), (c,2)}	(b,0) → (d,1) (b,1) ∧ (c,2) → (d,1) (a,0) ∧ (b,0) → (d,1) (a,0) ∧ (c,2) → (d,0) (a,0) → (d,0)	(a,0) → (d,0) (b,0) → (d,1) (b,1) ∧ (c,2) → (d,1)	4
{(a,0), (b,1), (c,1), (c,2)}	(c,1) → (d,1) (b,1) ∧ (c,2) → (d,1) (a,0) ∧ (c,1) → (d,1) (a,0) ∧ (c,2) → (d,0) (a,0) ∧ (b,1) → (d,0)	(c,1) → (d,1) (b,1) ∧ (c,2) → (d,1) (a,0) ∧ (c,1) → (d,1) (a,0) ∧ (c,2) → (d,0) (a,0) ∧ (b,1) → (d,0)	4
{(a,1), (b,0), (b,2), (c,0)}	(a,1) ∧ (b,0) → (d,1) (a,1) → (d,1) (b,0) → (d,1) (b,2) → (d,0) (c,0) → (d,0)	(a,1) → (d,1) (b,0) → (d,1) (b,2) → (d,0) (c,0) → (d,0)	4
{(a,1), (b,2), (c,0), (c,1)}	(a,1) ∧ (c,1) → (d,1) (a,1) → (d,1) (c,1) → (d,1) (b,2) → (d,0) (c,0) → (d,0)	(a,1) → (d,1) (b,2) → (d,0) (c,0) → (d,0) (c,1) → (d,1)	4

Similar reasoning can be done for attribute-object-value descriptors (a few steps are presented below).

$\mathcal{S}^{AOV}$={ (a,1,1), (a,2,1), (a,3,0), (a,4,0), (a,5,0), (b,1,0), (b,2,1), (b,3,0), (b,4,2), (b,5,1), (c,1,1), (c,2,2), (c,3,1), (c,4,2), (c,5,0) }

$IND(\mathcal{S}^{AOV})$ = { {1,1}, {1,2}, {1,3}, {2,1}, {2,2}, {2,3}, {3,1}, {3,2}, {3,3}, {4,4}, {4,5}, {5,4}, {5,5} }=$IND(\mathcal{S}^{AV})$

$IND(\{(a,1,1),(a,3,0)\}) = IND(\{(a,2,1),(a,4,0),(a,5,0)\}) =$
$IND(\{(a,2,1),(a,4,0)\}) = IND(\{(a,1,1),(a,5,0)\}) = IND(\{(a,0),(a,1)\})$

$IND(\{(c,1,1),(c,2,2),(c,4,0)\}) = IND(\{(c,0),(c,1),(c,2)\}) = IND(\mathcal{S}^{AOV})$

Note that in classical rough set approach [10], [9] this decision table has three relative reducts: {a,b}, {a,c}, {b,c}. Each reduct will induce five rules and each rule will consist of three attribute-value descriptors.

Table 2 contains sets of rules induced using classic rough set algorithms: based on relative reducts and reletive local reducts.

Table 2. Rules induced using classic rough sets algorithms

Algorithm details	Rules	Minimal Rules	Number of distinct descriptors
Relative reduct: {a,b}	(a,1) ∧ (b,0) → (d,1) (a,1) ∧ (b,1) → (d,1) (a,0) ∧ (b,0) → (d,1) (a,0) ∧ (b,2) → (d,0) (a,0) ∧ (b,1) → (d,0)	(a,1) ∧ (b,0) → (d,1) (a,1) ∧ (b,1) → (d,1) (a,0) ∧ (b,0) → (d,1) (a,0) ∧ (b,2) → (d,0) (a,0) ∧ (b,1) → (d,0)	5
Relative reduct: {a,c}	(a,1) ∧ (c,1) → (d,1) (a,1) ∧ (c,2) → (d,1) (a,0) ∧ (c,1) → (d,1) (a,0) ∧ (c,2) → (d,0) (a,0) ∧ (c,0) → (d,0)	(a,1) ∧ (c,1) → (d,1) (a,1) ∧ (c,2) → (d,1) (a,0) ∧ (c,1) → (d,1) (a,0) ∧ (c,2) → (d,0) (a,0) ∧ (c,0) → (d,0)	5
Relative reduct: {b,c}	(b,0) ∧ (c,1) → (d,1) (b,1) ∧ (c,2) → (d,1) (b,0) ∧ (c,1) → (d,1) (b,2) ∧ (c,2) → (d,0) (b,1) ∧ (c,0) → (d,0)	(b,0) ∧ (c,1) → (d,1) (b,1) ∧ (c,2) → (d,1) (b,0) ∧ (c,1) → (d,1) (b,2) ∧ (c,2) → (d,0) (b,1) ∧ (c,0) → (d,0)	6
Local reletive reducts	(a,1) → (d,1) (b,0) → (d,1) (c,1) → (d,1) (b,2) → (d,0) (c,0) → (d,0)	(a,1) → (d,1) (b,0) → (d,1) (c,1) → (d,1) (b,2) → (d,0) (c,0) → (d,0)	5

7 Results

As it was mentioned before presented approaches properly classify all objects from deterministic decision table used for rule induction. Induced sets of rules contain minimal number of appropriate descriptors. In the tests we wanted to check the effectiveness of classification of objects from test datasets using the concept of reduct of descriptors not only in a training set but also in a test dataset. Thus we built different classifiers and compare its coverage and classification accuracy, number of rules and its length and finally number of distinct descriptors and attributes in rules. The classification accuracy is the percent of objects properly classified, whereas the coverage is the percent of objects which may be classified using induced rules [4].

We used a few datasets from UCI repository [1]. There were two different groups of datasets. First one contained datasets which were explicitly divided into two decision tables: training and test (e.g. monks-1, monks-2, monks-3).

Table 3. Coverage and classification accuracy

Dataset	Coverage				Accuracy			
	AV	AOV	EXH	LEM2	AV	AOV	EXH	LEM2
irys	0.65	0.80	0.96	0.53	0.91	0.93	0.84	0.85
lung	0.53	0.83	—	0.30	0.44	0.48	—	0.40
lymn	0.59	0.91	0.98	0.56	0.78	0.79	0.79	0.84
monks-1	0.97	0.97	1.00	0.76	1.00	0.99	0.87	0.98
monks-2	0.39	0.98	1.00	0.75	1.00	0.76	0.74	0.83
monks-3	0.76	0.98	1.00	0.75	0.94	0.95	0.94	0.92
zoo	0.89	0.90	0.96	0.85	0.93	0.95	0.94	0.95

Table 4. Number of rules and average rule length

Dataset	Number of rules				Average rule length			
	AV	AOV	EXH	LEM2	AV	AOV	EXH	LEM2
irys	28.61	26.78	133.28	28.78	1.37	1.17	1.39	1.58
lung	10.92	16.17	—	5.75	2.58	1.09	—	4.48
lymn	40.56	48.11	2904.83	21.06	4.6	1.88	3.42	8.56
monks-1	35.00	39.00	161.00	45.00	3.00	2.90	3.70	4.50
monks-2	169.00	99.00	247.00	94.00	6.00	3.70	4.00	5.20
monks-3	82.00	38.00	135.00	41.00	4.00	2.70	3.40	4.50
zoo	10.28	15.39	521.00	7.39	3.47	1.80	3.42	6.59

Table 5. Number of distinct descriptors and attributes in rules

Dataset	Number of distinct descriptors				Number of distinct attributes			
	AV	AOV	EXH	LEM2	AV	AOV	EXH	LEM2
irys	23.89	28.44	93.17	34.17	3.78	3.11	4.00	3.78
lung	6.58	17.50	—	20.75	6.17	15.67	—	18.42
lymn	13.83	37.89	54.89	33.78	8.72	15.11	17.61	17.22
monks-1	10.00	17.00	17.00	17.00	3.00	6.00	6.00	6.00
monks-2	17.00	17.00	17.00	17.00	6.00	6.00	6.00	6.00
monks-3	13.00	17.00	17.00	17.00	4.00	6.00	6.00	6.00
zoo	8.44	16.94	34.22	19.83	6.11	11.56	15.78	12.33

Second one contained only one decision table per one dataset (e.g. irys, lung, lymn, zoo). In this case we divided original dataset into two decision tables: for training and test purposes. Division was made with several ratios (from 0.1 to 0.9 with step: 0.1) of number of objects of training decision table to test decision table. Ratio of objects from different categories was constant. Each division was repeated 5 times. Presented results are average values from all divisions and repetitions. We tested the following algorithms:

- **AV** - inducing rules from minimal reduct of attribute-value descriptors
- **AOV** - inducing rules from minimal reduct of attribute-object-value descriptors

- **EXH** - exhaustive algorithm [4] (inducing rules by calculating all local relative reducts)
- **LEM2** - LEM2 algorithm [6]

The algorithms EXH and LEM2 are implemented in RSES [3]. We used GLPK [2] for finding minimal reduct of descriptors. LEM2 algorithm was parameterized to cover whole training set. Exhaustive algorithm didn't find solution to set of lung decission tables in appropriate time. Conflicts in classification where resolved by *Standard voting* procedure (each rule votes with strength equals to its support). Coverage and classification accuracy of tested algorithm is shown in Table 3. We also calculated number of rules and average rule length found by each algorithm (Table 4). Comparison of number of distinct descriptors and attributes in rules is presented in Table 5.

Performed experiments led us to some interesting conclusions. As long as coverage and accuracy of classification is concerned in the most cases LEM2 was the worst algorithm. Exhaustive algorithm provided best coverage of all algorithms but also had the biggest number of rules containing the biggest number of distinct descriptors.

Attribute-object-value descriptor method had better coverage (also shorter average rule length) and better accuracy (except for 2 cases) than attribute-value approach. On the other hand rules induced by AV algorithm had the lowest number of distinct descriptors.

8 Conclusions

In this paper we proposed two new algorithms for rules induction based on rough set theory. The most characteristic feature of sets of induced rules is the fact that they are minimal in respect of number of appropriate descriptors (attribute-value in first algorithm and attribute-object-value in second one). The obtained results confirmed that the proposed methods give comparable or better results than other algorithms i.e EXH and LEM2. Our algorithms are the best suitable for situations where a small number of short rules is needed.

Proposed concept of minimal reduct of descriptors can be also used in a different way. For instance instead of inducing rules from one minimal reduct of set of descriptors one may induced rules using all minimal reducts or even all reducts (not only minimal). Another interesting idea to investigate would be to relax one of the conditions of reduct i.e. $IND(\mathcal{S}) = p * IND(\mathcal{R})$, where $p \in (0, 1)$. Such approach would induce rules with fewer number of distinct descriptors (shorter rules as well) providing better coverage of objects.

References

1. Data files from uci repository of machine learning databases and domain theories. http://www.ics.uci.edu.
2. Glpk (gnu linear programming kit). http://www.gnu.org/software/glpk/glpk.html.

3. Rses rough set exploration system. http://logic.mimuw.edu.pl/~rses/.
4. Jan G. Bazan, Hung Son Nguyen, Sinh Hoa Nguyen, Piotr Synak, and Jakub Wróblewski. Rough set algorithms in classification problem. *Rough set methods and applications: new developments in knowledge discovery in information systems*, pages 49–88, 2000.
5. Andrzej Dominik, Pawel Terlecki, and Zbigniew Walczak. Lagrangean relaxation in the minimal reduct problem. In *Proceedings of the Eighth National Conference on Evolutionary Computation and Global Optimization*, pages 55–62, 2005.
6. Jerzy W. Grzymala-Busse. A new version of the rule induction system lers. *Fundam. Inf.*, 31(1):27–39, 1997.
7. Marzena Kryszkiewicz and Katarzyna Cichon. Towards scalable algorithms for discovering rough set reducts. In *T. Rough Sets*, pages 120–143, 2004.
8. Zdzislaw Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, 1992.
9. Zdzislaw Pawlak and Andrzej Skowron. A rough set approach for decision rules generation. In *Proceedings of the Workshop W12: The Management of Uncertainty in AI at the 13th IJCAI*, Singapore, 1993. Springer-Verlag.
10. Andrzej Skowron. Boolean reasoning for decision rules generation. In *ISMIS '93: Proceedings of the 7th International Symposium on Methodologies for Intelligent Systems*, pages 295–305, London, UK, 1993. Springer-Verlag.
11. Zbigniew Walczak, Andrzej Dominik, and Pawel Terlecki. Space decomposition in the minimal reduct problem. In *Proceedings of the Seventh National Conference on Evolutionary Computation and Global Optimization*, pages 193–201, 2004.
12. Jakub Wroblewski. Finding minimal reducts using genetic algorithm. pages 186–189, 1995. University of Warsaw - Institute of Mathematics.

Comparison of Information Theoretical Measures for Reduct Finding

Szymon Jaroszewicz and Marcin Korzeń

Faculty of Computer Science and Information Systems
Technical University of Szczecin,
ul. Żołnierska 49, 71-210, Szczecin, Poland
{sjaroszewicz, mkorzen}@wi.ps.pl

Abstract. The paper discusses the properties of an attribute selection criterion for building rough set reducts based on discernibility matrix and compares it with Shannon entropy and Gini index used for building decision trees. It has been shown theoretically and experimentally that entropy and Gini index tend to work better if the reduct is later used for prediction of previously unseen cases, and the criterion based on the discernibility matrix tends to work better for learning functional relationships where generalization is not an issue.

1 Introduction

An *information system* [8] is a pair (U, A), where $U = \{u_1, \ldots, u_n\}$ is a set of *objects* (also called *records*), and $A = \{a_1, \ldots, a_m\}$ is a set of attributes. An attribute $a \in A$ is a function $a : U \to D_a$ assigning to each object a value of that attribute. D_a is the *domain* of attribute a. Without loss of generality let us assume that $D_a = \{1, \ldots, k\}$ for all $a \in A$.

A *decision system* is an information system $(U, A \cup \{d\})$ which contains a distinguished attribute d called *decision*. Let us assume that the domain of d is $D_d = \{1, \ldots, l\}$. A *relative reduct*, is a minimal (in the sense of set inclusion) set of attributes not containing d, which functionally determines the decision d, see [8] for a formal definition. For a decision system $(U, A \cup \{d\})$ we define relative discernibility matrix as

$$M_{ij}^d = \{a \in A : a(u_i) \neq a(u_j) \text{ and } d(u_i) \neq d(u_j)\}.$$

Some reduct finding algorithms, e.g. [1,12] are based on explicitly constructing the relative discernibility matrix. The size of this matrix is $O(n^2 m)$ so generating it is a serious performance limitation.

The reduct finding algorithm given in [1] starts with an empty set of attributes and heuristically adds new attributes one by one, in a greedy way, until a superreduct is constructed. Each time the attribute present in the largest number of cells of the relative discernibility matrix is added. This is equivalent to choosing the attribute which 'discerns' the largest number of pairs of objects with different decisions. Full details of the algorithm can be found in [1].

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 518–527, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Discernibility matrices are used mainly for counting the number of object pairs discerned by a given attribute. The number of the those pairs can however be obtained from the distribution of each attribute using combinatorial methods.

Let $S(d)$ be the number of pairs of objects which differ on attribute d. Let $S(d|a=i)$ be the number of pairs objects (u_1, u_2) such that $a(u_1) = a(u_2) = i$ and $d(u_1) \neq d(u_2)$. Intuitively, this is the number of pairs objects differing in d within the subset of objects having $a = i$. Define $S(d|a) = \sum_{i=1}^{k} S(d|a=i)$, it is the number of pairs of objects discerned by d after taking a into account.

In [6] the following combinatorial identities were proved (similar results have been presented earlier in [7,2]).

$$S(d) = n^2 - \sum_{j=1}^{l} {n_{\cdot j}}^2$$

$$S(d|a=i) = {n_{i\cdot}}^2 - \sum_{j=1}^{l} {n_{ij}}^2$$

$$S(d|a) = \sum_{i=1}^{k} {n_{i\cdot}}^2 - \sum_{i=1}^{k}\sum_{j=1}^{l} {n_{ij}}^2,$$

where $n_{i\cdot} = |\{u \in U : a(u) = i\}|$, $n_{\cdot j} = |\{u \in U : d(u) = j\}|$, $n_{ij} = |\{u \in U : a(u) = i \wedge d(u) = j\}|$ are the counts of events $a = i$, $d = j$ and $a = i \wedge d = j$, respectively, in the decision system.

Denote also $p_{i\cdot} = \frac{n_{i\cdot}}{n}$, $p_{\cdot j} = \frac{n_{\cdot j}}{n}$, $p_{ij} = \frac{n_{ij}}{n}$. These are estimates of the probabilities of events $a = i$, $d = j$ and $a = i \wedge d = j$, respectively, from the decision system.

After taking attribute a into account, the number of pairs discerned by d will decrease by

$$I_S(d, a) = S(d) - S(d|a).$$

Therefore, the attribute a added to the reduct in each step by the algorithm presented in [1] is the one which maximizes $I_S(d, a)$.

2 Information Theoretical Measures and Finding Reducts

In [6] we have introduced the following quantities

$$G(d) = 1 - \sum_{j=1}^{l} {p_{\cdot j}}^2$$

$$G(d|a=i) = 1 - \sum_{j=1}^{l} \left(\frac{p_{ij}}{p_{i\cdot}}\right)^2$$

$$G(d|a) = \sum_{i=1}^{k} {p_{i\cdot}}^2 G(d|a=i),$$

and shown that

$$S(d) = n^2 \left(1 - \sum_{j=1}^{l} {p_{\cdot j}}^2\right) = n^2 G(d)$$

$$S(d|a=i) = {n_{i\cdot}}^2 \left(1 - \sum_{j=1}^{l} \left(\frac{p_{ij}}{p_{i\cdot}}\right)^2\right) = {n_{i\cdot}}^2 G(d|a=i)$$

$$S(d|a) = n^2 \sum_{i=1}^{k} {p_{i\cdot}}^2 G(d|a=i) = n^2 G(d|a).$$

Notice that $G(d)$ is identical to the well known Gini index used in constructing decision trees [3] ($G_{CART}(d) = G(d)$). However $G(d|a)$ differs from the conditional Gini index in that the probabilities $p_{i\cdot}$ are squared in $G(d|a)$. Similar results have been previously obtained in [7].

It can thus be seen that many reduct finding algorithms based on discernibility matrices are in fact using this modified version of the Gini index to select attributes added to the reduct. It is thus important to understand the properties of G and that's what this paper is devoted to. Below we compare the properties of G with Shannon entropy H and conditional Gini index defined in the standard way G_{CART}, used for example in the CART decision tree builder [3].

A general class of entropies has been analyzed in [10] defined as

$$G_\alpha(d) = \frac{1}{2^{1-\alpha} - 1} \left(\sum_{j=1}^{l} {p_{\cdot j}}^\alpha - 1\right)$$

$$G_\alpha(d|a) = \sum_{i=1}^{k} {p_{i\cdot}}^\alpha G_\alpha(d|a=i).$$

It is interesting that G is a special case for $\alpha = 2$ and appears naturally when we count the number of discerned pairs of objects, also in the conditional case. When $\alpha \to 1$, G_α tends to the standard Shannon entropy.

2.1 Properties of G

Below we present some interesting properties of G, more can be found in [10].

(P1) Minimum and maximum value. We have $0 \leq G(d) \leq \frac{l-1}{l}$. The first inequality becomes an equality iff d is constant. The second inequality becomes an equality iff all values of d occur with identical probabilities $\frac{1}{l}$, as is the case with Shannon entropy.

Proof. Since for $p_{\cdot j} \in [0,1]$ we have $p_{\cdot j} \geq {p_{\cdot j}}^2$. It follows that

$$G(d) = 1 - \sum_{j=1}^{l} {p_{\cdot j}}^2 \geq 1 - \sum_{j=1}^{l} p_{\cdot j} = 1 - 1 = 0.$$

Suppose now that $G(d) = 0$. It follows that

$$1 = \sum_{j=1}^{l} {p_{\cdot j}}^2 = \left(\sum_{j=1}^{l} p_{\cdot j} \right)^2 - \sum_{1 \le j_1 \ne j_2 \le l} p_{\cdot j_1} p_{\cdot j_2},$$

which implies

$$\sum_{1 \le j_1 \ne j_2 \le l} p_{\cdot j_1} p_{\cdot j_2} = 0.$$

This is possible only if at most one $p_{\cdot j}$ is greater than 0, and this in turn is possible only if d is constant. The second inequality can easily be shown by maximizing $G(d)$ using the method of Lagrange multipliers. The details are omitted due to lack of space. ∎

(P2) Relation between joint and conditional measures. In case of Shannon entropy we have $H(da) = H(d|a) + H(a) = H(a|d) + H(d)$. This property does not hold for G_{CART}. But it does hold for G

$$G(ad) = 1 - \sum_{i=1}^{k} \sum_{j=1}^{l} {p_{ij}}^2 = 1 - \sum_{i=1}^{k} {p_{i\cdot}}^2 \sum_{j=1}^{l} \left(\frac{p_{ij}}{p_{i\cdot}} \right)^2 + \sum_{i=1}^{k} {p_{i\cdot}}^2 - \sum_{i=1}^{k} {p_{i\cdot}}^2 \quad (1)$$

$$= G(a) - \sum_{i=1}^{k} {p_{i\cdot}}^2 \left(1 - \sum_{j=1}^{l} \left(\frac{p_{ij}}{p_{i\cdot}} \right)^2 \right) = G(d|a) + G(a). \quad (2)$$

Of course $G(ad) = G(a|d) + G(d)$ is also true.

(P3) Symmetry of information gain. Define the information gain $I_G(d, a)$ for G as

$$I_G(d, a) = G(d) - G(d|a).$$

This is an analogue of Shannon entropy gain I_H [9] or Gini gain $I_{G_{CART}}$ [3]. We have (using (P2)) $I_G(d, a) = G(d) - G(d|a) = G(d) + G(a) - G(ad) = G(a) - G(a|d) = I_G(a, d)$, that is $I_G(a, d)$ is symmetric. The same property holds for entropy gain but not for $I_{G_{CART}}(d, a) = G_{CART}(d) - G_{CART}(d|a)$.

Also note that the Shannon entropy gain is equal to Shannon mutual information between a and d, defined as $I_H(a, d) = H(a) + H(d) - H(ad)$. Analogous property holds for G but not for G_{CART}.

(P4) Inequality between unconditional and conditional measures. The inequality $G(d) \ge G_{CART}(d|a) \ge G(d|a)$ always holds. Moreover, the second inequality is strict unless a or d are constant.

Proof. Assume all $p_{i\cdot} > 0$, and a and d not constant. Obviously all $p_{i\cdot} < 1$ or a would have been constant. Define $f(x) = x(1 - x)$. We have

$$G(d) = 1 - \sum_{j=1}^{l} {p_{\cdot j}}^2 = \sum_{j=1}^{l} p_{\cdot j} - \sum_{j=1}^{l} {p_{\cdot j}}^2 = \sum_{j=1}^{l} p_{\cdot j} \left(1 - p_{\cdot j} \right)$$

$$= \sum_{j=1}^{l} f\left(p_{\cdot j}\right) = \sum_{j=1}^{l} f\left(\sum_{i=1}^{k} p_{ij}\right) = \sum_{j=1}^{l} f\left(\sum_{i=1}^{k} p_{i\cdot}\frac{p_{ij}}{p_{i\cdot}}\right)$$
$$\geq \sum_{j=1}^{l}\sum_{i=1}^{k} p_{i\cdot} f\left(\frac{p_{ij}}{p_{i\cdot}}\right) = \sum_{i=1}^{k} p_{i\cdot}\sum_{j=1}^{l} f\left(\frac{p_{ij}}{p_{i\cdot}}\right) = \sum_{i=1}^{k} p_{i\cdot} G(d|a=i)$$
$$= G_{CART}(d|a) \geq \sum_{i=1}^{k} {p_{i\cdot}}^2 G(d|a=i) = G(d|a).$$

The first inequality follows from Jensen's inequality. Let us now consider the second inequality. Suppose that for some i, $G(d|a=i) > 0$, then $p_{i\cdot}^2 G(d|a=i) < p_{i\cdot} G(d|a=i)$ since $0 < p_{i\cdot} < 1$, and the strict inequality holds. If, on the other hand, for all i, $G(d|a=i) = 0$ then $G(d|a) = 0$ and the inequality is strict since for a non-constant d, $G(d) > 0$. ∎

(P5) Independent variables. It can be seen that G, contrary to G_{CART}, shares many properties of Shannon entropy. Interestingly, there is an important property which G does not have.

Suppose that d and a are statistically independent, that is $p_{ij} = p_{i\cdot} \cdot p_{\cdot j}$. In this case $H(d|a) = H(d)$, $G_{CART}(d|a) = G_{CART}(d)$ but, as Property $(P4)$ states, $G(d) > G(d|a)$ if a or d are not constant. Indeed, after elementary algebraic transformations we get

$$G(d|a) = \sum_{i=1}^{k} {p_{i\cdot}}^2 \left(1 - \sum_{j=1}^{l}\left(\frac{p_{ij}}{p_{i\cdot}}\right)^2\right)$$
$$= \sum_{i=1}^{k} {p_{i\cdot}}^2 \left(1 - \sum_{j=1}^{l} {p_{\cdot j}}^2\right) = G(d)(1 - G(a)),$$

which is different from $G(d)$ unless either a or d are constant. As a consequence $I_G(d,a) = G(d)G(a) > 0$. Also, it can be easily seen that

$$G(ad) = G(d) + G(a) - G(a)G(d)$$

and is not in general equal to $G(d) + G(a)$ for independent variables.

Notice further that both $I_H(d,a)$ and $I_{G_{CART}}(d,a)$ are minimal for independent a and d. This is not the case for $I_G(d,a)$. Consider for example attributes a_1, a_2, d with two values each. Suppose that a_1 and d are statistically independent and both follow probability distribution $(\frac{1}{2}, \frac{1}{2})$. Suppose now that the joint distribution of a_2 and d is

$$\begin{pmatrix} a_2 = 1, d = 1 & a_2 = 1, d = 2 & a_2 = 2, d = 1 & a_2 = 2, d = 2 \\ \frac{1}{8} & 0 & \frac{3}{8} & \frac{4}{8} \end{pmatrix}.$$

Clearly a_2 and d are not independent (e.g. $a_2 = 1$ implies $d = 1$) but

$$I_G(d, a_2) = G(d) - G(d|a_2) = \tfrac{1}{2} - \left(\tfrac{7}{8}\right)^2 \left(1 - \left(\tfrac{3}{7}\right)^2 - \left(\tfrac{4}{7}\right)^2\right) = \tfrac{1}{8} < \tfrac{1}{4} = I_G(d, a_1).$$

On the other hand $0 = I_H(d, a_1) < I_H(d, a_2) \approx 0.138$, and $0 = I_{G_{CART}}(d, a_1) < I_{G_{CART}}(d, a_2) \approx 0.0714$.

This fact has important implications for attribute selection. Entropy and G_{CART} treat independent attributes as having zero predictive value. They will never be added to a reduct when attributes not independent from d are present. On the other hand, there are cases when G, (and thus many algorithms based on discernibility matrices [1,2,12]) favor statistically independent attributes.

If the dataset is probabilistic, and we want the reduct to generalize to previously unseen data, adding independent attributes is clearly undesirable (except for rare special cases like the XOR function). For such cases we should use Shannon entropy or G_{CART} and not measures based on discernibility matrix.

If the dataset describes a deterministic function, i.e. generalization is not a factor, measures based on discernibility matrix should be used since (as we shall see in the next section) they tend to give shorter reducts.

2.2 Conditioning on More Attributes

In reality, when building reducts we are faced with a situation when some attributes are already in the reduct and we are adding another attribute to the ones already present. Let R be the set of attributes already in the reduct and a the new attribute to be added. Attributes are then added based on

$$I_G(d, a|R) = G(d|R) - G(d|aR). \tag{3}$$

Of course G can be replaced by Shannon entropy of G_{CART}.

It can be seen that properties $(P1) - (P4)$ all hold after conditioning on R. Due to lack of space we show this only for $(P2)$ but the proofs for $(P1)$, $(P3)$, $(P4)$ are similar.

We want to show that $G(da|R) = G(d|aR) + G(a|R)$. For every $r \in D_R$ we have $G(da|R = r) = G(d|a, R = r) + G(a|R = r)$. After multiplying by $Pr(R = r)^2$ and summing over $r \in D_R$ we get $G(da|R) = G(d|a, R) + G(a|R)$.

The important part of $(P5)$ can be generalized as $I_G(d, a|R) > 0$ if a and d are independent conditioned on R, i.e. for all $r \in D_R$, $Pr(ad|R = r) = Pr(a|R = r) \cdot Pr(d|R = r)$. For entropy and G_{CART} the gain would be 0 in this case.

3 Experimental Evaluation

We now present an evaluation of the algorithm on synthetic and real data.

We created a dataset with 50 mutually independent attributes and 4000 records. The attributes have randomly selected domain sizes of 2 or 3 and come from randomly selected probability distributions. We added a decision which was a function of 8 of the attributes. The function used was $d = \sum_{i=11}^{18} {a_i}^2 (\text{mod } 5)$. To model noise we randomly set the value of the decision in 25% of cases. The dataset is constructed in such a way that there is a very strong dependency between d and $a_{11}, \ldots, a_{18}$ jointly, and a weak, but detectable, dependency of d with each of them separately.

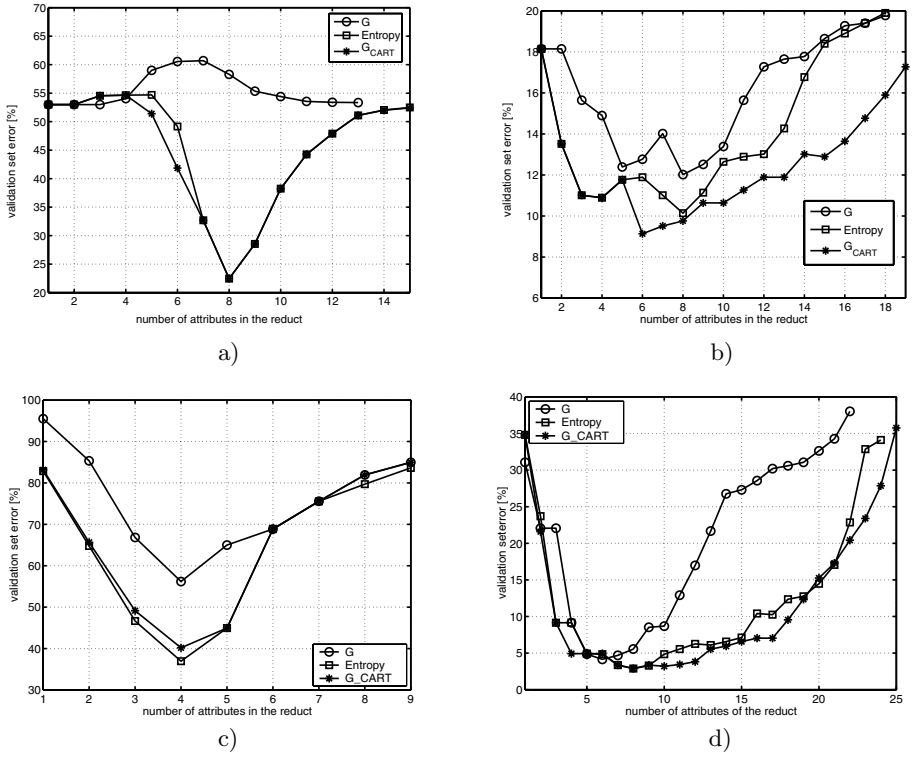


Fig. 1. Relation between reduct length and validation error for various attribute selection criteria, for artificial data (a), e-mail data (b), UCI `letter-recognition` data (c) and UCI chess endgame `KRvsKP` data (d)

We split the dataset into a training and validation datasets with 2000 records each. To use a reduct for prediction we construct a set of rules from it. For each set of values of attributes present in the reduct we construct a rule predicting the majority class for this case (based on the training set). For values of attributes not present in the training set we pick the decision at random according to the distribution of d in the training set. More advanced rule construction methods can also be used, see e.g. [11,5,4], but the simple method described above is sufficient for illustration purposes.

We then created reducts on the training set by adding attributes to the reduct one by one based on I_G, $I_{G_{CART}}$, I_H and Equation 3 computed on the training set. Attributes were added until a reduct was found on the training set. After each attribute was added we computed the reducts accuracy on the validation set. This way we get a sequence of errors on the validation set after adding each new attribute to the reduct based on the I_G, $I_{G_{CART}}$, I_H measures computed on the training set. Figure 1a shows how the error on the validation set changes after adding each attribute.

It can be seen that entropy and G_{CART} both perform quite well and quickly identify the correct attributes and their validation error decreases. Only after the correct attributes were picked, independent attributes are added to the reduct and the validation error begins to grow. On the other hand, G keeps picking wrong attributes right from the start and its error on the validation set does not decrease as the reduct grows. However G finds a reduct shorter by 2 attributes, so if generalization to future data is not an issue it gives better results.

Figure 1b shows results on a real dataset of *e*-mail messages. The set contains data about 1600 *e*-mails. There are attributes for 536 most frequently occurring words. The value of an attribute is 1 if a given word is present in the message and 0 otherwise. The set has been divided into training and validation sets of equal sizes. There are two decisions possible for each message: *spam* or *not spam*. The first 5 attributes selected by Shannon entropy and G_{CART} are identical, which shows that those measures are similar. The validation set error for entropy and G_{CART} remained lower than that of G throughout the learning process, which proves that entropy and G_{CART} have better generalization properties.

All measures pick the same first attribute. However it can be seen that Shannon entropy and G_{CART} pick a correct second attribute, while G picks an attribute which is independent from d, and does not reduce the validation error. Although entropy and G_{CART} were the first to pick an attribute visibly decreasing validation accuracy, it happened only in step 5 when the data was already split into relatively small classes by the already included attributes. It can be seen that a highly effective spam filtering system can be built using only 6 or 8 keywords found using $I_{G_{CART}}$ or I_H respectively.

Similar results can be seen for two of standard UCI benchmark datasets, see Figure 1c,d. Of course we cannot guarantee that the properties will hold for every dataset. In some cases I_G can give better predictions and I_H shorter reducts, however our experiments confirm that the trends described above are usually true.

Why G produces shorter reducts on the training set. We will now give some hints on why the G measure often picks a shorter reduct on the training set. Assume the decision d takes values in $\{0, 1\}$. Consider a number of attributes $a_1, a_2, \ldots$ statistically independent from the decision d and between each other. Suppose also the attributes are all binary with distributions $(\frac{1}{2}, \frac{1}{2})$. The attributes partition the training set into equivalence classes of exponentially decreasing sizes. That decrease will cause all decision values to be predicted perfectly *on the training set* by a reduct whose size is logarithmic in the size of the data. Of course such a reduct has no prediction power on unseen data. Suppose also that there is a number of attributes $b_1, b_2, \ldots$ such that if any of them is 1 the decision is also 1, but the probability that each $b_i = 1$ is very small, e.g. there is only one such example in the table. d is dependent on each b_i and a predictive reduct can be built from them. This reduct will however be much longer than that built based in independent attributes. We have

$$G(d|a_i) = \left(\frac{1}{2}\right)^2 G(d|a_i = 0) + \left(\frac{1}{2}\right)^2 G(d|a_i = 1) = \frac{1}{2}G(d),$$

Table 1. Example table for which I_G gives short and I_H very long reducts (case $n = 3$)

no.	a_1	a_2	a_3	b_1	b_2	b_3	b_4	$d = \sum a_i (\text{mod} 2)$
1	0	0	0	0	0	0	0	0
2	0	0	1	1	0	0	0	1
3	0	1	0	0	1	0	0	1
4	0	1	1	0	0	0	0	0
5	1	0	0	0	0	1	0	1
6	1	0	1	0	0	0	0	0
7	1	1	0	0	0	0	0	0
8	1	1	1	0	0	0	1	1

on the other hand

$$G(d|b_i) = \Pr(b_i = 0)^2 G(d|b_i = 0) + \Pr(b_i = 1)^2 G(d|b_i = 1) \approx 1 \cdot G(d) + 0 \cdot 0 = G(d),$$

so statistically independent attribute a_i would be picked leading to a shorter reduct.

In other words, there are two aspects which can be considered when picking attributes for the reduct: predicting the decision and quickly partitioning the sample space into small blocks. Entropy only considers the first aspect and completely ignores the second. The G measure on the other hand considers both those aspects.

We will now show an extreme example when I_G picks a reduct which is logarithmic in the size of the table and I_H and $I_{G_{CART}}$ pick a reduct with size linear in the size of the table. Let the table have 2^n records, n attributes $a_1, \ldots, a_n$ taking all possible different binary vectors of length n as values. The decision d is the exclusive-OR of all a_i's. This way the a_i's are statistically independent from each other and from d. Let us also add 2^{n-1} attributes b_i each of which has value 1 in only one record such that for every record with $d = 1$ there is a distinct b_i having the value of 1 in this record.

Table 1 shows the situation for $n = 3$. In the example $I_{G_{CART}}$ picks the same attributes as I_H so it's omitted. In the first step we have $I_G(d, a_i) = 0.25 > 0.125 = I_G(d, b_i)$ and $I_H(d, b_i) = 0.138 > 0 = I_H(d, a_i)$. So I_G picks one of a_i's and I_H picks one of b_i's. In the second step $I_G(d, a_{j\neq i}|a_i) = 0.125 > 0.0625 = I_G(d, b_j|a_i)$ and $I_H(d, b_{j\neq i}|b_i) = 0.173 > 0.018 = I_H(d, a_j|b_i)$. The last attribute picked by each method can be either an a_i or a b_i but the length of the reducts will be as described above.

Intuitively we expect that I_G will build a logarithmic (in the number of objects) length reduct using attributes a_i and I_H and $I_{G_{CART}}$ a reduct of length 2^{n-1} made from the b_i's. Unfortunately after several attributes have been added to the reduct, the exact behavior becomes difficult to analyze but experiments have shown that for n up to 10, I_G always found the shortest possible reduct (containing only a_i's), while I_H produced super-reducts containing all b_i's and some a_i's.

4 Conclusions

An analysis of the properties of attribute selection criteria for building rough set reducts based on discernibility matrix has been presented. It has been shown that this criterion is in fact a modified version of the well known Gini index.

A comparison of the criterion with standard Gini index and Shannon entropy has been presented. It has been shown theoretically and experimentally that entropy and traditional Gini index are better for discovering probabilistic relationships, while the criterion based on the discernibility of pairs of objects works better for deterministic cases. In other words the criterion G learns well a specific dataset while Entropy and classical Gini index have better generalization properties.

References

1. J. Bazan. *Methods of approximating infrence for synthesis of decision algorithms.* PhD thesis, Warsaw University, 1999. (in Polish).
2. J. Bazan, H.S. Nguyen, S.H. Nguyen, P. Synak, and J. Wróblewski. Rough set algorithms in classification problems. In *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems*, volume 56 of *Studies in Fuzziness and Soft Computing*, pages 49–88. Physica-Verlag, 2000.
3. L. Breiman, R. A. Olshen, J. H. Friedman, and C. J. Stone. *Classification and Regression Trees.* CRC Press, 1984.
4. J. Grzymala-Busse. LERS—a system for learning from examples based on rough sets. In *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory.*, pages 3–18. Kluwer Academic Publishers, Dordrecht, Boston, London, 1992.
5. J. Grzymala-Busse. LERS - a data mining system. In *The Data Mining and Knowledge Discovery Handbook*, pages 1347–1351. Springer, 2005.
6. M. Korzeń and S. Jaroszewicz. Finding reducts without building the discernibility matrix. In *Proc. of the 5th Int. Conf. on Intelligent Systems Design and Applications (ISDA'05)*, pages 450–455, 2005.
7. S.H. Nguyen and H.S. Nguyen. Some efficient algorithms for rough set methods. In *Proceedings of the Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU'96*, pages 1451–1456, Granada, Spain, July 1996.
8. Z. Pawlak. *Rough sets: Theoretical aspects of reasoning about data.* Dordrecht: Kluwer, 1991.
9. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.
10. D. Simovici and S. Jaroszewicz. A new metric splitting criterion for decision trees. *Journal of Parallel, Emerging and Distributed Computing.* to appear.
11. J. Stefanowski. On rough set based approaches to induction of decision rules. In A. Skowron and L. Polkowski, editors, *Rough Sets in Knowledge Discovery*, volume 1, pages 500–529. Physica Verlag, Heidelberg, 1998.
12. J. Zhang, J. Wang, D. Li, H. He, and J. Sun. A new heuristic reduct alghorithm base on rough sets theory. *Lecture Notes in Computer Sciences*, 2762:247–253, 2003.

Rough Approximation Operations Based on IF Sets

Anna Maria Radzikowska

Faculty of Mathematics and Information Science, Warsaw University of Technology,
Plac Politechniki 1, 00-661 Warsaw, Poland
and
System Research Institute, Polish Academy of Science
Newelska 6, 01–447 Warsaw, Poland

Abstract. Intuitionistic fuzzy sets, originally introduced by Atanassov, allow for representation both degrees of membership and degrees of non–membership of an element to a set. In this paper we present a generalisation of Pawlak's rough approximation operations taking Atanassov's structures as a basis. A special class of residuated lattices is taken as a basic algebraic structure. In the signature of these algebras we have abstract counterparts of two main classes of fuzzy implications. We show that basing on these lattices we can express degrees of weak and strong certainties and possibilities of membership and non–membership of an element to a set.

Keywords: Rough sets, Fuzzy sets, IF sets, Residuated lattices, Fuzzy logical connectives.

1 Introduction

In many applications the available information is in general both incomplete and imprecise. Rough set theory ([13],[14]) provides tools for representing and processing incomplete information, while fuzzy set theory ([23]) offers a wide variety of techniques for analysing imprecise data. During recent decade many researchers combined techniques developed within both theories and proposed hybrid fuzzy–rough structures for analysing incomplete and imprecise information ([12],[15],[16],[17],[21]).

In fuzzy set theory the central notion is the degree of *membership* of an element to a set and a degree of non–membership is technically calculated using a suitable fuzzy negation. Since fuzzy negations are non–increasing functions, higher degrees of membership make degrees of non–membership lower. In 1986 Atanassov ([1],[2]) introduced *intuitionistic fuzzy sets*, which allow us for representation degrees of both membership and non–membership of an element x (of the domain in discourse) to the set F by two fuzzy sets F_1 and F_2, with the additional assumption that for each x, $F_1(x) + F_2(x) \leq 1$. The value $1-F_1(x)-F_2(x)$, called a hesitation degree, represents lack of information about belongingness of x to F. Consequently, there is no strict monotonicity requirement between both functions, which admits more flexibility in representation.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 528–537, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In this paper we continue our investigations on fuzzy generalisations of rough sets and present rough approximation operations based on Atanassov's structures. Due to the terminological problems with the term "intuitionistic fuzzy sets" recently widely discussed in the literature ([9]), these structures will be referred to as *IF sets*. In contrast to many existing approaches ([3],[5],[6]), where the unit interval $[0,1]$ and fuzzy logical connectives are used, some class of *residuated lattices* ([8]) will be taken as a basic algebraic structure. Our approach is motivated by the following reasons. Firstly, rough approximations of sets are based on similarities between objects which, in general, involves incomparability (e.g. a baby is similar to his mother and to his father, but it is often hard to say to which of his parents he is more/less similar). To model such situations, some lattice structures seems to be more adequate. Secondly, taking a general algebraic structure we show that lattice–based IF sets can be viewed as specific L–fuzzy sets ([10]). Finally, in the signature of our lattices we have counterparts of main fuzzy logical connectives, in particular two main classes of fuzzy implications. We show that, depending on fuzzy implications determining lower rough approximations, we can express degrees of weak and strong certainties and possibilities of membership (resp. non–membership) of an element to a set.

2 Algebraic Foundations

A *monoid* is a structure $(M, \otimes, \varepsilon)$ such that M is a non–empty set, $\otimes$ is an associative operation in M (i.e. $a \otimes (b \otimes c) = (a \otimes b) \otimes c$) for all $a, b, c \in M$), and $\varepsilon \in M$ is a distinguished element satisfying $a \otimes \varepsilon = \varepsilon \otimes a = a$ for every $a \in M$. A monoid is called commutative iff $\otimes$ is commutative, i.e. $a \otimes b = b \otimes a$ for all $a, b \in M$.

Typical examples of commutative monoid operations are triangular norms and triangular conorms ([20]). Recall that a *triangular norm* (t–norm, for short) is a mapping $\otimes : [0,1] \times [0,1] \to [0,1]$, associative and commutative, non–decreasing in both arguments, and satisfying the border condition $a \otimes 1 = a$ for every $a \in [0,1]$. Three most popular triangular norms are:

- the Zadeh's t–norm $a \otimes_Z b = \min(a, b)$ for all $a, b \in [0,1]$,
- the algebraic product $a \otimes_P b = a \cdot b$ for all $a, b \in [0,1]$, and
- the Łukasiewicz t–norm $a \otimes_L b = \max(0, a + b - 1)$ for all $a, b \in [0,1]$.

A *triangular conorm* (t–conorm) is a mapping $\oplus : [0,1] \times [0,1] \to [0,1]$, associative and commutative, non–decreasing in both arguments, and satisfying $0 \oplus a = a$ for every $a \in [0,1]$. Three well–known triangular conorms are:

- the Zadeh's t–conorm $a \oplus_Z b = \max(a, b)$ for all $a, b \in [0,1]$,
- the bounded sum $a \oplus_P b = a + b - a \cdot b$ for all $a, b \in [0,1]$, and
- the Łukasiewicz t–conorm $a \oplus_L b = \min(1, a + b)$ for all $a, b \in [0,1]$.

A t–norm (resp. t–conorm) is called *left–continuous* (resp. *right–continuous*) iff it has left–continuous (resp. right–continuous) partial mappings.

Let $(L, \leq)$ be a poset and let $\circ : L \times L \to L$ be a commutative operation in L. By the *residuum* $\to$ *of* $\circ$ and the *dual residuum* $\leftarrow$ *of* $\circ$ we mean binary operations in L satisfying the following conditions for all $a, b, c \in L$,

$$a \circ b \leq c \iff a \leq b \to c$$
$$a \circ b \geq c \iff a \geq b \leftarrow c.$$

It can be easily proved that if the residuum of $\circ$ and the dual residuum of $\circ$ exist, then

$$a \to b = \sup\{c \in L : c \circ a \leq b\}$$
$$a \leftarrow b = \inf\{c \in L : c \circ a \geq b\}.$$

It is well–known that a t–norm (resp. t–conorm) has its residuum (resp. dual residuum) iff it is left–continuous (resp. right–continuous). The residua of left–continuous triangular norms are called *residual implications* ([11]). Three most popular residual implications, determined respectively by t_Z, t_P, and t_L, are given by: for all $a, b \in [0, 1]$,

- $\diamond$ the Gödel implication $a \to_Z b = 1$ iff $a \leq b$ and $a \to_Z b = b$ otherwise,
- $\diamond$ the Gaines implication $a \to_P b = 1$ iff $a \leq b$, and $a \to_P b = \frac{b}{a}$ otherwise,
- $\diamond$ the Łukasiewicz implication $a \to_L b = \min(1, 1 - a + b)$.

The dual residua of the three most popular triangular conorms, $\oplus_Z$, $\oplus_P$, and $\oplus_L$, are respectively given by: for all $a, b \in [0, 1]$,

- $\diamond$ $a \leftarrow_Z b = 0$ iff $b \leq a$ and $a \leftarrow_Z b = b$ otherwise,
- $\diamond$ $a \leftarrow_P b = 0$ iff $b \leq a$ and $a \leftarrow_P b = \frac{b-a}{1-a}$ otherwise
- $\diamond$ $a \leftarrow_L b = \max(0, b - a)$.

Let a poset $(L, \leq)$ be given. A unary operation $\sim : L \to L$ is called *antitone* iff for all $a, b \in L$, $a \leq b$ implies $\sim b \leq \sim a$; it is said to be *involutive* iff $\sim\sim a = a$ for every $a \in L$.

Definition 1. [16] *By an* ***extended residuated lattice*** *(**ER–lattice**, for short) we mean a system* $(L, \wedge, \vee, \otimes, \to, \sim, 0, 1)$ *such that*

- **(i)** $(L, \wedge, \vee, 0, 1)$ *is a bounded lattice with the greatest element* 1 *and the least element* 0,
- **(ii)** $(L, \otimes, 1)$ *is a commutative monoid,*
- **(iii)** $\to$ *is the* residuum of $\otimes$, *and*
- **(iv)** $\sim$ *is an antitone involution.*

The operation $\otimes$ *of an ER–lattice is called its product.* □

An ER–lattice is complete iff the underlying lattice $(L, \wedge, \vee, 0, 1)$ is complete.

Given an ER–lattice $(L, \wedge, \vee, \otimes, \to, \sim, 0, 1)$, we define the following additional operations for all $a, b \in L$,

$$a \oplus b = \sim(\sim a \otimes \sim b),$$
$$a \Rightarrow b = \sim a \oplus b,$$
$$a \leftarrow b = \inf\{c \in L : a \oplus c \geq b\}.$$

The operations $\otimes$ and $\oplus$ of an ER–lattice are the algebraic counterparts of a t–norm and a t–conorm, respectively. The residuum $\rightarrow$ of $\otimes$ corresponds to a residual implication based on a t–norm $\otimes$, $\Rightarrow$ is a counterpart of a fuzzy S–implication[1] ([11]), and $\sim$ corresponds to an involutive fuzzy negation. In order to find the intuitive meaning of the dual residuum $\leftarrow$ of $\oplus$, note first that

$$a \leftarrow b = \sim(\sim a \rightarrow \sim b).$$

Assume that $\sim$ and $\rightarrow$ are respectively the classical negation and the classical implication. Since $\sim(\sim a \rightarrow \sim b) = \sim(b \rightarrow a) = b \wedge \sim a$, $a \leftarrow b$ is in fact a fuzzy generalisation of the classical conjunction $\sim a \wedge b$.

Main properties of ER–lattices can be found in [16] and [18].

Example 1. Let $\otimes$ be a left–continuous t–norm, $\rightarrow$ be the residual implication determined by $\otimes$, and let $\sim$ be the standard fuzzy negation $\sim a = 1 - a$ for every $a \in [0,1]$. The algebra $\mathcal{L} = ([0,1], \min, \max, \otimes, \rightarrow, \sim, 0, 1)$ is an ER–lattice. □

Now we present a specific example of ER–lattice. Let $(L, \wedge, \vee, \otimes, \rightarrow, \sim, 0, 1)$ be an ER–lattice and let the set $L^\star$ be given by

$$L^\star = \{(a_1, a_2) \in L \times L : a_1 \leq \sim a_2\}.$$

Define the ordering relation $\leq^\star$ in $L^\star$ as: for all $a, b \in L^\star$, $a = (a_1, a_2), b = (b_1, b_2)$,

$$a \leq^\star b \quad \Longleftrightarrow \quad a_1 \leq b_1 \ \ \& \ \ b_2 \leq a_2.$$

Also, define the following binary operations in $L^\star$ as: for all $a, b \in L^\star$, $a = (a_1, a_2)$ and $b = (b_1, b_2)$,

$$a \wedge^\star b = (a_1 \wedge b_1, a_2 \vee b_2) \tag{1}$$
$$a \vee^\star b = (a_1 \vee b_1, a_2 \wedge b_2) \tag{2}$$
$$a \otimes^\star b = (a_1 \otimes b_1, a_2 \oplus b_2) \tag{3}$$
$$a \oplus^\star b = (a_1 \oplus b_1, a_2 \otimes b_2) \tag{4}$$
$$a \rightarrow^\star b = \sup\{c \in L^\star : c \otimes^\star a \leq^\star b\} \tag{5}$$
$$a \Rightarrow^\star b = (a_2 \oplus b_1, a_1 \otimes b_2) \tag{6}$$

and a unary operation in $L^\star$ by:

$$\sim^\star (a_1, a_2) = (a_2, a_1). \tag{7}$$

Moreover, two constants $1^\star$ and $0^\star$ are defined by

$$0^\star = (0, 1) \tag{8}$$
$$1^\star = (1, 0). \tag{9}$$

In [7] it was shown that the structure $(L^\star, \wedge^\star, \vee^\star, 0^\star, 1^\star)$ with the operations (1), (2) and the constants (8), (9), is a bounded lattice.

[1] Recall that an S–implication based on a t–conorm $\oplus$ and a fuzzy negation $\neg$ is a mapping $a \Rightarrow b = \neg a \oplus b$ for all $a, b \in [0,1]$.

We also have:

Proposition 1. *The residuum* $\rightarrow^\star$ *of* $\otimes^\star$ *is given by: for all* (a_1, a_2), $(b_1, b_2) \in L^\star$: $(a_1, a_2) \rightarrow^\star (b_1, b_2) = ((a_1 \rightarrow b_1) \wedge (\sim a_2 \rightarrow \sim b_2)\,,\, a_2 \leftarrow b_2)$. □

Moreover,

Theorem 1. *Let* $(L, \wedge, \vee, \otimes, \rightarrow, \sim, 0, 1)$ *be an ER–lattice. Then*

(i) $(L^\star, \otimes^\star, 1^\star)$ *and* $(L^\star, \oplus^\star, 0^\star)$ *are monoids*
(ii) $(L^\star, \wedge^\star, \vee^\star, \otimes^\star, \rightarrow^\star, \sim^\star, 0^\star, 1^\star)$ *is the ER–lattice.* □

Let $(L, \wedge, \vee, \otimes_L, \rightarrow_L, \sim, 0, 1)$ be an ER–lattice, where $\otimes_L$ and $\rightarrow_L$ are respectively the Łukasiewicz t–norm and the Łukasiewicz implication, and $\sim$ is the standard fuzzy negation. Using the definitions (3), (4), (6), and Proposition 1, we get the following Łukasiewicz $\mathcal{L}_\star$–logical connectives, also referred to as the *Łukasiewicz IF logical connectives*, presented in Table 1 below.

Table 1. Łukasiewicz IF logical connectives

$(a_1, a_2) \otimes^\star_L (b_1, b_2)$	$(\max(0, a_1 + b_1 - 1), \min(1, a_2 + b_2))$
$(a_1, a_2) \oplus^\star_L (b_1, b_2)$	$(\min(0, a_1 + b_1 - 1), \max(1, a_2 + b_2))$
$(a_1, a_2) \rightarrow^\star_L (b_1, b_2)$	$(\min(1, 1 - a_1 + b_1, 1 - b_2 + a_2), \max(0, b_2 - a_2))$
$(a_1, a_2) \Rightarrow^\star_L (b_1, b_2)$	$(\min(1, a_2 + b_1), \max(0, a_1 + b_2 - 1))$

3 Rough Sets, Fuzzy Sets, IF Sets

Rough sets. Let X be a non–empty domain and let $R \subseteq X \times X$ be a relation on X. Traditionally, R represents indiscernibilities (or similarities) among objects in X and is assumed to be (at least) reflexive. A system $\Sigma = (X, R)$ is called an *approximation space*. Given $\Sigma = (X, R)$, two operations $\underline{\Sigma}, \overline{\Sigma} : 2^X \rightarrow 2^X$ are defined as: for every $A \subseteq X$,

$$\underline{\Sigma}(A) = \{x \in X : (\forall y \in X)\ (x, y) \in R \Longrightarrow y \in A\} \tag{10}$$

$$\overline{\Sigma}(A) = \{x \in X : (\exists y \in X)\ (x, y) \in R\ \&\ y \in A\}. \tag{11}$$

$\underline{\Sigma}(A)$ is called a lower rough approximation of A in Σ, while $\overline{\Sigma}(A)$ is an upper rough approximation of A in Σ. It can be easily proved that reflexivity of R gives for every $A \subseteq X$,

$$\underline{\Sigma}(A) \subseteq A \subseteq \overline{\Sigma}(A).$$

Therefore, the operations (10) and (11) are called approximation operations. One can easily note that these operations correspond respectively to classical modal operations of *certainty* and *possibility* ([4]). Accordingly, $x \in \underline{\Sigma}(A)$ is read as "*x certainly belongs to A*", and $x \in \overline{\Sigma}(A)$ is read as "*x possibly belongs to A*".

A *rough set* is a pair $(\mathfrak{L}, \mathfrak{U})$ of subsets of X such that $\mathfrak{L} = \underline{\Sigma}(A)$ and $\mathfrak{U} = \overline{\Sigma}(A)$ for some $A \subseteq X$.

Fuzzy sets. Let $(L, \wedge, \vee, \otimes, \rightarrow, \sim, 0, 1)$ be an ER–lattice and let X be a non–empty universe. An *L–fuzzy set in X* is a mapping $F : X \rightarrow L$. For any $x \in X$, $F(x)$ is the degree of membership of x to F. Two specific L–fuzzy sets in X are defined as: $\emptyset(x) = 0$ and $X(x) = 1$, $x \in X$. An L–fuzzy complement of an L–fuzzy set A wrt $\sim$ is defined as: $(\sim A)(x) = \sim A(x)$, $x \in X$. For two L–fuzzy sets A and B, we write $A \subseteq_L B$ iff $A(x) \leq B(x)$ for any $x \in X$. An *L–fuzzy relation on X* is an L–fuzzy set in $X \times X$. An L–fuzzy relation R on X is called

- *reflexive* iff $R(x, x) = 1$ for every $x \in X$,
- *irreflexive* iff $R(x, x) = 0$ for every $x \in X$,
- *symmetric* iff $R(x, y) = R(y, x)$ for all $x, y \in X$.

IF sets. Let $(L^\star, \wedge^\star, \vee^\star, \otimes^\star, \rightarrow^\star, \sim^\star, 0^\star, 1^\star)$ be an ER–lattice defined in Section 2. An *L–IF set in X* is an $L^\star$–fuzzy set in X. For any $x \in X$, $F(x) = (F_1(x), F_2(x))$ is interpreted as: $F_1(x)$ is the degree of membership of x to F and $F_2(x)$ is the degree of non–membership of x to F. Note that $\sim F_2(x)$ (resp. $\sim F_1(x)$) can be viewed as the degree, to which x *potentially* belongs (resp. does not belong) to F. For two L–IF sets in X, $A = (A_1, A_2)$ and $B = (B_1, B_2)$, $A \subseteq_{L^\star} B$ means that $A_1 \subseteq_L B_1$ and $B_2 \subseteq_L A_2$. The family of all L–IF sets in X will be denoted by $IF_{L^\star}(X)$. An *L–IF relation on X* is an $L^\star$–fuzzy relation on X. For all $x, y \in X$, $R(x, y) = (R_1(x, y), R_2(x, y))$ is interpreted as: $R_1(x, y)$ is the degree to which x is R–related with y, while $R_2(x, y)$ is the degree to which x is not R–related with y. Note that an L–IF relation $R = (R_1, R_2)$ is

- reflexive iff R_1 is reflexive and R_2 is irreflexive,
- symmetric iff both R_1 and R_2 are symmetric.

4 IF Rough Approximation Operations

Let $(L^\star, \wedge^\star, \vee^\star, \otimes^\star, \rightarrow^\star, \sim^\star, 0^\star, 1^\star)$ be a complete ER–lattice as defined in Section 2, let $X \neq \emptyset$ be a set of objects, and let R be an L–IF relation on X representing (dis)similarities between objects: for all $x, y \in X$, $R(x, y) = (R_1(x, y), R_2(x, y))$, $R_1(x, y)$ is the degree to which x is R–similar to y, while $R_2(x, y)$ is the degree to which x is R–dissimilar to y. A system $\Sigma = (L^\star, X, R)$ is called an *IF approximation space*. Given Σ, define two mappings $\Sigma_{\triangleright^\star}, \Sigma^{\otimes^\star} : IF_{L^\star}(X) \rightarrow IF_{L^\star}(X)$ by: for every $A = (A_1, A_2) \in IF_{L^\star}(X)$ and for every $x \in X$,

$$\Sigma_{\triangleright^\star}(A)(x) = \inf_{y \in X}(R(x, y) \triangleright^\star A(y)) \tag{12}$$

$$\Sigma^{\otimes^\star}(A)(x) = \sup_{y \in X}(R(x, y) \otimes^\star A(y)), \tag{13}$$

where $\triangleright^\star$ stands for either $\rightarrow^\star$ or $\Rightarrow^\star$. $\Sigma_{\triangleright^\star}(A)$ is called an *IF–lower rough approximation of A*, while $\Sigma^{\otimes^\star}(A)$ is an *IF–upper rough approximation of A*.

Depending on the arrow operation used in (12) we have two classes of IF–lower rough approximation operations, $\Sigma_{\rightarrow^\star}$ and $\Sigma_{\Rightarrow^\star}$, respectively.

Some main properties of the operations (12)–(13) are given in the following theorem.

Theorem 2. *Let* $(L^\star, \wedge^\star, \vee^\star, \otimes^\star, \rightarrow^\star, \sim^\star, 0^\star, 1^\star)$ *be a complete ER–lattice and let* $\Sigma = (L^\star, X, R)$ *be an IF approximation space. Then*

(i) *Monotonicity:*

for all $A, B \in IF_{L^\star}(X)$, $A \subseteq_{L^\star} B$ *implies*

$$\Sigma_{\rightarrow^\star}(A) \subseteq_{L^\star} \Sigma_{\rightarrow^\star}(B)$$
$$\Sigma_{\Rightarrow^\star}(A) \subseteq_{L^\star} \Sigma_{\Rightarrow^\star}(B)$$
$$\Sigma^{\otimes^\star}(A) \subseteq_{L^\star} \Sigma^{\otimes^\star}(B)$$

(ii) *Duality:*

for $A \in IF_{L^\star}(X)$,

$$\Sigma_{\Rightarrow^\star}(A) = {\sim^\star}\Sigma^{\otimes^\star}({\sim^\star}A)$$
$$\Sigma^{\otimes^\star}(A) = {\sim^\star}\Sigma_{\Rightarrow^\star}({\sim^\star}A)$$

(iii) *Approximation property:*

for every $A \in IF_{L^\star}(X)$,

$$\begin{array}{lll} R \text{ is reflexive} & \text{iff} & \Sigma_{\rightarrow^\star}(A) \subseteq_{L^\star} A \\ & \text{iff} & \Sigma_{\Rightarrow^\star}(A) \subseteq_{L^\star} A \\ & \text{iff} & A \subseteq_{L^\star} \Sigma^{\otimes^\star}(A). \end{array}$$

□

In the literature similarity relations are usually fuzzy equivalence relations (i.e. reflexive, symmetric and $\otimes$–transitive, where $\otimes$ is a t–norm). However, many researchers argue that transitivity and symmetry do not adequately characterize the notion of similarity. It seems that reflexivity is the natural feature of this notion, since any object x is totally similar to itself. By the approximation property of Theorem 2, for any $A = (A_1, A_2)$, if $\Sigma_{\triangleright^\star}(A) = (\mathcal{L}_1, \mathcal{L}_2)$ and $\Sigma^{\otimes^\star}(A) = (\mathcal{U}_1, \mathcal{U}_2)$, then

$$\mathcal{L}_1 \subseteq_L A_1 \subseteq_L \mathcal{U}_1$$
$$\mathcal{U}_2 \subseteq_L A_2 \subseteq_L \mathcal{L}_2.$$

Hence, IF–rough approximation operations allow us to approximate both degrees of membership and non–membership of an object to a set: x *certainly* (resp. *possibly*) belongs to A to the degree $\mathcal{L}_1(x)$ (resp. $\mathcal{U}_1(x)$), and x *certainly* (resp. *possibly*) does not belong to A to the degree $\mathcal{U}_2(x)$ (resp. $\mathcal{L}_2(x)$).

4.1 Weak and Strong Modalities

We show now that, depending on the class of IF–lower rough approximations determined respectively either by $\rightarrow^\star$ or $\Rightarrow^\star$, we get interesting interpretations of possibilities and certainties of membership and non–membership of an element to a set.

Consider first the class of IF–lower approximation operations determined by $\rightarrow^\star$. For any L–IF set $A = (A_1, A_2) \in IF_{L^\star}(X)$, put $\Sigma_{\rightarrow^\star}(A) = (\mathcal{L}_1, \mathcal{L}_2)$. By Proposition 1, for any $x \in X$,

$$\mathcal{L}_1(x) = \inf_{y \in X}(R_1(x,y) \rightarrow A_1(y)) \wedge \inf_{y \in X}({\sim}R_2(x,y) \rightarrow {\sim}A_2(y))$$
$$\mathcal{L}_2(x) = \sup_{y \in X}(R_2(x,y) \leftarrow A_2(y)).$$

Intuitively, for $x \in X$, $\mathcal{L}_1(x)$ is the degree, to which all objects similar to y are in A and all objects *potentially* similar to y are *potentially* in A. Then $\mathcal{L}_1(x)$ might be viewed as the degree of *weak certainty of membership* of x to A. Since $R_2(x,y) \leftarrow A_2(y)$ is a generalisation of the classical conjunction $A_2(y) \wedge \sim R_2(x,y)$, $\mathcal{L}_2(x)$ is then the degree, to which some object $y \in X$ *potentially* similar to x does not belong to A. In other words, $\mathcal{L}_2(x)$ is the degree of *weak possibility of non–membership* of x to A.

Now, consider the second class of IF–lower approximation operations (determined by $\Rightarrow^\star$) For any $A = (A_1, A_2) \in IF_{L^\star}(X)$, put $\Sigma_{\Rightarrow^\star}(A) = (\mathcal{L}'_1, \mathcal{L}'_2)$. By the definition of $\Rightarrow^\star$ we easily get for every $x \in X$,

$$\mathcal{L}'_1(x) = \inf_{y \in X}(\sim R_2(x,y) \Rightarrow A_1(y))$$
$$\mathcal{L}'_2(x) = \sup_{y \in X}(R_1(x,y) \otimes A_2(y)).$$

Intuitively, $\mathcal{L}'_1(x)$ is the degree, to which all objects potentially similar to y are in A. We can say then that $\mathcal{L}'_1(x)$ reflects *strong certainty of membership* of x to A. Similarly, $\mathcal{L}'_2(x)$ is the degree, to which some object similar to x does not belong to A – here we have a *strong possibility of non–membership* of x to A.

Finally, let us consider IF–upper rough approximation operations. For any $A \in IF_{L^\star}(X)$, $A = (A_1, A_2)$, put $\Sigma^{\otimes^\star}(A) = (\mathcal{U}_1, \mathcal{U}_2)$. Then for every $x \in X$,

$$\mathcal{U}_1(x) = \sup_{y \in X}(R_1(x,y) \otimes A_1(y))$$
$$\mathcal{U}_2(x) = \inf_{y \in X}(R_2(x,y) \oplus A_2(y)).$$

Intuitively, $\mathcal{U}_1(x)$ is the degree, to which some object similar to x belongs to A. Then $\mathcal{U}_1(x)$ represents the degree of *strong possibility of membership* of x to A. Since $R_2(x,y) \oplus A_2(y) = \sim R_2(x,y) \Rightarrow A_2(y)$, $\mathcal{U}_2(x)$ is the degree, to which all objects potentially similar to x are not in A. This reflects *strong certainty of non–membership* of x to A.

It is worth noting that the duality property of Theorem 2 implies that strong possibility and strong certainty are strictly dual. This fact seems intuitively justified.

Example 2. Let $X = \{John, Bob, Al, Jim\}$ be a set of patients and let relationships among these patients, determined by their symptoms, have a form of similarities and dissimilarities. These relationships are represented by an IF relation given in Table 2. According to a doctor decision, for each patient it was determined to what extent his symptoms fit (resp. do not fit) to some disease d. The doctor's decision is represented by an IF set D given in Table 3.

Moreover, let $\mathcal{L}^\star = ([0,1]^2, \wedge^\star, \vee^\star, \otimes^\star_L, \rightarrow^\star_L, \sim^\star, 0^\star, 1^\star)$ be an ER–lattice, where $\wedge^\star$, $\vee^\star$, and $\sim^\star$ are respectively defined by (1), (2), and (7), and $\otimes^\star_L$ and $\rightarrow^\star_L$ are the Łukasiewicz IF–connectives given in Table 1.

Consider the IF approximation space $\Sigma = (\mathcal{L}^\star, X, R)$. By simple calculations we get $\Sigma_{\rightarrow^\star_L}(D)$, $\Sigma_{\Rightarrow^\star_L}(D)$, and $\Sigma^{\otimes^\star_L}(D)$ given in Table 3. In particular, Al's illness coincides with the disease d to the degree 0.3 and does not coincide with

Table 2. IF relation R

$\boldsymbol{R}$	$John$	Bob	Al	Jim
$John$	(1,0)	(0.3,0.2)	(0.2,0.6)	(0.8,0.0)
Bob	(0.6,0.1)	(1,0)	(0.3,0.4)	(0.9,0.1)
Al	(0.1,0.0)	(0.0,0.3)	(1,0)	(0.6,0.2)
Jim	(0.7,0.3)	(0.9,0.0)	(0.7,0.1)	(1,0)

Table 3. Decision D and its IF–rough approximations

$\boldsymbol{X}$	$\boldsymbol{\Sigma}_{\rightarrow_L^\star}(\boldsymbol{D})$	$\boldsymbol{\Sigma}_{\Rightarrow_L^\star}(\boldsymbol{D})$	$\boldsymbol{D}$	$\boldsymbol{\Sigma}^{\otimes_L^\star}(\boldsymbol{D})$
$John$	$(0.1, 0.8)$	$(0.1, 0.8)$	$(0.1, 0.8)$	$(0.7, 0.1)$
Bob	$(0.3, 0.7)$	$(0.2, 0.4)$	$(0.4, 0.0)$	$(0.8, 0.0)$
Al	$(0.2, 0.8)$	$(0.1, 0.5)$	$(0.3, 0.5)$	$(0.5, 0.3)$
Jim	$(0.4, 0.5)$	$(0.4, 0.5)$	$(0.9, 0.1)$	$(0.9, 0.0)$

d to the degree 0.5. Moreover, his illness certainly fits to d: in a weak sense to the degree 0.2 and in the strong sense to the degree 0.1. Also, it certainly does not fit to d (in the strong sense) to the degree 0.3. Finally, Al's illness possibly fits to d to the degree 0.5 (in the strong sense) and possibly does not fit to d: in the weak sense to the degree 0.8 and in the strong sense to the degree 0.5.

5 Conclusions

In this paper we have presented rough approximation operations based on IF sets. A specific class of residuated lattices has been taken as a basic algebraic structure. In particular, it has been shown that IF sets and IF relations can be viewed as specific L–fuzzy sets and L–fuzzy relations, respectively. We have shown that this approach allows us for representing weak/strong certainties and possibilities of membership and non–membership of elements to a set. Some main properties of IF rough approximation operations have been given.

References

1. Atanassov K. T. (1986). Intuitionistic fuzzy sets, Fuzzy Sets and Systems 20, 87–96.
2. Atanassov K. T. (1999). *Intuitionistic Fuzzy Sets*, Physica–Verlag, Heidelberg.
3. Chakrabarty K., Gedeon T., Koczy L. (1998). Intuitionistic fuzzy rough sets, Proceedings of 4th Joint Conference on Information Sciences JCIS'98, 211–214.
4. Chellas B. F. (1980). *Modal Logic: An Introduction*, Cambridge University Press, Cambridge.
5. Coker D. (1998). Fuzzy rough sets and intuitionistic L–fuzzy sets, Fuzzy Sets and Systems 96, 381–383.

6. Cornelis C., De Cock M., Kerre E. E. (2003). Intuitionistic Fuzzy Rough Sets: at the Crossroads of Imperfect Knowledge, Expert Systems 20(5), 260–269.
7. Deschrijver G. and Kerre E. E. (2003). On the relationship between some extensions of fuzzy set theory, Fuzzy Sets and Systems 133(2), 227–235.
8. Dilworth R. P. and Ward N. (1939). Residuated lattices, Transactions of the American Mathematical Society 45, 335–354.
9. Dubois D., Gottwald S., Hajek P., Kacprzyk J., Prade H. (2005). Terminological difficulties in fuzzy set theory – The case of "Intuitionistic Fuzzy Sets", Fuzzy Sets and Systems 156, 485–491.
10. Gougen J. A. (1967). L–fuzzy sets, Journal of Mathematical Analysis and Applications 18, 145–174.
11. Klir G. J. and Yuan B. (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice–Hall, Englewood Cliffs, NJ.
12. Pal S. K. and Skowron A. (1999). *Rough Fuzzy Hybridization: A New Trend in Decision Making*, Springer-Verlag.
13. Pawlak Z. (1982). Rough sets, Int. Journal of Computer and Information Science 1(5), 341–356.
14. Pawlak Z. (1991). *Rough Sets — Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers.
15. Radzikowska A. M. and Kerre E. E. (2002). A comparative study of fuzzy rough sets, Fuzzy Sets and Systems 126, 137–155.
16. Radzikowska A. M. and Kerre E. E. (2004). On L–valued fuzzy rough sets, in: *Artificial Intelligence and Soft Computing ICAISC'2004*, Rutkowski L., Siekmann J., Tadeusiewicz R., Zadeh L. A. (eds), Lecture Notes in Computer Science 3070, Springer–Verlag, 526–531.
17. Radzikowska A. M. and Kerre E. E. (2004). Fuzzy rough sets based on residuated lattices, in: *Transactions on Rough Sets II*, Peters J. F., Skowron A., Dubois D., Grzymala–Busse J., Inuiguchi M., Polkowski L. (eds), Lecture Notes in Computer Science 3135, Springer-Verlag, 278–297.
18. Radzikowska A. M. and Kerre E. E. (2004). Lattice–based fuzzy information relations and operators, Proceedings of *EUROFUSE–2004*, De Baets B., De Caluve R., Kacprzyk J., De Tré G., Zadrożny S. (eds), EXIT, Warsaw, Poland, 433–443.
19. Radzikowska A. M. and Kerre E. E. (2005). Characterisations of main classes of fuzzy relations using fuzzy modal operators, Fuzzy Sets and Systems 152(2), 223–247.
20. Schweizer B. and Sklar A. (1983). *Probabilistic Metric Spaces*, North Holland, Amsterdam.
21. Thiele H. (1993). On the definition of modal operators in fuzzy logic, Proceedings of International Symposium of Multiple–Valued Logics ISMVL'1993, 62–67.
22. Turunen E. (1999). *Mathematics Behind Fuzzy Logic*, Springer–Verlag.
23. Zadeh L. A. (1965). Fuzzy Sets, Information and Control 8, 338–358.

Relationships Between Concept Lattice and Rough Set*

Hong Wang[1,2] and Wen-Xiu Zhang[1]

[1] Institute for Information and System Sciences, Faculty of Science
Xi'an Jiaotong University, Xi'an, Shaan'xi 710049, P.R. China
[2] College of Mathematics and Computer Science,
Shanxi Normal University, Linfen, Shanxi 041004, P.R. China

Abstract. Formal concept analysis and rough set theory provide two different methods for data analysis and knowledge processing. In this paper, we discuss some basic relationships between the extensions of concepts and the equivalence class in rough set theory. And by introducing the term of anti-chain formal context, we study their relation between the two theories more deeply. Finally, we study the relation between the reduction of formal context in concept lattice and attribute reduction in rough set theory.

Keywords: Formal Context, concept lattice, rough set, information system, equivalence class.

1 Introduction

Formal concept analysis [4] and rough sets theory [8] offer related and complementary approaches for data analysis. They have very important meaning in activating mathematical thinking for conceptual data analysis and knowledge processing [1,4,8]. Formal concept analysis is a field of applied mathematics based on the mathematization of concept and conceptual hierarchy. It was firstly introduced by Wille [10] in 1982. Formal concept analysis, which also called concept lattice, is based on the notion of a formal context, which is a binary relation between a set of objects and a set of attributes. From the binary relation, one can construct hierarchical structure of concept, each concept is the unification of objects and attributes. It realizes the relationship of generalization and the specialization among concepts through Hasse diagram. As a kind of very effective methods for data analysis, formal concept analysis has been wildly applied to various fields, such as machine learning, information retrieval, software engineering, and knowledge discovery.

The theory of rough set is a new mathematical tool to deal with inexact, uncertain or vague knowledge. The basic concepts are that of an equivalence relation on a set of objects called the universe. Through the lower and upper approximation of subsets of discourse, it provides a mathematical method of knowledge discovery, and has many important applications in various fields. For

* Supported by 973 program of China (No.2002CB312200).

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 538–547, 2006.
© Springer-Verlag Berlin Heidelberg 2006

example, rough set theory has been applied to knowledge acquisition [8] and lattice theory [2]. In these domains, some useful results have been obtained using the principle of rough approximation.

Formal concept analysis and rough set theory use data as the central tool for the development of decision aids. Many efforts have been made to compare and combine the two theories [2,3,5-7, 9, 11-13]. For example, Duntsch and Gediga presented various forms of set approximations via the unifying concept of modal-style operators [2,3]. Saquer and Deogun studied approximations of a set of objects, a set of properties, and a pair of a pair of a set of objects, based on the system of formal concepts in the concept lattice [9]. Yao [12] presents a comparative study of rough set theory and formal concept analysis, and gives some concept correspondence relation. Wolff [11] studies the differences between the "partition oriented" rough set theory and the "order oriented" formal concept analysis. The fundamental connection of the two theories is that the knowledge bases of rough set theory and scaled many-valued contexts of formal concept analysis are shown to be nearly equivalent.

In this paper, we discuss some basic relationship between the extension of a formal concept and equivalence classes in a special information system. By introducing the concept of antichain formal context, we study their relation more deeply. In the process of classification and forming concept, it always exists some superfluous attributes, we then study the relationships between the reduction in a formal context and in an information system.

2 Basic Concepts

2.1 Formal Context and Concept Lattice

Definition 2.1. A formal context is a triple (U, A, I), where $U = \{x_1, x_2, \ldots, x_n\}$ is a non-empty, finite set of objects called the universe, and $A = \{a_1, a_2, \ldots, a_m\}$ is the set of attributes, I is binary relation between U and A, i.e., $I \subseteq U \times A$. In order to express that an object x is in a relation I with an attribute a, we write xIa or $(x, a) \in I$, I is called regular if it satisfies the following conditions:

(1)$\forall\, x \in U, \exists a_1, a_2 \in A, (x, a_1) \in I, (x, a_2) \notin I$;
(2) $\forall a \in A, \exists x_1, x_2 \in U, (x_1, a) \in I, (x_2, a) \notin I$.

In this paper, if $(x, a) \in I$, the intersection of the row labelled by x and the column labelled by a contains 1; otherwise it contains 0.

Definition 2.2[4]**.** Let (U, A, I) be a formal context, for a set $X \subseteq U$ of objects and a set B of attributes, we define

$$X^* = \{\ a \in A \mid \forall x \in X,\ (x, a) \in I\}$$

$$B^* = \{\ x \in U \mid \forall a \in B,\ (x, a) \in I\}$$

X^* is the set of common attributes of objects in X, B^* is the set of objects possessing all the attributes in set B. For $x \in U, a \in A$, we denote $x^* = \{x\}^*, a^* = \{a\}^*$.

Definition 2.3[14]**.** A pair $(X, B), X \subseteq U, B \subseteq A$, is called a formal concept of the context (U, A, I), if $X^* = B$ and $B = X^*$. We call X the extent and B the intent of the concept (X, B). The set of all concepts in (U, A, I) is denoted by $L(U, A, I)$.

Theorem 2.1[4]**.** Let (U, A, I) be a formal context, $X, X_1, X_2 \subseteq U, B, B_1, B_2 \subseteq A$, then

(1)$X_1 \subseteq X_2, \Longrightarrow X_2^* \subseteq X_1^*$;
(2)$B_1 \subseteq B_2, \Longrightarrow B_2^* \subseteq B_1^*$;
(3)$X \subseteq X^{**}$, $B \subseteq B^{**}$;
(4)$X^* = X^{***}$, $B^* = B^{***}$;
(5)$(X_1 \cup X_2)^* = X_1^* \cap X_2^*$, $(B_1 \cup B_2)^* = B_1^* \cap B_2^*$.

Definition 2.4. Let (U, A, I) be a formal context, (X_1, B_1) and (X_2, B_2) be concepts of the context, (X_1, B_1) is subconcept of (X_2, B_2), written $(X_1, B_1) \leq (X_2, B_2)$, or (X_2, B_2) is a superconcept of (X_1, B_1) if and only if $X_1 \subseteq X_2$(which is equivalent to $B_2 \subseteq B_1$).

Theorem 2.2[4]**.** Let (U, A, I) be a formal context, then $L(U, A, I)$ is a complete lattice. Its infimum and supremum are given by:

$$\bigwedge_{i\in\tau}(X_i, B_i) = (\bigcap_{i\in\tau} X_i,\ (\bigcup_{i\in\tau} B_i)^{**}),$$

$$\bigvee_{i\in\tau}(X_i, B_i) = ((\bigcup_{i\in\tau} X_i)^{**},\ \bigcap_{i\in\tau} B_i).$$

Where τ is an index set and for every $t \in \tau$, (X_t, B_t) is a formal concept.

Let $L(U, A, I)$ be a concept lattice, for any $Y \subseteq U$, denote

$$\underline{Apr}(Y) = ex(\bigvee\{(X, B) \in L(U, A, I) |\ X \subseteq Y\},$$

$$\overline{Apr}(Y) = ex(\bigwedge\{(X, B) \in L(U, A, I) |\ Y \subseteq X\}.$$

$(\underline{Apr}(Y), (\underline{Apr}(Y))^*)$ and $(\overline{Apr}(Y), (\overline{Apr}(Y))^*)$ are , respectively, referred to the lower and upper approximation concept of Y Wrt $L(U, A, I)$.

Definition 2.5. Let $L(U, A_1, I_1)$ and $L(U, A_2, I_2)$ be two concept lattices. If for any $(X, B) \in L(U, A_2, I_2)$, there exists $(X', B') \in L(U, A_1, I_1)$ such that $X' = X$, then $L(U, A_1, I_1)$ is said finer than $L(U, A_2, I_2)$, denoted by

$$L(U, A_1, I_1) \leq L(U, A_2, I_2).$$

If in addition $L(U, A_2, I_2) \leq L(U, A_1, I_1)$, we say that the two concept lattices are isomorphic, denoted by

$$L(U, A_1, I_1) \cong L(U, A_2, I_2).$$

Definition 2.6. Let (U, A, I) be a formal context, $D \subseteq A$. D is referred to as a consistent set of (U, A, I) if $L(U, D, I \cap (U \times D)) \cong L(U, A, I)$. If D is a consistent set and no proper subset of D is consistent, then D is referred to as a reduct of (U, A, I).

2.2 Information Systems and Reduction

In rough set theory, an information system plays similar role as context in concept lattice.

An information system is a triple (U, A, F), where $U = \{x_1, x_2, \ldots x_n\}$ is a nonempty, the finite set of objects and $A = \{a_1, a_2, \ldots a_m\}$ is a nonempty, finite set of attributes, F is a set of functions between U and A, i.e. $F = \{a_l : U \to V_l,\ a_l \in A\}$, V_l is the domain of a_l.

Every nonempty subset $B \subseteq A$ determines an equivalence relation as follows:

$$R_B = \{(x_i, x_j) \in U \times U : a_l(x_i) = a_l(x_j), \forall a_l \in B\}.$$

Since R_B is an equivalence relation on U, it partitions U into a family of disjoint subsets U/R_B called a quotient set of U, $U/R_B = \{[x_i]_B : x_i \in U\}$, where $[x_i]_B$ denote the equivalence class determined by x_i with respect to B, i.e., $[x_i]_B = \{x_j \in U :\ (x_i, x_j) \in R_B\}$. Denote by $\sigma(U/R_B)$ the algebra generated by U/R_B, that is,

$$\sigma(U/R_B) = \{\bigcup_{x \in E} [x]_B : E \subseteq U\} \cup \{\emptyset\}.$$

Let $X \subseteq U$, $B \subseteq A$, denote

$$\underline{R_B}(X) = \{x \in U : [x]_B \subseteq X\} = \bigcup\{[x]_B : [x]_B \subseteq X\};$$

$$\overline{R_B}(X) = \{x \in U : [x]_B \cap X \neq \emptyset\} = \bigcup\{[x]_B : [x]_B \cap X \neq \emptyset\}.$$

$\underline{R_B}(X)$ and $\overline{R_B}(X)$ are referred to the lower and the upper approximation of X Wrt $\sigma(U/R_B)$, respectively. The lower approximation $\underline{R_B}(X)$ is the set of objects that belong to X with certainty, while the upper approximation $\overline{R_B}(X)$ is the set of objects that possibly belong to X.The pair$(\underline{R_B}(X), \overline{R_B}(X))$ is referred to as the Pawlak rough set of X wrt. B.

Definition 2.7. Let (U, A, F) be an information system, $B \subseteq A$. Then B is referred to as a partition consistent set of (U, A, F) if $R_B = R_A$. If B is a consistent set and no proper subset of B is partition consistent, then B is referred to as a partition reduct of (U, A, F). Let $\{B_k,\ k \leqslant r\}$ be the family of partition reducts of (U, A, F), if $C_A = \bigcap_{k=1}^{r} B_k \neq \emptyset$, then we called C_A a partition core of (U, A, F).

Actually, from the above definition we know that a formal context (U, A, I) is a special information system in which $V_l = \{0, 1\}$, and the following relations hold:

(1)$(x, a) \in I \Longleftrightarrow a_(x) = 1$;
(2)$(x_i, x_j) \in R_A \Longleftrightarrow x_i^* = x_j^*$.

In the following, we discuss the relationships between the concept based on a formal context and the equivalence class of information systems derived from the context.

3 Relationships Between Concept Lattice and Rough Set

For the sake of simplicity, we denote $L_u(U, A, I) = \{X;\ (X, B) \in L(U, A, I)\}$.

Theorem 3.1. Let (U, A, I) be a formal context. For any $X \in L_u(U, A, I)$, we have

$$X = \bigcup_{x \in X} [x]_A \in \sigma(U/R_A)$$

Proof. Clearly, $\emptyset, U \in \sigma(U/R_B)$.

For $x \in X$, since R_A is reflexive, we have $x \in [x]_A$, which implies that $x \in \bigcup_{x \in X} [x]_A$. Therefore, $X \subseteq \bigcup_{x \in X} [x]_A$.

Conversely, since $X \in L_u(U, A, I), X = (\bigcup_{x \in X} x)^{**} = (\bigcap_{x \in X} x^*)^* = (\bigcap_{x \in X} [x]_A^*)^* = (\bigcup_{x \in X} [x]_A)^{**} \supseteq \bigcup_{x \in X} [x]_A$. Hence $X = \bigcup_{x \in X} [x]_A \in \sigma(U/R_A)$.

Theorem 3.2. Let (U, A, I) be a formal context, $B \subseteq A$. Denoted by

$$f^{-1}(B) = \{x \in U,\ x^* = B\},$$
$$J = \{f^{-1}(B) \neq \emptyset\}.$$

Then J is a partition of U, and $J = U/R_A$.

Proof. If $B_1 \neq B_2$, then $f^{-1}(B_1) \cap f^{-1}(B_2) = \emptyset$.

Clearly $\bigcup_{B \subseteq A} f^{-1}(B) \subseteq U$.

If $x \in U$, then there exists $B \subseteq A$ such that $x^* = B$, i.e., $x \in f^{-1}(B)$. Hence $x \in \bigcup_{B \subseteq A} f^{-1}(B)$. Thus we obtain that $U = \bigcup_{B \subseteq A} f^{-1}(B)$. Therefore J is a partition of U.

If $x_1, x_2 \in f^{-1}(B)$, then $x_1^* = x_2^*$, i.e., $(x_1, x_2) \in R_A$. Hence $J = U/R_A$.

Theorem 3.3. Let (U, A, I) be a formal context, and $(X_i, B_i)(i \in \tau)$ be the nonempty minimum element of $L(U, A, I)$, then

(1)$X_i \in U/R_A$,
(2)$\bigcap_{i \in \tau_0} X_i = \emptyset\ (\tau_0 \subseteq \tau)$.

Proof. (1)In order to prove $X_i \in U/R_A$, by Theorem 3.2, we have to prove that $X_i \in J$, i.e.,$X_i = f^{-1}(B_i)$.

If $x \in f^{-1}(B_i)$, then $x^* = B_i$, so for any $b \in B$, we have $(x, b) \in I$, thus $x \in B_i^* = X_i$. Consequently $f^{-1}(B_i) \subseteq X$.

Conversely, if $x \in X_i = B_i^*$, then $x^{**} \subseteq B_i^{***} = B_i^* = X_i$. It follows from the conditions that $x^{**} = X_i$, and so $x^* = B_i$ i.e., $x \in f^{-1}(B_i)$. Hence $X_i \subseteq f^{-1}(B_i)$. Thus we obtain $X_i = f^{-1}(B_i) \in J$.

(2)It follows directly from (1).

Theorem 3.4. Let (U, A, I) be a formal context. For any $(X, B) \in L(U, A, I)$, we denote

$$L(X, B) = \{(X', B'), X' \subseteq X\}$$
$$L^0(X, B) = X - \bigcup\{X', (X', B') \in L(U, A, I)\}$$

Then $U/R_B = \{L^0(X, B) \neq \emptyset.\ (X, B) \in L(U, A, I)\}$.

Proof. Clearly $X - \bigcup\{X', (X', B') \in L(U, A, I)\} \subseteq U$.

Firstly, we suppose $(X_1, B_1), (X_2, B_2) \in L(U, A, I)$. If $(X_1 - \bigcup\{X_1', (X_1', B_1') \in L(X_1, B_1)\}) \bigcap (X_2 - \bigcup\{X_2', (X_2', B_2') \in L(X_1, B_1)\}) \neq \emptyset$, then there exists at least one $x \in U$, such that $x \in (X_1 - \bigcup\{X_1', (X_1', B_1') \in L(X_1, B_1)\})$ and $x \in (X_2 - \bigcup\{X_2', (X_2', B_2') \in L(X_1, B_1)\})$, hence $x \in X_1 \cap X_2, x \notin \bigcup\{X_1', (X_1', B_1') \in L(X_1, B_1)\}$, and $x \notin \bigcup\{X_2', (X_2', B_2') \in L(X_1, B_1)\}$. Since $X_1 \cap X_2 \subseteq X_i (i = 1, 2)$, which is a contradiction.

Secondly, for any $x \in U$. Since I is regular, there exists $(X, B) \in L(U, A, I)$, such that $x \in X$, i.e., $x \in X - \bigcup\{X', (X', B') \in L(X, B)\}$ or $x \in \bigcup\{X', (X', B') \in L(X, B)\}$. If $x \in X' - \bigcup\{X', (X', B') \in L(X, B)\}$, then conclusion is obvious. If $x \in \bigcup\{X', (X', B') \in L(X, B)\}$, then there exists at least $(X'', B'') \in L(X, B)$ such that $x \in X''$. It is similar to the above methods, then we can prove that for any $x \in U$ there exists at least one $(X, B) \in L(X, B)$, such that $x \in X - \bigcup\{X', (X', B') \in L(X, B)\}$. Hence, $\{L^0(X, B) \neq \emptyset.\ (X, B) \in L(U, A, I)\}$ is a partition. Finally, since for any $x_1, x_2 \in X - \bigcup\{X', (X', B') \in L(U, A, I)\}$, $x_1, x_2 \in X$ and $x_1, x_2 \notin \bigcup\{X', (X', B') \in L(U, A, I)\}$. Hence $x_1^* = x_2^*$, i.e., $(x_1, x_2) \in R_A$.

Example 3.1. A formal context (U, A, I) is given as Table 1, where $U = \{x_1, x_2, \ldots, x_n\}$ and $A = \{a_1, a_2, \ldots, a_m\}$.

Table 1. A formal context

	a	b	c	d	e
1	1	0	1	1	1
2	1	0	1	0	0
3	0	1	0	0	1
4	0	1	0	0	1
5	1	0	0	0	0
6	1	1	0	0	1

For the formal context given by Table 1, the corresponding concept lattice is given by Figure 1. For simplicity, a set is denoted by listing its elements. For example, the set $\{1, 2, 5, 6\}$ is denoted by 1256.

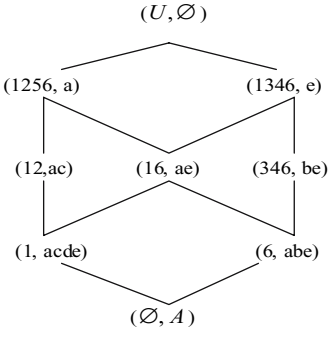


Fig. 1. The concept lattice of (U, A, I)

where

$$1^* = \{a, c, d, e\},\ 2^* = \{a, c\},\ 3^* = 4^* = \{b, e\},\ 5^* = \{a\}.$$

The set of all extensions of formal concepts of the context (U, A, I) is

$$L_u(U, A, I) = \{U, 1256, 1346, 12, 16, 346, 1, 6, \emptyset\}$$

Table 1 as an information system, it can be calculated that

$$U/R_A = \{C_1, C_2, C_3, C_4, C_5\}$$

where $C_1 = [1]_A = \{1\}$, $C_2 = [2]_A = \{2\}$, $C_3 = [3]_A = [4]_A = \{3, 4\}$, $C_4 = [5]_A = \{5\}$, $C_5 = [6]_A = \{6\}$. For any equivalence class, it may not necessarily be the extension of a formal concept, but there are strong connections between them.

If $X = \{1, 3, 4, 6\} \in L_u(U, A, I)$, then

$$X = [1]_A \cup [3]_A \cup [6]_A.$$

We can easily notice $(1, acde)$ and $(6, abe)$ are two nonempty minimum elements of $L(U, A, I)$, their extensions is two equivalence classes of the information system, and intersection of extensions is emptyset.

Definition 3.1. Let (U, A, I) be a formal context, we denote

$$R_A^{<} = \{(x_i, x_j),\ x_i^* \subset x_j^*\}.$$

If $R_A^{<} = \emptyset$, then the formal context is called an antichain formal context.

Example 3.2. Table 2 gives an antichain formal context and Figure 2 gives the corresponding concept lattice.

Table 2. An antichain formal context (U, A, I)

	a	b	c	d	e
1	1	1	0	0	1
2	1	1	1	0	0
3	0	0	0	1	1
4	1	0	1	0	1

From Table 2, we have $1^* = \{a, b, e\}$, $2^* = \{a, b, c\}$, $3^* = \{d\}$, $4^* = \{a, c, e\}$. Obviously, $R_A^< = \emptyset$, which means that (U, A, I) is an antichain formal context.

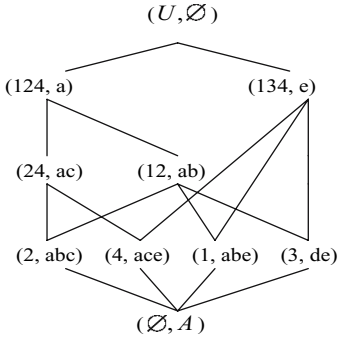


Fig. 2. The concept lattice of (U, A, I)

The set of all extensions of formal concepts of the context (U, A, I) is

$$L_u(U, A, I) = \{U, 124, 134, 24, 12, 1, 2, 3, 4, \emptyset\}$$

Table 2 as an information system, it can be calculated that

$$U/R_A = \{C_1, C_2, C_3, C_4, \}$$

where $C_1 = [1]_A = \{1\}$, $C_2 = [2]_A = \{2\}$, $C_3 = [3]_A = \{3, \}$, $C_4 = [4]_A = \{4\}$.

Obviously, for any equivalence class, it must be the extension of a formal concept.

Theorem 3.4. Let (U, A, I) be a formal context. If $(x_i, x_j) \in R_A^<$, then $[x_i]_A \notin L_u(U, A.I)$.

Proof. Since $(x_i, x_j) \in R_A^<$, we have $x_i^* \subset x_j^*$. Hence $x_i^* = x_i^* \cap x_j^* = (x_i \cup x_j)^*$. Simultaneously, since $(x_i^{**}, x_i^*) \in L(U, A, I)$, $x_j \in x_i^{**}$, we have $x_j \notin [x_i]_A$. Therefore $[x_i]_A \notin L_u(U, A.I)$.

Theorem 3.5. Let (U, A, I) be a formal context, we denote

$$U_0 = \{x_j; \text{ not exists } x_k, (x_j, x_k) \in R_A^<\}.$$

For any $x_j \in U_0$, then $[x_j]_A \in L_u(U, A, I)$.

Proof. Since $x_j^{**} = \{x; x^* \supseteq x_j^*\} = \{x; x^* = x_j^*\} = [x_j]_A$, we have $[x_j]_A \in L_u(U, A, I)$.

Theorem 3.6. Let (U, A, I) be a formal context. Then $R_A^< = \emptyset \Longleftrightarrow$ Every equivalence class is a formal concept.

Proof. If $R_A^< = \emptyset$. Then the conclusion follows from Theorem 3.5.

If every equivalence class is a concept, then $[x_i]_A = [x_i]_A^{**}$. Since $[x_i]_A^{**} = \{x_j; x_j^* \supseteq x_i^*\}$, there exists no x such that $x^* \supset x_i^*$. Therefore $R_A^< = \emptyset$.

Theorem 3.7. Let (U, A, I) be a formal context. If $D \subseteq A$ is a consistent set of the formal context (U, A, I), then it is a partition consistent set of (U, A, I).

Proof. If D is not a partition consistent set of (U, A, I), then $R_A \nsubseteq R_D$, there exists $(x_i, x_j) \in R_D$ such that $(x_i, x_j) \notin R_A$. Hence for the formal context $(U, D, I \cap (U \times D)), x_i^* = x_j^*$, and for the formal context $(U, A, I), x_i^* \neq x_j^*$. Thus D is not partition consistent set. This is a contradiction. Hence D is a partition consistent set of (U, A, I).

Corollary 3.7. Let (U, A, I) be a formal context, and C_A a partition core of (U, A, I). If $D \subseteq A$ is a reduct of the formal context (U, A, I), then $C_A \subseteq D$.

Example 3.3. From Table 1, by using the method in [15] , We conclude that $D = \{abce\}$ is a reduct of the formal context. It can also be calculated by the discernibility matrix method [14] that the information system has two reducts $B_1 = \{ace\}$, and $B_2 = \{abcd\}$. therefore, the core is $C_A = \{ac\}$. It follows that D is a partition consistent set of (U, A, I), and $C_A \subseteq D$, but B_1, B_2 aren't the reducts of the formal context.

4 Conclusion

Formal concept analysis and rough set theory provide two different methods for data analysis. By studying the relationship between these two theories and combining them, we may provide more insights into data analysis. Because they contact with each other, it is very significant to compare these two theories. In this paper, based on a data table, we discuss the connections between extensions of the formal concept and equivalent class of rough set, and also discuss the relations between the reduction in the two theories.

References

1. Duntsch, I., Gediga, G.: Approximation operators in qualitative data analysis, in: Theory and Application of Relational Structures as Knowledge Instrumwnts, Deswart, H., Orlowska, E., Schmidt, G. and Roubens, M. (Eds.), Springer, Heidelberg, (2003) 216-233

2. Gediga, G. and Duntsch, I.: Modal-style operators in qualitative data analysis, Proceedings of the 2002 IEEE International Conference on Data Mining, (2002)155-162
3. Gediga, G. and Duntsch, I.:Skills and knowledge structures. British Journal of Mathematical and Statistical Psychology, 48(1995) 9-27
4. Ganter,B., Wille,R.:Formal concept analysis, Mathematical Foundations Springer, Belin (1999)
5. Hu, K., Sui, Y., Lu, Y., Wang, J., Shi, C.: Concept approximation in concept lattice. Knowledge Discovery and Data Mining. Proceedings of the 5th Pacific-Asia Conference, PAKDD 2001, Lecture Notes in Computer Science 2035, 2001,167-173
6. Kent,R.E.:Rough concept analysis: a synthesis of rough sets and formal concept analysis,Fundamenta Informaticae,27(1996)169-181
7. Pagliani,P.,:From concept lattices to approximation spaces:algebraic structures of some spaces of partial objects,Fundamenta Informaticae,18(1993)1-18
8. Pawlak,Z.,: Rough sets: Theoretical aspects of Reasoning about Data, Kluwer Academic Publishers, Boston, 1991
9. Saquer, J., Deogun, J.S.: Formal rough concept analysis, New directions in Rough Sets,Data Mining, and Granular-Soft Computer Science 1711, Springer, Berlin,(1999)91-99
10. Wille,R.,: Restructuring lattice lattice Theory:an approach based on hierarchies of concept,in:Ordered Sets,Rival,1.(Ed.),Reidel,Dordrecht-Boston,(1982) 445-470
11. Wolff,K.E.,:A comceptual view of knowledge bases in rough set theory,Rough Sets and Current Trends in Computing, Second International Conference, RSCTC 2000,Lecture Notes in Computer Science 2005,Springer, Berlin,(2001)220-228
12. Yao,Y.Y.,:A comparative study of formal concept analysis and rough set theory in Data analysis, Rough Sets and Current Trends in Computing, 4th International Conference (RSCTC 2004) Proceedings, Lecture Notes in Computer Science 3066, Tsumoto, S., Slowinski, R., Jan Komorowski, H., Grzymala-Busse, J.W. (Eds.), Springer, Berlin, Uppsala, Sweden, June 1-5, 2004, pp. 59-68.
13. Yao, Y.Y., Chen, Y.H.,: Rough set approximation in formal concept analysis. In: Dick, S.,Kurgan, L., Pedrycz, W., Reformat, M. (eds.): Proceedings of 2004 Annual Meeting of the North American Fuzzy Information Processing Society. IEEE (2004)73-78
14. Zhang,W.X., Leung,Y., Wu,W.Z.,:Information Systems and Knowledge Discovery, Science Press, Beijing, 2003.
15. Zhang, W.X. and Qiu, G.F.: Uncertain Decision Making Based on Rough sets, Tsinghua University Press, Beijing, (2005).

Extended SMART Algorithms for Non-negative Matrix Factorization

Invited Paper

Andrzej Cichocki[1,⋆], Shun-ichi Amari[2],
Rafal Zdunek[1,⋆⋆], Raul Kompass[1,⋆⋆⋆], Gen Hori[1], and Zhaohui He[1,†]

[1] Laboratory for Advanced Brain Signal Processing
[2] Amari Research Unit for Mathematical Neuroscience,
BSI, RIKEN, Wako-shi Japan

Abstract. In this paper we derive a family of new extended SMART (Simultaneous Multiplicative Algebraic Reconstruction Technique) algorithms for Non-negative Matrix Factorization (NMF). The proposed algorithms are characterized by improved efficiency and convergence rate and can be applied for various distributions of data and additive noise. Information theory and information geometry play key roles in the derivation of new algorithms. We discuss several loss functions used in information theory which allow us to obtain generalized forms of multiplicative NMF learning adaptive algorithms. We also provide flexible and relaxed forms of the NMF algorithms to increase convergence speed and impose an additional constraint of sparsity. The scope of these results is vast since discussed generalized divergence functions include a large number of useful loss functions such as the Amari α– divergence, Relative entropy, Bose-Einstein divergence, Jensen-Shannon divergence, J-divergence, Arithmetic-Geometric (AG) Taneja divergence, etc. We applied the developed algorithms successfully to Blind (or semi blind) Source Separation (BSS) where sources may be generally statistically dependent, however are subject to additional constraints such as non-negativity and sparsity. Moreover, we applied a novel multilayer NMF strategy which improves performance of the most proposed algorithms.

1 Introduction and Problem Formulation

NMF (Non-negative Matrix Factorization) called also PMF (Positive Matrix Factorization) is an emerging technique for data mining, dimensionality reduction, pattern recognition, object detection, classification, gene clustering, sparse nonnegative representation and coding, and blind source separation (BSS)

⋆ On leave from Warsaw University of Technology, Poland.
⋆⋆ On leave from Institute of Telecommunications, Teleinformatics and Acoustics, Wroclaw University of Technology, Poland.
⋆⋆⋆ Freie University, Berlin, Germany.
† On leave from the South China University, Guangzhou, China.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 548–562, 2006.
© Springer-Verlag Berlin Heidelberg 2006

[1, 2, 3, 4, 5, 6]. The NMF, first introduced by Paatero and Trapper, and further investigated by many researchers [7, 8, 9, 10, 4, 11, 12], does not assume explicitly or implicitly sparseness, smoothness or mutual statistical independence of hidden (latent) components, however it usually provides quite a sparse decomposition [1, 13, 9, 5]. NMF has already found a wide spectrum of applications in PET, spectroscopy, chemometrics and environmental science where the matrices have clear physical meanings and some normalization or constraints are imposed on them (for example, the matrix $\boldsymbol{A}$ has columns normalized to unit length) [7, 2, 3, 5, 14, 15]. Recently, we have applied NMF with temporal smoothness and spatial constraints to improve the analysis of EEG data for early detection of Alzheimer's disease [16]. A NMF approach is promising in many applications from engineering to neuroscience since it is designed to capture alternative structures inherent in data and, possibly to provide more biological insight. Lee and Seung introduced NMF in its modern formulation as a method to decompose patterns or images [1, 13].

NMF decomposes the data matrix $\boldsymbol{Y} = [\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(N)] \in \mathbb{R}^{m \times N}$ as a product of two matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{X} = [\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(N)] \in \mathbb{R}^{n \times N}$ having only non-negative elements. Although some decompositions or matrix factorizations provide an exact reconstruction of the data (i.e., $\boldsymbol{Y} = \boldsymbol{AX}$), we shall consider here decompositions which are approximative in nature, i.e.,

$$\boldsymbol{Y} = \boldsymbol{AX} + \boldsymbol{V}, \qquad \boldsymbol{A} \geq 0, \;\; \boldsymbol{X} \geq 0 \tag{1}$$

or equivalently $\boldsymbol{y}(k) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{v}(k), \;\; k = 1, 2, \ldots, N$ or in a scalar form as $y_i(k) = \sum_{j=1}^{n} a_{ij} x_j(k) + \nu_i(k), \;\; i = 1, \ldots, m, \;\;$ with $a_{ij} \geq 0$ and $x_{jk} \geq 0$ where $\boldsymbol{V} \in \mathbb{R}^{m \times N}$ represents the noise or error matrix (depending on applications), $\boldsymbol{y}(k) = [y_1(k), \ldots, y_m(k)]^T$ is a vector of the observed signals (typically positive) at the discrete time instants[1] k while $\boldsymbol{x}(k) = [x_1(k), \ldots, x_n(k)]^T$ is a vector of nonnegative components or source signals at the same time instant [17]. Due to additive noise the observed data might sometimes take negative values. In such a case we apply the following approximation: $\hat{y}_i(k) = y_i(k)$ if $y_i(k)$ is positive and otherwise $\hat{y}_i(k) = \varepsilon$, where ε is a small positive constant. Our objective is to estimate the mixing (basis) matrix $\boldsymbol{A}$ and sources $\boldsymbol{X}$ subject to nonnegativity constraints of all entries of $\boldsymbol{A}$ and $\boldsymbol{X}$. Usually, in BSS applications it is assumed that $N >> m \geq n$ and n is known or can be relatively easily estimated using SVD or PCA. Throughout this paper, we use the following notations: $x_j(k) = x_{jk}$, $y_i(k) = y_{ik}$ and $z_{ik} = [\boldsymbol{AX}]_{ik}$ means ik-th element of the matrix $(\boldsymbol{AX})$, and the ij-th element of the matrix $\boldsymbol{A}$ is denoted by a_{ij}.

The main objective of this contribution is to derive a family of new flexible and improved NMF algorithms that allow to generalize or combine different criteria in order to extract physically meaningful sources, especially for biomedical signal applications such as EEG and MEG.

[1] The data are often represented not in the time domain but in a transform domain such as the time frequency domain, so index k may have different meanings.

2 Extended Lee-Seung Algorithms and Fixed Point Algorithms

Although the standard NMF (without any auxiliary constraints) provides sparseness of its component, we can achieve some control of this sparsity as well as smoothness of components by imposing additional constraints in addition to non-negativity constraints. In fact, we can incorporate smoothness or sparsity constraints in several ways [9]. One of the simple approach is to implement in each iteration step a nonlinear projection which can increase the sparseness and/or smoothness of estimated components. An alternative approach is to add to the loss function suitable regularization or penalty terms. Let us consider the following constrained optimization problem:

Minimize:

$$D_F^{(\alpha)}(\boldsymbol{A},\boldsymbol{X}) = \frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X}\|_F^2 + \alpha_A J_A(\boldsymbol{A}) + \alpha_X J_X(\boldsymbol{X}) \tag{2}$$

$$s.t.\ \ a_{ij} \geq 0, \quad x_{jk} \geq 0, \ \ \forall\, i,j,k, \tag{3}$$

where α_A and $\alpha_X \geq 0$ are nonnegative regularization parameters and terms $J_X(\boldsymbol{X})$ and $J_A(\boldsymbol{A})$ are used to enforce a certain application-dependent characteristics of the solution. As a special practical case we have $J_X(\boldsymbol{X}) = \sum_{jk} f_X(x_{jk})$, where $f(\cdot)$ are suitably chosen functions which are the measures of smoothness or sparsity. In order to achieve sparse representation we usually choose $f(x_{jk}) = |x_{jk}|$ or simply $f(x_{jk}) = x_{jk}$, or alternatively $f(x_{jk}) = x_{jk}\ln(x_{jk})$ with constraints $x_{jk} \geq 0$. Similar regularization terms can be also implemented for the matrix $\boldsymbol{A}$. Note that we treat both matrices $\boldsymbol{A}$ and $\boldsymbol{X}$ in a symmetric way. Applying the standard gradient descent approach, we have

$$a_{ij} \leftarrow a_{ij} - \eta_{ij}\frac{\partial D_F^{(\alpha)}(\boldsymbol{A},\boldsymbol{X})}{\partial a_{ij}}, \quad x_{jk} \leftarrow x_{jk} - \eta_{jk}\frac{\partial D_F^{(\alpha)}(\boldsymbol{A},\boldsymbol{X})}{\partial x_{jk}}, \tag{4}$$

where η_{ij} and η_{jk} are positive learning rates. The gradient components can be expressed in a compact matrix form as:

$$\frac{\partial D_F^{(\alpha)}(\boldsymbol{A},\boldsymbol{X})}{\partial a_{ij}} = [-\boldsymbol{Y}\boldsymbol{X}^T + \boldsymbol{A}\boldsymbol{X}\boldsymbol{X}^T]_{ij} + \alpha_A\frac{\partial J_A(\boldsymbol{A})}{\partial a_{ij}}, \tag{5}$$

$$\frac{\partial D_F^{(\alpha)}(\boldsymbol{A},\boldsymbol{X})}{\partial x_{jk}} = [-\boldsymbol{A}^T\boldsymbol{Y} + \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{X}]_{jk} + \alpha_X\frac{\partial J_X(\boldsymbol{X})}{\partial x_{jk}}. \tag{6}$$

Here, we follow the Lee and Seung approach to choose specific learning rates [1, 3]:

$$\eta_{ij} = \frac{a_{ij}}{[\boldsymbol{A}\boldsymbol{X}\boldsymbol{X}^T]_{ij}}, \quad \eta_{jk} = \frac{x_{jk}}{[\boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{X}]_{jk}}, \tag{7}$$

that leads to a generalized robust multiplicative update rules:

$$a_{ij} \leftarrow a_{ij} \frac{\left[\boldsymbol{Y}\,\boldsymbol{X}^T]_{ij} - \alpha_A \varphi_A(a_{ij})\right]_\varepsilon}{[\boldsymbol{A}\,\boldsymbol{X}\,\boldsymbol{X}^T]_{ij} + \varepsilon}, \tag{8}$$

$$x_{jk} \leftarrow x_{jk} \frac{\left[[\boldsymbol{A}^T\,\boldsymbol{Y}]_{jk} - \alpha_X\ \varphi_X(x_{jk})\right]_\varepsilon}{[\boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{X}]_{jk} + \varepsilon}, \tag{9}$$

where the nonlinear operator is defined as $[x]_\varepsilon = \max\{\varepsilon, x\}$ with a small positive ε and the functions $\varphi_A(a_{ij})$ and $\varphi_X(x_{jk})$ are defined as

$$\varphi_A(a_{ij}) = \frac{\partial J_A(\boldsymbol{A})}{\partial a_{ij}}, \quad \varphi_X(x_{jk}) = \frac{\partial J_X(\boldsymbol{X})}{\partial x_{jk}}. \tag{10}$$

Typically, $\varepsilon = 10^{-16}$ is introduced in order to ensure non-negativity constraints and avoid possible division by zero. The above Lee-Seung algorithm can be considred as an extension of the well known ISRA (Image Space Reconstruction Algorithm) algorithm. The above algorithm reduces to the standard Lee-Seung algorithm for $\alpha_A = \alpha_X = 0$. In the special case, by using the l_1-norm regularization terms $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$ for both matrices $\boldsymbol{X}$ and $\boldsymbol{A}$ the above multiplicative learning rules can be simplified as follows:

$$a_{ij} \leftarrow a_{ij} \frac{\left[[\boldsymbol{Y}\,\boldsymbol{X}^T]_{ij} - \alpha_A\right]_\varepsilon}{[\boldsymbol{A}\,\boldsymbol{X}\,\boldsymbol{X}^T]_{ij} + \varepsilon}, \qquad x_{jk} \leftarrow x_{jk} \frac{\left[[\boldsymbol{A}^T\,\boldsymbol{Y}]_{jk} - \alpha_X\right]_\varepsilon}{[\boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{X}]_{jk} + \varepsilon}, \tag{11}$$

with normalization in each iteration as follows $a_{ij} \leftarrow a_{ij} / \sum_{i=1}^{m} a_{ij}$. Such normalization is necessary to provide desired sparseness. Algorithm (11) provides a sparse representation of the estimated matrices and the sparseness measure increases with increasing values of regularization coefficients, typically $\alpha_X = 0.01 \sim 0.5$.

It is worth to note that we can derive as alternative to the Lee-Seung algorithm (11) a Fixed Point NMF algorithm by equalizing the gradient components of (5)-(6) (for l_1-norm regularization terms) to zero [18] :

$$\nabla_X D_F^{(\alpha)}(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}) = \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{X} - \boldsymbol{A}^T\boldsymbol{Y} + \alpha_X = 0, \tag{12}$$

$$\nabla_A D_F^{(\alpha)}(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}) = \boldsymbol{A}\boldsymbol{X}\boldsymbol{X}^T - \boldsymbol{Y}\boldsymbol{X}^T + \alpha_A = 0. \tag{13}$$

These equations suggest the following fixed point updates rules:

$$\boldsymbol{X} \leftarrow \max\left\{\varepsilon, \left[(\boldsymbol{A}^T\boldsymbol{A})^+(\boldsymbol{A}^T\boldsymbol{Y} - \alpha_X)\right]\right\} = \left[(\boldsymbol{A}^T\boldsymbol{A})^+(\boldsymbol{A}^T\boldsymbol{Y} - \alpha_X)\right]_\varepsilon, \tag{14}$$

$$\boldsymbol{A} \leftarrow \max\left\{\varepsilon, \left[(\boldsymbol{Y}\boldsymbol{X}^T - \alpha_A)(\boldsymbol{X}\boldsymbol{X}^T)^+\right]\right\} = \left[(\boldsymbol{Y}\boldsymbol{X}^T - \alpha_A)(\boldsymbol{X}\boldsymbol{X}^T)^+\right]_\varepsilon, \tag{15}$$

where $[\boldsymbol{A}]^+$ means Moore-Penrose pseudo-inverse and *max* function is componentwise. The above algorithm can be considered as nonlinear projected Alternating Least Squares (ALS) or nonlinear extension of EM-PCA algorithm.

Furthermore, using the Interior Point Gradient (IPG) approach an additive algorithm can be derived (which is written in a compact matrix form using MATLAB notations):

$$\boldsymbol{A} \leftarrow \boldsymbol{A} - \boldsymbol{\eta}_A * (\boldsymbol{A} \,./\, (\boldsymbol{A} * \boldsymbol{X} * \boldsymbol{X}')) \,.*\, ((\boldsymbol{A} * \boldsymbol{X} - \boldsymbol{Y}) * \boldsymbol{X}'), \tag{16}$$

$$\boldsymbol{X} \leftarrow \boldsymbol{X} - \boldsymbol{\eta}_X * (\boldsymbol{X}./(\boldsymbol{A}' * \boldsymbol{A} * \boldsymbol{X})) \,.*\, (\boldsymbol{A}' * (\boldsymbol{A} * \boldsymbol{X} - \boldsymbol{Y})), \tag{17}$$

where operators .* and ./ mean component-wise multiplications and division, respectively, and $\boldsymbol{\eta}_A$ and $\boldsymbol{\eta}_X$ are diagonal matrices with positive entries representing suitably chosen learning rates [19].

Alternatively, the mostly used loss function for the NMF that intrinsically ensures non-negativity constraints and it is related to the Poisson likelihood is based on the generalized Kullback-Leibler divergence (also called I-divergence):

$$D_{KL1}(\boldsymbol{Y}||\boldsymbol{AX}) = \sum_{ik} \left(y_{ik} \, \ln \frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} + [\boldsymbol{AX}]_{ik} - y_{ik} \right), \tag{18}$$

On the basis of this cost function we proposed a modified Lee-Seung learning algorithm:

$$x_{jk} \leftarrow \left(x_{jk} \, \frac{\sum_{i=1}^{m} a_{ij} \, (y_{ik}/[\boldsymbol{A}\,\boldsymbol{X}]_{ik})}{\sum_{q=1}^{m} a_{qj}} \right)^{1+\alpha_{sX}}, \tag{19}$$

$$a_{ij} \leftarrow \left(a_{ij} \, \frac{\sum_{k=1}^{N} x_{jk} \, (y_{ik}/[\boldsymbol{A}\,\boldsymbol{X}]_{ik})}{\sum_{p=1}^{N} x_{jp}} \right)^{1+\alpha_{sA}}, \tag{20}$$

where additional small regularization terms $\alpha_{sX} \geq 0$ and $\alpha_{sX} \geq 0$ are introduced in order to enforce sparseness of the solution, if necessary. Typical values of the regularization parameters are $\alpha_{sX} = \alpha_{sA} = 0.001 \sim 0.005$.

Raul Kompass proposed to apply beta divergence to combine the both Lee-Seung algorithms (11) and (19)-(20) into one flexible and elegant algorithm with a single parameter [10]. Let us consider beta divergence in the following generalized form as the cost for the NMF problem [10, 20, 6]:

$$D_K^{(\beta)}(\boldsymbol{Y}||\boldsymbol{AX}) = \sum_{ik} \left(y_{ik} \, \frac{y_{ik}^{\beta} - [\boldsymbol{AX}]_{ik}^{\beta}}{\beta(\beta+1)} + [\boldsymbol{AX}]_{ik}^{\beta} \frac{[\boldsymbol{AX}]_{ik} - y_{ik}}{\beta+1} \right) + \alpha_X \|\boldsymbol{X}\|_1 + \alpha_A \|\boldsymbol{A}\|_1, \tag{21}$$

where α_X and α_A are small positive regularization parameters which control the degree of smoothing or sparseness of the matrices $\boldsymbol{A}$ and $\boldsymbol{X}$, respectively, and l_1-norms $||\boldsymbol{A}||_1$ and $||\boldsymbol{X}||_1$ are introduced to enforce sparse representation of solutions. It is is interesting to note that for $\beta = 1$ we obtain the square Euclidean distance expressed by Frobenius norm (2), while for the singular cases $\beta = 0$ and $\beta = -1$ the beta divergence has to be defined as limiting cases as $\beta \to 0$ and $\beta \to -1$, respectively. When these limits are evaluated one gets for

$\beta \to 0$ the generalized Kullback-Leibler divergence (called I-divergence) defined by equations (18) and for $\beta \to -1$ the Itakura-Saito distance can be obtained:

$$D_{I-S}(\boldsymbol{Y}||\boldsymbol{AX}) = \sum_{ik} \left[\ln(\frac{[\boldsymbol{AX}]_{ik}}{y_{ik}}) + \frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} - 1 \right]. \tag{22}$$

The choice of the β parameter depends on statistical distribution of data and the beta divergence corresponds to the Tweedie models [21, 20]. For example, the optimal choice of the parameter for the normal distribution is $\beta = 1$, for the gamma distribution is $\beta \to -1$, for the Poisson distribution $\beta \to 0$, and for the compound Poisson $\beta \in (-1, 0)$.

From the beta generalized divergence we can derive various kinds of NMF algorithms: Multiplicative based on the standard gradient descent or the Exponentiated Gradient (EG) algorithms (see next section), additive algorithms using Projected Gradient (PG) or Interior Point Gradient (IPG), and Fixed Point (FP) algorithms.

In order to derive a flexible NMF learning algorithm, we compute the gradient of (21) with respect to elements of matrices $x_{jk} = x_j(k) = [\boldsymbol{X}]_{jk}$ and $a_{ij} = [\boldsymbol{A}]_{ij}$. as follows

$$\frac{\partial D_K^{(\beta)}}{\partial x_{jk}} = \sum_{i=1}^{m} a_{ij} \ \left([\boldsymbol{AX}]_{ik}^{\beta} - y_{ik} \ [\boldsymbol{AX}]_{ik}^{\beta-1} \right) + \alpha_X, \tag{23}$$

$$\frac{\partial D_K^{(\beta)}}{\partial a_{ij}} = \sum_{k=1}^{N} \left(\ [\boldsymbol{AX}]_{ik}^{\beta} \ - y_{ik} [\boldsymbol{AX}]_{ik}^{\beta-1} \right) \ x_{jk} + \alpha_A. \tag{24}$$

Similar to the Lee and Seung approach, by choosing suitable learning rates:

$$\eta_{jk} = \frac{x_{jk}}{\sum_{i=1}^{m} a_{ij} [\boldsymbol{AX}]_{ik}^{\beta}}, \quad \widetilde{\eta}_{ij} = \frac{a_{ij}}{\sum_{k=1}^{N} [\boldsymbol{AX}]_{ik}^{\beta} \ x_{jk}}, \tag{25}$$

we obtain multiplicative update rules [10, 6]:

$$x_{jk} \leftarrow x_{jk} \frac{[\sum_{i=1}^{m} a_{ij} \ (y_{ik} / [\boldsymbol{AX}]_{ik}^{1-\beta}) - \alpha_X]_{\varepsilon}}{\sum_{i=1}^{m} a_{ij} \ [\boldsymbol{AX}]_{ik}^{\beta}}, \tag{26}$$

$$a_{ij} \leftarrow a_{ij} \ \frac{[\sum_{k=1}^{N} (y_{ik} / [\boldsymbol{AX}]_{ik}^{1-\beta}) \ y_{jk} - \alpha_A]_{\varepsilon}}{\sum_{k=1}^{N} [\boldsymbol{AX}]_{ik}^{\beta} \ x_{jk}}, \tag{27}$$

where again the rectification defined as $[x]_{\varepsilon} = \max\{\varepsilon, x\}$ with a small ε is introduced in order to avoid zero and negative values.

3 SMART Algorithms for NMF

There are two large classes of generalized divergences which can be potentially used for developing new flexible algorithms for NMF: the Bregman divergences

and the Csiszár's φ-divergences [22, 23, 24]. In this contribution we limit our discussion to the some generalized entropy divergences.

Let us consider at beginning the generalized K-L divergence dual to (18):

$$D_{KL}(\boldsymbol{AX}||\boldsymbol{Y}) = \sum_{ik} \left([\boldsymbol{AX}]_{ik} \ln \left(\frac{[\boldsymbol{AX}]_{ik}}{y_{ik}} \right) - [\boldsymbol{AX}]_{ik} + y_{ik} \right) \tag{28}$$

subject to nonnegativity constraints (see Eq. (18)) In order to derive the learning algorithm let us apply multiplicative exponentiated gradient (EG) descent updates to the loss function (28):

$$x_{jk} \leftarrow x_{jk} \exp\left(-\eta_{jk} \frac{\partial D_{KL}}{\partial x_{jk}} x_{jk} \right), \quad a_{ij} \leftarrow a_{ij} \exp\left(-\eta_{ij} \frac{\partial D_{KL}}{\partial a_{ij}} a_{ij} \right), \tag{29}$$

where

$$\frac{\partial D_{KL}}{\partial x_{jk}} = \sum_{i=1}^{m} (a_{ij} \ln[\boldsymbol{AX}]_{ik} - a_{ij} \ \ln y_{ik}) \tag{30}$$

$$\frac{\partial D_{KL}}{\partial a_{ij}} = \sum_{k=1}^{N} (x_{jk} \ln[\boldsymbol{AX}]_{ik} - x_{jk} \ \ln y_{ik}) \,. \tag{31}$$

Hence, we obtain the simple multiplicative learning rules:

$$x_{jk} \leftarrow x_{jk} \exp\left(\sum_{i=1}^{m} \overline{\eta}_{jk} a_{ij} \ln \left(\frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} \right) \right) = x_{jk} \prod_{i=1}^{m} \left(\frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} \right)^{\overline{\eta}_{jk} a_{ij}} \tag{32}$$

$$a_{ij} \leftarrow a_{ij} \exp\left(\sum_{k=1}^{N} \widetilde{\eta}_{ij} x_{jk} \ln \left(\frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} \right) \right) = a_{ij} \prod_{k=1}^{N} \left(\frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} \right)^{\widetilde{\eta}_{ij} x_{jk}} \tag{33}$$

The nonnegative learning rates $\overline{\eta}_{jk}$ and $\widetilde{\eta}_{ij}$ can take different forms. Typically, for simplicity and in order to guarantee stability of the algorithm we assume that $\overline{\eta}_{jk} = \overline{\eta}_j = \omega \left(\sum_{i=1}^{m} a_{ij} \right)^{-1}$, $\widetilde{\eta}_{ij} = \widetilde{\eta}_j = \omega \left(\sum_{k=1}^{N} x_{jk} \right)^{-1}$, where $\omega \in (0, 2)$ is an over-relaxation parameter. The EG updates can be further improved in terms of convergence, computational efficiency and numerical stability in several ways.

In order to keep weight magnitudes bounded, Kivinen and Warmuth proposed a variation of the EG method that applies a normalization step after each weight update. The normalization linearly rescales all weights so that they sum to a constant. Moreover, instead of the exponent function we can apply its re-linearizing approximation: $e^u \approx \max\{0.5, 1 + u\}$. To further accelerate its convergence, we may apply individual adaptive learning rates defined as $\eta_{jk} \leftarrow \eta_{jk} c$ if the corresponding gradient component $\partial D_{KL} / \partial x_{jk}$ has the same sign in two consecutive steps and $\eta_{jk} \leftarrow \eta_{jk} / c$ otherwise, where $c > 1$ (typically $c = 1.02 - 1.5$) [25].

The above multiplicative learning rules can be written in a more generalized and compact matrix form (using MATLAB notations):

$$\boldsymbol{X} \leftarrow \boldsymbol{X} \ .* \ \exp\left(\boldsymbol{\eta_X} \ .* \ (\boldsymbol{A}' * \ln(\boldsymbol{Y}./(\boldsymbol{A} * \boldsymbol{X} + \epsilon))) \right) \tag{34}$$

$$\boldsymbol{A} \leftarrow \boldsymbol{A} \ .* \ \exp\left(\boldsymbol{\eta}_A \ .* \ (\ln(\boldsymbol{Y}./(\boldsymbol{A} * \boldsymbol{X} + \epsilon)) * \boldsymbol{X}') \right), \tag{35}$$

$$\boldsymbol{A} \leftarrow \boldsymbol{A} * \mathrm{diag}\{1./sum(\boldsymbol{A}, 1)\}, \tag{36}$$

where in practice a small constant $\varepsilon = 10^{-16}$ is introduced in order to ensure positivity constraints and/or to avoid possible division by zero, and $\boldsymbol{\eta}_A$ and $\boldsymbol{\eta}_X$ are non-negative scaling matrices representing individual learning rates. The above algorithm may be considered as an alternating minimization/projection extension of the well known SMART (Simultaneous Multiplicative Algebraic Reconstruction Technique) [26, 27]. This means that the above NMF algorithm can be extended to MART and BI-MART (Block-Iterative Multiplicative Algebraic Reconstruction Technique) [26].

It should be noted that the parameters (weights) $\{x_{jk}, a_{ij}\}$ are restricted to positive values, the resulting updates rules can be written:

$$\ln(x_{jk}) \leftarrow \ln(x_{jk}) - \eta_{jk} \frac{\partial D_{KL}}{\partial \ln x_{jk}}, \quad \ln(a_{ij}) \leftarrow \ln(a_{ij}) - \eta_{ij} \frac{\partial D_{KL}}{\partial \ln a_{ij}}, \tag{37}$$

where the natural logarithm projection is applied component-wise. Thus, in a sense, the EG approach takes the same steps as the standard gradient descent (GD), but in the space of logarithm of the parameters. In other words, in our current application the scalings of the parameters $\{x_{jk}, a_{ij}\}$ are best adapted in log-space, where their gradients are much better behaved.

4 NMF Algorithms Using Amari α-Divergence

It is interesting to note, that the above SMART algorithm can be derived as a special case for a more general loss function called Amari α-divergence (see also Liese & Vajda, Cressie-Read disparity, Kompass generalized divergence and Eguchi-Minami beta divergence)[2] [29, 28, 23, 22, 10, 30]):

$$D_A(\boldsymbol{Y}||\boldsymbol{AX}) = \frac{1}{\alpha(\alpha-1)} \sum_{ik} \left(y_{ik}^{\alpha} z_{ik}^{1-\alpha} - \alpha y_{ik} + (\alpha-1) z_{ik} \right) \tag{38}$$

We note that as special cases of the Amari α-divergence for $\alpha = 2, 0.5, -1$, we obtain the Pearson's, Hellinger and Neyman's chi-square distances, respectively, while for the cases $\alpha = 1$ and $\alpha = 0$ the divergence has to be defined by the limits $\alpha \to 1$ and $\alpha \to 0$, respectively. When these limits are evaluated one obtains for $\alpha \to 1$ the generalized Kullback-Leibler divergence defined by equations (18) and for $\alpha \to 0$ the dual generalized KL divergence (28).

The gradient of the above cost function can be expressed in a compact form as

$$\frac{\partial D_A}{\partial x_{jk}} = \frac{1}{\alpha} \sum_{i=1}^{m} a_{ij} \left[1 - \left(\frac{y_{ik}}{z_{ik}} \right)^{\alpha} \right], \quad \frac{\partial D_A}{\partial a_{ij}} = \frac{1}{\alpha} \sum_{k=1}^{N} x_{jk} \left[1 - \left(\frac{y_{ik}}{z_{ik}} \right)^{\alpha} \right]. \tag{39}$$

[2] Note that this form of α–divergence differs slightly with the loss function of Amari given in 1985 and 2000 [28, 23] by the additional term. This term is needed to allow de-normalized variables, in the same way that extended Kullback-Leibler divergence differs from the standard form (without terms $z_{ik} - y_{ik}$) [24].

However, instead of applying the standard gradient descent we use the projected (linearly transformed) gradient approach (which can be considered as generalization of exponentiated gradient):

$$\Phi(x_{jk}) \leftarrow \Phi(x_{jk}) - \eta_{jk}\frac{\partial D_A}{\partial \Phi(x_{jk})}, \quad \Phi(a_{ij}) \leftarrow \Phi(a_{ij}) - \eta_{ij}\frac{\partial D_A}{\partial \Phi(a_{ij})}, \tag{40}$$

where $\Phi(x)$ is a suitable chosen function.

Hence, we have

$$x_{jk} \leftarrow \Phi^{-1}\left(\Phi(x_{jk}) - \eta_{jk}\frac{\partial D_A}{\partial \Phi(x_{jk})}\right), \tag{41}$$

$$a_{ij} \leftarrow \Phi^{-1}\left(\Phi(a_{ij}) - \eta_{ij}\frac{\partial D_A}{\partial \Phi(a_{ij})}\right). \tag{42}$$

It can be shown that such nonlinear scaling or transformation provides stable solution and the gradients are much better behaved in Φ space. In our case, we employ $\Phi(x) = x^\alpha$ and choose the learning rates as follows

$$\eta_{jk} = \alpha^2\Phi(x_{jk})/(x_{jk}^{1-\alpha}\sum_{i=1}^{m} a_{ij}), \quad \eta_{ij} = \alpha^2\Phi(a_{ij})/(a_{ij}^{1-\alpha}\sum_{k=1}^{N} x_{jk}), \tag{43}$$

which leads directly to the new learning algorithm [3]: (the rigorous convergence proof is omitted due to lack of space)

$$x_{jk} \leftarrow x_{jk}\left(\frac{\sum_{i=1}^{m} a_{ij}\,(y_{ik}/z_{ik})^\alpha}{\sum_{q=1}^{m} a_{qj}}\right)^{1/\alpha}, \; a_{ij} \leftarrow a_{ij}\left(\frac{\sum_{k=1}^{N}(y_{ik}/z_{ik})^\alpha\, x_{jk}}{\sum_{t=1}^{N} x_{jt}}\right)^{1/\alpha} \tag{44}$$

This algorithm can be implemented in similar compact matrix form using the MATLAB notations:

$$\boldsymbol{X} \leftarrow \boldsymbol{X} .* \left(\boldsymbol{A}' * ((\boldsymbol{Y} + \varepsilon)./\,(\boldsymbol{A} * \boldsymbol{X} + \varepsilon)).^\alpha\right).^{1/\alpha}, \tag{45}$$

$$\boldsymbol{A} \leftarrow \boldsymbol{A} .* \left(((\boldsymbol{Y} + \varepsilon)./\,(\boldsymbol{A} * \boldsymbol{X} + \varepsilon)).^\alpha * \boldsymbol{X}'\right).^{1/\alpha}, \tag{46}$$

$$\boldsymbol{A} \leftarrow \boldsymbol{A} * \mathrm{diag}\{1./sum(\boldsymbol{A}, 1)\}.$$

Alternatively, applying the EG approach, we can obtain the following multiplicative algorithm:

$$x_{jk} \leftarrow x_{jk}\exp\left\{\overline{\eta}_{jk}\sum_{i=1}^{m} a_{ij}\,\left[\left(\frac{y_{ik}}{z_{ik}}\right)^\alpha - 1\right]\right\}, \tag{47}$$

$$a_{ij} \leftarrow a_{ij}\exp\left\{\widetilde{\eta}_{ij}\sum_{k=1}^{N}\left[\left(\frac{y_{ik}}{z_{ik}}\right)^\alpha - 1\right] x_{jk}\right\}. \tag{48}$$

[3] For $\alpha = 0$ instead of $\Phi(x) = x^\alpha$ we have used $\Phi(x) = \ln(x)$.

5 Generalized SMART Algorithms

The main objective of this paper is to show that the learning algorithm (32) and (33) can be generalized to the following flexible algorithm:

$$x_{jk} \leftarrow x_{jk} \exp\left[\sum_{i=1}^{m} \overline{\eta}_{jk}\, a_{ij}\, \rho(y_{ik}, z_{ik})\right],\ a_{ij} \leftarrow a_{ij} \exp\left[\sum_{k=1}^{N} \widetilde{\eta}_{ij}\, x_{jk}\, \rho(y_{ik}, z_{ik})\right] \tag{49}$$

where the error functions defined as

$$\rho(y_{ik}, z_{ik}) = -\frac{\partial D(\boldsymbol{Y}||\boldsymbol{AX})}{\partial z_{ik}} \tag{50}$$

can take different forms depending on the chosen or designed loss (cost) function $D(\boldsymbol{Y}||\boldsymbol{AX})$ (see Table 1).

As an illustrative example let us consider the Bose-Einstein divergence:

$$BE_\alpha(\boldsymbol{Y}||\boldsymbol{AX}) = \sum_{ik} y_{ik} \ln\left(\frac{(1+\alpha)y_{ik}}{y_{ik}+\alpha z_{ik}}\right) + \alpha z_{ik} \ln\left(\frac{(1+\alpha)z_{ik}}{y_{ik}+\alpha z_{ik}}\right). \tag{51}$$

This loss function has many interesting properties:

1. $BE_\alpha(\boldsymbol{y}||\boldsymbol{z}) = 0$ if $\boldsymbol{z} = \boldsymbol{y}$ almost everywhere.
2. $BE_\alpha(\boldsymbol{y}||\boldsymbol{z}) = BE_{1/\alpha}(\boldsymbol{z}||\boldsymbol{y})$
3. For $\alpha = 1$, BE_α simplifies to the symmetric Jensen-Shannon divergence measure (see Table 1).
4. $\lim_{\alpha\to\infty} BE_\alpha(\boldsymbol{y}||\boldsymbol{z}) = KL(\boldsymbol{y}||\boldsymbol{z})$ and for α sufficiently small $BE_\alpha(\boldsymbol{y}||\boldsymbol{z}) \approx KL(\boldsymbol{z}||\boldsymbol{y})$.

The gradient of the Bose-Einstein loss function in respect to z_{ik} can be expressed as

$$\frac{\partial BE_\alpha(\boldsymbol{Y}||\boldsymbol{AX})}{\partial z_{ik}} = -\alpha \ln\left(\frac{y_{ik}+\alpha z_{ik}}{(1+\alpha)z_{ik}}\right) \tag{52}$$

and in respect to x_{jk} and a_{ij} as

$$\frac{\partial BE_\alpha}{\partial x_{jk}} = -\sum_{i=1}^{m} a_{ij} \frac{\partial BE_\alpha}{\partial z_{ik}}, \qquad \frac{\partial BE_\alpha}{\partial a_{ij}} = -\sum_{k=1}^{N} x_{jk} \frac{\partial BE_\alpha}{\partial z_{ik}}. \tag{53}$$

Hence, applying the standard (un-normalized) EG approach (29) we obtain the learning rules (49) with the error function $\rho(y_{ik}, z_{ik}) = \alpha \ln((y_{ik}+\alpha z_{ik})/((1+\alpha)z_{ik}))$. It should be noted that the error function $\rho(y_{ik}, z_{ik}) = 0$ if and only if $y_{ik} = z_{ik}$.

6 Multi-layer NMF

In order to improve performance of the NMF, especially for ill-conditioned and badly scaled data and also to reduce risk to get stuck in local minima of non-convex minimization, we have developed a simple hierarchical and multi-stage

Table 1. Extended SMART NMF adaptive algorithms and corresponding loss functions

$a_{ij} \leftarrow a_{ij} \exp\left(\sum_{k=1}^{N} \widetilde{\eta}_{ij}\, x_{jk}\, \rho(y_{ik}, z_{ik})\right),$ $\quad a_j = \sum_{i=1}^{m} a_{ij} = 1, \ \forall j, \quad a_{ij} \geq 0$

$x_{jk} \leftarrow x_{jk} \exp\left(\sum_{i=1}^{m} \overline{\eta}_{jk}\, a_{ij}\, \rho(y_{ik}, z_{ik})\right)$ $\quad y_{ik} > 0, \ z_{ik} = [\boldsymbol{AX}]_{ik} > 0, \ x_{jk} \geq 0$

Minimization of loss function	Corresponding error function $\rho(y_{ik}, z_{ik})$
1. K-L I-divergence, $D_{KL}(\boldsymbol{AX}\|\|\boldsymbol{Y})$	
$\sum_{ik} \left(z_{ik} \ln \frac{z_{ik}}{y_{ik}} + y_{ik} - z_{ik} \right)$	$\rho(y_{ik}, z_{ik}) = \ln\left(\frac{y_{ik}}{z_{ik}}\right)$
2. Relative A-G divergence $AG_r(\boldsymbol{Y}\|\|\boldsymbol{AX})$	
$\sum_{ik} \left((y_{ik} + z_{ik}) \ln\left(\frac{y_{ik} + z_{ik}}{2y_{ik}}\right) + y_{ik} - z_{ik} \right)$	$\rho(y_{ik}, z_{ik}) = \ln\left(\frac{2y_{ik}}{y_{ik} + z_{ik}}\right)$
3. Symmetric A-G divergence $AG(\boldsymbol{Y}\|\|\boldsymbol{AX})$	
$2\sum_{ik} \left(\frac{y_{ik} + z_{ik}}{2} \ln\left(\frac{y_{ik} + z_{ik}}{2\sqrt{y_{ik} z_{ik}}}\right) \right)$	$\rho(y_{ik}, z_{ik}) = \frac{y_{ik} - z_{ik}}{2z_{ik}} + \ln\left(\frac{2\sqrt{y_{ik} z_{ik}}}{y_{ik} + z_{ik}}\right)$
4. Relative Jensen-Shannon divergence	
$\sum_{ik} \left(2y_{ik} \ln\left(\frac{2y_{ik}}{y_{ik} + z_{ik}}\right) + z_{ik} - y_{ik} \right)$	$\rho(y_{ik}, z_{ik}) = \frac{y_{ik} - z_{ik}}{y_{ik} + z_{ik}}$
5. Symmetric Jensen-Shannon divergence	
$\sum_{ik} y_{ik} \ln\left(\frac{2y_{ik}}{y_{ik} + z_{ik}}\right) + z_{ik} \ln\left(\frac{2z_{ik}}{y_{ik} + z_{ik}}\right)$	$\rho(y_{ik}, z_{ik}) = \ln\left(\frac{y_{ik} + z_{ik}}{2z_{ik}}\right)$
6. Bose-Einstein divergence $BE(\boldsymbol{Y}\|\|\boldsymbol{AX})$	
$\sum_{ik} y_{ik} \ln\left(\frac{(1+\alpha)y_{ik}}{y_{ik} + \alpha z_{ik}}\right) + \alpha z_{ik} \ln\left(\frac{(1+\alpha)z_{ik}}{y_{ik} + \alpha z_{ik}}\right)$	$\rho(y_{ik}, z_{ik}) = \alpha \ln\left(\frac{y_{ik} + \alpha z_{ik}}{(1+\alpha)z_{ik}}\right)$
7. J-divergence $D_J(\boldsymbol{Y}\|\|\boldsymbol{AX})$	
$\sum_{ik} \left(\frac{y_{ik} - z_{ik}}{2} \ln\left(\frac{y_{ik}}{z_{ik}}\right) \right)$	$\rho(y_{ik}, z_{ik}) = \frac{1}{2} \ln\left(\frac{y_{ik}}{z_{ik}}\right) + \frac{y_{ik} - z_{ik}}{2z_{ik}}$
8. Triangular Discrimination $D_T(\boldsymbol{Y}\|\|\boldsymbol{AX})$	
$\sum_{ik} \left\{ \frac{(y_{ik} - z_{ik})^2}{y_{ik} + z_{ik}} \right\}$	$\rho(y_{ik}, z_{ik}) = \left(\frac{2y_{ik}}{y_{ik} + z_{ik}}\right)^2 - 1$
9. Amari's α divergence $D_A(\boldsymbol{Y}\|\|\boldsymbol{AX})$	
$\frac{1}{\alpha(\alpha-1)} \sum_{ik} \left(y_{ik}^{\alpha} z_{ik}^{1-\alpha} - y_{ik} + (\alpha - 1)(z_{ik} - y_{ik}) \right)$	$\rho(y_{ik}, z_{ik}) = \frac{1}{\alpha} \left[\left(\frac{y_{ik}}{z_{ik}}\right)^{\alpha} - 1 \right]$

procedure in which we perform a sequential decomposition of nonnegative matrices as follows: In the first step, we perform the basic decomposition (factorization) $\boldsymbol{Y} = \boldsymbol{A}_1 \boldsymbol{X}_1$ using any available NMF algorithm. In the second stage, the results obtained from the first stage are used to perform the similar decomposition: $\boldsymbol{X}_1 = \boldsymbol{A}_2 \boldsymbol{X}_2$ using the same or different update rules, and so on. We continue our decomposition taking into account only the last achieved components. The process can be repeated arbitrarily many times until some stopping criteria are satisfied. In each step, we usually obtain gradual improvements of the performance. Thus, our model has the form: $\boldsymbol{Y} = \boldsymbol{A}_1 \boldsymbol{A}_2 \cdots \boldsymbol{A}_L \boldsymbol{X}_L$, with the basis nonnegative matrix defined as $\boldsymbol{A} = \boldsymbol{A}_1 \boldsymbol{A}_2 \cdots \boldsymbol{A}_L$. Physically, this means that we build up a system that has many layers or cascade connections of L mixing subsystems. The key point in our novel approach is that the learning (update) process to find parameters of sub-matrices $\boldsymbol{X}_l$ and $\boldsymbol{A}_l$ is performed sequentially, i.e. layer by layer[4]. In each step or each layer, we can use the same cost (loss) functions, and consequently, the same learning (minimization) rules, or completely different cost functions and/or corresponding update rules. This can be expressed by the following procedure:

(**Multilayer NMF Algorithm**)

Set: $\boldsymbol{X}_0 = \boldsymbol{Y}$,

For $l = 1, 2, \ldots\ L,$ **do** :

Initialize randomly $\boldsymbol{A}_l^{(0)}$ and/or $\boldsymbol{X}_l^{(0)}$,

For $k = 1, 2, \ldots, K$, **do** :

$$\boldsymbol{X}_l^{(k)} = \arg\min_{\mathbf{X}_l \geq 0} \left\{ D_l \left(\boldsymbol{X}_{l-1} || \boldsymbol{A}_l^{(k-1)} \boldsymbol{X}_l \right) \right\},$$

$$\boldsymbol{A}_l^{(k)} = \arg\min_{\mathbf{A}_l \geq 0} \left\{ \widetilde{D}_l \left(\boldsymbol{X}_{l-1} || \boldsymbol{A}_l \boldsymbol{X}_l^{(k)} \right) \right\},$$

$$\boldsymbol{A}_l^{(k)} \leftarrow \left[\frac{a_{ij}}{\sum_{i=1}^{m} a_{ij}} \right]_l^{(k)} ,$$

End

$$\boldsymbol{X}_l = \boldsymbol{X}_l^{(K)}, \quad \boldsymbol{A}_l = \boldsymbol{A}_l^{(K)},$$

End

7 Simulation Results

All the NMF algorithms discussed in this paper (see Table 1) have been extensively tested for many difficult benchmarks for signals and images with various statistical distributions. Simulations results confirmed that the developed algorithms are stable, efficient and provide consistent results for a wide set of parameters. Due to the limit of space we give here only one illustrative example: The five (partially statistically dependent) nonnegative source signals shown in Fig.1 (a) have been mixed by randomly generated uniformly distributed nonnegative

[4] The multilayer system for NMF and BSS is subject of our patent pending in RIKEN BSI, March 2006.

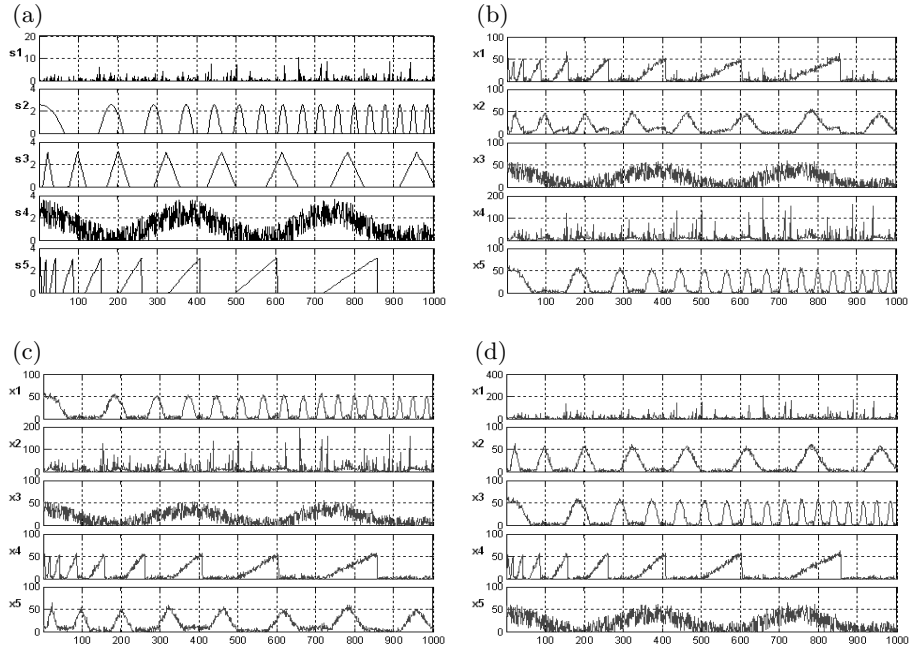


Fig. 1. Example 1: (a) The original 5 source signals; (b) Estimated sources using the standard Lee-Seung algorithm (8) and (9) with SIR = 8.8, 17.2, 8.7, 19.3, 12.4 [dB]; (c) Estimated sources using 20 layers applied to the standard Lee-Seung algorithm (8) and (9) with SIR = 9.3, 16.1, 9.9, 18.5, 15.8 [dB], respectively; (d) Estimated source signals using 20 layers and the new hybrid algorithm (15) with (49) with the Bose Shannon divergence with $\alpha = 2$; individual performance for estimated source signals: SIR = 15, 17.8, 16.5, 19, 17.5 [dB], respectively

matrix $\boldsymbol{A} \in \mathbb{R}^{50\times 5}$. To the mixing signals strong uniform distributed noise with SNR=10 dB has been added. Using the standard multiplicative NMF Lee-Sung algorithms we failed to estimate the original sources. The same algorithm with 20 layers of the multilayer system described above gives better results – see Fig.1 (c). However, even better performance for the multilayer system provides the hybrid SMART algorithm (49) with Bose-Einstein cost function (see Table 1) for estimation the matrix $\boldsymbol{X}$ and the Fixed Point algorithm (projected pseudo-inverse) (15) for estimation of the matrix $\boldsymbol{A}$ (see Fig.1 (d). We also tried to apply the ICA algorithms to solve the problem but due to partial dependence of the sources the performance was poor. The most important feature of our approach consists in applying multi-layer technique that reduces the risk of getting stuck in local minima, and hence, a considerable improvement in the performance of NMF algorithms, especially projected gradient algorithms.

8 Conclusions and Discussion

In this paper we considered a wide class of loss functions that allowed us to derive a family of robust and efficient novel NMF algorithms. The optimal choice of a loss function depends on the statistical distribution of the data and additive noise, so different criteria and algorithms (updating rules) should be applied for estimating the matrix $\boldsymbol{A}$ and the matrix $\boldsymbol{X}$ depending on *a priori* knowledge about the statistics of the data. We derived several multiplicative algorithms with improved performance for large scale problems. We found by extensive simulations that multilayer technique plays a key role in improving the performance of blind source separation when using the NMF approach.

References

[1] Lee, D.D., Seung, H.S.: Learning of the parts of objects by non-negative matrix factorization. Nature **401** (1999) 788–791.
[2] Cho, Y.C., Choi, S.: Nonnegative features of spectro-temporal sounds for classification. Pattern Recognition Letters **26** (2005) 1327–1336.
[3] Sajda, P., Du, S., Parra, L.: Recovery of constituent spectra using non-negative matrix factorization. In: Proceedings of SPIE – Volume 5207, Wavelets: Applications in Signal and Image Processing (2003) 321–331.
[4] Guillamet, D., Vitri'a, J., Schiele, B.: Introducing a weighted nonnegative matrix factorization for image classification. Pattern Recognition Letters **24** (2004) 2447 – 2454
[5] Li, H., Adali, T., W. Wang, D.E.: Non-negative matrix factorization with orthogonality constraints for chemical agent detection in Raman spectra. In: IEEE Workshop on Machine Learning for Signal Processing), Mystic USA (2005)
[6] Cichocki, A., Zdunek, R., Amari, S.: Csiszar's divergences for non-negative matrix factorization: Family of new algorithms. Springer LNCS **3889** (2006) 32–39
[7] Paatero, P., Tapper, U.: Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values. Environmetrics **5** (1994) 111–126
[8] Oja, E., Plumbley, M.: Blind separation of positive sources using nonnegative PCA. In: 4th International Symposium on Independent Component Analysis and Blind Signal Separation, Nara, Japan (2003)
[9] Hoyer, P.: Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research **5** (2004) 1457–1469.
[10] Kompass, R.: A generalized divergence measure for nonnegative matrix factorization, Neuroinfomatics Workshop, Torun, Poland (2005)
[11] Dhillon, I., Sra, S.: Generalized nonnegative matrix approximations with Bregman divergences. In: NIPS -Neural Information Proc. Systems, Vancouver Canada. (2005)
[12] Berry, M., Browne, M., Langville, A., Pauca, P., Plemmons, R.: Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics and Data Analysis (2006) http://www.wfu.edu/˜plemmons/papers.htm.
[13] Lee, D.D., Seung, H.S.: Algorithms for nonnegative matrix factorization. Volume 13. NIPS, MIT Press (2001)

[14] Novak, M., Mammone, R.: Use of nonnegative matrix factorization for language model adaptation in a lecture transcription task. In: Proceedings of the 2001 IEEE Conference on Acoustics, Speech and Signal Processing. Volume 1., Salt Lake City, UT (2001) 541–544
[15] Feng, T., Li, S.Z., Shum, H.Y., Zhang, H.: Local nonnegative matrix factorization as a visual representation. In: Proceedings of the 2nd International Conference on Development and Learning, Cambridge, MA (2002) 178–193
[16] Chen, Z., Cichocki, A., Rutkowski, T.: Constrained non-negative matrix factorization method for EEG analysis in early detection of Alzheimer's disease. In: IEEE International Conference on Acoustics, Speech, and Signal Processing,, ICASSP-2006, Toulouse, France (2006)
[17] Cichocki, A., Amari, S.: Adaptive Blind Signal And Image Processing (New revised and improved edition). John Wiley, New York (2003)
[18] Cichocki, A., Zdunek, R.: NMFLAB Toolboxes for Signal and Image Processing *www.bsp.brain.riken.go.jp*, JAPAN (2006)
[19] Merritt, M., Zhang, Y.: An interior-point gradient method for large-scale totally nonnegative least squares problems. Technical report, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, USA (2004)
[20] Minami, M., Eguchi, S.: Robust blind source separation by beta-divergence. Neural Computation **14** (2002) 1859–1886
[21] Jorgensen, B.: The Theory of Dispersion Models. Chapman and Hall (1997)
[22] Csiszár, I.: Information measures: A critical survey. In: Prague Conference on Information Theory, Academia Prague. Volume A. (1974) 73–86.
[23] Amari, S., Nagaoka, H.: Methods of Information Geometry. Oxford University Press, New York (2000)
[24] Zhang, J.: Divergence function, duality and convex analysis. Neural Computation **16** (2004) 159–195.
[25] Schraudolf, N.: Gradient-based manipulation of non-parametric entropy estimates. IEEE Trans. on Neural Networks **16** (2004) 159–195.
[26] Byrne, C.: Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative (RBI) methods. IEEE Transactions on Image Processing **7** (1998) 100 – 109.
[27] Byrne, C.: Choosing parameters in block-iterative or ordered subset reconstruction algorithms. IEEE Transactions on Image Progressing **14** (2005) 321–327
[28] Amari, S.: Differential-Geometrical Methods in Statistics. Springer Verlag (1985)
[29] Amari, S.: Information geometry of the EM and em algorithms for neural networks. Neural Networks **8** (1995) 1379–1408.
[30] Cressie, N.A., Read, T.: Goodness-of-Fit Statistics for Discrete Multivariate Data. Springer, New York (1988)

MAICS: Multilevel Artificial Immune Classification System

Michal Bereta[1] and Tadeusz Burczynski[1,2]

[1] Cracow University of Technology, Institute of Computer Modeling, Artificial Intelligence Department, Warszawska 24, 31-155 Cracow, Poland
beretam@torus.uck.pk.edu.pl,
tburczyn@pk.edu.pl
[2] Department for Strength of Materials and Computational Mechanics, Silesian University of Technology, Konarskiego 18a, 44-100 Gliwice, Poland
Tadeusz.Burczynski@polsl.pl

Abstract. This paper presents a novel approach to feature selection and multiple-class classification problems. The proposed method is based on metaphors derived from artificial immune systems, clonal and negative selection paradigms. A novel clonal selection algorithm – Immune K-Means, is proposed. The proposed system is able to perform feature selection and model identification tasks by evolving specialized subpopulations of T- and B-lymphocytes. Multilevel evolution and real-valued coding enable for further extending of the proposed model and interpreting the subpopulations of lymphocytes as sets of evolving fuzzy rules.

1 Introduction

Multiple-class discrimination is an important task in machine learning. However, it demands many steps, each of which is a difficult problem itself. The main steps are feature selection, model identification and classifier training. It is obvious that systems that combine all these steps in one learning procedure could bring significant benefits. In this paper a novel classification system which performs all these main steps in one multilevel evolution process is proposed. The proposed system utilizes the main paradigms of the Artificial Immune Systems (AIS) [1], the clonal and negative selection, together with the novel suppression mechanism. The model is an extension of the Two-Level AIS model, first proposed in [2]. To further extend the model, a novel clonal selection algorithm which is described in this paper was developed. The paper is organized as follows. In Section 2 a short description of the AIS and a brief overview of the previous work is given. In Section 3, the limitations of the Two-Level AIS are pointed out and possible improvements and extensions are proposed. In Section 4 the novel clonal selection algorithm – Immune K-Means – is described and utilized in building of the proposed model MAICS (Multilevel Artificial Immune Classification System) in Section 5. Some conclusions are drawn in Section 6.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 563–572, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Two-Level Artificial Immune System

Artificial immune systems [1] try to imitate the real immune systems. The main task of the immune system is to defend the organism against the pathogens. Different types of cells cooperate to give a reliable system able to efficiently adapt itself to changing environment. The AIS use only the main ideas of the real immune systems, the clonal and negative selection, which deal with the evolution of B- and T-cells, respectively.

2.1 General Concept of Artificial Immune Systems

B-cells with different receptors' shapes try to bind to antigens (training and testing data). The best fitted B-cells become stimulated and start to proliferate and produce clones, which are mutated at very high rates (somatic hyper-mutation). These steps are repeated and it is likely that there will emerge a better B-cell (better solution). The whole process is called the clonal selection. T-cells undergo a different type of evolution. They are created in thymus and learned to recognize none of the self-cells presented to them. If a T-cell recognizes any of the self-cells it is destroyed in the thymus. The mature T-lymphocytes do not react on self-cells and thus can protect the organism from auto-destruction. This evolution is know as a negative selection and it has been used by several researches for such problems as computer security, novelty or anomaly detection. These tasks can be considered as two-class discrimination problems. The clonal selection paradigm has been mainly used for data compression, data and web mining, clustering and optimization.

These two techniques, the clonal and negative selections, have been usually used separately. However, it should be kept in mind that the real power of the immune system is the cooperation of different types of cells. In particular, B- and T-cells cooperate in the way that B-cells wait for the signal from T-cells before they start to perform cloning. This simplified concept of cooperation between T- and B-cells is utilized in the model. The system AIRS proposed in [3] is an application of the clonal selection for multiple-class classification problems. The AIRS is based on a concept of ARB (artificial recognition ball) which is generalized concept of B-cell based on limitation of resources available for the evolving system. However, the AIRS model differs much from that proposed in this paper.

2.2 Evolutionary Methods for Classification Problems

Evolutionary methods (genetic algorithms, evolutionary strategies) have also been applied to classification tasks in different ways (evolving classification model, parameters' tuning, etc.). They have also been applied to the problem of evolving a population of fuzzy rules for classifications purposes [4]. The concept of applying evolutionary methods for creating and/or tuning of classification systems is not new, however, as it will be presented in the next sections, AIS, especially the one proposed in our work, can bring significant benefits, as the

individuals in AIS are detectors which undergo different kind of evolution, thus are able to explore the search space in a different manner, resulting in interesting possibilities for building complex in behavior, thus conceptually simple systems.

2.3 Two-Level AIS for Feature Selection and Classification

A system for feature selection and classification based on immune metaphors has been proposed in [2]. It uses both the clonal and negative selections. Its main characteristics are: a localization, a two-level evolution and a novel suppression mechanism. On the highest level of the system there is a population of binary strings in which "1's" indicate selected features. Each binary string describes a subpopulation of T-cells which are created by means of the negative selection based only on the features indicated by the binary string of that subpopulation. As it was shown in [2] this approach enables T-cells to focus on selected features (localization) and thus, to give better results. The two-level evolution, as depicted in Fig. 1, stands for the use of both the negative selection (for T-detectors creation and evaluation of selected features) and the clonal selection (for exploration of the feature space as the evolution of the population of binary stings describing subpopulations). The detectors from each subpopulation are able to detect different numbers of antigens as they use different subsets of features, some of which are potentially more useful. Those subpopulations, which recognize bigger number of antigens, are promoted during the evolution and can produce more clones. A novel suppression mechanism, based on the usefulness of a given subpopulation for the whole system, rather then on the similarity between binary strings, allowed not to remove useful subpopulation and to reach high percentage of recognized antigens while keeping the population size on constant, dynamically selected level. The subpopulation is removed if its loss does not result in decrease in the total number of recognized antigens by the whole system. This idea comes from the fact that slightly different subsets of features can have different discrimination abilities. The tests showed that this type of suppression does not result in an explosion in the total number of subpopulations [2]. The system is able to dynamically evolve a proper number of subpopulations and gives better results as when the traditional suppression is used.

3 Limitations of the Two-Level AIS and Proposed Improvements

The most important limitation of the Two-Level AIS is that it performs two-class discrimination only. Also the knowledge gained by the system is not so useful while trying to extract it as a set of rules, which are much easier to understand for humans. There are two main reasons for that. The first one is that the created detectors were artificial T-cells and as such they are able to answer to the question "when the feature vector does not belong to the self-class" rather then to the question "when the feature vector does belong to the self class".

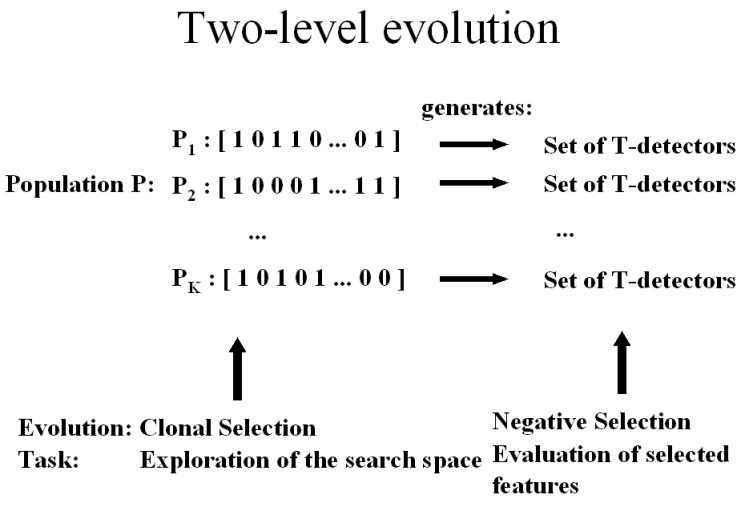


Fig. 1. The main concept of the Two-Level AIS [2]. Each P_i describes features used by a subpopulation of T-cells.

The reason is that the T-cells are trained to recognize all but not the samples presented to them during evolution. The second reason is that the T-cells are binary coded. This form is considered [5] as being too low-level coded and thus losing its ability for being easily understood by humans.

The answer for the aforementioned limitations is using the real-coded detectors and creating subpopulations of T-cells for each class treated as a self-class. Also, specialized subpopulations of real-valued B-cells should be created in the same fashion as the subpopulations of T-cells but using the clonal selection method. Thus, the proposed method incorporates a new level of evolution by means of the clonal selection. But this clonal selection is of different type that the clonal selection of the population of binary strings describing subpopulations. The subpopulations are still described by binary strings as they are easily interpreted as indicators of selected features. However, the B-cells in subpopulations are real-coded and require different type of the clonal selection, adequate for real-coded lymphocytes. An efficient algorithm of the clonal selection is proposed in the next section.

4 Immune K-Means – A Novel Clonal Selection Algorithm

A new clonal selection algorithm is proposed. It is a simple but robust clustering algorithm. It is suited especially for use in the proposed system MAICS as the clonal selection used for creating subpopulations of real-valued B-cells is repeated several times during the evolution of MAICS. The proposed algorithm resembles in some parts the well known k-means clustering algorithm, which has some limitations thought, like the need of determining the number of clusters in advance,

however, it is fast and easy to implement. It is especially useful when the training data set is big and using more sophisticated clustering methods is computationally prohibited. On the other hand clonal selection algorithms are able to evolve a proper number of clusters. Combining these two methods results in a clustering method that has all the positives and is free of the limitations. The biggest advantage of the new algorithm is that each B-cell creates only one clone in each iteration. The most important concept in this novel algorithm is a proper suppression mechanism which is able to decide when to remove useless B-cells. The concept of the suppression based on the usefulness of the given cell rather then based on the similarity among the cells was adopted. This approach to performing suppression emerged during developing of the Two-Level AIS [2]. In that system, starting from the worst subpopulation, the whole subpopulation was temporarily removed and the system was evaluated whether there is any loss in the number of recognized antigens by the rest of subpopulations. If there was no loss, the subpopulation was removed permanently. The main concept of removing only the cells that are useless for the whole system can be easily adopted here.

The current implementation of the proposed method assumes that both antigens and B-cells are real valued vectors. The algorithm goes as follows:

1. Generate an initial population of B-cells as a set of random real valued vectors.
2. For each antigen a_i (a sample in the training set) find its nearest B-cell.
3. For each B-cell create its clone as a mean vector of all antigens for which a given B-cell is the nearest neighbor. If a clone is the same as the parent, mutate the clone by adding to each dimension a random value from range [*-mut, mut*]. It allows the clones to escape from the places where there are no clusters. Add the clones to the population of B-cells.
4. For each antigen find its new nearest B-cell.
5. Count the stimulation level of each B-cell. The stimulation level for the j-th B-cell is counted according to (1):

$$stimulation_level(B_j) = \sum_{a_i : a_i \in NN(B_j)} \exp^{-beta * euc_dist(B_j, a_i)} \tag{1}$$

   where $NN(B_j)$ is a subset of antigens for which B_j is the nearest neighbor, *beta* is a positive constant and *euc_dist* stands for the Euclidean distance.
6. Sort B-cells in a descending order according to their stimulation level.
7. Perform the suppression.
8. Repeat from step 3 until the termination condition is satisfied (in the current implementation it is a given number of iterations).

The first proposed suppression goes as follows:

Suppression I. Starting from the less stimulated B_j, for each $a_i \in NN(B_j)$ find B_{new} as a_i's new NN (nearest neighbor) among B-cells (except for the B_j) and calculate the distance to B_{new}. If a condition (2)

$$(new_dist - old_dist) \leq alpha \tag{2}$$

is satisfied for each $a_i \in NN(B_j)$, remove B_j and find new NN for all $a_i \in NN(B_j)$. The values *old_dist* and *new_dist* are the Euclidean distances between the given antigen a_i and B-cells B_j and B_{new}, respectively.

The training samples are normalized to the range [0, 1]. The parameters of the algorithm are easily tuned. As tests showed, the parameter *beta* does not influence the evolution much and can be set to 1 and thus removed from tuning. The parameter *mut* should decrease over the iteration to 0, with a small starting value (like 0.05). The most important parameter is *alpha*. It indicates the maximum allowed change in distance to the NN of each antigen while trying to remove a given B-cell. The initial value of this parameter should be small (like 0.001), and then, which can be confusing at first, should grow over the iterations. The final value depends on the training set and it can be considered as a density measure of the B-cells' population. The bigger the final value, the less B-cells there will be in the final population.

Fig.2 depicts a result of applying the Immune K-means algorithm to two-dimensional data. The algorithm is able to learn the structure of data easily.

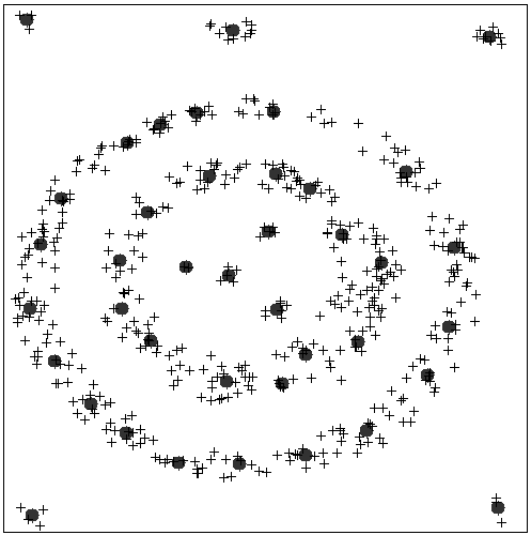


Fig. 2. Immune K-Means algorithm is easily able to evolve a population of B-cells (*black dots*) that properly represent the structure of data (*crosses*). The starting value of the parameter *alpha* was 0.001, the final value was 0.07. The iterations number was set to 25. There was only one B-cell in the initial population.

The proposed algorithm Immune K-Means is an unsupervised learnign algorithm, as it does not use class information during training. In order to use the B-cells from the final population as classifiers it is necessery to label each of the final B-cells as representing one of the classes. The simplest way of doing this

is to label each B-cell depending on the number of training samples (antigens) it binds (for which it is the nearest neighbor). The B-cell gets the label of the class, from which it binds the biggest number of antigens.

However, if the lymphocytes are to be good classifiers, using class information during learning seems to be a good idea. For that reason another type of suppression is proposed. While performing this type of suppression during learning it is necessary to count the class labels for each lymphocyte everytime when the nearest neighbors are found for the antigens. The difference is also while creating offspring of each B-cell: each cell creates one clone for each class as the mean of antigens (from that class) for which the given B-cell is the nearest neighbor. The suppression goes as follows:

Suppression II. For each lymphocyte B_j , for each $a_i \in NN(B_j)$ find its new NN, B_{new}, among B-cells (except for the B_j) and check whether the condition (3)

$$C(B_j) = C(a_i) \quad AND \quad C(B_{new}) \neq C(a_i) \tag{3}$$

is satisfied. If the condition (3) is satisfied for at least one $a_i \in B_j$, B_j cannot be removed from the population, otherwise, permanently remove B_j and find a NN for each $a_i \in B_j$. $C(B_j)$ and $C(a_i)$ are the class labels of the B-cell and the antigen, respectively. This mechanism does not allow removing B-cells in the situation when at least one training sample was classified correctly by a given B-cell and it would be misclassified by another B-cell while trying to remove the first one. A given B-cell B_j is removed permanently only when its removal does not cause the growth of the total number of misclassified training samples.

The suppression II utilizes the class information during the learning and thus makes the Immune K-Means algorithm a supervised learning algorithm. Both variations of the proposed clonal selection algorithm share most of the steps. The most important difference is in the suppression step. Additionally, one should easily observe that the parameter *alpha* does not play any role while using suppression II and thus it can be eliminated from tuning. Also, as tests revealed, sorting of B-cells according to their bounding degree seems to have no influence on the behavior of the algorithm and its final results when using suppression II. Sorting is a necessary step while using suppression I, as it leads to an attempt of removing the B-cells from the edges of data clusters and thus enables the evolution of cells representing the inner distribution of data. On the contrary, suppression II is expected to focus the learning on the class discrimination and place the B-cells near the class boundaries. Of course, it is possible to use both types of suppression in the same time, which would result in developing B-cells of two types: those describing the inner distribution of data and those somewhat describing the class boundaries.

An example result of applying the proposed Immune K-Means algorithm with the suppression II to artificial two-dimensional data is presented in Fig. 3.

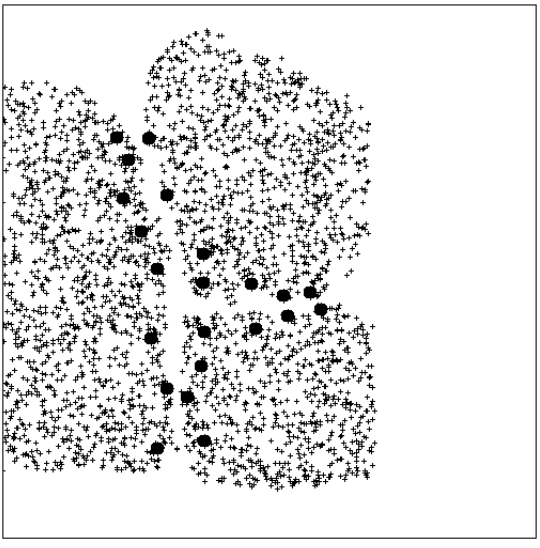


Fig. 3. Immune K-Means algorithm with suppression II. B-cells (*blackdots*) near the boundaries of three distinct clusters (three classes) are the result of different meaning of *usefulness* in suppression II. The starting value of the parameter *alpha* was 0.004, the final value was 0.13. The iterations number was set to 20. There were 5 B-cells in the initial population.

5 Classification with MAICS

The proposed clonal selection method is incorporated in the proposed system extending its capabilities. Subpopulations of B-cells are allowed to evolve in a similar fashion as the subpopulations of T-cells. Fig. 4 depicts the idea. The subpopulations of B-cells evolve by means of the Immune K-Means by using samples from all classes. After the Immune K-Means algorithm is finished, each B-cell in the given subpopulation is labeled as recognizing antigens from a given class, by a simple voting scheme – the subpopulation is treated as the kNN classifier and the number of recognized antigens from each class for each B-cell is checked. Each B-cell can then be assigned a certainty degree as the ratio of the number of recognized antigens from the class it is assigned to, to the total number of recognized antigens from other classes. The average certainty degree of the B-cells from a given subpopulation ebables to rate the subpopulations and to reward the best of them during the evolution by allowing them to produce bigger number of clones. Also, as it was mentioned earlier, subpopulations of T-detectors are created for each class treated as the self-class.

Having evolved subpopulations of both types of lymphocytes MAICS is able to perform multiple-class discrimination. Having an unknown sample the system uses B-cells as k-nearest-neighbor classifiers. Different types of voting schemes

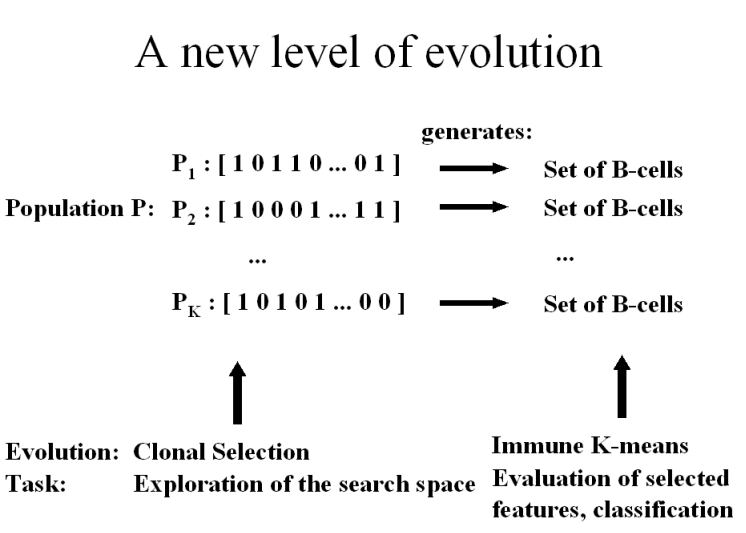


Fig. 4. A new level of evolution is added. The proposed Immune K-Means algorithm is incorporated to create subpopulations of B-cells based on the selected features.

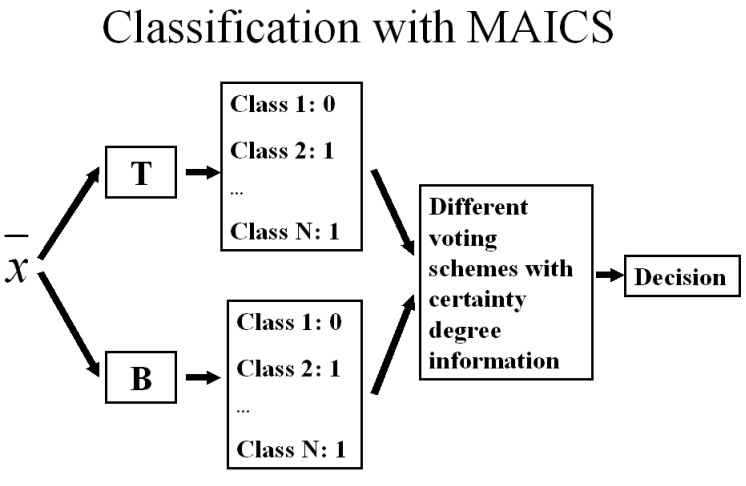


Fig. 5. Classification scheme of MAICS. Given an unknown sample, MAICS uses information from T- and B-cells populations. T-cells subpopulations give binary information of whether a given sample belongs or not to a given class. Available certainty degree of each B-cell allows to develop different schemes of classifications.

can be adopted and tested. Subpopulations of T-detectors cooperate with subpopulations of B-cells and provide additional classification criteria, leading to possible decrease in classification error. As the detectors are real-coded it is possible to interpret them as subpopulations of evolving fuzzy rules. Thus, the proposed system can be used for developing fuzzy inference systems for classification,

which is done in the same time as the feature selection process. The concept of classification process in MAICS is depicted in Fig. 5.

6 Conclusion

The proposed system MAICS tries to combine different type of metaphors derived from artificial immune methods in one multilevel evolutionary system able to perform the model identification, feature selection and detectors creation in one powerful, but conceptually simple, process. The novel clonal selection algorithm and the suppression mechanisms are proposed. The proposed system will be further tested and compared to other methods for classification problems.

Acknowledgement

The research is financed from The Foundation For Polish Science.

References

1. de Castro, L., Timmis, J.: Artificial Immune Systems: A New Computational Approach. Springer-Verlag, London. UK. (2002)
2. Bereta, M., Burczyński, T.: Hybrid immune algorithm for feature selection and classification of ECG signals. In Burczyński, T., Cholewa, W., Moczulski, W., eds.: Recent Developments in Artificial Intelligence Methods. AI-METH Series, Gliwice (2005) 25–28
3. Goodman, D., Boggess, L., Watkins, A.: Artificial immune system classification of multiple-class problems. In: Artificial Neural Networks in Engineering (ANNIE-2002). (2002)
4. Hoffmann, F.: Combining boosting and evolutionary algorithms for learning of fuzzy classification rules. Fuzzy Sets and Systems **141**(1) (2004) 47–58
5. Ji, Z., Dasgupta., D.: Real -valued negative selection algorithm with variable-sized. In et al., D.K., ed.: International Conference on Genetic and Evolutionary Computation (GECCO-2004), Seattle, Washington USA, Springer-Verlag (2004) 287–298

Selection of Prototype Rules: Context Searching Via Clustering

Marcin Blachnik[1], Włodzisław Duch[2,3], and Tadeusz Wieczorek[1]

[1] Division of Computer Methods, Department of Electrotechnology and Metallurgy, The Silesian University of Technology, Krasińskiego 8, 40-019 Katowice, Poland
{marcin.blachnik, tadeusz.wieczorek}@polsl.pl
[2] Department of Informatics, Nicolaus Copernicus University, Grudziądzka 5, Toruń, Poland
[3] School of Computer Engineering, Nanyang Technological University, Singapore
WWW home page: Google "Duch"

Abstract. Prototype-based rules are an interesting alternative to fuzzy and crisp logical rules, in many cases providing simpler, more accurate and more comprehensible description of the data. Such rules may be directly converted to fuzzy rules. A new algorithm for generation of prototype-based rules is introduced and a comparison with results obtained by neurofuzzy systems on a number of datasets provided.

1 Introduction

Similarity-based approaches that developed from the k-Nearest Neighbors (kNN) algorithm [1,2] are still one of the most useful and popular algorithms in pattern recognition. They are also at the foundation of the Artificial Intelligence learning methods, such as the Case Based Reasoning or Memory Based Reasoning methods, allowing for comparison of complex objects that cannot be easily represented in a feature space with fixed number of dimensions.

Nearest neighbor algorithms, or more generally Similarity-Based Learning (SBL) framework [2,3], may provide not only predictive models, but also prototype-based logical rules (P-rules) [4,5] for data understanding. Knowledge hidden in many datasets can be captured in a small set of prototypes using appropriate similarity measures. As shown in [6] this type of rules are equivalent to fuzzy rules (F-rules). P-rules seem to be more general because they support all types of features (discrete, nominal or symbolic), while the use of F-rules is restricted to continuous or ordinal attributes. Moreover, non-additive distance functions may include explicit models of correlation between variables and cannot be converted into fuzzy rules. Algorithms for selection of good prototypes and similarity measures are in a direct competition to the popular neurofuzzy approaches and fuzzy modeling techniques [7], and therefore their further development is an important issue.

Selection of reference vectors, also called prototypes, is very important especially for large datasets, because storing the whole training set in memory is then

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 573–582, 2006.
© Springer-Verlag Berlin Heidelberg 2006

prohibitively expensive and searching for nearest neighbors becomes computationally very expensive, making similarity-based algorithms completely useless for real time applications. The need to overcome these limitations brought the development of many approaches aimed at increase of the speed and reduction of the memory requirement of the basic nearest neighbor algorithms. As a result many powerful algorithms for prototype selection were created [8,9,10,11] that may also find an application in creation of P-rules.

These algorithms may be divided into four groups: noise filters, data condensing methods, clustering, and prototype optimization methods. Most of these approaches try to reduce the number of prototypes by getting rid of irrelevant training vectors [10]. Some algorithms try also to identify and remove outliers that decrease accuracy. More advanced algorithms like WITS, DROP [1,2,3,4,5,11], remove also vectors which lie close to the decision border and are not necessary for classification, increasing generalization abilities. Another group of methods starts with a small number of prototypes optimizing their position using such techniques as learning vector quantization (LVQ). This is a powerful method, but the results strongly depend on the starting position of prototypes. Results presented in [8,9] lead to the conclusion that LVQ approach should be used as second step following other prototype selection methods.

The last group of methods is based on clustering algorithms that search for interesting groupings of vectors in a dataset [1]. Usually clustering is done separately for each class, so that each class has an independent number of clusters. In this approach spherical clusters are preferred like obtained from Hard C-means (HCM), Fuzzy C-Means (FCM) algorithm or Gaussian Mixture Model (GMM). Unfortunately these clustering methods search for clusters among all vectors belonging to one of the classes, producing a subset of irrelevant prototypes that are often far away from the decision boundary and that do not participate in the decision process.

This problem is especially important for P-rules where a small subset of reference vectors that can be interpreted as prototypes is searched for [2]. These observations allow for construction of a new prototype selection algorithm based on the context dependent clustering approach, for example the Conditional FCM (CFCM) with weighting of the training vectors [12]. In the next section heterogeneous distance functions are described, in the third section CFCM algorithm is presented, in the fourth section creation of an appropriate condition values to determine clustering context is described, the algorithm for the selection of optimal number of prototypes is presented in section five, followed by empirical experiments in section six, and the final section contains conclusions.

2 Heterogeneous Distance Functions

Real word classification problems usually involve different types of features, some may be continuous or linearly ordered, some may be discrete and some symbolic. This constitutes a real problem for many algorithms, including statistical, neural or neurofuzzy approaches (ANFIS). Similarity-Based Learning framework

[2,3] defines many variants of similarity evaluations, including different distance functions for different types of attributes. The most popular and frequently used Minkovsky distance function is a generalization of Euclidian distance and is applicable only for numerical attributes. Symbolic features require different distance functions that may be defined using conditional probabilities to evaluate similarity of each symbolic feature value from the point of view of class distributions, as it is done in the Value Distance Metric (VDM) [15]. Numerical and probabilistic distance measures may be combined in a general distance metric, usually called the Heterogeneous Distance Metric (HDM), and in particular case of the VDM metric the Heterogeneous Value Distance Metric (HVDM). Such distance functions were used by Wilson and Martinez [16] who simply added scaled contributions from different features:

$$D\left(\mathbf{x},\mathbf{r}\right)^{\alpha}=\sum_{i=1}^{n}d\left(x_i,r_i\right)^{\alpha} \tag{1}$$

where $D\left(\mathbf{x},\mathbf{r}\right)$ is total distance between two vectors $\mathbf{x}$ and $\mathbf{r}$ with n features, $d(x_i,r_i)$ are distances calculated for single features, and α is an exponent (in more general case exponents on the left and right side of Eq. 1 may be different [3]). Explicit account of correlations between features is done in the simplest way by introducing covariance matrices. HDM assumes that different types of features are independent, and for the real-valued or ordered discrete features Minkovsky's distances are used, while for symbolic features probabilistic distances based on conditional probabilities are used.

$$D_{Mink}\left(\mathbf{x},\mathbf{r}\right)^{\alpha}=\sum_{i=1}^{n_1}\left|x_i-r_i\right|^{\alpha} \tag{2}$$

$$D_{VMD}\left(\mathbf{x},\mathbf{r}\right)^{\alpha}=\sum_{i=1}^{n_2}\sum_{j=1}^{C}\left|p\left(c_j|x_i\right)-p\left(c_j|r_i\right)\right|^{\alpha} \tag{3}$$

where n_1 is the number of numerical and n_2 of symbolic values, C is the number of classes and posterior probabilities $p\left(c_j|x_i\right)$ and $p\left(c_j|r_i\right)$ are calculated as:

$$p\left(c_j|x_i\right)=Nx_{ij}/Nx_i \tag{4}$$

where Nx_i is number of instances in the training set that have value x for feature i, and Nx_{ij} is the number of training vectors from class j that have value x for feature i.

Because HVDM is additive it can be written as the sum of two distance functions that depend on different attribute types:

$$D\left(\mathbf{x},\mathbf{r}\right)^{\alpha}=D_{Mink}\left(\mathbf{x_a},\mathbf{r_a}\right)^{\alpha}+D_{VDM}\left(\mathbf{x_b},\mathbf{r_b}\right)^{\alpha} \tag{5}$$

where $\mathbf{x}_a$ and $\mathbf{r}_a$ are subsets of numerical attributes and $\mathbf{x}_b$ and $\mathbf{r}_b$ are subsets of their symbolic features. An important problem with such heterogeneous

functions is to define a common distance scale. D_{Mink} takes values in the range $(0, +\infty)$ while D_{VDM} in the range $\left(0, n_2^{1/\alpha}\right)$. Features should be properly normalized to assure that each distance component has the some or comparable contribution. Wilson and Martinez [16] proposed one type of normalization for numerical attributes (6) and three different normalization methods for VDM distance (7), where the choice depends on particular problem:

$$d_{Mink}(x, r) = \frac{|x - r|}{4\sigma} \tag{6}$$

$$\begin{aligned} N1 &: d_{VMD}(x, r) = \sum_{j=1}^{C} \left| \frac{Nx_j}{Nx} - \frac{Nr_j}{Nr} \right| \\ N2 &: d_{VMD}(x, r) = \sqrt{\sum_{j=1}^{C} \left| \frac{Nx_j}{Nx} - \frac{Nr_j}{Nr} \right|^2} \\ N3 &: d_{VMD}(x, r) = \sqrt{C \cdot \sum_{j=1}^{C} \left| \frac{Nx_j}{Nx} - \frac{Nr_j}{Nr} \right|^2} \end{aligned} \tag{7}$$

In $N1$ the distance between one dimensional probabilities is calculated using the Manhattan norm, while in $N2$ and $N3$ the Euclidean norm is used.

In [16] authors suggest that normalization $N2$ "favors having all of the class correlations fairly similar rather then having some very close and some very different". The difference between $N2$ and $N3$ normalizations may have an influence on classification using kNN method, but for P-rules, with a small number of reference vectors common feature weighting may be replaced by optimization of their positions. The parameter α that occurs in all distance functions has significant influence on the shape of decision borders [3]. There is no specific value of this parameter that will always lead to the best classification results, therefore the optimal α value should be determined using meta-learning algorithms [17]. Another possibility is to set the value of α depending on particular needs. This is one of the P-rules advantages, because the results are equivalent to fuzzy rules with various types of membership functions, for example Gaussian for α=2 or crisp logical rules for α=∞.

3 Conditional Fuzzy C-Means

In classification tasks the area between samples from different classes is most important [1], and the goal of pattern recognition algorithms is to construct the optimal shape of this border to assure maximum generalization, that is the lowest error rate for the test data. This observation encourages searching for prototypes similar to "support vectors" [18] in the border area. Classical clustering algorithms search for groups of data in the whole data space without any knowledge of class distribution. Of course one may cluster each class independently, however the centers of the clusters will lead to prototypes far from the decision borders. Another disadvantage of classical clustering algorithms follows

from the fact that useless prototypes are created for data clusters far from decision borders, as show in Fig. 1. Optimization methods from the LVQ family cannot move such prototypes to areas where they could take an important part in the decision process.

One possible solution of this problem is based on conditional or context dependent clustering algorithms. One of the examples of such methods is the Conditional Fuzzy C-Means method (CFCM) [12]. It is an extension of the FCM clustering algorithm where data are grouped under some external conditions defined for every training vector. These conditions are specified by an external variable $y_k$which corresponds to each training vector x_k. This variable y_k has an associated membership function $\mu(y)$ or in other words weight that creates clustering condition f_k=$\mu_A(y)$ $\in$[0,1] defined for every vector x_k. This condition allows for clustering related data, where f_k defines strength of the relation between data vectors x_k and the external variable y_k.

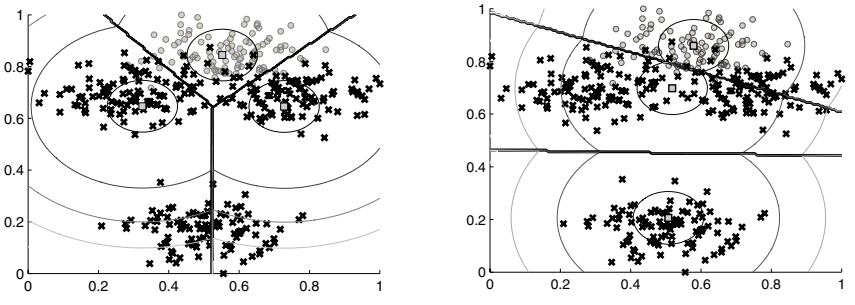


Fig. 1. Decision border and prototype activation areas for prototypes selected by a) Conditional Fuzzy C-means and b) Fuzzy C-means algorithm

In our approach this external variable is defined as a measure of distance between each data vector and a possible decision border. The task is to find clusters near to this border. FCM and CFCM are based on minimization of a cost function defined as:

$$J_m(\mathbf{U},\mathbf{V}) = \sum_{i=1}^{C}\sum_{k=1}^{N}(u_{ik})^m \left\|x_k - v_i\right\|_A^2 \tag{8}$$

where C is the number of clusters centered at v_i, N is the number of vectors, m >1 is a parameter, and $\mathbf{U}$=(u_{ik}) is a $C \times N$ dimensional membership matrix with elements $u_{ik} \in$ [0,1] defining the degree of membership of the k-th vector in the i-th class. The matrix $\mathbf{U}$ has to fulfill three conditions:

1^o each vector x_k belongs to the i-th cluster to a degree between 0 to 1:

$$\bigvee_{1\le i\le C}\ \bigvee_{1\le k\le N} u_{ik} \in [0,1] \tag{9}$$

2^o sum of the membership values of k-th vector x_k in all clusters is equal to f_k

$$\bigvee_{1\le k\le N} \sum_{i=1}^{C} u_{ik} = f_k \tag{10}$$

3^o no clusters are empty.

$$\bigvee_{1\le i\le C} 0 < \sum_{k=1}^{N} u_{ik} < N \tag{11}$$

Cost function (8) is minimized under these conditions by [12]:

$$\bigvee_{1\le i\le C} v_i = \sum_{k=1}^{N} (u_{ik})^m x_k \Bigg/ \sum_{k=1}^{N} (u_{ik})^m \tag{12}$$

$$\bigvee_{\substack{1\le i\le C\\ 1\le k\le N}} u_{ik} = f_k \Bigg/ \sum_{j=1}^{C} \left(\frac{\|x_k - \nu_i\|}{\|x_k - \nu_j\|} \right)^{2/(m-1)}. \tag{13}$$

4 Determining the Context

Searching for optimal position of prototypes for P-rules is a difficult problem; moreover, a balance between the number of prototypes (simplicity of the rules) and the accuracy of the whole system is very important. Irrelevant prototypes, that is prototypes that do not have any influence on classification, should be removed. This assumption allows for search of prototypes that lie close to one of the opposite classes, that is close to the possible decision border.

To determine position of prototypes using conditional clustering algorithm a coefficient w_k is defined, evaluating for a given vector x_k the ratio of a scatter between this vector and all vectors from the same class, divided by a scatter for all vectors from the remaining classes:

$$w_k = \sum_{j,\omega(x_j)=\omega(x_k)} \|x_k - x_j\|^2 \Bigg/ \sum_{l,\omega(x_l)\neq\omega(x_k)} \|x_k - x_l\|^2 \tag{14}$$

Here $\omega(x_k)$ denotes class label of the vector x_k. This coefficients is then normalized to be in the range [0,1]:

$$w_k = \Big(w_k - \min_k (w_k) \Big) \Big/ \Big(\max_k (w_k) - \min_k (w_k) \Big) \tag{15}$$

Normalized w_k coefficients reach values close to 0 for vectors inside large homogenous clusters, and close to 1 if the vector x_k is near the vectors of the

opposite classes and far from other vectors from the same class (for example if it is an outlier). To avoid effects related to multiple density peaks in non-homogenous data distributions a cut-off point for distance of the order of standard deviation of the whole data could be introduced, although in our tests it was not necessary. These normalized weights determine the external variable which then is filtered to assign appropriate context or condition for CFCM clustering. Filtering can be also understood as assigning appropriate linguistic term in the fuzzy sense. In the algorithm used below a Gaussian filter was used:

$$f_k = \exp\left(\frac{w_k - \mu}{\sigma^2}\right) \tag{16}$$

with the best parameters in the range of μ=0.6-0.8 and σ=0.6-0.9, determined empirically for a wide range of datasets. The μ parameter controls where the prototypes will be placed; for small μ they are closer to the center of the cluster and for larger μ closer to the decision borders. The range in which they are sought is determined by the σ parameter.

5 Prototype Racing Algorithm for Optimizing Number of Prototypes

Most algorithms for clustering require specification of the number of clusters. This is especially important for the C-means clustering group of algorithms, where premises for determining C do not exist, while in the hierarchical clustering they can be obtained by analyzing dendrogram tree [19]. In our approach clusterization of data from each class is done in an independent way, therefore determination of the number of prototypes requires control of several C_i parameters, one for each class. Moreover, these parameters may be strongly correlated. The simplest way to build a model with optimal number of reference vectors is to check its performance for all possible combinations of C_i with optimization of final positions of all prototypes but this will lead to a very high computational costs.

One possible solution is provided by the racing algorithm [13] based on the results of candidate model comparison in cross-validation task. To speed up the learning process Hoeffding or Bayesian estimation of error bounds of analyzed models is used, rejecting those that are significantly worse than the current best model during the optimization procedure. Another possible improvement leading to reduction of computational costs, especially in optimizing balanced accuracy (BER), is based on the heuristics for increasing the number of prototypes only for the class that has worst accuracy. Adding new prototype to the class with lowest accuracy increases the accuracy for that class but may also reduce overall balanced accuracy. To avoid such problems the racing algorithm keeps only the best models, growing the number of prototypes per class as long as the gain in balanced accuracy justifies the added complexity. This approach significantly increases optimization speed but unfortunately not always finds optimal combination of prototypes per class.

6 Experiments and Results

In numerical experiments presented below prototype selection scheme described above was compared with results obtained from classical clustering algorithm without external conditions. All results are from the 10-fold crossvalidation tests with the racing algorithm for determining the number of prototypes. Maximum number of iterations was limited to 15 for both FCM and CFCM clustering. Positions of prototypes obtained from clustering were then optimized with the LVQ1 algorithm [19]. Test results summarized in Tab. 1 were obtained for six benchmark datasets from UCI repository [7], with different number of classes and types of features: *Cleveland Heart Disease, Glass, Ionosphere, Pima Indians Diabetes,* and the *Iris* dataset with all 4 and with two most important features. A small medical dataset *Appendicitis* was also used, with two most relevant features selected using SSV decision tree (description of these datasets may be found in [4]).

Crisp rules generated by decision tree using Gini index were used to generate C-rules as a reference for accuracy and complexity comparison. Features selected by the tree were used with the CFCM approach. Results are also compared with the state-of-the-art NefClass [7] neurofuzzy system that generates F-rules. It is based on greedy partitioning for initialization of fuzzy sets. In NefClass the number of fuzzy sets and the maximum number of rules are selected manually. Several available shapes of fuzzy membership functions have been used with different number of fuzzy sets and rules, and the best balanced accuracy results are reported in Tab.1. Crossvalidation results are used to select the model and to estimate expected accuracy, and this model is then trained on all data to obtain the number of rules and premises.

Appendicitis dataset is quite small (21+85 samples) and thus standard deviation for balanced accuracy is very large, therefore the differences are not significant. All solutions are quite simple, including the C-rules from the tree: If F7<7520.5 and F4<12 then Class 2, else Class 1. For the remaining five datasets best balanced accuracy was achieved using the CFCM algorithm, although for such small datasets standard deviations are quite large and thus even 8% differences (Ionosphere) between P-rules and F-rules are not statistically significant. P-rules generated in an automatic way using the CFCM approach have comparable complexity to the manually optimized F-rules. P-rules may be directly converted to the F-rules [6] and therefore algorithms for their generation offer a competitive approach to the neurofuzzy algorithms.

7 Conclusions

In many data mining applications understanding and transparency of the results are of great importance. Rule-based systems should be flexible to represent data accurately, and should be easy to understand, implying a small number of accurate rules. Although both P-rules and F-rules may fulfill these conditions for some reason P-rules are not so popular, mistakenly believed to be difficult

Table 1. Classification results for 6 datasets: balanced accuracy, standard deviation, the number of premises and rules

Dataset	Decision Tree			CFCM			FCM			NefClass		
	bacc	std	Prem/ C-rule	bacc	std	Prem/ P-rule	bacc	std	P-rule	bacc	std	Prem/ F-rule
Appendicitis	76.0	19.0	2/1	81.1	13.0	2/3	79.7	19.7	2	81.9	12.3	1/4
Heart	78.9	7.7	3/5	76.3 78.9	3.4 5.9	13/4 3/2	75.4	6.8	5	76.6	6.6	1/2
Glass	62.4	9.1	6/8	71.2 70.6	11,6 11.1	9/16 6/14	68.6	10.0	16	64.8	9.4	3/13
Ionosphere	88.7	4.8	2/2	90,1 85.9	5.3 6.8	33/7 2/6	84.8	6.5	6	81.8	9.8	2/4
Diabetes	69.9	4.4	2/2	74.9 73.9	3.1 5.9	8/7 2/3	74.1	4.7	7	71.1	4.9	1/2
Iris	93.3	7.0	2/2	96.0 97.3	4.7 4.7	4/5 2/5	95.3	5.5	5	93.3	4.4	2/3

to understand. On the other hand complex sets of fuzzy rules that no-one even tries to understand are hailed as comprehensible just because they are logical rules. Comparison with neurofuzzy systems shows that P-rule algorithms are frequently capable of generating simpler and more accurate description of data in terms of prototypes. The balance between transparency (rule simplicity and their number) and accuracy should be determined in each task individually, depending on the application.

In this paper a new approach based on the conditional fuzzy C-means approach with determination of the distance to the decision border has been introduced and used for selecting good prototypes, optimizing their position and number. This algorithm has several advantages: it limits the search area to the most probable space, facilitates searching for good prototypes, reduces influence of outliers, reduces the number of irrelevant prototypes, creating only prototypes that have important meaning for classification, and it automatically determines optimal number of prototypes for each class.

Although this approach introduces two new parameters that should be tuned experiments showed that it is not sensitive to their values and leaving these values in the μ=0.6-0.8 and σ=0.6-0.9 range is sufficient. The results presented in Tab.1 show that the proposed coefficient for context clustering may significantly improve the prototype selection based on clustering. For some datasets like Glass or Ionosphere the classification quality increased by almost 20% comparing to the normal clustering, while for the other datasets the increase was more modest.

So far the context dependent clustering algorithm was used only with the CFCM clustering, but applications to other algorithms look also very promising and will be investigated in our future work. Combination of this approach with feature selection based on Relief index should lead to a P-rule system that should easily compete with the best neurofuzzy systems.

Acknowledgement. WD is grateful for the support by the Polish Committee for Scientific Research, research grant 2005-2007.

References

1. Duda R.O., Hart P.E., Stork D.G.: Pattern Classification, New York: John Wiley & Sons, 2nd ed, 2001.
2. Duch W. : Similarity based methods: a general framework for classification, approximation and association. *Control and Cybernetics,* vol. 29(4), pp. 937-968, 2000.
3. Duch W., Adamczak R., Diercksen G.H.F.: Classification, Association and Pattern Completion using Neural Similarity Based Methods. Applied Mathematics and Computer Science vol. 10(4), pp. 101-120, 2000.
4. Duch W., Setiono R., Zurada J.M.: Computational intelligence methods for understanding of data. Proc. of the IEEE, vol. 92(5), pp. 771- 805, 2004.
5. Duch W., Grudziński K.: Prototype based rules - a new way to understand the data. Proc. of IJCNN 2001, Washington D.C., USA, pp. 1858-1863.
6. Duch W., Blachnik M.: Fuzzy rule-based system derived from similarity to prototypes, Lecture Notes in Computer Science, vol. 3316, pp. 912-917, 2004.
7. Nauck D., Klawonn F., Kruse R.: Foundations on Neuro-Fuzzy Systems. Wiley, Chichester, 1997.
8. Jankowski N., Grochowski M.: Comparison of Instance Selection Algorithms I. Algorithms Survey, Lecture Notes in Artificial Intelligence, vol. 3070, pp. 598-603, 2004.
9. Grochowski M., Jankowski N.: Comparison of Instance Selection Algorithms II. Results and Comments, Lecture Notes in Artificial Intelligence, vol. 3070, pp. 580-585, 2004.
10. Bezdek J.C., Kuncheva L.I.: Nearest prototype classifier designs: An experimental study, International Journal of Intelligent Systems, vol. 16(12), pp. 1445-1473, 2001.
11. Wilson D.R., Martinez T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning vol. 38, pp. 257-268, 2000.
12. Pedrycz W.: Conditional Fuzzy C-Means, Pattern Recognition Letters, vol. 17, 625-632, 1996.
13. Maron O., Moore A.: The Racing Algorithm: Model Selection for Lazy Learners. Artificial Intelligence Review, vol. 11, 193-225, 1997.
14. Mertz C.J., Murphy P.M.: UCI repository of machine learning databases, www.ics.uci.edu/pub/machine-learning-databases.
15. Stanfill C., Waltz D.: Toward memory-based reasoning. Communications of the ACM, 29(12), 1986, pp 1213-1228
16. Wilson D. R., Martinez T.R.: Improved Heterogeneous Distance Functions, Journal of Artificial Intelligence Research 6, 1997
17. Duch W., Grudziński K.: Meta-learning via search combined with parameter optimization. Intelligent Information Systems, Advances in Soft Computing, Physica Verlag (Springer) 2002, pp. 13-22
18. Cristianini N., Shawe-Taylor J.: An introduction to support vector machines (and other kernel-based learning methods). Cambridge University Press, 2000.
19. Theodoridis S., Koutroumbas K.: Pattern Recognition. Academic Press, 3rd Ed, 2006.

Committee Machine for Road-Signs Classification

Bogusław Cyganek

AGH - University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
cyganek@agh.edu.pl

Abstract. The paper presents a system for the road signs recognition which is based on an ensemble of the non Euclidean distance neural networks and an arbitration unit. The input to this system constitutes a binary pictogram of a sign which is supplied from the detection module. The classifier is composed of a mixture of experts - the Hamming neural networks - each working with a single group of deformed reference pictograms. The ensemble of experts is controlled by an arbitration module operating in the winner-takes-all mode. Additionally it is equipped with a promoting mechanism that favours the most populated group of unanimous experts. The presented classifier is characterized by the fast training and very fast response times which features make it suitable for the driving assistant systems. The presented concepts have been verified experimentally. Their results and conclusions are also discussed.

1 Introduction

The paper presents a compound neural classifier designed to recognize the rectangular shaped road signs (RSs) of the group "D". These are the so called 'information' signs frequently encountered on our roads. This classifier is a part of a more complex driving assisting system which is able to detect and recognize road signs during driving. This is a model-free recognition system with prototypes in a form of deformable images created from the predefined data base (DB). Each set of deformable training patterns is used to train a single 1-nearest-neighbour expert which, in our case, is a Hamming neural network (HNN). All outputs of the experts are then fed into the arbitration module, operating in the winner-takes-all mode, and which was additionally extended by a mechanism that amplifies response of the unanimous experts (a democratic like voting system).

Detection and recognition of the road signs have gained much attention and research [2,6,11,13]. Some shape detectors are described in [1,2,3,6]. RS recognition with the back-propagation NN is proposed in [1,6]. However, there are many ways of image preprocessing to feed the input neurons. For example in [5] the corners are detected and their mutual relations are analyzed, then the final classification is done by the NN. The Kohonen NN is proposed in [9] - this network is trained considering rotations and occlusions. The other NNs for the sign recognition were also proposed: the receptive field NN [10], the radial basis function RBF NN [15], and the adaptive resonance ART NN [5].

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 583–592, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Structure of the Road Sign Classification System

Fig. 1 depicts architecture of the proposed classification system for the RS detection and recognition. This is a general scheme which can fit many classification systems and which behaviour depends on the chosen building blocks. In our case the process begins with the image acquisition. We use the Olympus C70 camera for image acquisition. Then the input image is fed into the detector of rectangular shapes, built upon the structural tensor [3]. It operates on colour images with different scales. However we do not use colour information in other parts of the system. Proper operation of these modules has a significant influence on the rest of the system.

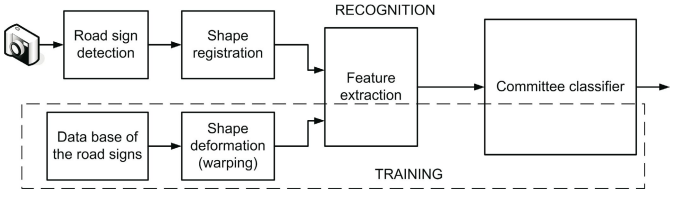


Fig. 1. Overall view of the system for road sign detection and recognition

The detection is followed by the shape registration phase. Its purpose is to recover a rectified view of an object which in natural scenes has been projectively distorted due to physical parameters of the image acquisition system. The road signs have been designed to be easily discernible on the roads. Therefore they are characterized by modal histograms. We utilize this attribute in the feature extraction module (Fig. 1) which binarizes the detected pictograms. Finally, the binary features are fed into the committee classifier depicted in Fig. 2.

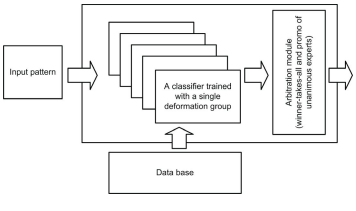


Fig. 2. The hybrid system for the road sign classification

The road sign data base and the shape deformation modules are used during the training phase of the system. The big advantage of the proposed system is its ability to use a data base in the form of the predefined printed patterns instead of natural examples which require much more effort. We use the predefined bitmaps (Fig. 1a) of the Polish law regulations for traffic signalization [12]. However, nothing hinders from using real examples as well. The shape deformation module

in Fig. 1 is based on the inverse warping algorithm that generates predefined horizontally and vertically shifted versions of the training patterns from the data base. Each set of the patterns deformed in one way is fed into one expert of the committee machine in Fig. 2.

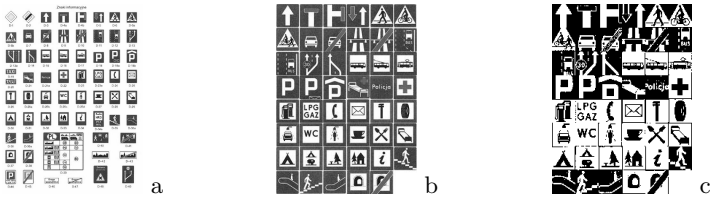


Fig. 3. The formal specification of the "D" group of the Polish RSs (a). The DB created from the specification (b). The pictograms after binarization and sampling (c).

Fig. 2 depicts architecture of the neural committee classifier (see Fig. 1). This is a mixture of experts which are separate Hamming NNs [4]. Each expert is responsible for recognition within a group of a single deformation.

3 Shape Registration and Feature Extraction

The image registration is accomplished by the following affine transformation:

$$\mathbf{Ax} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \hat{x_1} \\ \hat{x_2} \\ 1 \end{bmatrix} = \hat{\mathbf{x}}, \tag{1}$$

where $x = (x_1, x_2, x_3)^T$ is a point in homogeneous coordinate system, A is the warping (affine) matrix, and $\hat{\mathbf{x}}$ is the position of a point $\mathbf{x}$ after the warping.

A value at a point is determined by the bilinear interpolation. The matrix $\mathbf{A}$ is computed from the system of linear equations set from the three non collinear pairs of matched points (the three corners of a sign).

From experiments on real scenes we noticed that binarization around the mean intensity value gives acceptable results for most of the scenes under different

Fig. 4. Example of the real road scene: Each of the detected "D-6" signs is marked by the three white cross pointers (a), sampling scheme of a registered pattern (b)

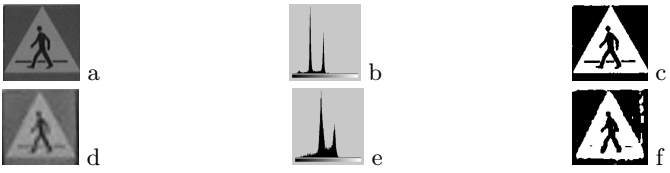


Fig. 5. The two "D-6" signs detected from the scene in Fig. 4a and registered to the 64×64 frame (a,d). Histograms of the registered images (b,e). Mean binarization (c,f).

lighting conditions. Results of this process are presented in Fig. 5 and Fig. 3. The group of "D" road signs (Fig. 1a) contains square, rectangular and special shaped signs (diamonds D-1 and D-2, as well as D-39 and D-42). The latter are dealt separately by the special detection module and are not considered here. We noticed that the system can be further simplified by cropping the rectangular signs to a common square of $d_1 \times d_1$ pixels. Thus, all the patterns fit into a square frame which is further shrunk by a horizontal a vertical offsets (in practice 4-6 pixels) and sampled as shown in Fig. 4b.

4 A Single Expert-Classifier Module

Having binary input signals (i.e. the sign pictograms) the natural choice for an expert is a binary NN. We choose the Hamming NN which is an extension of the well known Hopfield NN and which originally was proposed by Lippmann [8].

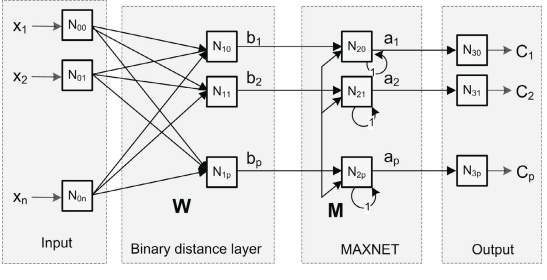


Fig. 6. The hybrid system for the road sign classification

This four (auto-associative) or five (hetero-associative) layers recursive NN has many interesting properties and overcomes some problems encountered in the Hopfield NN, as for example small pattern capacity. It is much faster during the training phase and also in the recognition stage. The very good and thorough analysis of the auto- and hetero-associative memories build upon the HNN, with in-depth analysis of the network convergence and computational complexity, can be found in [7]. The HNN has been quite often used for image classification. In this case however, the best results are obtained when the network is not directly fed with image intensity values but rather with binarized or non-parametrically

preprocessed input signals. The Hamming NN (Fig. 6) directly realizes the one-nearest-neighbor classification rule [4]. For each of the training data vectors we assign a single class C_i. In the recognition stage the one nearest (in the sense of the Hamming distance) vector to the input pattern $\mathbf{x}$ is found and its class C_i is returned. In this type of NN the four layers of neurons can be distinguished:

1. The input binary layer $N_{0(.)}$ which accepts binary vector with components from the set {-1, +1}. The modification to this can include the new "neutral" state with value 0 which can be used to indicate the unknown state [7].
2. The layer $N_{1(.)}$ which computes the binary distance between the input pattern and each reference pattern, already stored in the weights $\mathbf{W}$;
3. The recursive winner-takes-all layer $N_{2(.)}$ selects the winning neuron. The characteristic is a self connection of a neuron with the weight 1.0.
4. The output layer $N_{3(.)}$ which at the stable state contains at most one neuron with non-zero output - a winner which indicates a class of the input vector.

In the hetero-associative version there is an another final layer $N_{4(.)}$ which returns a new output vector $\mathbf{y}$ which the network associated with $\mathbf{x}$. Training of the binary NN is fast and simple, at least when compared to other NNs. It consists of filling in the matrix $\mathbf{W_{pn}}$ in Fig. 6, as follows:

$$\mathbf{w_i} = \mathbf{x_i}, \quad 1 \leq i \leq p, \tag{2}$$

where p is the number of input patterns $\mathbf{x}$, of length n, w_i is the *i-th* row of the matrix $\mathbf{W_{p \times n}}$. The computation time is linear with the size and number of input patterns p. The recursive layer $N_{2(.)}$ selects a winning neuron. However, prior to this its weights need to be initialized. This group of neurons is characterized by a self connection of a neuron to itself with a weight $m_{ii} = 1$ for all $l \leq i \leq p$. At the same time all other weights are kept negative. Thanks to this strategy a neuron amplifies itself while all other neurons try to lower its value by amount which is proportional to the connecting weights and values of these neurons. Initialization of the $N_{2(.)}$ layer is done by assigning negative values to the square matrix $\mathbf{M_{p \times p}}$ except the main diagonal. Originally Lippmann proposed the following formula for initialization [8]:

$$m_{kl} = -\frac{1}{p-1} + \xi_{kl} \quad for \quad k \neq l, \quad 1 \quad for \quad k = l, \quad where \; 1 \leq k, l \leq p, p > 1, \tag{3}$$

ξ is a random value for which $|\xi| \ll (p-1)^{-1}$. A modification to (8) was proposed by Floréen [7] which consists of assigning ε_k to all m_{kl} for $k \neq l$, as follows:

$$m_{kl} = \varepsilon_k = -\frac{1}{p-1}\left(1 - \frac{1}{n}\right)^{\frac{k-1}{p-1}}, \quad k \neq l, \; 1 \; for \; k = l, \quad 1 \leq k, l \leq p, p > 1, \tag{4}$$

Computed this way values ε_k are near-optimal in terms of the network convergence. The computational complexity of this scheme is of order $O(p \cdot log(np))$ in general case and $O(p \cdot log(n))$ if there exists a unique stored vector which is the nearest to the input pattern [7]. The other advantage of using (4) is significant

saving of the memory necessary to store the matrix $\mathbf{M}$ which in this case reduces to a single vector of length p. However, it appears that the most efficient and still convergent solution consists of setting equal weights for all neurons $N_{2(.)}$ which are then modified at each step during classification phase, as follows [7]:

$$m_{kl} = \varepsilon_k(t) = -\frac{1}{p-t}, \quad k \neq l,\ 1\ for\ k = l, \quad 1 \leq k, l \leq p, p > 1, \tag{5}$$

where t is a classification time step. In this case the convergence is achieved in *p-1-r* steps, where $r > 1$ stands for a number of nearest vectors stored in $\mathbf{W}$ [7]. During the classification phase the group $N_{1(.)}$ of neurons computes the binary distance measure between the input pattern $\mathbf{z}$ and the training patterns stored in $\mathbf{W}$. Most often than not this is the Hamming distance (i.e. amount of mismatched positions in the two vectors), given as follows [7]:

$$b_i(\mathbf{z}, \mathbf{W}) = 1 - \frac{1}{n} D_H(\mathbf{z}, \mathbf{w_i}), \quad 1 \leq i \leq p, \tag{6}$$

where $b_i \in [0, 1]$ is a value of an *i-th* neuron in $N_{1(.)}$, $D_H(\mathbf{z}, \mathbf{w_i}) \in \{0, 1, \ldots, n\}$ is the Hamming distance between the input patterns $\mathbf{z}$ and the *i-th* stored pattern w_i (i.e. the *i-th* row of $\mathbf{W}$). It is interesting to observe that values of D_H take on only discrete values whereas values of b_i are in the real set $[0, 1]$. Binary values of the input patterns influence only computation of D_H in (6). Usually the vectors are assumed to have coefficients from the set {-1,+1}. In this case (6) can be computed as follows:

$$b_i(\mathbf{z}, \mathbf{W}) = \frac{1}{2}(n^{-1}\mathbf{w_i}\mathbf{z} + 1) = \frac{1}{2}(n^{-1}\sum_{r=1}^{n} w_{ir} z_r + 1), \quad 1 \leq i \leq p, \tag{7}$$

where $\mathbf{z}$ is a column vector. For the vectors with values from the set {0,1}, which is more common in image processing, we have the following formula:

$$b_i(\mathbf{z}, \mathbf{W}) = 1 - n^{-1}\mathbf{w_i}\mathbf{z} = 1 - n^{-1}\sum_{r=1}^{n} w_{ir} z_r, \quad 1 \leq i \leq p, \tag{8}$$

In [7] it is proposed to extend the allowable values from the binary set {-1,+1} to the ternary {-1,0,+1} where the new middle value 0 indicates a "don't know" state. During classification $N_{2(.)}$ operates recursively to select a winner:

$$a_i[t+1] = \varphi(\sum_{j=1}^{n} m_{ij} a_j[t]) = \varphi(a_i[t] + \sum_{j=1, i \neq j}^{n} m_{ij} a_j[t]), \tag{9}$$

where $a_i[t]$ is an output of the *i-th* neuron of the $N_{2(.)}$ layer at the iteration step t, ϕ denotes a threshold function which is given as follows:

$$\varphi(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}. \tag{10}$$

Choosing different schemes (3-5) for computation of the m_{ij} in (9) results in different dynamics of the classification stage. The goal of the iterative process (9) is to run up until only one neuron has value different than 0 (a winner).

5 The Arbitration Unit

The arbitration unit selects a winner of the whole system. It operates after E competing experts select their local winners (Fig. 2). This unit follows the same MAXNET scheme, analogous to the $N_{2(.)}$ layer in Fig. 6. Inputs to this layer come from the experts $e \in \{1, \ldots, E\}$ after reaching their final states and selecting their winners w_e. Their output follow (9) when reaching the final step t_F: $a_w^e[t_w^e] \in \{0, 1\}$. The weights of the arbitration MAXNET layer are initialized in accordance with (3) and with p=E. Randomized scheme (3) appeared to be the most effective for initialization, especially for similar output values a_w coming from the expert modules. Finally, an answer is accepted iff a winning neuron has its score greater than a predefined threshold t_H. Choice of its value depends on many factors, such as: size of the pictograms in DBs, number of experts, assumed level of acceptance/rejection ratio, etc.

A mechanism has been developed to cope with situations when many experts select the same winning class. This is more probable for small number of experts. Such common voting can be a clue for selecting a dominating class (similar to human experts or democratic voting). We propose the promotion mechanism which operates as follows: the most populous group of unanimous experts (the ones voting for the same class) relax their inhibition weights to the neurons belonging to their group. This way a cooperating group of support is emerged. If such a dominating group does not exist then this mechanism is not applied. To apply this rule the weights (5) of neurons from a group of support are modified as follows:

$$m_{kl} = \begin{cases} m_{kl} + \gamma_G & \text{if} \quad m_{kl} < -\gamma_G \\ m_{kl} & \text{otherwise} \end{cases} \quad k \neq l,\ (k,l) \in G\,, \tag{11}$$

where γ_G is a positive value related to the mean of the cooperating group G:

$$\gamma_G = \frac{c}{N} \sum_{(k,l) \in G, k \neq l} |m_{kl}|, \tag{12}$$

where N is a number of indices $\{k,l\}$ such that $\{k, l\} \in G, k \neq l$, c is a constant value (in practice 0.05-0.3).

6 Experimental Results

For implementation we used the Microsoft© Visual C++ 6.0 and Intel© C++ 9.0 compiler (code optimized for speed). The computer platform consists of the IBM PC© with Pentium 4© 3.4G and 2 GB RAM, controlled by the Windows XP Prof. The system has two types of detectors for the signs of group "D". The first is based on the structural tensor [9] and is used during classification. The second is a manual detector where a user points the three corner points of a sign (Fig. 4a). It was used to create the non-deformed reference data base (Fig. 3b) of the Polish "D" road signs from their formal specification (Fig. 3a) [12]. We also tested influence of the size of the prototype patterns on the classification quality.

For this purpose we created two reference DBs with patterns of size 44×44 and 64×64, respectively. This gives 1056 and 2756 input bits per single pattern (with 6×6 border margins - Fig. 4b), respectively. This is also a number of input neurons for each of the experts (Fig. 6). The smaller sizes of DBs gave much worse results because of lack of sufficient salient features. In this case classification was also more problematic since many different patterns had the same binary distance. The question is how to configure the system classification with many deformable models of reference patterns. In this case we have p classes, however each class is represented by up to q deformable versions of the main reference pattern. There are two possible configurations of the system: Create p experts, each responsible for classification of a single sign based on its q deformable reference patterns. The second option is to create q experts, each classifying all road sign under a single deformation. Since we changed q from 1 to 289 deformations we choose the latter, which is also more suitable for incremental build of the system, i.e. adding step by step new experts that work on a new data base with differently deformed images. As already mentioned, the deformations were created by combinations of horizontal and vertical shifts of the reference images. The maximum tested version of such deformation was with ± 16 pixels shifts with 2 pixels step. This gives $17^2 = 289$ possibilities and this is the maximal amount of experts that were tested. Due to quite precise automatic registration of the input patterns to the lower base segment of the "D" signs, we avoided generation of the rotated versions of DBs which would significantly increase number of experts.

Quality of the system was measured in terms of Precision vs. Recall on the patterns from the two data-bases: The first one contains only "D" RS objects - it is created on the base of the reference DB with "D" road signs that was already used during the training. We draw a test pattern from this DB, adding the Gaussian noise and random deformation however. The second DB is composed of 50% "D" RSs and 50% of non-RS objects. The latter are randomly created from other images that do not contain any RSs. As previously, a drawn object is deformed and noised.

Table 1 presents results of our system operating on the first DB. In this experiment we tested an influence of the resolution of the pictograms, as well

Table 1. Classification results on the DB of "D" road signs with different resolutions and the arbitration thresholds. Horz/vert deformations ± 16 pixels with 2 pixels step

Pat.size ((t_H))	44×44 (0.01)		64×64 (0.01)		44×44 (0.05)		64×64 (0.05)	
PSNR[dB]	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
100	1.000	0.958	1.000	1.000	1.000	0.939	1.000	0.987
50	1.000	0.922	1.000	0.928	1.000	0.830	1.000	0.682
44	1.000	0.860	1.000	0.895	1.000	0.699	1.000	0.454
40	1.000	0.906	1.000	0.940	1.000	0.729	1.000	0.490
38	1.000	0.764	1.000	0.927	1.000	0.710	1.000	0.735

as threshold value of the arbitration MAXNET module on the precision and recall of the system. As we have expected the best results were obtained for larger resolutions of the pictograms used for training (the more discriminative features) and smaller threshold value (less requirements on a winning neuron). The test patterns were gradually deteriorated with noise and random deformations (horz/vert shifts). Although PSNR values (computed from the expectation value of a difference of a 'pure' image and its noised version) indicate significant deterioration of the patterns, the system performs well.

Table 2. Classification results on a DB with 50/50 true/false examples of the "D" RSs (randomly cropped from provided images). All pictograms 64×64.

$t_H(c)$	0.0275 (c=0.2)		0.0275 (c=0.35)		0.07 (c=0.15)		0.07 (c=0.35)	
PSNR[dB]	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
100	0.937	1.000	0.882	1.000	1.000	0.930	1.000	0.964
80	1.000	0.696	0.939	0.628	1.000	0.659	1.000	0.693
50	1.000	0.329	0.991	0.327	1.000	0.320	1.000	0.359
40	1.000	0.202	0.842	0.260	1.000	0.247	1.000	0.213
30	1.000	0.154	0.617	0.185	1.000	0.124	1.000	0.159

Classification results obtained on the second DB and with 64×64 pictograms presents Table 2. In this experiment we tested robustness on the system in terms of true vs. false patterns in terms of the arbitration threshold and constant c in (12). We noticed a more rapid deterioration of the recall value with added noise and distortions to the test images. However, the precision parameter is high (especially for $t_H = 0.07$) what indicates that almost all 'good' objects were detected. We also noticed a positive influence of the proposed unanimous arbitration mechanism on the recall factor. The conclusion from these experiments is that the performance of the system is satisfactory taking into consideration that - when built in the whole driving assistant system - its input is coming from the precisely tuned RS detector which acts as a pre-classification mechanism. This also follows the principle of boosted cascade of simple classifiers proposed e.g. in [14]. Currently we are developing the additional classifier which operates in the log-polar space and is responsible for rejection of non road sign patterns before they are put into the classifier described in this paper.

7 Conclusions

The paper presents the committee machine for classification of the "D" road signs. The proposed classifier is a part of a system for driving assistance. Its works with the deformable prototypes from the reference DB. Each group of deformed patterns is processed by a single one-nearest-neighbour classifier for which we used the Hamming NN. The ensemble of experts is orchestrated by the arbitration unit which is a MAXNET layer that selects the winning neuron. The

arbitration is augmented by the mechanism of supporting groups of unanimous experts. The achieved results, as well as run times are quite promising. The system can be built in about 1 second, whereas its response is in order of ms.

Acknowlegement

This work was sponsored by the Polish scientific grant no. KBN 3T11C 045 26.

References

1. Aoyagi, Y., Asakura, T.: A study on traffic sign recognition in scene image using genetic algorithms and neural networks. IEEE Conf. Electronics (1996) 1838-1843
2. Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic Detection and Recognition of Signs From Natural Scenes. IEEE Trans. on Image Proc. 13(1) (2004) 87-99
3. Cyganek, B.: Object Detection in Multi-Channel and Multi-Scale Images Based on the Structural Tensor. Springer LNCS 3691 (2005) 570-578
4. Duch, W., Grudziski, K.: A framework for similarity-based methods. Second Polish Conference on Theory and Applications of Artificial Intelligence (1998) 33-60
5. Escalera, A., Moreno, L., Salichs, M., Armingol, J.: Road traffic sign detection and classification. IEEE Trans. Ind. Electron., v.44 (1997) 848-859
6. Escalera, A., Armingol, J.: Visual Sign Information Extraction and Identification by Deformable Models. IEEE Tr. On Int. Transportation, 5(2) (2004) 57-68
7. Floréen, P.: Computational Complexity Problems in Neural Associative Memories. PhD Thesis, University of Helsinki, Finland (1992)
8. Lippman, R.: An introduction to computing with neural nets. IEEE Transactions on Acoustic, Speech, and Signal Processing, v.ASSP-4 (1987) 4-22
9. Luo, R., Potlapalli, H., Hislop, D.: Neural network based landmark recognition for robot navigation. IEEE Int. Conf. Industrial Electronics (1992) 1084-1088
10. Luo, R., Potlapalli, H.: Landmark recognition using projection learning for mobile robot navigation. In Proc. IEEE Int. Conf. Neural Networks (1994) 2703-2708
11. Piccioli, G., Micheli, E.D., Parodi, P., Campani, M.: Robust method for road sign detection and recognition. Image and Vision Computing, v.14 (1996) 209-223
12. Road Signs and Signalization. Directive of the Polish Ministry of Infrastructure. Internal Affairs and Administration (Dz. U. Nr 170, poz. 1393) (2002)
13. Rehrmann, V., Lakmann, R., Priese, A.: A parallel system for real-time traffic sign recognition. Deimler-Benz Technical Report (1995)
14. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE Conf. on Computer Vision and Pattern Recognition (2001) 151-158
15. Zheng, Y. J., Ritter, W., Janssen, R.: An adaptive system for traffic sign recognition. Proc. IEEE Intelligent Vehicles Symp. (1994) 165-170

Cluster Analysis Via Dynamic Self-organizing Neural Networks

Marian B. Gorzałczany and Filip Rudziński

Department of Electrical and Computer Engineering
Kielce University of Technology
Al. 1000-lecia P.P. 7, 25-314 Kielce, Poland
{m.b.gorzalczany, f.rudzinski}@tu.kielce.pl

Abstract. The paper presents dynamic self-organizing neural networks with one-dimensional neighbourhood that can be efficiently applied to complex, multidimensional cluster-analysis problems. The proposed networks in the course of learning are able to disconnect their neuron chains into sub-chains, to reconnect some of the sub-chains again, and to dynamically adjust the overall number of neurons in the system; all of that - to fit in the best way the structures "encoded" in data sets. The operation of the proposed technique has been illustrated by means of three synthetic data sets, and then, this technique has been tested with the use of two real-life, complex and multidimensional data sets (*Optical Recognition of Handwritten Digits Database* and *Image Segmentation Database* of *Statlog Databases*) available from the ftp-server of the University of California at Irvine (ftp.ics.uci.edu).

1 Introduction

Cluster analysis in data sets is an essential issue in designing various systems for "making sense" of data in order to support the user in better understanding of data and thus making sensible decisions. In general, cluster analysis aims at partitioning a given set of data or objects into clusters (classes, groups) such that the elements of each cluster are as "similar" as possible to each other and as "different" as possible from those of the other clusters (see, e.g., [4] for review).

This paper presents a clustering technique that is an advanced and improved version of clustering method presented in [2]; the latter method, in turn, develops the approaches of [1]. Therefore, papers [1], [2], and the present one (by the same authors) can be put in a line demonstrating the evolution of the implementation of some new, general and flexible idea of clustering. This idea consists in designing a route coming through the data set at hand; then, an analysis of this route (by determining the histogram of nearness between neighbouring points along the route - see [1, 2] for details) provides the user with an image - on the plane - of the cluster distribution in the considered multidimensional data set.

This paper, first, illustrates - by means of a synthetic data set - some drawbacks of the method of [2] that operates on the modification of the conventional self-organizing neural network with one-dimensional neighbourhood (the neuron chain). The modification consists in embedding into the learning algorithm

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 593–602, 2006.
© Springer-Verlag Berlin Heidelberg 2006

of the network some mechanisms allowing the neuron chain to disconnect and reconnect in the course of learning to better fit the data structure in a given data set. Then, a generalization of the method of [2] - in the form of a dynamic self-organizing neural network - is formulated. The generalization aims at automatic adjustment of the number of neurons in the network in the course of its learning in order to further enhance its abilities to fit in the best way the cluster distribution in a given data set. In turn, an illustration of the operation of the dynamic self-organizing neural networks with the use of three synthetic data sets is provided. Finally, the advantages of using the dynamic self-organizing neural networks are demonstrated in the clustering of two real-life, complex and multidimensional data sets (*Optical Recognition of Handwritten Digits Database* and *Image Segmentation Database* of *Statlog Databases*) [5]. The application of the proposed technique to big-scale problems of WWW-document clustering is presented in [3].

2 Illustration of Drawback of Modified Self-organizing Neural Network of [2] in Cluster Analysis

Fig. 1 shows a synthetic data set (1000 points of the plane) with three visible "parallel" clusters and several routes in this set determined by modified self-organizing neural networks of [2] for different numbers of neurons in the neuron chains. It is clear that the number of neurons - fixed and experimentally selected at the beginning of the learning - is the parameter that critically affects the performance of the system. Too small number of neurons in the chain unables its free spread in the whole data set (Fig. 1b), whereas too large number of neurons results in a relatively big number of low-active neurons that affect the correct operation of the sub-chain disconnection- and reconnection mechanisms of the considered technique (Fig. 1d). In both cases the system provides wrong image of the cluster distribution in a given data set; see Fig. 1e - the envelope of the nearness histogram (see [1, 2] for details) for the route of Fig. 1d suggesting the occurrence of four clusters.

3 Dynamic Self-organizing Neural Networks for Cluster Analysis

Since the predefinition of the number of neurons in the neuron chain is the critical parameter of the approach of [2], the main goal of its generalization (in the form of a dynamic self-organizing network) is to solve that problem by introducing a mechanism that automatically adjust the number of neurons in the course of learning. This mechanism (it has three components - see below) supports two already existing sub-chain disconnection- and reconnection mechanisms (see [2] for the discussion of their rationale) in fitting by the neuron chain - in the best way - the data structure (the cluster distribution) in a given data set. In order to achieve this goal, five successive operations are activated (under some condition) after each learning epoch of the network:

1) the removal of single, low-active neurons,
2) the disconnection of a neuron chain (sub-chain) into two sub-chains,
3) the removal of short neuron sub-chains,
4) the insertion of additional neurons into the neighbourhood of high-active neurons in order to take over some of their activities,
5) the reconnection of two selected sub-chains.

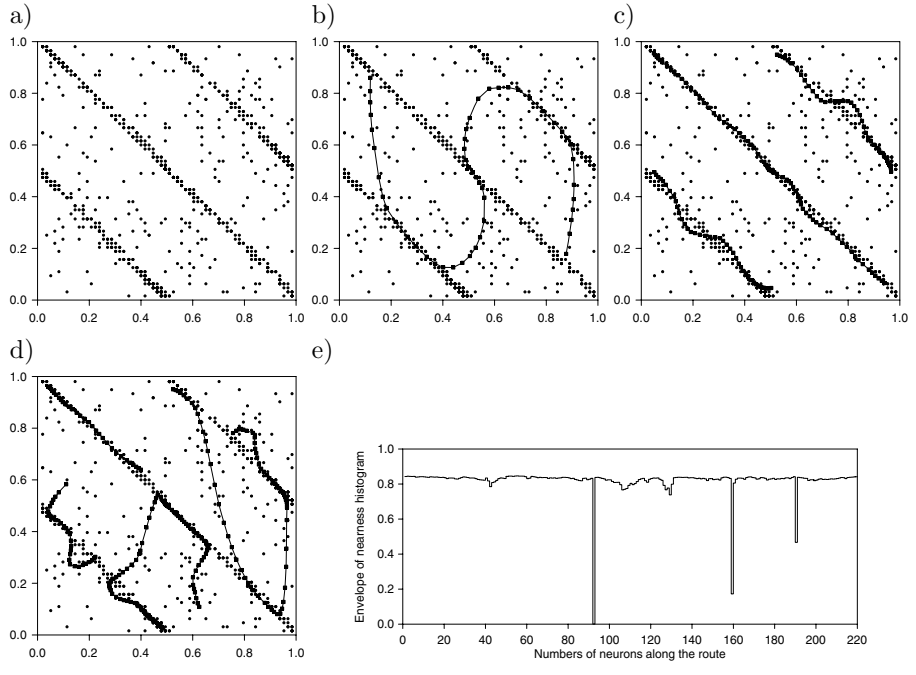


Fig. 1. Synthetic data set (a) and the routes in it determined by modified self-organizing neural networks with: b) 50, c) 150, and d) 220 neurons as well as the envelope of nearness histogram for the route of Fig. 1d (e)

The operations nos. 1, 3, and 4 are the components of the afore-mentioned mechanism for automatic adjustment of the number of neurons in the chain. Based on experimental investigations, the following conditions for particular operations have been formulated (numberings of conditions and operations are the same). Possible operation takes place between neuron no. i and neuron no. $i+1$; $i \in \{1, 2, \ldots, r-1\}$, where r is the number of neurons in the original neuron chain or a given sub-chain.

Condition 1: $win_i < \beta_1$, where win_i is the number of wins of i-th neuron and β_1 is experimentally selected parameter (usually $\beta_1 \in \{3, 4, \ldots, 7\}$). This condition allows to remove single neuron whose activity (measured by the number of its wins) is below an assumed level represented by parameter β_1.

Condition 2: $d_{i,i+1} > \alpha_1 \frac{\sum_{j=1}^{r-1} d_{j,j+1}}{r}$, where $d_{i,i+1}$ is the distance between the neurons no. i and no. $i+1$ (see [1] for details) and α_1 is experimentally selected parameter (usually $\alpha_1 \in [2, 4]$). This condition prevents the excessive disconnection of the neuron chain or sub-chain by allowing to disconnect only relatively remote neurons.

Condition 3: $r_S < \beta_2$, where r_S is the number of neurons in sub-chain S and β_2 is experimentally selected parameter (usually $\beta_2 \in \{3, 4\}$). This condition allows to remove r_S-element neuron sub-chain S that length is shorter than assumed acceptable value β_2.

The operation of the insertion of additional neurons into the neighbourhood of high-active neurons in order to take over some of their activities covers 3 cases denoted by 4a, 4b, and 4c, respectively.

Condition 4a (the insertion of new neuron (n) between two neighbouring high-active neurons no. i and no. $i+1$): IF $win_i > \beta_3$ AND $win_{i+1} > \beta_3$ THEN weight vector $\boldsymbol{w}_{(n)}$ of new neuron (n) is calculated as follows: $\boldsymbol{w}_{(n)} = \frac{\boldsymbol{w}_i + \boldsymbol{w}_{i+1}}{2}$, where win_i, win_{i+1} are as in Condition 1 and β_3 is experimentally selected parameter (usually $\beta_3 \in \{5, 6, \ldots, 9\}$).

Conditions 4b (the replacement of high-active neuron no. i - accompanied by low-active neurons no. $i-1$ and no. $i+1$ - by two new neurons: (n) and $(n+1)$): IF $win_i > \beta_3$ AND $win_{i-1} < \beta_3$ AND $win_{i+1} < \beta_3$ THEN weight vectors $\boldsymbol{w}_{(n)}$ and $\boldsymbol{w}_{(n+1)}$ of new neurons (n) and $(n+1)$ are calculated as follows: $\boldsymbol{w}_{(n)} = \frac{\boldsymbol{w}_{i-1} + \boldsymbol{w}_i}{2}$ and $\boldsymbol{w}_{(n+1)} = \frac{\boldsymbol{w}_i + \boldsymbol{w}_{i+1}}{2}$ (β_3 - as in Condition 4a).

Condition 4c (the insertion of new neuron in the neighbourhood of an end-chain high-active neuron accompanied by low-active neighbour; r-th neuron case will be considered; 1st neuron case is analogous): IF $win_r > \beta_3$ AND $win_{r-1} < \beta_3$ THEN weight vector $\boldsymbol{w}_{r+1}$ of new neuron $(r+1)$ is calculated as follows: $\boldsymbol{w}_{r+1} = \boldsymbol{w}_r + \frac{\boldsymbol{w}_r - \boldsymbol{w}_{r-1}}{d_{r,r-1}} d_{avr}$, where $d_{avr} = \frac{1}{r-1} \sum_{j=1}^{r-1} d_{j,j+1}$ (β_3 - as in Condition 4a and $d_{j,j+1}$ - as in Condition 2).

Condition 5: $d_{e_{S1},e_{S2}} < \alpha_2 [\frac{1}{2} (\frac{\sum_{j=1}^{r_{S1}-1} d_{j,j+1}}{r_{S1}} + \frac{\sum_{j=1}^{r_{S2}-1} d_{j,j+1}}{r_{S2}})]$, where $S1$ and $S2$ are two sub-chains (containing r_{S1} and r_{S2} neurons, respectively) whose appropriate ends $e_{S1} \in \{1, r_{S1}\}$ and $e_{S2} \in \{1, r_{S2}\}$ are closest to each other; sub-chains $S1$ and $S2$ are the candidates for the connection by combining their ends e_{S1} and e_{S2} ($d_{j,j+1}$ - as in Condition 2, α_2 - experimentally selected parameter (usually $\alpha_2 \in [0.5, 4]$)). This condition allows to connect two sub-chains not only with closest ends but also with relatively close to each other neighbouring neurons that correspond to compact pieces of the same cluster of data.

Conditions 1 through 5 are checked after each learning epoch. The condition that is fulfilled activates the appropriate operation. As far as the network learning rule is concerned, the Winner-Takes-Most (WTM) approach has been applied. It uses the Gaussian-type neighbourhood function for the winning neurons as well as the Euclidean distance measure between learning- and weight vectors in determining these neurons (see also [1, 3]).

Fig. 2 shows the performance of the dynamic self-organizing neural network applied to the synthetic data set of Fig. 1a considered earlier in this paper.

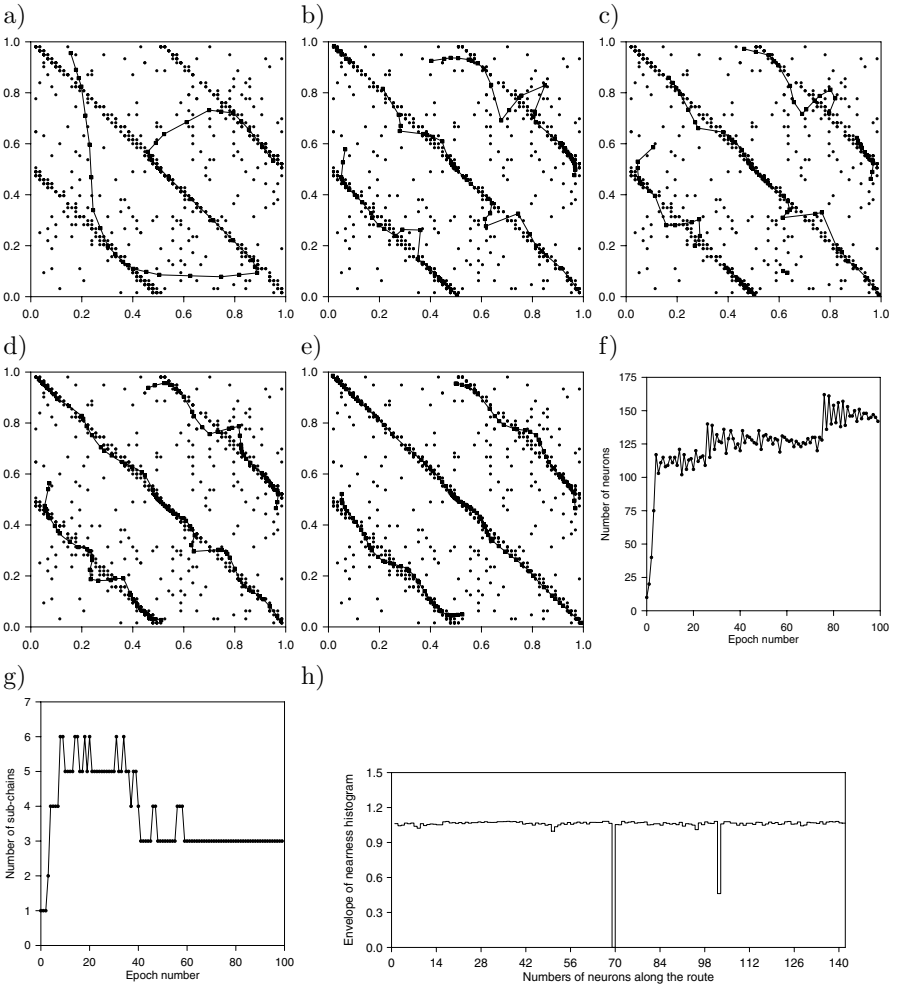


Fig. 2. Synthetic data set of Fig. 1a and the route in it in learning epochs: a) no. 2, b) no. 5, c) no. 15, d) no. 50, and e) no. 100 (end of learning), as well as the plots of number of neurons (f) and number of sub-chains (g) vs. epoch number, and the envelope of nearness histogram for the route of Fig. 2e (h)

This time, the system automatically adjusts the number of neurons in the network starting from arbitrarily chosen 10 neurons at the beginning of learning. In the final stage of learning the number of neurons oscillates around an optimal value, to stabilize finally on 124 neurons (see Fig. 2f). The number of neuron sub-chains in the course of learning is presented in Fig. 2g. The proposed approach - as it can be seen in Fig. 2e and Fig. 2h (the envelope of nearness histogram for the route of Fig. 2e) - provides a clear and correct image of the cluster distribution in the problem under consideration.

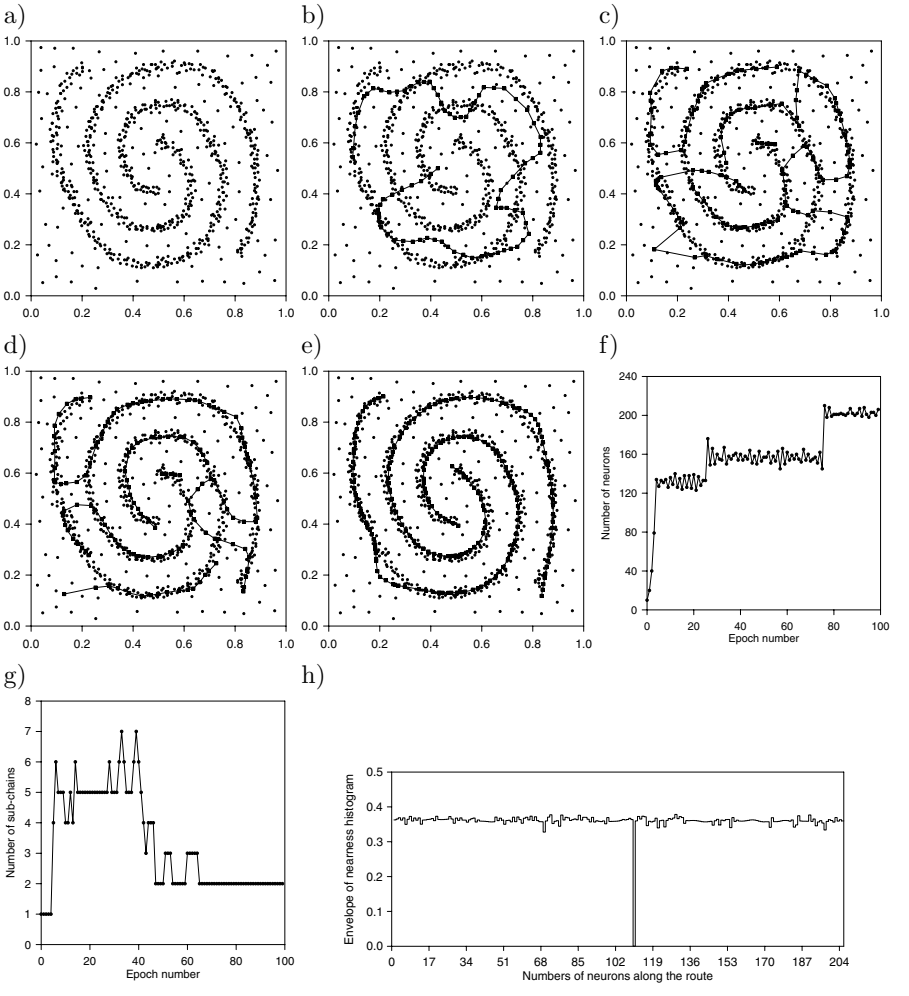


Fig. 3. Synthetic data set (a) and the route in it in learning epochs: b) no. 3, c) no. 10, d) no. 20, and e) no. 100 (end of learning), as well as the plots of number of neurons (f) and number of sub-chains (g) vs. epoch number, and the envelope of nearness histogram for the route of Fig. 3e (h)

Figs. 3 and 4 are further illustrations of the performance of the proposed clustering technique applied to two synthetic data sets presented in Figs. 3a and 4a, respectively.

Data set of Fig. 3a contains clusters in the form of two spirals "wound" on each other, whereas data set of Fig. 4a contains as many as fourteen clusters of different sizes and shapes. Such a diversified collection of data sets (including also two real-life, multidimensional data sets considered later in this paper) enables us to carry out a comprehensive test of the performance of the proposed clustering technique.

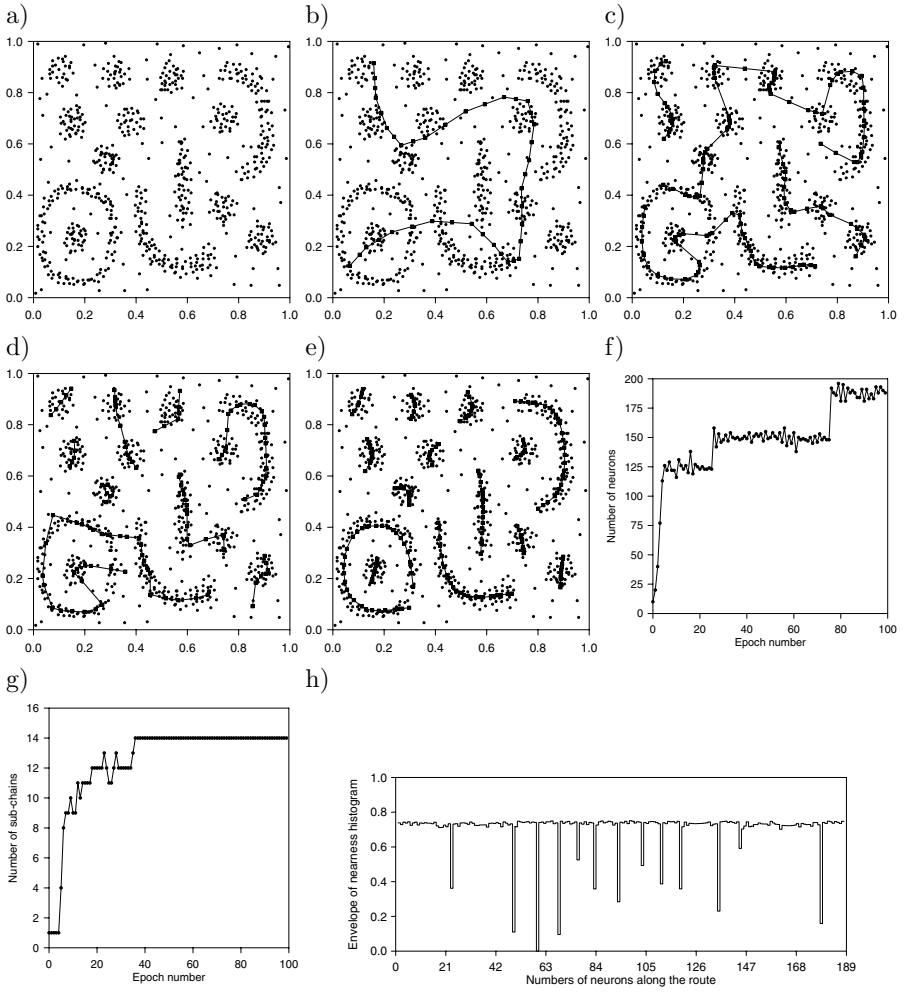


Fig. 4. Synthetic data set (a) and the route in it in learning epochs: b) no. 2, c) no. 5, d) no. 10, and e) no. 100 (end of learning), as well as the plots of number of neurons (f) and number of sub-chains (g) vs. epoch number, and the envelope of nearness histogram for the route of Fig. 4e (h)

As far as experiments regarding data sets of Figs. 3a and 4a are concerned, in both cases the dynamic neural systems automatically adjust the numbers of their neurons starting - as in experiment of Fig. 2 - from arbitrarily chosen 10 neurons at the beginning of learning. In the final stage of learning the numbers of neurons oscillate around optimal values, finally stabilizing on 206 neurons for two-spiral problem (see Fig. 3f), and 188 neurons for the problem of Fig. 4a (see Fig. 4f). The numbers of neuron sub-chains in the course of learning are presented in Figs. 3g and 4g, respectively. As it can be seen in Fig. 3e and Fig. 3h (the envelope of nearness histogram for the route of Fig. 3e) in the

case of two-spiral problem, and, analogously, in Fig. 4e and Fig. 4h in the case of fourteen-cluster problem, the proposed clustering technique provides correct, clear and easily-verified image of the cluster distribution in both problems under consideration.

4 Application to Complex, Multidimensional Cluster-Analysis Problems

The dynamic self-organizing neural networks as the clustering technique will now be tested with the use of two real-life, complex and multidimensional data sets (*Optical Recognition of Handwritten Digits Database* and *Image Segmentation Database* of *Statlog Databases*) [5]. The first data set contains 1797 records (numerical descriptions of images of handwritten digits); each record is described by as many as 64 numerical attributes. The second data set consists of 4435 records (numerical descriptions of 3x3-pixel neighbourhoods in a satellite image) described by 36 attributes. Due to complexity of both data sets and high dimensionality of their attribute spaces, their graphical presentation by means of e.g. a well-known Sammon's mapping is of no use.

Figs. 5 and 6 present the performance of the proposed clustering technique in both considered data sets. As the learning progresses, both dynamic neural systems adjust the overall numbers of neurons in their networks (Figs. 5a and 6a, respectively) that finally are equal to 163 (the first data set) and 886 (the second

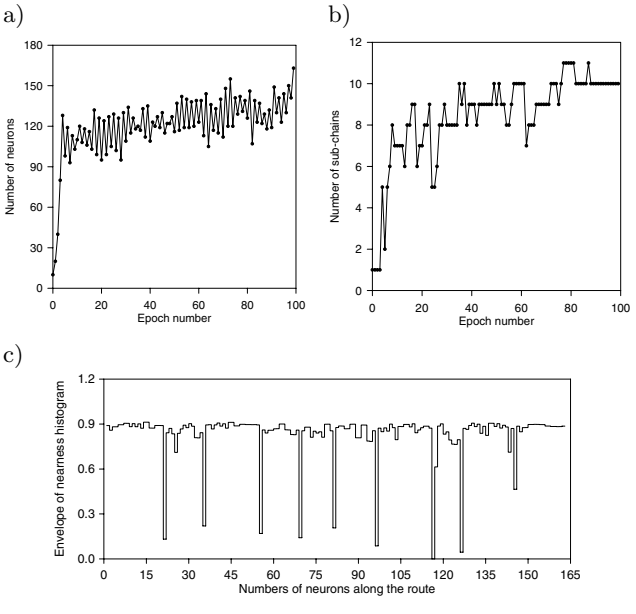


Fig. 5. The plots of number of neurons (a) and number of sub-chains (b) vs. epoch number, and c) the envelope of nearness histogram for the route in the attribute space of *Optical Recognition of Handwritten Digits Database*

data set) as well as the numbers of sub-chains (Figs. 5b and 6b, respectively) finally achieving the values equal to 10 (the first data set) and 6 (the second data set). The number of sub-chains is equal to the number of clusters detected in a given data set. The envelopes of the nearness histograms for the routes in the attribute spaces of both considered data sets (Figs. 5c and 6c, respectively) reveal perfectly clear images of the cluster distributions in them, including the number of clusters and the cluster boundaries (indicated by 9 local minima on the plot of Fig. 5c and 5 local minima on the plot of Fig. 6c). Since the number of classes and class assignments are known in both original data sets, a direct verification of the obtained results is also possible. The percentages of correct decisions, equal to 91.65% (the first data set) and 75.33% (the second data set), regarding the class assignments are very high (especially that they have been achieved by the unsupervised-learning systems operating on complex and multidimensional data sets).

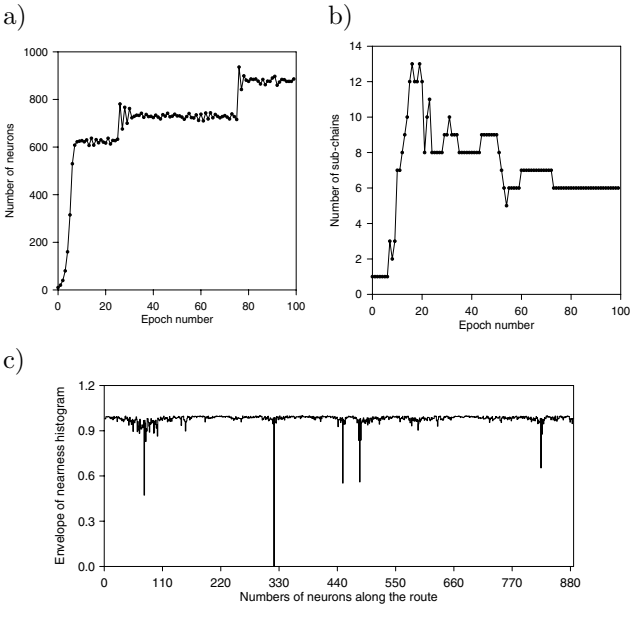


Fig. 6. The plots of number of neurons (a) and number of sub-chains (b) vs. epoch number, and c) the envelope of nearness histogram for the route in the attribute space of *Image Segmentation Database* of *Statlog Databases*

5 Conclusions

The dynamic self-organizing neural networks with one-dimensional neighbourhood that can be efficiently applied in complex, multidimensional cluster-analysis problems have been presented in this paper. The proposed networks in the course of learning are able to disconnect their neuron chains into sub-chains and to reconnect some of the sub-chains again as well as to dynamically adjust the overall

number of neurons in the system. These features enable them to fit in the best way the structures "encoded" in data sets. The operation of the proposed technique has been illustrated by means of three diversified synthetic data sets, and then, this technique has been tested with the use of two real-life, complex and multidimensional data sets (*Optical Recognition of Handwritten Digits Database* and *Image Segmentation Database* of *Statlog Databases*) [5]. The application of the proposed technique to WWW-document clustering is presented in [3].

References

1. Gorzałczany M.B., Rudziński F.: Application of genetic algorithms and Kohonen networks to cluster analysis, in L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L.A. Zadeh (Eds.), Artificial Intelligence and Soft Computing - ICAISC 2004, Proc. of 7th Int. Conference, Lecture Notes in Artificial Intelligence 3070, Springer-Verlag, Berlin, Heidelberg, New York, 2004, pp. 556-561.
2. Gorzałczany M.B., Rudziński F.: Modified Kohonen networks for complex cluster-analysis problems, in L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L.A. Zadeh (Eds.), Artificial Intelligence and Soft Computing - ICAISC 2004, Proc. of 7th Int. Conference, Lecture Notes in Artificial Intelligence 3070, Springer-Verlag, Berlin, Heidelberg, New York, 2004, pp. 562-567.
3. Gorzałczany M.B., Rudziński F.: Application of dynamic self-organizing neural networks to WWW-document clustering, International Journal of Information Technology and Intelligent Computing, to appear (also presented at 8th Int. Conference on Artificial Intelligence and Soft Computing ICAISC 2006, Zakopane).
4. Höppner F., Klawonn F., Kruse R., Runkler T.: Fuzzy Cluster Analysis, J.Wiley&Sons, Chichester, 1999.
5. Machine Learning Database Repository, University of California at Irvine (ftp.ics.uci.edu).

Learning Vector Quantization Classification with Local Relevance Determination for Medical Data

B. Hammer[1], T. Villmann[2], F.-M. Schleif[3],
C. Albani[2], and W. Hermann[4]

[1] Clausthal University of Technology, Institute of Computer Science,
Clausthal-Zellerfeld, Germany
[2] University of Leipzig, Clinic for Psychotherapy, Leipzig, Germany
[3] University of Leipzig, Institute of Computer Science, Germany
[4] Paracelsus Hospital Zwickau, Germany

Abstract. In this article we extend the global relevance learning vector quantization approach by local metric adaptation to obtain a locally optimized model for classification. In this sense we make a step in the direction of quadratic discriminance analysis in statistics where classwise variance matrices are used for class adapted discriminance functions. We demonstrateb the performance of the model for a medical application.

1 Introduction

Classification is an important tool in data processing. Popular methods are traditional statistical approaches like Fishers linear or quadratic discriminant analysis (LDA/QDA) or Learning Vector Quantization (LVQ). Fishers discriminant assumes global constant variance for the linear case and classwise constant variance for the quadratic case. The variance is directly involved in the classification decision by the discriminance function. In this sense, the QDA decision is based on classwise local variance analysis. However, LDA/QDA is a linear classifier, only. LVQ belongs to the prototype based classifiers, i.e. one or more prototypes as representatives for the several class distributions are generated. It has the disadvantage that the adaptation process does not follow a gradient descent of a cost function. The problem is solved by Sato&Yamada proposing the generalized LVQ (GLVQ). As Fishers approach, GLVQ is based on the Euclidean metric. Yet, it can be extended to handle general similarity measures, which may depend on additional parameters. In this way a *global* metric adaptation has been established resulting in relevance GLVQ (GRLVQ). However, the prototypes reflect the *local* properties of their respective classes. Hence, a local metric would be preferable.

In this paper we extend GRLVQ to locally adapt the metric for each prototype. Thus the above mentioned conflict is solved. Choosing a class dependent metric the approach becomes a non-linear extension of QDA, in the same way as GRLVQ can be seen as non-linear extension of LDA.

In mathematical terms, GRLVQ can be interpreted as large margin optimization. The idea of margin optimization has been pioneered in the well-known,

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 603–612, 2006.
© Springer-Verlag Berlin Heidelberg 2006

powerful support vector machine. We extend the margin analysis of GRLVQ to the local variant, introduced in this article. The efficiency of the method is demonstrated in a real life example from medicine.

2 Prototype Based Classification

From a mathematical point of view, we are interested in general classification tasks. Data $X = \{x^i \in \mathbb{R}^n : | : i = 1, \ldots, m\}$, whereby the input vectors x^i are characterized by n features, are to be classified into C given classes. Components of a vector $x \in \mathbb{R}^n$ are referred to by subscripts, i.e., $x = (x_1, \ldots, x_n)$. Prototype based classifiers constitute a particularly intuitive way of classification by means of typical locations of known class allocation which characterize local regions of the data space. Every class c is represented by a set $W(c)$ of weight vectors (prototypes) in $\mathbb{R}^n$. Weight vectors are denoted by w^r and their respective class label is referred to by c_r. A new signal $x \in \mathbb{R}^n$ is classified by the winner-takes-all rule of the classifier, i.e.

$$x \mapsto c(x) = c_r \text{ such that } d(x, w^r) \text{ is minimum.}$$

Thereby, $d(x, w^r)$ is chosen as the squared Euclidean distance

$$d(x, w^r) = \|x - w^r\|^2 = \sum_{i=1}^{n} (x_i - w_i^r)^2$$

of the data point x to the prototype w^r. The respective closest prototype w^r is called winner or best matching unit for x. The subset

$$\Omega_r = \{x^i \in X \mid d(x^i, w^r) \text{ is minimum}\}$$

is called receptive field of neuron w^r. Thus, data point x^i is mapped to the class $c(x^i)$.

Usually, one is interested in finding a prototype based classifier which matches a given training set and its underlying regularity as accurately as possible. A training set consists of a collection of data points together with their known class allocations $\{(x^i, y_i) \in \mathbb{R}^n \times \{1, \ldots, C\} \mid i = 1, \ldots, m\}$. Training aims at minimizing the classification error on the given training set. I.e., prototype locations have to be found such that the difference between the set of points belonging to the cth class, $\{x^i \in X \mid y_i = c\}$ and the receptive fields of the corresponding prototypes, $\bigcup_{w^r \in W(c)} \Omega_r$, is minimized by the adaptation process.

Learning vector quantization (LVQ) as proposed by Kohonen [7] constitutes a popular and simple learning algorithm which forms the base for several extensions and alternatives. The LVQ learning rule consists in heuristically motivated Hebbian learning which leads to a stochastic gradient descent on the cost function

$$\text{Cost}_{\text{LVQ}} = \sum_{x^i \in X} f_{\text{LVQ}}(d_{r_+}, d_{r_-}) \,.$$

d_{r_+} denotes the squared Euclidean distance of x^i to the closest prototype w^{r+} labeled with $c_{r_+} = y_i$, and d_{r_-} denotes the squared Euclidean distance to the closest prototype w^{r-} labeled with a label c_{r_-} different from y_i. For standard LVQ, the decisiion function is

$$f_{\text{LVQ}}(d_{r_+}, d_{r_-}) = \begin{cases} d_{r_+} & \text{if } d_{r_+} \leq d_{r_-} \\ -d_{r_-} & \text{otherwise} \end{cases}$$

Obviously, this cost function is highly discontinuous, and instabilities arise for overlapping data distributions. The decision function of LVQ2.1 is

$$f_{\text{LVQ2.1}}(d_{r_+}, d_{r_-}) = I_w(d_{r_+} - d_{r_-}),$$

whereby I_w yields the identity inside a window where LVQ2.1 adaptation takes place, and I_w vanishes outside. Still this choice might produce an instable dynamic, and the window where adaptation takes place must be chosen carefully. Generalized LVQ (GLVQ) has been proposed by Sato&Yamada as a stable alternative to LVQ2.1 [8]. The respective cost function can be obtained by setting

$$f_{\text{GLVQ}}(d_{r_+}, d_{r_-}) = \text{sgd}\left(\frac{d_{r_+} - d_{r_-}}{d_{r_+} + d_{r_-}}\right)$$

whereby

$$\text{sgd}(x) = (1 + \exp(-x))^{-1}$$

denotes the logistic function. The update rule can be achieved by taking the derivatives [4]

$$\triangle w^{r_+} = 2\epsilon^+ \cdot \text{sgd}'_{\mu(x^i)} \cdot \xi^+ \cdot (x^i - w^{r_+}) \text{ and } \triangle w^{r_-} = -2\epsilon^- \cdot \text{sgd}'_{\mu(x^i)} \cdot \xi^- \cdot (x^i - w^{r_-})$$

where ϵ^+ and $\epsilon^- \in (0, 1)$ are the learning rates, the logistic function is evaluated at position $\mu(x^i) = (d_{r_+} - d_{r_-})/(d_{r_+} + d_{r_-})$, and

$$\xi^+ = \frac{2 \cdot d_{r_-}}{(d_{r_+} + d_{r_-})^2} \quad \text{and} \quad \xi^- = \frac{2 \cdot d_{r_+}}{(d_{r_+} + d_{r_-})^2} \tag{1}$$

denote the derivatives of $f_{\text{GLVQ}}(d_{r_+}, d_{r_-})$ with respect to d_{r_+} and d_{r_-}, respectively. This procedure still has the drawback that it is very sensitive to initialization of prototypes because of the multiple optima of the cost function. This can be widely avoided by integrating neighborhood cooperation resulting in supervised relevance neural gas, whereby the cost function of GLVQ is obtained in the limit for vanishing neighborhood cooperativness.

2.1 Metric Adaptation

Prototype based classifiers crucially depend on the metrics. Since an appropriate metric is usually not clear prior to learning, learning metrics which are automatically adapted during training according to the information contained in the data

are particularly interesting. Since GLVQ is formulated as general cost minimization algorithm, any differentiable similarity measure can be integrated into its cost function instead of the Euclidean metric yielding update rules for the prototypes where the Hebbian terms $(x^i - w)$ are substituted by the derivative of the respective similarity measure with respect to the prototype w, as demonstrated in [4]. The same optimization mechanism can be used to adapt metric parameters during training. Then the prototype update is accompanied by a simultaneous adaptation of the metric parameters. One simple but powerfull choice is the *local scaled Euclidean metric*

$$d_r^{\lambda^r}(x, w^r) = \|x - w^r\|_{\lambda^r}^2 = \sum_{i=1}^{n} \lambda_i^r (x_i - w_i^r)^2 \text{ with } \lambda^r = (\lambda_1^r, \ldots, \lambda_n^r)$$

with the constraint $\lambda_i^r \geq 0$ and $\sum_i \lambda_i^r = 1$ attached to prototype r. Thus, the relevance factors λ_i^r are assigned to a specific prototype and the respective local region of the data space. They can be adapted independently for each local region of the data space. Classification is performed extending the winner takes all rule to this situation

$$x \mapsto c(x) = c_r \text{ such that } \|x - w^r\|_{\lambda^r}^2 \text{ is minimum}\,. \tag{2}$$

Note that now, the receptive fields of prototypes need no longer be convex since no global metric is used for classification. Training can be achived by taking the derivative of the extended cost function

$$\text{Cost}_{LGRLVQ} = \sum_{x^i \in X} \text{sgd}\left(\frac{d_{r_+}^{\lambda^{r_+}} - d_{r_-}^{\lambda^{r_-}}}{d_{r_+}^{\lambda^{r_+}} + d_{r_-}^{\lambda^{r_-}}}\right)$$

We refer to this *local relevance* GLVQ as LGRLVQ. Standard GRLVQ is obtained by $\lambda^r \equiv \lambda, \forall r$. As for GLVQ neighborhood cooperation during learning can easily be established to avoid local optima.

The updates for the prototypes and local relevance terms are achieved taking the derivatives as beforehand. The relevance terms are adapted by

$$\triangle \lambda_l^{r_+} = -\epsilon_\lambda \cdot \text{sgd}'|_{\mu^r(x^i)} \cdot \xi^+ \cdot (w_l^{r_+} - x_l^i)^2$$

and

$$\triangle \lambda_l^{r_-} = \epsilon_\lambda \cdot \text{sgd}'|_{\mu^r(x^i)} \cdot \xi^- \cdot (w_l^{r_-} - x_l^i)^2$$

whereby the local distance measure has to be used in (1). A normalization for each λ^r is added after an adaptation step.

2.2 Generalization Ability

Due to the *local* relevance parameters of LVQ, the generalization bounds of standard GRLVQ [5] do not longer hold for this more powerful setting. Here, we derive large margin generalization bounds for LGRLVQ. Thereby, we derive bounds for general function classes given by the winner takes all rule with

adaptive local metric as defined in equation (2). In addition, we show that the denominator of the cost function function of LGRLVQ characterizes the margin and directly influences the generalization bound. Thus, LGRLVQ can be interpreted as large margin algorithm just as GRLVQ.

Generally speaking, the generalization ability of a classifier refers to the comparison of the training error with the expected error for new data [2,11]. Here, we consider binary classification problems, with classes labeled 1 and -1. Assume an (unknown) probability measure P is given on $\mathbb{R}^n \times \{-1, 1\}$. Training samples (x^i, y_i) are drawn independently and identically distributed (i.i.d. for short) from $\mathbb{R}^n \times \{-1, 1\}$. P^m refers to the product of P if m examples (x^1, y_1), ..., (x^m, y_m) are chosen. The unknown regularity shall be learned by a LGRLVQ-network or some other prototype-based classifier with adaptive local diagonal metric and p prototypes $w^k \in \mathbb{R}^n$ and the respective relevance terms λ^1, ..., λ^p which describe the local weighted metrics. The function computed by the classifier is given by the winner-takes-all rule defined in (2). Denote by

$$F = \{f : \mathbb{R}^n \to \{-1, 1\} \mid f \text{ is given by (2) depending on : } w^1, \ldots, w^p, \lambda^1, \ldots, \lambda^p \in \mathbb{R}^n\}$$

the class of functions which can be computed by such a network. The goal of learning is to find a function $f \in F$ for which the probability

$$E_P(f) := P(y \neq f(x))$$

is minimum. Since the underlying regularity P is not known and only examples (x^i, y_i) are available for characterizing this regularity, training tries to minimize the empirical training error

$$\hat{E}_m(f) := \sum_{i=1}^{m} 1_{y_i \neq f(x^i)}/m$$

whereby $1_{y_i \neq f(x^i)}$ indicates whether x^i is mapped to the desired class y_i or not. Generalization means that $\hat{E}_m(f)$ is representative for $E(f)$ with high probability if the examples are chosen according to P^m such that optimization of the empirical training error will eventually approximate the underlying regularity.

Given a point x with desired output y, we define the margin as the value

$$M_f(x, y) := -d_{r_+}^{\lambda^{r_+}} + d_{r_-}^{\lambda^{r_-}},$$

i.e. x is classified incorrectly iff $M_f(x, y)$ is negative. Otherwise, x is classified correctly with 'security' margin $M_f(x, y)$. Due to the choice of the cost function of LGRLVQ which involves this term within the denominator, LGRLVQ aims at maximizing this margin. Following the approach [2] we define the loss function

$$L : \mathbb{R} \to \mathbb{R}, t \mapsto \begin{cases} 1 & \text{if } t \leq 0 \\ 1 - t/\rho & \text{if } 0 < t \leq \rho \\ 0 & \text{otherwise} \end{cases}$$

for fixed $\rho > 0$. The term

$$\hat{E}_m^L(f) := \sum_{i=1}^{m} L(M_f(x^i, y_i))/m$$

accumulates the number of errors made by f and, in addition, punishes all correctly classified points, if their margin is smaller than ρ. It can be shown using the theory provided in [2] that this modified empirical error, which also includes the margin, is representative for the true error with high probability, whereby a bound is obtained, which is independent of the dimensionality of the input space: We assume that the support of the probability measure P is bounded, i.e. $\forall x$ the inequality $\|x\| \leq B$ holds for some $B > 0$ and further $\|w\| \leq B$, using the standard Euclidean metric.

According to [2](Theorem 7) we can estimate for all $f \in F$ with probability at least $1 - \delta/2$

$$E_P(f) \leq \hat{E}_m^L(f) + \frac{2K}{\rho} \cdot G_m(F) + \sqrt{\frac{\ln(4/\delta)}{2m}}$$

whereby K is a universal positive constant and $G_m(F)$ is the so-called Gaussian complexity of the considered function class which we now define. The empirical Gaussian complexity is given by

$$\hat{G}_m(F) = E_{g_1,\ldots,g_m}\left(\sup_{f \in F}\left|\frac{2}{m}\sum_{i=1}^{m} g_i \cdot f(x^i)\right|\right)$$

for which expectation is taken with respect to independent Gaussian variables $g_1, \ldots, g_m$ with zero mean and unit variance. The Gaussian complexity is the expectation over the i.i.d. points x^i according to the marginal distribution induced by P:

$$G_m(F) = E_{x^1,\ldots,x^m}\hat{G}_m(F).$$

Both complexities measure the richness of the function class F and constitute convenient alternatives to the standard VC-dimension which can also be estimated for prototype-based classifiers.

The classification given by the winner-takes-all rule (2) can be reformulated as fixed Boolean formula over terms of the form $d_i^{\lambda^i} - d_j^{\lambda^j}$ with $d_i^{\lambda^i}$ and $d_j^{\lambda^j}$ constituting the weighted squared Euclidean distance of a given input x to two prototypes w^i and w^j with different class labels. Note that the number of such terms is upper bounded by $p \cdot (p-1)/2$ since p prototypes are available within the classifier. According to [2](Theorem 16) we find

$$G_m(F) \leq p \cdot (p-1) \cdot G_m(F_{ij})$$

whereby F_{ij} denotes the restricted class of classifiers which can be implemented with only two prototypes w^i and w^j with different class label. Define by Λ^i the diagonal matrix with entries λ_j^i. For fixed i and j, we find

$$
\begin{aligned}
& d_i^{\lambda^i} - d_j^{\lambda^j} \leq 0 \\
\iff & (x - w^i)^t \cdot \Lambda^i \cdot (x - w^i) - (x - w^j)^t \cdot \Lambda^j \cdot (x - w^j) \leq 0 \\
\iff & x^t \cdot \Lambda^i \cdot x - x^t \cdot \Lambda^j \cdot x \\
& - 2 \cdot (\Lambda^i \cdot w^i - \Lambda^j \cdot w^j)^t x + (w^i)^t \cdot \Lambda^i \cdot w^i - (w^j)^t \cdot \Lambda^j \cdot w^j \leq 0
\end{aligned}
$$

Hence, every function from F_{ij} can be written as the sum of a function from the set $F_i = \{x \mapsto x^t \cdot \Lambda^i \cdot x\}$, a function form the set $-F_j$, and a function implemented by a simple perceptron, i.e. linear classifier. According to [2](Theorem 12), it holds

$$G_m(c \cdot F) = c \cdot G_m(F)$$

and

$$G_m(\sum_i F_i) \leq \ln m \sum_i G_m(F_i).$$

Thus it is sufficient to independently estimate the Gaussian complexity of linear and quadratic functions of this form.

For linear functions, the estimation follows immediately: since $\|x\| \leq B$, the length of inputs to the linear classifier can be restricted by $B + 1$ (including the bias term). Since all prototypes w are restricted by $\|w\| \leq B$ and the relevance terms add up to 1, the size of the weights of the linear classifier is restricted by $4B + 2B^2$. The empirical Gaussian complexity of this class of linear classifiers can be estimated according to [2](Lemma 22) by

$$\frac{4 \cdot B \cdot (B+1) \cdot (B+2) \cdot \sqrt{m}}{m}.$$

The empirical Gaussian complexity and the Gaussian complexity differ by more than ϵ with probability at most $2 \cdot \exp(-\epsilon^2 m/8)$ according to [2] (Theorem 11).

Since we can interprete the mapping $(x \mapsto (x_1^2, \ldots, x_n^2)$ as feature map of a kernel, an estimation of the Gaussian complexity for the consisdered quadratic functions is also possible: for $x \mapsto \sum \lambda_i^j x_i^2$ with $\|\lambda^j\| \leq 1$ we can estimate the empirical Gaussian complexity by

$$\frac{2 \cdot B^2 \cdot \sqrt{m}}{m}$$

because of [2](Lemma 22), using again the fact $\|x\| \leq B$.

Thus, the overall error bound

$$
\begin{aligned}
E_P(f) &\leq \hat{E}_m^L(f) + \frac{4K \cdot p(p-1)(2B(B+1)(B+2) + B^2) \ln m}{\rho \cdot \sqrt{m}} \\
&\quad + \left(1 + \frac{8K \cdot p(p-1) \cdot \ln m}{\rho}\right) \sqrt{\frac{\ln 4/\delta}{2m}} \\
&\leq \hat{E}_m^L(f) + \frac{\ln m}{\rho \cdot \sqrt{m}} \cdot \sqrt{\ln(1/\delta)} \cdot \mathrm{O}(Kp^2B^3)
\end{aligned}
$$

with probability of at least $1 - \delta$ arises. This term limits the generalization error for all classifiers of the form (2) with adaptive metric if only two classes are dealt with and inputs and weights are restricted by B. Note that this bound is independent of the dimensionality n of the data. It scales inversely to the margin ρ, i.e. the larger the margin the better the generalization ability.

This bound indicates that LGRLVQ includes the objective of structural risk minimization during training because the terms $M_f(x, y)$ which characterize the margin are directly contained in the cost function of LGRLVQ. Naturally, only the extremal margin values need to be limited and thus a restriction of the respective update to extremal pairs of prototypes would suffice. Thus, this argument even proposes schemes for active data selection if a fixed and static pattern set is available for training to speed the algorithm and improve its convergence.

3 Experiments and Concluding Remarks

We apply LGRLVQ to medical data. Prototype based classification plays an important role in medical application because of its intuitive understanding and robust behaviour [9]. We investigate data of psychotherapy patients which underwent a psychodynamic psychotherapy at the University Hospital for Psychotherapy of Leipzig university between 1997 and 2005 [10]. Psychotherapy patients have to give a self-adjudgemet of their recent psychotic stage at several times during therapy. The resepctive questionaire is the *Symptom-Check-List-90-R* (SCL), which was developed to describe psychosomatic impairments [3]. It consists of 90 items, which are collected in 9 scales describing different clinical aspects:

- somatization (12 items)
- obsessive-compulsive (10 items)
- interpersonal sensitivity (9 items)
- depression (13 items)
- anxiety (10 items)
- anger-hostility (6 items)
- phobic anxiety (7 items)
- paranoid ideation (6 items)
- psychoticism (10 items)
- rest scale (7 items)

The rest scale is not under consideration, thus the questionaires data dimension is 9. We investigate two data sets, whereby the patients data were collected at the begin of a stationary psychotherapy in our psychosomatic clinic. The first data set comprise all patients between 1997 and 2001 ($N_1 = 484$), wheras the second one is from 2001 to 2005 ($N_2 = 575$). The statistic of both sets does not differ, such that the first can serve as training set, the other one as test. The patient data should varying according to patients impairment.

The clinical impairment level of the patients was judged by a discretized *global-severity-index* (GSI) level. The GSI is slightly influenced by the age and

Table 1. GSI-levels and clinical meaning using the threshold $\tau = 0.6$

GSI - value	meaning
$GSI \leq 0.6$	without disturbances
$0.6 < GSI \leq 1.2$	weakly disturbed
$1.2 < GSI \leq 1.8$	moderately disturbed
$1.8 < GSI$	heavily disturbed

Table 2. Classification accuracies obtained by the the several methods for training and test

data set	LDA	QDA	GLVQ	GRLVQ	LGRLVQ
training	71.39%	84.46%	83.12%	85.71%	87.28%
test	67.27%	82.13%	82.06%	85.54%	86.02%

the gender. Yet, the relation is that the higher the GSI-value the heavier the disturbances, in general, and the influence of age and gender is only marginally [1]. So it is neglected in our investigations. Roughly, one can identify several level classes of severity using the threshold $\tau = 0.6$, which are depicted in Tab.1. Thus the level ranges between $1 \ldots 4$ with inceasing disorder.

We compare LDA/QDA with GLVQ/GRLVQ/LGRLVQ whereby for the LVQ-algorithms 3 prototypes per class are used. The results are depicted in Tab. 2

We see that the metric adapted prototype based classifiers outperform LDA/QDA. Further, as expected, LGRLVQ achieves best accuracy for training as well as test. Thus LGRVQ can be seen as a very powerful non-linear alternative to QDA. Moreover, the good experimental behavior is accompanied by a mathematical counterpart putting the algorithm into the class of large margin optimizers.

References

1. C. Albani and G. Blaser and U. Rietz and T. Villmann and M. Geyer. Die geschlechtsspezifische Erfassung körperlicher Beschwerden bei PsychotherapiepatientInnen mit dem "Gießener Beschwerdebogen" (GBB). In A. Hinz and O. Decker (Eds.) Gesundheit im gesellschaftlichen Wandel. Psychosozial-Verlag, Gießen 2006.
2. P.L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning and Research* 3:463-482, 2002.
3. G.H. Franke. Möglichkeiten und Grenzen im Einsatz der Symptom-Check-List-90-R. *Verhaltenstherapie & psychosoziale Praxis* 33:475–485, 2001.
4. B. Hammer, M. Strickert, and T. Villmann. Supervised neural gas with general similarity measure. *Neural Processing Letters* 21:21–44, 2005.

5. B. Hammer, M. Strickert, and T. Villmann. On the generalization ability of GRLVQ networks. *Neural Processing Letters* 21:109–120, 2005.
6. B. Hammer and T. Villmann. On the generalization ability of localized GRLVQ. Technical Report *Clausthal University of Technology, Institute for Computer Science*, 2005.
7. T. Kohonen. Self-Organizing Maps. Springer-Verlag, 1997.
8. A.S. Sato and K. Yamada. Generalized learning vector quantization. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 423–429, MIT Press, 1995.
9. T. Villmann. Neural Maps for Faithful Data Modelling in Medicine - State of the Art and Exemplary Applications. *Neurocomputing* 48:229–250, 2002.
10. T. Villmann and G. Blaser and A. Körner and C. Albani. Relevanzlernen und statistische Diskriminanzverfahren zur ICD-10 Klassifizierung von SCL90-Patienten-Profilen bei Therapiebeginn. In G. Plöttner (ed.). Psychotherapeutische Versorgung und Versorgungsforschung. Leipziger Universitätsverlag, Leipzig, Germany, 2004.
11. V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications* 16(2): 264-280, 1971.

Genetically Evolved Trees Representing Ensembles

Ulf Johansson[1], Tuve Löfström[1], Rikard König[1], and Lars Niklasson[2]

[1] School of Business and Informatics, University of Borås, Sweden
ulf.johansson@hb.se, tuve.lofstrom@hb.se, rikard.konig@hb.se
[2] School of Humanities and Informatics, University of Skövde, Sweden
lars.niklasson@his.se

Abstract. We have recently proposed a novel algorithm for ensemble creation called GEMS (Genetic Ensemble Member Selection). GEMS first trains a fixed number of neural networks (here twenty) and then uses genetic programming to combine these networks into an ensemble. The use of genetic programming makes it possible for GEMS to not only consider ensembles of different sizes, but also to use ensembles as intermediate building blocks. In this paper, which is the first extensive study of GEMS, the representation language is extended to include tests partitioning the data, further increasing flexibility. In addition, several micro techniques are applied to reduce overfitting, which appears to be the main problem for this powerful algorithm. The experiments show that GEMS, when evaluated on 15 publicly available data sets, obtains very high accuracy, clearly outperforming both straightforward ensemble designs and standard decision tree algorithms.

1 Introduction

This paper focuses on predictive modeling, which is the task of learning a target function that maps each attribute set to an output value. If the output (target) value is continuous, the problem is referred to as *regression.* Predictive *classification*, on the other hand, deals with target values representing predefined class labels. Fig. 1 on the following page shows a schematic picture of predictive modeling. Here, data from both a data warehouse and operational databases is fed to the data mining algorithm. The data mining algorithm uses a score function to produce a model, which in turn is used on novel data (a *test* or *production* set) to produce the actual predictions.

More technically, the predictive model is a mapping from a vector input to a scalar output which is learnt from samples. The training data thus consists of pairs of measurements, each having an input vector $\mathbf{x}$(i) and a corresponding target value y(i). The predictive model is an estimation of the function y=f($\mathbf{x}$;Θ), used to predict a value y, given an input vector of measured values $\mathbf{x}$ and a set of estimated parameters Θ for the model f. The process of finding the best Θ values, using the score function, is the core of the data mining technique.

The primary goal, when performing predictive modeling, is, of course, to obtain models that are likely to generalize well; thus exhibiting high accuracy when

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 613–622, 2006.
© Springer-Verlag Berlin Heidelberg 2006

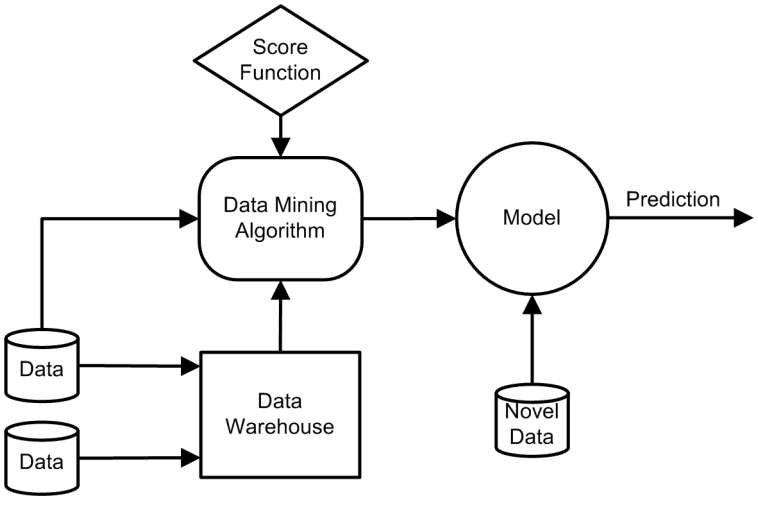


Fig. 1. Predictive modeling

applied to novel data. The two most common data mining techniques are, arguably, artificial neural networks (ANNs) and decision trees (DTs). ANNs are normally the technique of choice if there is no explicit demand requiring a transparent model. The motivation is that ANNs are known to often produce very accurate models in many diverse domains. Within the research community it is, however, a well-known fact that the use of ANN ensembles normally results in even higher accuracy; see e.g. [1] or [2]. Despite this, the use of ensembles in applications is still limited. Two possible reasons for this are insufficient knowledge about the benefits of using ensembles and limited support in most data mining tools. In addition, even when ensembles are used, very simple variants are often preferred. A typical choice would be to train exactly five (or ten) ANNs with identical topology and simply average the output. With this in mind, algorithms for constructing accurate ensembles should be of significant interest to both researchers and practitioners within the data mining community. The overall purpose of this paper is to extend and further evaluate a recently proposed technique for the creation of ANN ensembles, called GEMS (Genetic Ensemble Member Selection).

2 Background and Related Work

Although any algorithm for constructing ensembles must somehow determine ensemble members, the actual selection could be performed in many different ways. Standard techniques like bagging, introduced in [3], and boosting, introduced in [4], rely on resampling techniques to obtain different training sets for each of the classifiers. Another option is to train each classifier independently (most often using common data) and then either combine all classifiers or select a subset to form the actual ensemble.

Several approaches try to use genetic algorithms (GAs) to search for optimal ensembles. Zhou et al. [5] - [6] proposed an approach named GASEN, where several ANNs are trained before GAs are used to select an optimal subset of individual networks. In GASEN, the optimization is performed on individual ANNs and each ANN is coded (in the chromosome) as a real number denoting the benefit of including that ANN. The optimization criterion (the fitness) is rather technical, but boils down to accuracy on a hold-out set. Opitz and Shavlik [7] proposed a method called ADDEMUP, where the GA is used for creating new ANNs as parts of an ensemble. The size of the ensemble is predetermined and fixed. ADDEMUP uses a fitness function directly balancing accuracy against diversity, also using a hold-out set.

We have recently proposed and evaluated a novel, yet simpler, approach [9], also based on GAs. Here several individual ANNs are trained separately, on the same data set, and then GAs are used to directly find an accurate ensemble from these ANNs. The optimization is performed on ensembles and the fitness is directly based on ensemble accuracy on training and/or hold-out sets. The number of ANNs in the ensemble can vary since optimization is performed on the ensemble level. More specifically; each chromosome is represented as a bitstring where each gene corresponds to a specific ANN. As expected, a 1 would indicate that the specific ANN should be included in the ensemble.

In that study we also evaluated several, more basic ways, of creating ensembles. Although our novel technique finished on top, an interesting result was that some extremely straightforward approaches came very close. More specifically; if we trained 50 ANNs and used the ten ANNs with highest individual accuracy on a hold-out set to form the ensemble, this ensemble turned out to have almost as high accuracy on the test set as the one created using GAs. Another observation was that just combining all 50 ANNs into an ensemble most often also resulted in very accurate ensembles. All in all, the main conclusion was that almost all ensembles clearly outperform single models, but it is very hard to find a technique constantly producing ensembles more accurate than even the most straightforward ensembles.

In our latest study [8], we introduced the novel algorithm GEMS. GEMS consists of two steps, each requiring several design choices and parameters. In the first step of GEMS, a number of ANNs are trained and stored in a pool. Each ANN uses a localist (1-of-C) coding; the number of output units is thus equal to the number of classes. The activation level of the output units for a specific ANN is termed its *result vector*. In the second step, Genetic Programming (GP) is used to create the actual ensemble. When using GP, the ensembles are coded as (genetic) programs, each individual representing a possible combination of the available ANNs. More specifically; each ensemble is represented as a tree, where the internal nodes contain operators, while the leaves must be either ANNs from the pool or (random) constants. In the first study, GEMS used only two operators; FACT and AVG. FACT was used to multiply a result vector with a constant and AVG averaged the result vectors from its children. The actual classification from a GEMS ensemble is determined from the maximum value of

the top-level result vector. If, for example, the problem has three classes, every result vector would include three values (each representing a specific class) and the ensemble would make the prediction based on these values on the top level. It should be noted that GEMS in fact builds ensembles using a mix of smaller ensembles and single ANNs as building blocks. Fig. 2 shows a GEMS ensemble coded in the tree format described above. This very small, sample, ensemble uses only three ANNs and the result is the average of ANN3 (multiplied with a factor 0.8) and the average of ANN1 and ANN2.

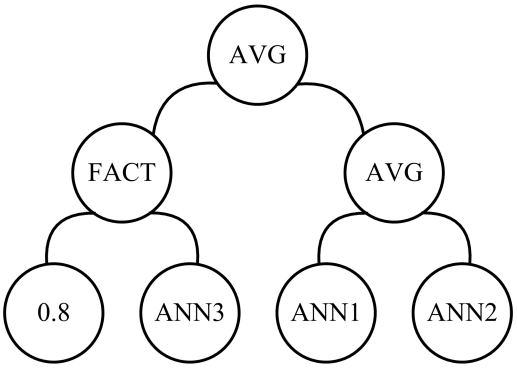


Fig. 2. GEMS ensemble

In the paper introducing GEMS [8], GEMS was evaluated in two different experiments, both using four data sets. The first experiment used a pool of 20 ANNs and the second 50 ANNs. In the experiments, GEMS was compared to the straightforward choice of choosing a fixed number of ANNs, based on hold-out set accuracy, from the mutual ANN pool. The GEMS fitness function was also based mainly on hold-out set accuracy. The main result was that GEMS, on two of the four data sets, was significantly more accurate than all fixed ensembles evaluated. The results on the other two data sets were, however, inconclusive since no statistically significant difference between the ensembles evaluated was found. To be more precise, GEMS in both experiments had almost identical accuracy to the best competing ensemble. Again it appeared almost impossible to constantly obtain ensembles significantly better than straightforward choices. Part of this is probably due to the fact that hold-out set accuracy does not seem to be the silver bullet it is often assumed to be. As a matter of fact, the standard procedure of using hold-out set accuracy when comparing models must be questioned. We all agree that the overall goal is to achieve high accuracy on unseen data, so naturally the best possible test seems to be to measure exactly that, accuracy on unseen data. This reasoning, however, has a devious shortcoming; the real issue is how a model chosen from accuracy on a hold-out set would perform on yet novel data. If we use a hold-out set to somehow choose one model over another, the underlying assumption must be that there is a high correlation between accuracy on that hold-out set and accuracy on

another set of unseen data; i.e. the actual test set. If this assumption does not hold, there is obviously little to gain from using a hold-out set as a basis for ensemble construction.

With this in mind, one very interesting observation was the fact that although GEMS consistently had much higher accuracy on the hold-out set (compared to the other ensembles), this property was not always preserved in the test set. Even though the fitness function used all available data (i.e. both training and hold-out data) and a length penalty was used to encourage smaller ensembles, the most probable explanation is that the GP, in spite of this, did overfit the test data. Just to iterate this important point; in the first study GEMS always had very high accuracy on the part of the data set covered by the fitness function, but this did not necessarily carry over to the production set. So, the main purpose of this study is to investigate if it is possible to avoid the effects of overfitting by making some alterations to GEMS.

3 Method

With the previous results in mind, we wanted to modify GEMS and the training scheme in several ways. First, we decided to dispose of hold-out sets altogether when constructing the ensembles. We believe that the best way of using the data set might be to use all available data for both ANN training and GP evolution. We also opted for using a micro technique to enforce some diversity among the networks. More specifically, each ANN was trained using only part of the available training data (here 70%) and the exact patterns were randomized for every ANN. In addition, we significantly altered some parameters, where all changes were aimed at reducing the risk of overfitting. More specifically; the pool now contained only 20 ANNs, GP was aborted after only 100 generations and evolution started with much less complex programs since the creation depth was reduced from eight to six. In addition, mutation was increased by a factor ten to introduce more variation in the population. The GP settings used are found in Table 1.

Table 1. GP parameters

Parameter	Value	Parameter	Value
Crossover rate	0.8	Elitism	Yes
Mutation rate	0.01	Persistence	25
Population size	500	Creation depth	6
Number of generations	100	Creation method	Ramped half-and-half

The most important difference, however, is that the representation language for GEMS was changed. Now the first levels of the tree contain splits (the conditions are expressed in original variables) similar to standard DTs. Function and

F = {*if, equals, <, >, avg, **}
T = {i_1, i_2, …, i_m, a_1, a_2, …, a_k, c_1, c_2, …, c_m, $\Re$}

DTree	**:-**	**(** ***if RExp Dtree Dtree*** **)** **|** ***BaseEns***
BaseEns	**:-**	***Avg*** **|** ***Ann***
Ens	**:-**	***Avg*** **|** ***Fact*** **|** ***Ann***
Avg	**:-**	**(** ***avg Ens Ens*** **)**
Fact	**:-**	**(** **** ConConst Ens*** **)**
Ann	**:-**	***Artificial neural network result vector***
RExp	**:-**	**(** ***ROp ConInp ConConst*** **)** **|** **(** ***equals CatInp CatConst*** **)**
ROp	**:-**	**<** **|** **>**
CatInp	**:-**	***Categorical input variable***
ConInp	**:-**	***Continuous input variable***
CatConst	**:-**	***Categorical attribute value***
ConConst	**:-**	$\Re$

Fig. 3. GEMS representation language

terminal sets together with the exact grammar for the representation language are given in Fig. 3. Using this grammar, GEMS becomes extremely flexible. Now, it is possible to mix tests partitioning the data with ensembles built using either individual ANNs or other ensembles. Fig. 4 shows a sample program in the syntax used by GEMS.

(***if*** **(** ***equals*** **X2 1.0)** **(** ***avg*** **ann:13 ann:10)** **(** ***avg*** **(** *** 0.3837 ann:8)** **(** ***avg*** **ann:10 ann:6)))**

Fig. 4. A sample ensemble in GEMS syntax

The fitness function used has two components: accuracy on the training set and a length penalty; see Eq. 1. The purpose of the length penalty is to encourage smaller ensembles to avoid complex and overly specialized ensembles.

$$f = \#correct\,in\,training\,set - \frac{1}{10}\,length \tag{1}$$

All ANNs in the pool are fully connected feed-forward ANNs. Of the 20 ANNs; five have no hidden layer, ten have one hidden layer and the remaining five have two hidden layers. The exact number of units in each hidden layer is slightly randomized, but is based on the number of inputs and classes in the current data set. For an ANN with one hidden layer, the number of hidden units is determined from Eq. 2.

$$h = \left\lfloor 2 \cdot rand \cdot \sqrt{v \cdot c} \right\rfloor \tag{2}$$

v is the number of input variables and c is the number of classes. rand is a random number in the interval [0, 1]. For ANNs with two hidden layers the number of units in each hidden layer is determined from Eq. 3 and Eq. 4.

$$h_1 = \left\lfloor \frac{\sqrt{v \cdot c}}{2} + 4 \cdot rand \cdot \frac{\sqrt{v \cdot c}}{c} \right\rfloor \quad (3)$$

$$h_2 = \left\lfloor rand \cdot \frac{\sqrt{v \cdot c}}{c} + c \right\rfloor . \quad (4)$$

3.1 Experiments

The 15 data sets used in this study are all publicly available from the UCI Repository [10]. One explicit goal of this study was to evaluate GEMS on a sufficiently large and diverse set of problems. For a summary of the characteristics; see Table 2.

To evaluate the performance of GEMS, we created two competing ensembles also built using the available ANNs. The first ensemble (called *S10*) uses the ten ANNs with highest individual training accuracy and the second (*all*) uses all 50 ANNs. In addition, results are also reported for the two standard decision tree algorithms C5 [11] and CART [12]. For the actual evaluation, we use standard ten-fold cross validation.

Table 2. Data set characteristics

Name	Abbreviation	Instances	Classes	Continuous	Categorical	Total
BUPA	BUP	345	2	6	0	6
CLEVE	CLE	303	2	6	7	13
CMC	CMC	1473	3	2	7	9
CRX	CRX	690	2	6	9	15
GERMAN	GER	1000	2	7	13	20
GLASS	GLA	214	7	9	0	9
LED7	LED	3200	10	0	7	7
PIMA	PIM	768	2	8	0	8
SAT	SAT	6435	6	36	0	36
SONAR	SON	208	2	60	0	60
TAE	TAE	151	3	1	4	5
TIC-TAC	TIC	958	2	0	9	9
WAVEFORM	WAV	5000	3	21	0	21
WBC	WBC	699	2	9	0	9
VEHICLE	VEH	846	4	18	0	18

4 Results

The results from the experiments are given in Table 3 on the following page.

As seen in Table 3 on the next page, all ensembles clearly outperform the two DT algorithms. Additionally, GEMS has the highest mean accuracy on 8 of 15 data sets. GEMS also has the highest overall mean accuracy; although an average of mean values from different data sets, is, admittedly a rather simplified measurement. Another very important observation, is the fact that GEMS' drop

in accuracy from the training set to the test set is much smaller, compared to the previous study, indicating that the problem with overfitting actually was reduced. To further compare the techniques, a series of standard pair wise t-tests over all ten folds was performed. The results are shown in Table 4.

Table 3. Accuracy on test set, mean values over 10 folds

Technique	C5	CART	S10		All		GEMS	
Data set	TestAcc	TestAcc	TrainAcc	TestAcc	TrainAcc	TestAcc	TrainAcc	TestAcc
BUP	.626	.640	.784	.733	.787	**.737**	.819	**.737**
CLE	.742	.765	.990	.806	.985	**.809**	.998	**.809**
CMC	.555	**.559**	.602	.534	.622	.543	.634	.544
CRX	**.871**	.842	.942	.861	.933	.848	.952	.849
GER	.716	.716	.956	**.734**	.937	.729	.958	.732
GLA	.632	.632	.838	.668	.835	.664	.884	**.708**
LED	.648	.652	.747	.731	.750	**.738**	.753	.734
PIM	.757	.758	.818	**.768**	.815	.765	.840	.767
SAT	.870	.863	.926	.904	.918	.897	.931	**.905**
SON	.789	.711	1.00	.826	1.00	.811	1.00	**.853**
TAE	.419	.488	.779	.550	.781	.531	.820	**.581**
TIC	.809	.827	.992	.913	.979	.888	.995	**.915**
WAV	.768	.766	.906	.864	.901	**.866**	.910	.855
WBC	.940	.933	.999	.964	.997	**.967**	1.00	.961
VEH	.724	.693	.918	.837	.903	.841	.933	**.851**
MEAN	**.724**	**.723**	.880	**.780**	.876	**.776**	.895	**.787**

Table 4. Statistically significant differences

Data sets	Results
GLA	GEMS significantly better than all other
SAT, TIC	GEMS and S10 significantly better than all other
BUP, CLE, LED, WBC, VEH	All ensembles significantly better than both DTs
TAE	GEMS significantly better than DTs
WAV	S10 and All significantly better than GEMS. GEMS significantly better than DTs
CMC, GER, PIM	No significant differences
SON	All ensembles significantly better than CART
CRX	C5 significantly better than all except S10

It should be noted that it is rather hard to obtain statistical significance using only ten observations (folds). Accordingly, we performed another series of experiments on Sonar, TAE and Vehicle. 40 runs on each data set were carried out; each using 80% of available data for ANN training and GP evolution and the remaining 20% for testing. The result, again using pair wise t-tests, was that GEMS, on these data sets, in fact had significantly higher accuracy than both S10 and All. One interesting observation is that many successful ensembles

turned out to be very small. As a matter of fact, most winning ensembles used a single split and, in addition, the two alternative ensembles were often also rather small.

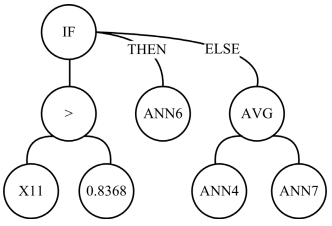


Fig. 5. Evolved GEMS ensemble (BUPA fold 10)

Fig. 5 shows one example, from the BUPA data set, where a single split determines if a specific ANN (ANN6) or a small ensemble (ANN4 and ANN7) should make the prediction.

5 Conclusions

The results of this study clearly illustrates GEMS' ability to create accurate classifiers (here ANN ensembles) for many diverse problems. The enhancements to GEMS introduced here successfully address the previously identified shortcoming of the original GEMS; i.e. the strong tendency to overfit data used for GP evolution. The most important change is the extension of the representation language to also include tests partitioning the data, further increasing flexibility. The excellent performance, coupled with the high flexibility, makes GEMS an interesting approach to develop further.

References

1. L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(10), pp. 993-1001, 1990.
2. A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems*, Volume 2, pp. 650-659, San Mateo, CA, Morgan Kaufmann, 1995.
3. L. Breiman. Bagging predictors, *Machine Learning*, 24(2), pp. 123-140, 1996.
4. R. Shapire. The strength of weak learnability, *Machine Learning*, 5(2), pp. 197-227, 1990.
5. Z.-H. Zhou, J.-X. Wu, Y. Jiang and S.-F. Chen. Genetic algorithm based selective neural network ensemble, *17th International Joint Conference of Artificial Intelligence*, vol. 2, pp. 797-802, Seattle, WA, 2001.
6. Z.-H. Zhou, J.-X. Wu, and W. Tang. Ensembling Neural Networks: Many Could Be Better Than All, *Artificial Intelligence*, vol. 137, no. 1-2, pp. 239-263, Elsevier, 2002.
7. D. Opitz and J. Shavlik. Actively searching for an effective neural-network ensemble, *Connection Science*, 8(3/4), pp. 337-353, 1996.

8. U. Johansson, T. Löfström, R. König and L. Niklasson. Introducing GEMS - a Novel Technique for Ensemble Creation, 19th Florida Artificial Intelligence Research Society Conference (FLAIRS) 06, Melbourne, FL, AAAI Press, 2006. To appear.
9. U. Johansson, T. Löfström and L. Niklasson, Obtaining Accurate Neural Network Ensembles, *International Conference on Computational Intelligence for Modelling Control and Automation - CIMCA'2005*, In Press.
10. C. L. Blake and C. J. Merz. *UCI Repository of machine learning databases*, University of California, Department of Information and Computer Science, 1998.
11. J. R. Quinlan, See5 version 2.02, http://www.rulequest.com, 2005.
12. L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*, Wadsworth International Group, 1984.

Sequential Classification Via Fuzzy Relations

Marek Kurzynski and Andrzej Zolnierek

Wroclaw University of Technology, Faculty of Electronics, Chair of Systems and Computer Networks, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
marek.kurzynski@pwr.wroc.pl, andrzej.zolnierek@pwr.wroc.pl

Abstract. In this paper there are developed and evaluated methods for performing sequential classification (SC) using fuzzy relations defined on product of class set and fuzzified feature space. First on the base of learning set, fuzzy relation in the proposed method is determined as a solution of appropriate optimization problem. Next, this relation in the form of matrix of membership degrees is used at successive instants of sequential decision process. Three various algorithms of SC which differ both in the sets of input data and procedure are described. Proposed algorithms were practically applied to the computer-aided recognition of patient's acid-base equilibrium states where as an optimization procedure the real-coded genetic algorithm (RGA) was used.

1 Introduction

In many pattern recognition problems there exist dependencies among patterns to be recognized. For instance, this situation is typical for character recognition, recognition of state in technological processes, image classification, medical diagnosis, to name only a few [1].

Among the different concepts and methods of using "contextual" information in pattern recognition, the Bayes decision theory is an attractive from the theoretical point of view and efficient approach. In this approach a classifying decision is made on one pattern at a time using information from the entire past and as a dependence model the Markov chain is adopted [2]. Unfortunately, this model has many disadvantages (e.g. to construct the effective decision rules the naive Bayes assumption should be made [4], it requires *a priori* knowledge of probability characteristics of compound statistical process) which seriously restrict its practical usefulness.

Methods developed in the field of computational intelligence such as neural networks, fuzzy logic and genetic algorithms are recently becoming increasingly popular in the pattern recognition as an attractive alternative to statistical approach. They can perform classification from both labeled and unlabeled training set as well as acquire and explore the human expert knowledge.

This paper presents a new approach to the problem of sequential classification which uses a fuzzy logic model. This method is based on a concept of fuzzy relation defined on Cartesian product of input data (appropriately formed feature space) and set of class numbers. This relation expressed as matrix of membership degrees can either be obtained from the experts directly or can be extracted from

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 623–632, 2006.
© Springer-Verlag Berlin Heidelberg 2006

available learning data containing a set of correctly classified patterns. Though there are several approaches to designing a classifier using the concept of fuzzy relation (e.g. [11], [13], [14], [15], [17]), the authors had focussed their attention on the classical recognition task without taking into account the "context", i.e. dependencies among the patterns to be recognized.

A specific feature of the explored classification task is the dependence between patterns at particular instants, which should be taken into account in SC algorithms. In other words, when constructing an appropriate decision algorithm we must not limit our approach to the narrow information channel that concerns just the features of current pattern but we have to consider all the available data instead, as they may contain important information about the class of pattern at a given instant. This methodological imperative applied to the fuzzy relation approach denotes new conceptual and technical problems which must be solved in the phase of classifier design. Particularly, it involves the determination of space of input data, the computation of fuzzy relation matrix (matrices) from learning data and its/their use in recognition procedure.

The rest of this paper is organized as follows. In Section 2 we introduce necessary background and formulate the problem of SC using fuzzy relations. Section 3 presents various concepts of SC which differ both in the sets of input data and classification procedure for particular instants of decision process. In Section 4 we discuss the results of application of proposed SC algorithms to computer-aided recognition of human acid-base equilibrium states.

2 Preliminaries and the Problem Statement

We will treat the SC as a discrete dynamical process. The object is at the n-th instant in the state $j_n \in \mathcal{M} = \{1, 2, \ldots, M\}$. The state j_n is unknown - what we can observe only is the indirect features by which a state manifests itself. We will denote a d-dimensional feature vector by $x_n = [x_n^{(1)}, x_n^{(2)}, ..., x_n^{(d)}] \in \mathcal{X}$, for features measured at the n-th instant (thus $\mathcal{X}$ is the feature space).

The current object state depends on the history and thus in the general case the decision algorithm must take into account the whole sequence of the preceding feature values, $\bar{x}_n = \{x_1, x_2, \ldots, x_n\}$. It must be underlined here that sometimes it may be difficult to include all the available data, especially for bigger n. In such cases we have to allow various simplifications (e.g. make allowance for only several recent values in the $\bar{x}_n$ vector), or compromises (e.g. substituting the whole object history segment that spreads as far back as the k-th instant, with data processed in the form of a decision established at that instant, say i_k).

Apart from the data measured for an object to be recognized, we need some more general information to take a valid classifying decision, namely the *a priori* knowledge concerning the general associations that hold between states (classes) and features. From now on we assume that this knowledge has the form of a so called training set, which in the considered decision task consists of m training sequences:

$$\mathcal{S} = \{S_1, S_2, ..., S_m\}, \tag{1}$$

where

$$S_k = ((x_{1,k}, j_{1,k}), (x_{2,k}, j_{2,k}), ..., (x_{N,k}, j_{N,k})) \tag{2}$$

denotes a sequence of feature observations and states (classes) for a learning object.

In consequence, the classification algorithm in the n-th step is of the following form:

$$\Psi_n(\bar{x}_n, \mathcal{S}) = i_n. \tag{3}$$

Obviously, the SC can be also treated as a sequence of single independent tasks without taking into account the associations that may occur between them. Such approach leads to the classical concept of recognition algorithm, which assigns a pattern at the nth instant to a class on the base of its features only, namely:

$$\Psi_n(x_n, \mathcal{S}) = i_n. \tag{4}$$

Application of fuzzy relation to the construction of classifier (4) from the learning set (1) containing $N \times k$ patterns (now the order of patterns in the sequences (2) is irrelevant) is well known in literature (see e.g. [11], [17], [13], [18]) and resulting procedure comprises the following items:

1. Cover the space $\mathcal{X}^{(l)}$ of the individual feature $x^{(l)}$ $(l = 1, 2, ..., d)$ by overlapping fuzzy sets corresponding to the linguistic "values" of this feature (e.g. *small*, *medium*, *big*, etc.). For each fuzzy set define its membership function. Obtained fuzzy sets state fuzzified feature space $\mathcal{X}_F^{(l)}$ of individual features. Create fuzzified feature space as a product $\mathcal{X}_F = \mathcal{X}_F^{(1)} \times \mathcal{X}_F^{(2)} \times ... \times \mathcal{X}_F^{(d)}$. Let its cardinality be equal to d_F - this value depends on number of partitions and the size of feature vector. For example, in the further practical medical diagnosis task, $d = 3$ and we used triangular fuzzy numbers with 3 regular partitions [3], which gave $d_F = 27$.
2. Determine observation matrix $O(\mathcal{S})$ of learning set $\mathcal{S}$, i.e. fuzzy relation defined on product of fuzzified feature space $\mathcal{X}_F$ and learning set $\mathcal{S}$. The ith row of $O(\mathcal{S})$ $(i = 1, 2, ..., N \times k)$ contains membership degrees of features of ith learning pattern to fuzzy sets of space $\mathcal{X}_F$. The number of columns of $O(\mathcal{S})$ is equal to d_F.
3. Determine decision matrix $D(\mathcal{S})$, i.e. relation defined on product of learning set $\mathcal{S}$ and the set of decisions (classes) $\mathcal{M}$. For the training data, where the classification is exactly known, the ith row is a fuzzy singleton set, i.e. a vector of all zeros except for a one at the place corresponding to the class number of ith learning pattern.
4. Find matrix $E(\mathcal{S})$ as a solution of so-called *fuzzy relational equation* ([11], [13]):

$$O(\mathcal{S}) \circ E(\mathcal{S}) = D(\mathcal{S}), \tag{5}$$

or - in approximate way - as a solution of the following optimization problem:

$$\rho(O(\mathcal{S}) \circ E(\mathcal{S}), D(\mathcal{S})) = min_E\, \rho(O(\mathcal{S}) \circ E, D(\mathcal{S})), \tag{6}$$

where criterion $\rho(A, B)$ evaluates difference between matrices A and B, i.e. $\rho(A, B) \geq 0$ and $\rho(A, B) = 0$ iff $A = B$. Operator $\circ$ denotes here max-min-norm composition of relations, i.e. multiplication of matrices O and E with $\times$ and $+$ operators replaced by min and max operators (more general by t-norm and s-norm operators)([7]). In the further practical example we decided to select the method of determination of matrix E, adopting

$$\rho(A, B) = \sum_{i,j} (a_{ij} - b_{ij})^2 \tag{7}$$

and applying as an optimization procedure real-coded genetic algorithm.

Matrix $E(\mathcal{S})$ is a fuzzy relation defined on product of decision set $\mathcal{M}$ and feature space $\mathcal{X}_F$, in which reflects knowledge contained in the learning set. To classify a new pattern x, first the row-matrix of fuzzy observation $O(x)$ is calculated from known vector of its features $[x^{(1)}, x^{(2)}, ..., x^{(d)}]$. Then matrix $E(\mathcal{S})$ is applied to compute an output row-matrix called *target vector* ([17]):

$$O(x) \circ E(\mathcal{S}) = T(x) = [t_1(x), t_2(x), ..., t_M(x)], \tag{8}$$

which gives a fuzzy classification in terms of membership degrees $t_i(x)$ of the pattern x to the given classes $i = 1, 2, ..., M$. When a crisp decision is required, defuzzification has to be applied, typically according to the maximum rule.

In the next section we will apply this procedure to the construction of SC algorithm (3), i.e. taking into account dependencies among patterns to be recognized.

3 Algorithms of Sequential Classification

Although, the main concept of the proposed methods of SC is the same as for independent patterns, there are many differences concerning details in procedure of construction of matrix E and the course of recognition process.

Three procedures have been proposed which differ exclusively in the relevant selection of input data. The first algorithm includes k-instant-backwards-dependence ($k < N$) with full measurement data. It means, that decision at the nth instant is made on the base of vector of features

$$\bar{x}_n^{(k)} = (x_{n-k}, x_{n-k+1}, ..., x_{n-1}, x_n). \tag{9}$$

In the second approach however, we also include k-instant-backward-dependence, but using the previous decisions in lieu of the features (9).

Before we will describe these algorithms let us first introduce set $\mathcal{S}^{(k)}$ containing sequences of $(k+1)$ learning patterns from $\mathcal{S}$ and set $\mathcal{S}^{(k)}_{\bar{j}^{(k)}}$ - as previously but in which at the first k position additionally the sequence of classes $\bar{j}^{(k)} \in \mathcal{M}^k$ appears.

3.1 Algorithm with *k*th Order Dependence (AkD)

The algorithm with the full measurement features can be presented according to the following points:

1. Create the fuzzified feature space $\mathcal{X}_F$ as in the procedure for independent patterns (see Section 2.)
2. Determine observation matrix $O^{(k)}$, i.e. fuzzy relation in the space $\mathcal{X}_F^k = \mathcal{X}_F \times \mathcal{X}_F \times \cdots \times \mathcal{X}_F$ (k times) and learning subset $\mathcal{S}^{(k)}$. The ith row of observation matrix contains memberships degrees of features $\bar{x}^{(k)}$ of ith learning sequence from $\mathcal{S}^{(k)}$ to the fuzzy sets of space $\mathcal{X}_F^k$.
3. Determine decision matrix $D^{(k)}$, i.e. relation defined on product of learning sequences $\mathcal{S}^{(k)}$ and the set of decisions (classes) $\mathcal{M}$. The ith row of $D^{(k)}$ is a vector of all zeros except for a one at the place corresponding to the last class number of ith sequence in the set $\mathcal{S}^{(k)}$.
4. Find matrix $E^{(k)}$, so as to minimize criterion

$$\rho(O^{(k)} \circ E^{(k)}, D^{(k)}). \tag{10}$$

   Matrix $E^{(k)}$ is a fuzzy relation defined on product of decision set $\mathcal{M}$ and feature space $\mathcal{X}_F^k$.

The manner of applying the matrix $E^{(k)}$ for decision making is obvious. At the nth step of sequential recognition first the row-matrix of fuzzy observation $O(\bar{x}_n^{(k)})$ is calculated from known sequence of feature observations (9). Then matrix $E^{(k)}$ is applied to compute a target vector of soft decisions:

$$O(\bar{x}_n^{(k)}) \circ E^{(k)} = T(\bar{x}_n^{(k)}), \tag{11}$$

and final crisp decision is obtained after defuzzification step.

It must be emphasized that proposed procedure leads to the very flexible sequential recognition algorithm due to optional value of k. In particular, the value of k need not be constant but it may dynamically change from step to step. It means next, that choice $k = n - 1$ for nth instant of sequential classification denotes the utilization of the whole available information according to the general form of decision rule (3). On the other side however, such concept - especially for bigger n - is rather difficult for practical realization.

3.2 Reduced Algorithm with *k*th Order Dependence and Crisp History (RkDC)

In this approach for classification at the nth instant, we substitute the whole object history segment which - as previously - covers the k last instances, i.e. $(x_{n-k}, x_{n-k+1}, ..., x_{n-1})$ values with data processed in the form of decisions established at these instances, say

$$\bar{i}_n^{(k)} = (i_{n-k}, i_{n-k+1}, ..., i_{n-1}). \tag{12}$$

Such a concept significantly simplifies the computational procedure since - the sequence of previous decisions plays plays the role of peculiar "switch" which allows to select appropriate fuzzy relation (matrix). Thus now, we determine the set of matrices $E^{(k)}_{\bar{i}^{(k)}}$ (for different sequences (12)) using identical procedure as previously in which set $\mathcal{S}^{(k)}$ is replaced with set $\mathcal{S}^{(k)}_{\bar{i}^{(k)}}$ and fuzzified feature space $\mathcal{X}^k_F$ with simply $\mathcal{X}_F$.

As a consequence for each sequence of decisions $\bar{i}^{(k)}$, on the base of learning sequences (1) via optimization procedure (6) the matrix $E^{(k)}_{\bar{i}^{(k)}}$ is determined. Next applied in the formula

$$O(x_n) \circ E^{(k)}_{\bar{i}^{(k)}_n} = T(x_n, \bar{i}^{(k)}_n) \tag{13}$$

leads to the vector of soft classifications at the nth instant and then after maximum defuzzification procedure, to the crisp result.

3.3 Reduced Algorithm with *k*th Order Dependence and Soft History (RkDS)

In the RkDC algorithm with crisp history, matrix E was univocally determined by observed sequence of previous diagnoses (12). In the concept of algorithm with soft history however, we take into account soft decisions at previous instances, i.e. sequence of decisions for previous instances before defuzzification procedure (target vectors T containing membership degrees for particular classifications) instead of sequence (12) of crisp decisions.

Let

$$T_{n-i} = (t^{(1)}_{n-i}, t^{(2)}_{n-i}, ..., t^{(M)}_{n-i}), \tag{14}$$

be the vector of membership degrees for all classes produced by classification algorithm at the $(n-i)$th instant $(i = 1, 2, ..., k)$.

In the RkDS algorithm at the nth instant we replace in (13) matrix $E^{(k)}_{\bar{i}^{(k)}_n}$ for observed sequence of previous decisions (12) with the weighted sum of matrices for all possible sequences $\bar{i}^{(k)}_n \in \mathcal{M}^k$, viz.

$$O(x_n) \circ \sum_{\bar{i}^{(k)}_n \in \mathcal{M}^k} w_{\bar{i}^{(k)}_n} \times E^{(k)}_{\bar{i}^{(k)}_n} = T(x_n, \bar{i}^{(k)}_n), \tag{15}$$

where weight coefficients are equal to product of elements of vectors (14) corresponding to the elements of vector (12), namely

$$w_{\bar{i}^{(k)}_n} = t^{(i_{n-k})}_{n-k} \cdot t^{(i_{n-k+1})}_{n-k+1} \cdot ... \cdot t^{(i_{n-1})}_{n-1}. \tag{16}$$

4 Practical Example and Concluding Remarks

All the decision algorithms that are depicted in the previous section have been experimentally tested in respect of the decision quality (frequency of correct

classifications) for real data that are concerned with recognition of human acid-base equilibrium states (ABE).

The ABE states diagnosis treated as a classification task contains 5 classes (patient's states): metabolic acidosis, respiratory acidosis, metabolic alkalosis, respiratory alkalosis, correct state and decision is made on the base of vector of 3 features (gasometric examinations): the pH of blood, the pressure of carbon dioxide, the current dioxide concentration.

Experiments have been worked out on the basis of evidence material that was collected in Neurosurgery Clinic of Medical Academy of Wroclaw and constitutes the set of training sequences (1). The material comprises 78 patients (78 sequences) with ABE disorders caused by intracranial pathological states for whom the data were regularly put down on the 12-hour basis. There were around 20 examination cycles for each patient, yielding the total of 1486 single examination instances.

In order to find matrix E the genetic algorithm was applied, which is a popular and powerful search technique. It is based on ideas borrowed from the theories of natural selection - the "survival of the fittest" ([9]). The genetic algorithm was proceeded as follows:

- *Coding method* - Although binary representation has been widely used for GA analysis, in recent years many researchers have been concentrated on the use of real-coded GA (RGA). It is robust, accurate and efficient approach because the floating-point representation is conceptually closest to the problems with real optimization parameters. The chromosome used by the RGA is a string of floating-point numbers (real-valued genes) of the same length as the solution vector. It means, that in our task, the elements of matrix $E = [e_{ij}]_{d_F \times M}$ were directly coded to the chromosome, namely:

$$C = [e_{11}, e_{12}, ..., e_{1M}, e_{21}, e_{22}, ..., e_{2M}, ..., e_{d_F M}] = [c_1, c_2, ..., c_L], \tag{17}$$

  where M is a number of classes and d_F denotes dimension of feature space.
- *The fitness function* - Each chromosome was evaluated based on the following fitness function:

$$Fit = \frac{1}{1 + \rho(A, B)}, \tag{18}$$

  where ρ is as in (7).
- *Initialization* - GA needs an initial individual population to carry out parallel multidirectional search of optimal solution. The RGA starts with constructing an initial population of individuals generated randomly within the search space. Since each chromosome contains L genes which directly correspond to the elements of matrix E, the generated random numbers have a range $[0, 1]$. The size of population after trials was set to 60.
- *Selection* - The probability of selecting a specific individual can be calculated by using the individuals fitness and the sum of population fitness. In this research a roulette wheel approach was applied. Additionally, an elitism policy, wherein the best individual from the current generation is copied directly to the next generation, was also used for faster convergence.

– *Crossover* - The crossover process defines how genes from the parents have been passed to the offspring. In experiments both arithmetic and directional crossover [10] were used and in each generation one of them was selected randomly. If C_p^1 and C_p^2 are parent chromosomes, then we get offspring chromosomes C_c^1 and C_c^2 according to the following formulas (α is a random number uniformly distributed in $[0, 1]$):

$$C_c^1 = \alpha \cdot C_p^1 + (1 - \alpha) \cdot C_p^2, \quad C_c^2 = \alpha \cdot C_p^2 + (1 - \alpha) \cdot C_p^1, \tag{19}$$

for arithmetic crossover,

$$C_c^1 = \alpha \cdot (C_p^1 - C_p^2) + C_p^1, \quad C_c^2 = \alpha \cdot (C_p^2 - C_p^1) + C_p^2, \tag{20}$$

for directional crossover.

These crossover procedures represent a promising way for introducing a correct exploration/exploitation balance in order to avoid premature convergence and reach approximate final solution.

– *Mutation* - Mutation is carried out by randomly perturbing genes of chromosomes after crossover. This operator provides GA with a mechanism for escaping local maxima and for protection against premature convergence. The mutation used in experiments is the random (uniform) procedure ([8]), i.e.:
  - an individual (in offspring population) is randomly selected,
  - a mutation site is randomly fixed in the interval $[1, L]$,
  - the selected gene (value) c_i is replaced by $\bar{c}_i$, randomly generated from $[0, 1]$ interval with uniform distribution.

  The probability of mutation was equal to 0.01.

– *Stop procedure* - evolution process was terminated after 1000 generations. In fact, the fitness value usually converged within this value. Fig. 1. shows the fitness change against generation number in one run of RGA.

To compare the classification accuracy of proposed concepts of SC algorithms and the performance of RGA, ten independent runs of RGA were carried out for each diagnostic algorithm with different random initial populations. The results are shown in Table 1. The values depicted in the Table are those of the best

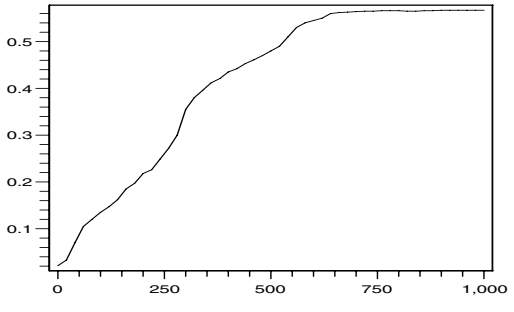


Fig. 1. The example of the course of the fitness value vs. number of generation

Table 1. Frequency of correct diagnosis for various diagnostic algorithms (in per cent)

Trial	A0D	A1D	A2D	R1DC	R2DC	R1DS	R2DS
1	80.6	89.6	91.8	85.1	85.7	82.7	90.3
2	82.2	86.5	91.9	82.7	89.6	88.6	86.7
3	79.4	87.2	88.9	89.3	85.3	85.9	85.8
4	78.5	85.9	92.6	87.2	85.7	83.8	90.4
5	80.9	90.3	91.9	82.5	85.4	84.2	84.5
6	82.1	89.7	91.6	86.8	89.8	89.1	83.8
7	81.9	88.1	89.4	85.2	90.6	82.8	90.4
8	78.3	87.2	89.0	87.9	86.1	83.3	89.7
9	78.5	90.7	92.9	84.3	89.9	87.5	90.8
10	81.1	88.7	92.8	88.1	89.2	88.3	89.6
Best	82.2	90.7	92.9	89.3	90.6	89.1	90.8
Mean	80.3	88.4	91.3	85.9	87.7	85.6	88.2
SD	1.44	1.58	1.49	2.20	2.12	2.43	2.57

solution obtained at the end of a RGA trial. Table 1 contains also the best result, the mean value and standard deviation for each SC algorithm.

These results imply the following conclusions:

1. Algorithm A0D that does not include the inter-state dependencies and treats the sequence of states as independent objects is worse than those that have been purposefully designed for the sequential medical diagnosis task, even for the least effective selection of input data. This confirms the effectiveness and usefulness of the conceptions and algorithm construction principles presented above for the needs of sequential diagnosis.
2. There occurs a common effect within each algorithm group: the model of the second order dependency (A2D, R2DC, R2DS) turns out to be more effective than the first order dependence approach (A1D, R1DC, R1DS).
3. Algorithms A1D and A2D that utilize the original data (i.e. gasometric examinations) yield always better results than those which substitute the data with diagnoses.
4. There is no essential difference between the algorithms with crisp and soft history.
5. The RGA method is capable of solving the problem of learning of SC algorithm for practical computer-aided medical diagnostic system. Results of RGA performances turn out to be quite repeatable and insensitive to initial conditions.

References

1. Toussaint G.: The Use of Context in Pattern Recognition. Pattern Recognition, Vol. 10 (1978) 189-204
2. Kurzynski, M.: Benchmark of Approaches to Sequential Diagnosis. In: Lisboa P., Ifeachor, J., Szczepaniak P. (eds.): Perspectives in Neural Computing. Springer-Verlag, Berlin Heidelberg New York (1998) 129-140

3. Kurzynski M.: Multistage Diagnosis of Myocardial Infraction Using a Fuzzy Relation. In: Rutkowski L., Tadeusiewicz R. (eds): Artificial Intelligence in Soft Computing, Lecture Notes in Artificial Intelligence, Vol. 3070, Springer-Verlag, Berlin Heidelberg New York (2004) 1014-1019
4. Zolnierek A.: The Empirical Study of the Naive Bayes Classifier in the Case of Markov Chain Recognition Task. In: Kurzynski M., Wozniak M. (eds.): Computer Recognition Systems CORES 05, Springer-Verlag, Berlin Heidelberg New York (2005) 329-336
5. Devroye L., Gyorfi P., Lugossi G.: A Probabilistic Theory of Pattern Recognition. Springer-Verlag, Berlin Heidelberg New York (1996)
6. Duda R., Hart P., Stork D.: Pattern Classification. John Wiley and Sons, New York (2001)
7. Czogala E., Leski J.: Fuzzy and Neuro-Fuzzy Intelligent Systems. Springer-Verlag, Berlin Heidelberg New York (2000)
8. Michalewicz Z.: Genetic Algorithms + Data Structure = Evolution Programs. Springer Verlag, New York (1996)
9. Goldberg D.: Genetic Algorithms in Search, Optimization and Machine Learning. Adison-Wesley, New York (1989)
10. Herrera F., Lozano M.: Gradual Distributed Real-Coded Genetic Algorithm. IEEE Trans. on Evolutionary Computing, Vol. 4 (2000) 43-63
11. Pedrycz W.: Fuzzy Sets in Pattern Recognition: Methodology and Methods. Pattern Recognition, Vol. 23 (1990) 121-146
12. Pedrycz W,: Genetic Algorithms for Learning in Fuzzy Relation Structures. Fuzzy Sets Syst., Vol. 69 (1995) 37-45
13. Ray K., Dinda T.: Pattern Classification Using Fuzzy Relational Calculus. IEEE Trans. SMC, Vol. 33 (2003) 1-16
14. Dinola A., Pedrycz W., Sessa S.: Fuzzy Relation Equations Theory as a Basis of Fuzzy Modelling: An Overview. Fuzzy Sets Syst., Vol. 40 (1991) 415-429
15. Ovchinnikov S., Riera T.: On Fuzzy Classifications. Fuzzy Sets Syst., Vol. 49 (1992) 119-132
16. Gottwald S.: Approximately Solving Fuzzy Relation Equations: Some Mathematical Results and Some Heuristic Proposals. Fuzzy Sets Syst., Vol. 66 (1994) 175-193
17. Setnes M., Babuska R.: Fuzzy Relational Classifier Trained by Fuzzy Clustering. IEEE Trans. on SMC, Vol. 29 (1999) 619-625
18. Acharya U., et al.: Classification of Heart Rate Data Using Artificial Neural Network and Fuzzy Equivalence Relation. Pattern Recognition, Vol. 36 (2002) 61-68

Attention Improves the Recognition Reliability of Backpropagation Network*

Zbigniew Mikrut and Agata Piaskowska

AGH University of Science and Technology, Institute of Automatics,
al.Mickiewicza 30, 30-059 Krakow, Poland
zibi@agh.edu.pl

Abstract. In the paper a method is presented for improving the recognition reliability of backpropagation-type networks, based on the attention shifting technique. The mechanism is turned on in cases when the reliability of the network's answer is low. The signals reaching the hidden layer are used for selection of image areas which are the most "doubtful" in the process of recognition by the network. Three methods have been proposed for appending the input vector after shifting the area where the attention is focused. The methods have been tested in the problem of hand-written digits recognition. Noticeable improvement of the recognition reliability has been obtained.

1 Introduction

Image recognition making use of the selective attention method is one of the most interesting methods, based on the biological visual systems. In the majority of models based on application of that technique the so called saliency maps are created. The choice, location and feature analysis for such maps depend on many factors, like the type of surrounding scene (background) or the color, contrast and orientation of a given feature [1]. Normally such features are analyzed in sequence: the attention shift to the next feature depends on the interpretation of the previous features. Another problem is the necessity for introducing the "blockade", i.e. avoiding places that have already been analyzed in the attention shifting process.

Examples of such realization can be found in the literature. In the paper [2] a hierarchical system has been presented for words recognition, based on the features of individual characters. The features of individual characters (detection of lines with various slopes, located within a rectangular matrix), forming level 1 of the network, influence the recognition of consecutive characters (level 2). These in further turn condition the recognition of words on level 3, the highest. These three levels are related by both top-down and bottom-up connections. The words from the top level are connected with the letters forming the given word, and the letters are connected with their characteristic features. This is also visible in the reverse direction: the features are connected with the respective letters, and the letters are connected with words, in which they are present. At the

* This work has been partially supported by the AGH UST grant No 10.10.120.39.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 633–642, 2006.
© Springer-Verlag Berlin Heidelberg 2006

beginning of the recognition process all nodes at the words level are active and the same probabilities (summing up to one) are attributed to individual nodes. The downward information flow models the individual probabilities for each letter and each feature. The recognition process consists of cycles, which are completed when one of the nodes is attributed probability equal to one, and all the other nodes - zero. In the first cycle one feature is investigated and on that basis the individual probabilities of encountering specific letters and words are modified. Some words are completely excluded. Then the information flows down and some letters and features are also excluded. After modification of the probabilities the attention is focused on the feature which is responsible for the greatest ambiguities: the algorithm determines the feature the presence or absence of which, affects most seriously the progress speed. The authors show that the recognition is achieved faster in comparison to the cases in which the whole word is being analyzed (in the average it takes 5.3 cycles for a database including 950 words).

In the papers [3][4] the saliency map is created in a similar way, and the detected features are analyzed in sequence. In the paper [3] other features like intensity, color or symmetry are also taken into account. Additionally the features that have lost their "importance" status during the analysis are removed from the saliency map - and for that task a modified ART neural network is used. In the paper [5] a neural network deals with both searching and recognition of the object. In addition to classical connections between the network layers there are connections within the layers and feedback connections to the input layer. The respective signals can be also taken from outside of the network. Some of the above-mentioned authors point out the necessity to include also the reduced resolution matrix representation in the recognition process.

The approach proposed in the present work differs from the ones presented above. The authors have decided to model the attention shifting process on the basis of two well-known mechanisms:

- well known and properly looking objects (e.g. letters or digits) are being recognized by humans "at a glance",
- doubts emerge, when the object is somehow different from the standard look (e.g. a digit has been written carelessly) - then the attention is focused on the areas, which contain features differing the objects being mixed up.

These two general rules have been implemented in a backpropagation type neural network, with the intended task to learn the recognition of hand-written digits (see figure 1). The accuracy of reproduction of the digit images has been reduced. The operation has been applied to both the learning and test sets images.

During recognition of the test set images the recognition reliability (see figure 1 - two small grey arrows, indicating the differences between two highest network outputs) has been monitored. If its value was too low the network switched on the mechanism of attention shifting to the doubtful image areas (in figure 1 - area a of digit "1" and α of digit "7"). The respective areas were reproduced with more accuracy and then the recognitions were redone in order to check

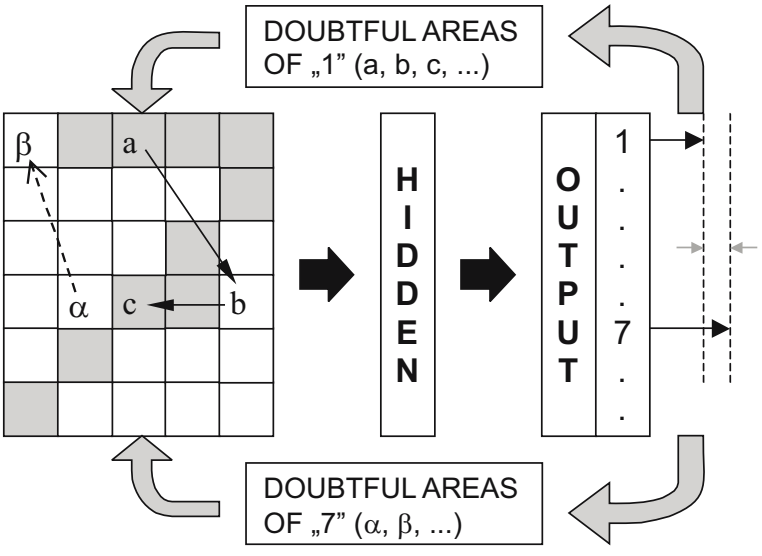


Fig. 1. The working principle of the mechanism of attention shifting based on the recognition results of the input character matrix elaborated by a backpropagation type network

whether the recognition reliability was improved enough. If the answer was "Yes" the recognition process was terminated. If the answer was "No" the attention focusing point was shifted to another image area (in figure 1 - areas *b*, *c* of digit "1" and β of digit "7") and the recognition process was repeated - until positive results were obtained or the specified number of iterations was reached. Obviously during the recognition process, or the iterative attention shifting and the related modification of the input information, the network was allowed to change its decisions.

2 Methods and Algorithms

In the present paper the main stress has been put on defining and testing of several methods responsible for modification of the network's input, during the attention shifting process. Therefore it has been decided that the basic platform for the experiments will be a simple neural network, with one hidden layer, trained by the fast Resilient Backpropagation algorithm.

A simple mechanism has been also determined for the attention shifting process. From the correctly recognized characters of the learning set averaged images have been created separately for each class. In combination with some of the network connection weights they provided the representation of the knowledge. For determination of consecutive attention shifting points the connection weights between the input and hidden layer have been used. The connections

between the hidden and output layer have not been taken into account. Such a solution limits the operation accuracy but it also drastically reduces the numerical complexity.

The iterative mechanism of the attention shifting is turned on when:

$$|y_r - y_q| \leq delta \tag{1}$$

where y_r and y_q are two highest signals of the network's output layer, and *delta* is the recognition reliability threshold, specified by the user. For the two characters related to the r and q indices two n-dimensional vectors are calculated (where n is the number of network inputs), which determine the contributions from the consecutive network inputs, which sum up to form the total signals stimulating the hidden layer elements:

$$R_r = \left[\left| (\bar{x}_1^r - x_1^r) \sum_{i=1}^{m} w_{1,i} \right| \ldots \left| (\bar{x}_n^r - x_n^r) \sum_{i=1}^{m} w_{n,i} \right| \right] \tag{2}$$

$$R_q = \left[\left| (\bar{x}_1^q - x_1^q) \sum_{i=1}^{m} w_{1,i} \right| \ldots \left| (\bar{x}_n^q - x_n^q) \sum_{i=1}^{m} w_{n,i} \right| \right] \tag{3}$$

where x are the respective input signals (averaged and current), w - connection weights between the inputs and the hidden layer elements, and m is the size of the hidden layer. Then the vectors are sorted in descending order. After reordering the indices of consecutive elements point to the areas, to which the attention should be sequentially shifted, with the accompanying modification of respective inputs. In order to realize the above procedure three methods, described below, have been proposed. Two of the methods are based on transforming the character image to the log-polar form, the third one - on the classical raster representation.

2.1 Log-Polar Transform

The log-polar transform [6] is a very simplified model of the arrangement of receptive fields in human retina. It is characterized by inhomogeneous reproduction of details: it is very accurate near the center and with the growing distance from the center the receptive fields become bigger and therefore the details become blurred (averaged). The receptor fields are limited by concentric circles with exponentially growing radii and split into angular sectors by radial lines (rays), dividing the 360^o angle into equal parts. Additionally there is a "blind-spot" in the center. The image part covered by the blind-spot area is not transformed into the log-polar space.

In figure 2 an example is shown of transforming the image of digit "5" to the log-polar form. The main advantage of the log-polar transform is the considerable reduction of the input data volume which however preserves the rough information about the object. After transforming a small raster image 139x70 (9730 pixels) to the form consisting of 9 rings and 24 sectors (216 pixels) the achieved compression is 45:1. As can be seen in the right-hand part of figure 2

(the inverse transform) part of the information is lost: it is therefore a lossy compression and it is hardly useful for image compression but it can be plausible in image recognition. The reduction of input data volume is particularly important when neural networks are used, because the construction and even more the learning of an oversized network is a task of high calculational complexity.

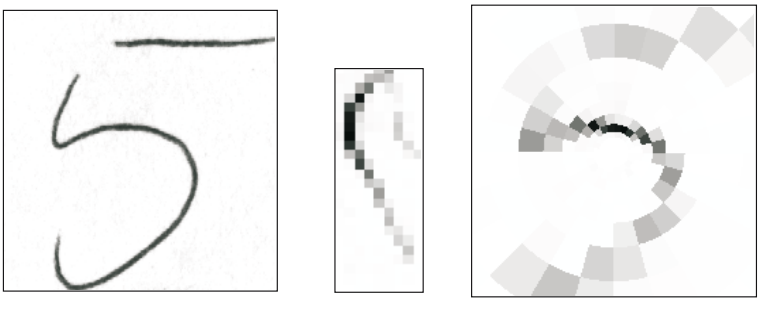


Fig. 2. Original image matrix, the image after log-polar transform (24x9 pixels) and the inverse transform (the log-polar image has been increased)

Additional advantage of that transform is the fact that two operations: input image rotation and its rescaling with respect to the transform center are realized in the log-polar space by parallel object translations along respective axes.

In spite of its multiple advantages the log-polar transform exhibits also some limitations. As mentioned before the transformation involves a lossy compression, and as a result inaccurate mapping of details located far from the center. The next problem is the loss of information contained near the transform center. This is an essential flaw, because the main feature of the transform is the fact that the area near the center of the image contains the best reproduced and sometimes most essential information. A direct consequence of that problem is another problem: the central point of the transform must be chosen very carefully as even a minor object shift with respect to the center generates a considerable change in the resulting output image.

The log-polar transformation was an inspiration for several practical applications: from realization of the object tracking tasks [7] through the design of a photo-sensitive chip [8] to the generation of object representations [9][10].

2.2 Method 1 – Moving the Log-Polar Center

In the discussed method the shifting of the attention focusing point is realized by shifting the center of the log-polar transform to the receptive field on which the attention should be focused. Then a new log-polar transform is calculated, in which the examined receptive field is reproduced more accurately, because now it is located near the transform center. The new values are determined from the averaged values of the "new" receptive fields, whereas the zero values are replaced by the values from the previous transform.

2.3 Method 2 – Changing Resolution of the Log-Polar Transform

Shifting the transformation center is numerically cumbersome, therefore in the second method it has been decided to leave the center where it was. Instead a four-fold increase of the resolution has been executed (doubling the number of the concentric circles, and doubling the number of rays dividing the receptive fields). The averaging over the new fields, located within the previous receptive field, is done in order to avoid the edge blurring: the maximum values are determined in the individual new fields, and then they are averaged with the neighboring fields, which usually exhibit comparable values.

2.4 Method 3 – Changing Resolution of the Image Matrix

In method 3 the log-polar transform has been abandoned in favor of input image scaling, using bicubic interpolation. Similarly to method 2 the shift of the attention focusing point is realized by carrying out the analysis of the interesting area magnified four times (twice in vertical direction and twice in horizontal). Averaging to the original matrix is realized in the same way as in method 2, what means that if there are bright spots within the analyzed field then the general intensity is increased (the same is true in the reverse direction).

3 Experiments and Results

In order to test the above-mentioned algorithms several backpropagation networks, with sigmoidal transfer function, have been built and learned. The networks were different in the size of the hidden layer (between 7 and 30 neurons) as well as initial conditions and learning parameters. The set used for learning and testing was a set of 4000 hand-written digits, collected during research work carried out in the Biocybernetic Laboratory. It contained images in 256 greyscale levels, with matrix sizes varying between 100x100 and almost 1000x1000 pixels. The set has been divided into two equal parts: learning and testing. Each part consisted of 200 representations of each character.

During the initial experiments the influence has been studied of the log-polar representation form on the results of the networks training and recognition. Log-polar representations of various sizes have been designed: the changes affected number of sectors (24-30), number of rings (8-10) and the minimum transformation radius (the dimensions of the blind spot). Additionally two possible locations of the transformation centers have been tested: in the character's center of gravity and in the geometrical center of character's outline. Better effectiveness has been achieved for a log-polar transform with the center located in the center of gravity of the analyzed character. The character representations obtained from such transform exhibited more reproducible shapes. It has also turned out that further increase of the matrix resolution in the log-polar coordinates is useless, because the recognition quality obtained for matrices greater than 30x10 pixels is worse than the results obtained for smaller matrices.

Finally two networks have been selected. Both networks include an input layer consisting of 216 elements, but they differ in the process of preparation of the hand-written digits images. The first network recognizes images (24x9) pixels, processed by applying the log-polar transform, the second network uses matrices downsized to (18x12) pixels, using the bicubic interpolation. Parameters of both networks and the obtained recognition rates are listed in Table 1. The application of methods 1 and 2 have not improved the recognition results, whereas the application of the third method have lead to small increase in the number of correct recognitions.

Further experiments have been carried out for three values of the rejection level *delta*: 0.4, 0.2 and 0.1. Obviously the higher the rejection level the more characters are rejected, but the results are similar, if one takes into account the ratio of the rejections number with and without attention shifting. In Table 2 the results for *delta*=0.1 are listed. As can be seen for all algorithms the recognition rate of the accepted data has decreased, while the recognition reliability has been increased. It means that for the case when the rejection level has been used, less undecided ("I don't know") answers have been obtained. If the "I don't know" answers are regarded as wrong, then the network's effectivity becomes better after application of the attention shifting (see the last row in Table 2).

Figure 3 presents an example of iterative shifting of the attention focusing point. Initially the network has recognized the character "9" as character "7", but with low recognition reliability (*delta*=0.1, see figure 3). In the course of

Table 1. Parameters and recognition rates of backpropagation networks

	Network 1 (log-polar)	Network 2 (bicubic)
Input layer (organization)	216 (24x9)	216 (18x12)
Hidden layer	22	23
Recognition rate (attention "off")	87.05%	88.3%
Recognition rate (att. "on" - methods 1,2)	86.95%	
No of iterations (avg)	8.4 /7.1	—
Recognition rate (att. "on" - method 3)	—	89.9%
No of iterations (avg)	4.75	

Table 2. Results for networks with and without attention shifting (*delta*=0.1)

	Network 1 (log-polar)	Network 1 method 1	Network 1 method 2	Network 2 (bicubic)	Network 2 method 3
Number of rejected data	7.35%	3.65%	3.50%	6.50%	0.70%
Recognition rate of accepted data	90.02%	88.53%	88.65%	91.55%	90.13%
Recognition rate of all the data	83.40%	85.30%	85.55%	85.60%	89.50%

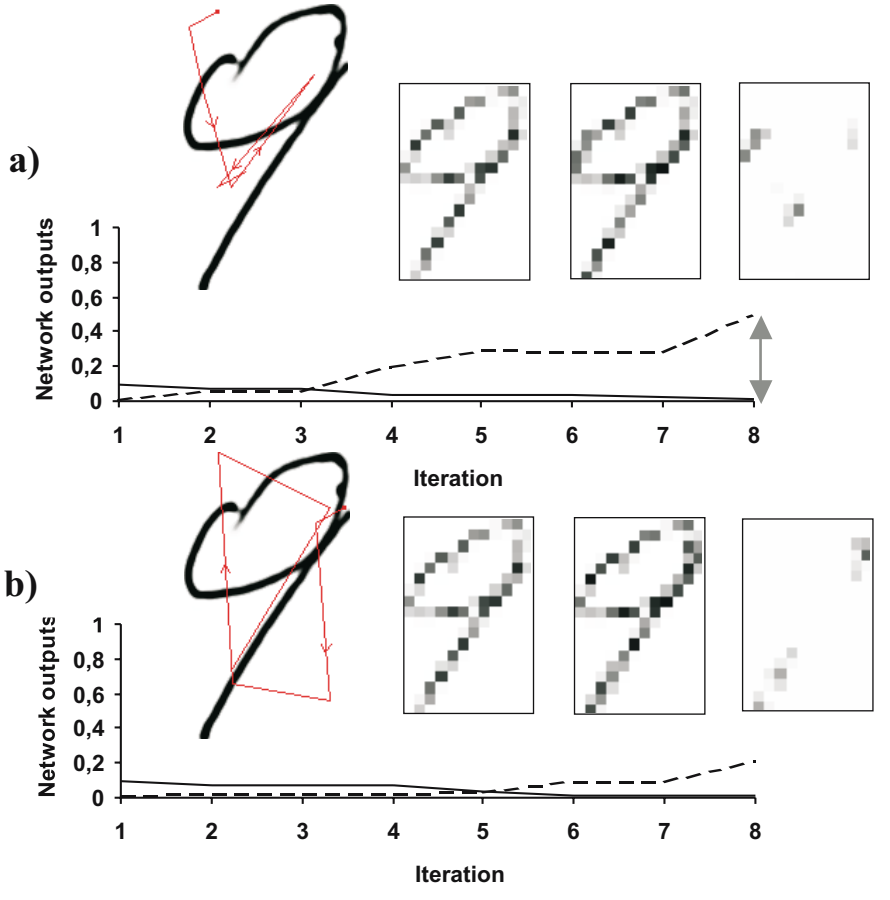


Fig. 3. Shifting of the attention to the areas inconsistent with a) digit "7", b) digit "9". The dashed line - network output attributed to "9", solid line - to "7". On the left-hand side the translation path of the attention focusing point has been plotted against the analyzed character. For better visibility the path has been slightly shifted upwards. The three (outlined) matrices on the right-hand side respectively present: the originally presented matrix of the character, the final (corrected by the shifting attention algorithm) matrix and their difference image.

consecutive iterations, carried out alternatively for the answers "7" and "9", and the respective modifications of the input vector, the network's answers have also varied. Finally in the 8-th iteration the character has been recognized as "9", because the difference of the highest output neuron answers has become higher than the predefined threshold *delta*=0.4 (see figure 3a, grey double arrow).

In figure 3 one can study the details of algorithm execution, having in mind that the path of the attention focusing point plotted against the original "9" character has been slightly shifted upwards. Without this translation most of that path would coincide with the elements of the original character. The attention

shifting order should be also noticed (marked by arrows): at first the most essential features are searched for. The shape of the "9" character is somewhat atypical, as it becomes similar to the shape of digit "7". Figure 3a shows as the algorithm searches for the missing characteristic points of the digit "7": first the horizontal "roof" and then the mid-height crossing line. Figure 3b shows that the missing features of the "9" character are the vertical and lower part of the "cane" and the upper left part of the "circle". The matrix elements changed during the translation of attention focusing point are presented in the right-hand side of the picture. The changes have influenced the output values from the network until a satisfactory recognition reliability has been achieved.

4 Summary and Conclusions

The purpose of this work was to check whether the mechanism of attention shifting to "doubtful" areas of the image of a hand-written digit will positively affect the recognition results of a backpropagation network. Algorithm has been constructed analyzing both the learning set data and the knowledge, contained in the connection weights between the first two layers of the network. The attention shifting mechanism has been turned on depending on the network's recognition reliability. The modifications of the input vector have been done by three different methods. Two algorithms have been based on the log-polar transform, the third one on resizing of the rectangular image matrix. The application of the first two mechanisms have not increased the recognition rate values (network without attention shifting 87.05%, with the attention shifting mechanism 86.95%), whereas the third method, in which the log-polar transform has been replaced by the matrix representation, has improved the results (without attention - 88.3%, with attention - 89.8%).

After testing the algorithms it has turned out that all of them have increased the recognition reliability. It means that when the answers with low reliability (low difference between the two highest output values) are considered as wrong the network's effectiveness is better after switching on the feedback mechanism. At the same time it usually leads to a considerable decrease in the rejection rate (see Table 2). The obtained results indicate that further development of this methods is recommended, with introduction of the necessary modification of the applied algorithms. It seems that the modifications should explore two main directions: making use of the information accumulated in all the network's connection weights and including the information contained in the neighborhood of the receptive field to be enhanced, by application of more advanced processing.

References

1. Itti, L., Koch, C.: Computational Modeling of Visual Attention. Nature Rev. Neuroscience, **2** (2001) 194–203
2. Graboi, D., Lisman, J.: Recognition by top-down and bottom-up processing in cortex: the control of selective attention, J. Neurophysiol., **90** (2003) 798–810

3. Alpaydın, E., Salah A.A., Akarun, L.: A Selective Attention Based Method for Visual Pattern Recognition with Application to Handwritten Digit Recognition and Face Recognition. IEEE Trans. PAMI, **24** (2002), 420–425
4. Choi, S., Ban, S., Lee, M.: Biologically Motivated Attention System Using Bottom-up Saliency Map and Top-down Inhibition. Neural Information Processing Letters and Reviews, **22** (2004)
5. Spratling, M.W., Johnson, M.H.: Pre-integration lateral inhibition enhances unsupervised learning. Neural Computation, **14** (2002), 2157-79
6. Schwartz, E.L.: Spatial mapping in the primate sensory projection: analytic structure and the relevance to perception. Biological Cybernetics, **25** (1977), 181-194
7. Weiman, C.F.R.: Tracking Algorithms Using Log-Polar Mapped Image Coordinates. SPIE Intelligent Robots and Computer Vision VIII **1192** (1989)
8. Sandini, G., Tistarelli, M.: Vision and Space-Variant Sensing. In: Wechsler, H. (ed.): Neural Networks for Perception. Academic Press, San Diego, (1992)
9. Mikrut, Z., Augustyn, G.: Influence of the object representation on the results of character recognition in the car's license plates. Proc. 5th Conf. on Neural Networks and Soft Computing, Zakopane, Poland, (2000), 241-246
10. Mikrut, Z.: Recognition of objects normalized in Log-polar space using Kohonen networks. Proc. 2nd Int. Symp. on Image and Signal Processing and Analysis, Pula, Croatia, (2001), 308-312

An Accurate MDS-Based Algorithm for the Visualization of Large Multidimensional Datasets

Antoine Naud

Department of Informatics, Nicolaus Copernicus University, Toruń, Poland
naud@phys.uni.torun.pl
http://www.phys.uni.torun.pl/~naud

Abstract. A common task in data mining is the visualization of multivariate objects on scatterplots, allowing human observers to perceive subtle inter-relations in the dataset such as outliers, groupings or other regularities. Least- squares multidimensional scaling (MDS) is a well known Exploratory Data Analysis family of techniques that produce dissimilarity or distance preserving layouts in a non-linear way. In this framework, the issue of visualizing large multidimensional datasets through MDS-based methods is addressed. An original scheme providing very accurate layouts of large datasets is introduced. It is a compromise between the computational complexity $O(N^{5/2})$ and the accuracy of the solution that makes it suitable both for visualization of fairly large datasets and preprocessing in pattern recognition tasks.

1 Introduction

The increasing amount of data available over the Internet gives rise to a need in efficient data analysis tools allowing an easier use of large databases. Data visualization is often a necessary step in a data analysis process that permits to detect the presence of clusters, outliers or various regularities in data. This paper focuses on dimensionality reduction methods as tools for the visualization of large multidimensional datasets, as well as a feature extraction of such data. These tasks have been successfully performed by neural networks as the Self-Organizing Maps [10], or by kernel methods [17], latent variable methods as the GTM [3] or multidimensional scaling (MDS) [7]. In order to improve the quality of layouts and to adapt them to the visualization of increasingly growing datasets, newly developed approaches to the above models include local dimensionality reduction and hierarchical visualization.

The visualization of large datasets using full MDS scaling is in general unpractical due to the computational complexity scaling exponentially with the number of mapped items N (one iteration of the minimization process scales with $O(N^2)$ and $O(N)$ such steps are needed, leading to an algorithmic complexity of $O(N^3)$). This limits such applications to a few thousands items sized datasets. A strategy to alleviate this constraint is to split the dimensionality reduction process into two steps: first a smaller dataset built from the data (obtained e.g. by clustering, or any other technique) is mapped and second, the input data is added in some way to the smaller dataset's layout obtained in the first step.

This general scheme for large scale dimensionality reduction has been proposed and implemented in various manners, by applying various approaches for building

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 643–652, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the smaller dataset (called sometimes Basis, as in the rest of this paper) and different choices for the dimensionality reduction technique. Let us mention here below only those approaches that are most closely related to our MDS-based proposal. Basalaj proposed *incremental scaling* [2], in which data points are incrementally added through a single least-squares scaling (points are added on a 1 by 1 basis). A Minimal Spanning Tree of the data is used to define the order in which points are added. The Basis is incrementally growing, leading to $O(N^{7/3})$ global complexity. Brodbeck and Girardin [5] use the clustering capability of SOM and a spring model to produce whether local layouts of cluster neighborhoods, or one global layout of the cluster centers. Morrison et al. [6] [11] [12] use a sample of $\sqrt{N}$ items instead of a data clustering followed by an interpolation strategy also proposed by Brodbeck and Girardin, achieving very low complexities: $O(N^2)$, $O(N\sqrt{N})$ and $O(N^{5/4})$, allowing to visualize a dataset of 108,000 14-dimensional objects. Schwenker et al. [18] combine in ACMDS adaptive c-means and classical scaling. Williams and Muntzer [19] designed a steerable and progressive MDS capable of visualizing 120,000 items and 294 dimensions in a few hours, using hierarchical structures to select subsets of interest and progressive, in-depth and localized layouts. There are also many approaches proposed to adapt linear dimensionality reduction techniques (such as classical scaling [7]) to the visualization of large datasets, let us mention among others FastMap [8] and Locally Linear Embeding [16].

Our approach is to take as a Basis the set of cluster centers resulting from k-means clustering of input data, and then map the cluster centers using standard least-squares MDS. In a second step, input data is added to the Basis layout using *relative MDS* [13]. This association scheme of k-means clustering and multidimensional scaling is introduced in next Section 2. In Section 3, experiments on artificial and real datasets show the validity of the proposed scheme. A short conclusion summarizes the paper.

2 A New Approach to the Association of MDS to k-Means Clustering

In least-squares MDS, the preservation of neighborhood relationships is ensured by the minimization of the Stress functional $S(\mathbf{Y})$ defined as

$$S(\mathbf{Y}) = \frac{1}{F_n} \sum_{i<j}^{N} w_{ij} \cdot (D_{ij} - d_{ij}(\mathbf{Y}))^2 \tag{1}$$

in which $\mathbf{Y}$ is the matrix of coordinates of N points representing the given N D-dimensional items in the output d-dimensional space. $\{D_{ij}\}$ are given dissimilarities or inter-item distances, $\{d_{ij}\}$ are inter-point distances in the output space and $\{w_{ij}\}$ are weighting factors that permit to tune the impact of large distance on the sum (w_{ij} is generally inversely proportional to D_{ij}). F_n is a normalization factor designed to keep Stress values in unit range $[0, 1]$. The minimization of functional $S(\mathbf{Y})$ wrt $N \times d$ variables (the coordinates in $\mathbf{Y}$) can be realized in various ways, these may be local or global optimization techniques, with more or less accurate and time consuming procedures. Our approach implements a steepest descent procedure which includes second order derivatives (increasing convergence and accuracy), in such a way that it is not as

computationally intensive as a real Newton method. It was found in practice to be a good compromise between accuracy of the solution and computational complexity.

In the first step, the N_B-sized Basis is build as the set of cluster centers obtained by a standard k-means clustering. The choice of this clustering algorithm results from a comparison with other techniques (Learning Vector Quantization or dendrograms) in this framework [14], where it was shown that k-means is best suited to this task (i.e. leading to layouts with lower final Stress values). Another small experiment also confirmed this result: 100 *random* Bases were generated as N_B-sized samples from the input data, these were mapped according to the proposed scheme and the resulting layouts were compared in terms of general Stress values (1). The best final layout obtained was with a Basis of points very close to the cluster centers obtained from k-means clustering. The association scheme presented in this paper is similar to the one presented in [14], but is gives generally better results (i.e. lower final stresses) for a relatively small computational cost. Let us recall here this scheme: in the first step, the N_B Basis points are mapped using standard least-squares MDS, that is minimizing the *Stress* functional $S_b(\mathbf{Y})$ defined as

$$S_b(\mathbf{Y}) = \frac{1}{F_n} \sum_{i<j}^{N_B} w_{ij} \cdot (D_{ij} - d_{ij}(\mathbf{Y}))^2 , \tag{2}$$

The second step (addition of data to the Basis layout) slightly differ from the one in [14], in which input data was added on a point by point basis (as in Basalaj's work). Here the data is added into K batches of N_C input data ($N_C = N_B$ for each batch, except for the last batch where $N_C \leq N_B$, and $K = \lceil N/N_C \rceil$). So each batch adds only a subset of N_C input data to the Basis layout using relative MDS. Each such batch mapping consists in the minimization of one Stress functional from the following series (for $k = 1, \cdots, K$):

$$S_{r,k}(\mathbf{Y}) = \frac{1}{F_n} \sum_{i<j}^{N_C} w_{ij} \cdot (D_{ij} - d_{ij}(\mathbf{Y}))^2 + \frac{1}{F_n} \sum_{i=1}^{N_C} \sum_{j=1}^{N_B} w_{ij} \cdot (D_{ij} - d_{ij}(\mathbf{Y}))^2 , \tag{3}$$

Intuitively (this is confirmed by experiments in Section 3), the layout resulting from this relative MDS mapping scheme cannot be as low as the one obtained by one full MDS of the entire dataset, because in relative MDS distances between points added in separate batches are never taken into account. For this reason, adding points in batches of small subsets, whose inter-points distances are included in Stress expression (3) should give better results than adding points one by one. We have at hand groupings from the clustering stage: the cluster centers neighborhoods (the set of points whose a center is the closest). Two strategies to form the data groups are as follows: i) form one group corresponding to each cluster center neighborhood (this is *intra-cluster* grouping) and ii) form one group by picking randomly one input data from each neighborhood (this is *inter-cluster* grouping). Expressions (2) and (3) give inherently more weight to larger distances, even more if squared distances are used. This suggests to apply the above second strategy (ii), in order to force having as much as possible larger distances taken into account in each relative mapping batch, to finally obtain a lower Stress. Experiments presented in the next section will show that this intuitive approach is well-founded.

In order to reduce computation times, the above Stress expressions have been simplified by using squared Stress (*SStress*) where all input and output distances are squared Euclidean distances. Besides this, all the weights $\{w_{ij}\}$ are taken equal to 1 and the normalization factor is $F_n = \sum_{i<j}^{N} D_{ij}^2$, which leads to a Stress functional called here below SS_1. The stopping criterion for the minimizations iterative process was the gradient length per variable, that is divided by $N_m \times d$, ($N_m = \{N_B, N_C\}$ is the number of points mapped in one minimization process). The stopping threshold value was $\epsilon_G = 1.0e-12$ for full MDS and $\epsilon_G = 1.0e-8$ for relative MDS, which yields in general to a number of iterations of the same order as N_m.

As proven by expressions (4-6) and shown in next section's experiments, minimizing expressions (2) and (3) is faster than minimizing the original expression (1). We assume that the number of iterations needed in one standard MDS minimization process is proportional to N, and proportional to $\sqrt{N}$ in relative MDS. Computational complexities CC_{full} of full MDS on input data, CC_{single} of relative MDS in single batches and CC_{groups} of relative MDS of N_C-sized batches can be assessed as follows (following Chalmers et al. [11], we set $N_B = \sqrt{N}$ to simplify the expressions, and neglecting the clustering stage):

$$CC_{full} = O\left(N^3\right), \tag{4}$$

$$CC_{single} = CC_{step1} + CC_{step2} = O\left(N_b^3\right) + O\left(N_b^2 N\right) \approx O\left(N^2\right), \tag{5}$$

$$CC_{groups} = CC_{step1} + CC_{step2} = O\left(N_b^3\right) + O\left((N/N_b)^2 N_b N\right) \approx O\left(N^{5/2}\right). \tag{6}$$

3 Experiments on Artificial and Real Datasets

The proposed mapping schemes have been checked and compared on 10 artificial and real datasets summarized in Table 1. Four artificial datasets were generated from high dimensional simplices as follows: a D-dimensional simplex is first built as a set of $D+1$ vertices such that $D_{ij} = 1, \{i, j\} \in [1, ..., D+1]$, then N_G points are generated by spherical Gaussian distributions centered on each simplex vertex (with fixed variance $\sigma = 0.01$). Four such intrinsically high-dimensional datasets (IDs`1-4` in Table 1 were generated for $D = 10, 11, 20, 50$ and $N_G = 500, 1000, 100, 120$ respectively. Dataset `5-psychometric`, provided by J. Gomula from UMK Academic Dispensary in Torun, Poland, contains psychometric profiles of patients featured by 13 numerical scales ($\in [20, 120]$) derived from the MMPI tests. The next 3 datasets, `6-abalone`, `7-satimage` and `8-segment` are from the UCI repository [4]. In order to assess the accuracy of the results obtained by our approach, we need to compare their Stress values to the ones obtained by full scaling of the entire datasets. These datasets have similar sizes (N values of order a few thousand items), which allows full scaling of the entire datasets. In order to evaluate its scalability, the proposed scheme was also tested on 2 larger datasets: `9-KDD cup2004` from which the Quantum Physics dataset (file phy_train) was used and only 65 out of the 70 available features were taken, ignoring the 5 features having missing values. Finally, dataset ID`10-UCI-KDD texture` is a part of the Corel Image Features dataset from the UCI KDD Archive. From the 4 subsets

Table 1. Datasets used for tests: data whose names are followed by * have been standardized prior to process. Computation times (in seconds) for Full MDS and k-means + relative MDS mappings with Basis size $N_B = 100$ on CPUs: Pentium 4 3.4GHz (s3) or Dual Core AMD Opteron 2.65GHz (s4).

ID	Dataset name	N (# items)	D (# features)	CPU	Full MDS NOC	k-means + relative MDS N_{it}	KMS	BAS	1B1	ICG
1	simplex10r500	5000	10	s4	152230	10	671	30	105	919
2	simplex11r1000	11000	11	s4	–	10	2985	30	262	3349
3	simplex20r100	2000	20	s4	5527	10	140	7	71	1490
4	simplex50r120	6000	50	s4	47004	10	1232	7	635	5400
5	psychometric	1606	13	s3	11094	10	116	16	22	364
6	abalone	4177	7	s3	65500	20	913	90	432	4583
7	satimage	4435	36	s3	60196	20	1958	5	42	182
8	segment*	2310	16	s3	19800	10	103	31	23	546
9	KDD cup2004*	50000	65	s4	–	10	17733	26	1022	29226
10	UCI-KDD texture	68040	16	s4	–	20	13722	15	968	3913

of features available, only the texture co-occurrence 16 coefficients (file CoocTexture) were used.

The datasets were clustered using the standard iterative k-means procedure of Matlab (compiled in C). Since k-means and relative MDS are not deterministic algorithms, they were run several times for each mapping (10 times for MDS and relative MDS, and N_{it} times for k-means), finally keeping the best solution (with lowest Stress for MDS or lowest sum of distances for k-means). Although k-means is generally considered to be a fast clustering method, its application to our scheme showed that it is the computation time bottleneck when $N_B > 100$, and its use is prohibitive for Bases of size $N_B > 500$. Table 1 presents execution times (in seconds) for 3 different mapping schemes: a) Full MDS mapping (column NOC, for no clustering), b) k-means combined to relative MDS of data added one by one (column 1B1) and c) k-means combined to relative MDS of data added in inter-cluster groups (column ICG). For these two last schemes, execution times of k-means clustering (column KMS) and MDS mapping of Basis (column BAS) are detailed. Full MDS is always very time consuming, and it was not performed on datasets IDs2, 11 and 12 for obvious prohibitive time and memory requirements. It can be seen that k-means execution times grow proportionally to D as well as N and N_B, and become very important for larger datasets. The relative MDS runs shown in Table 1 were for $N_B = 100$. Execution times (in seconds) for varying Basis sizes are presented in Fig. 3 for two datasets. The durations differences between the datasets for relative MDS can vary from one Basis size to another. These performances can be reduced if we decrease the iterations stopping criterion ϵ_G, at the cost of less accurate final layouts.

The performances of the different mappings schemes were assessed by computing Stress values using expr. (1) for the whole datasets. These values were compared to what should be their optimal values: the Stress obtained by Full MDS mapping of the entire datasets. Final Stress values for varying Basis sizes $N_B \in [10, 500]$ are presented in Fig. 1 (datasets 1-5) and Fig. 2 (datasets 6-10). Note that the Stress values scales do

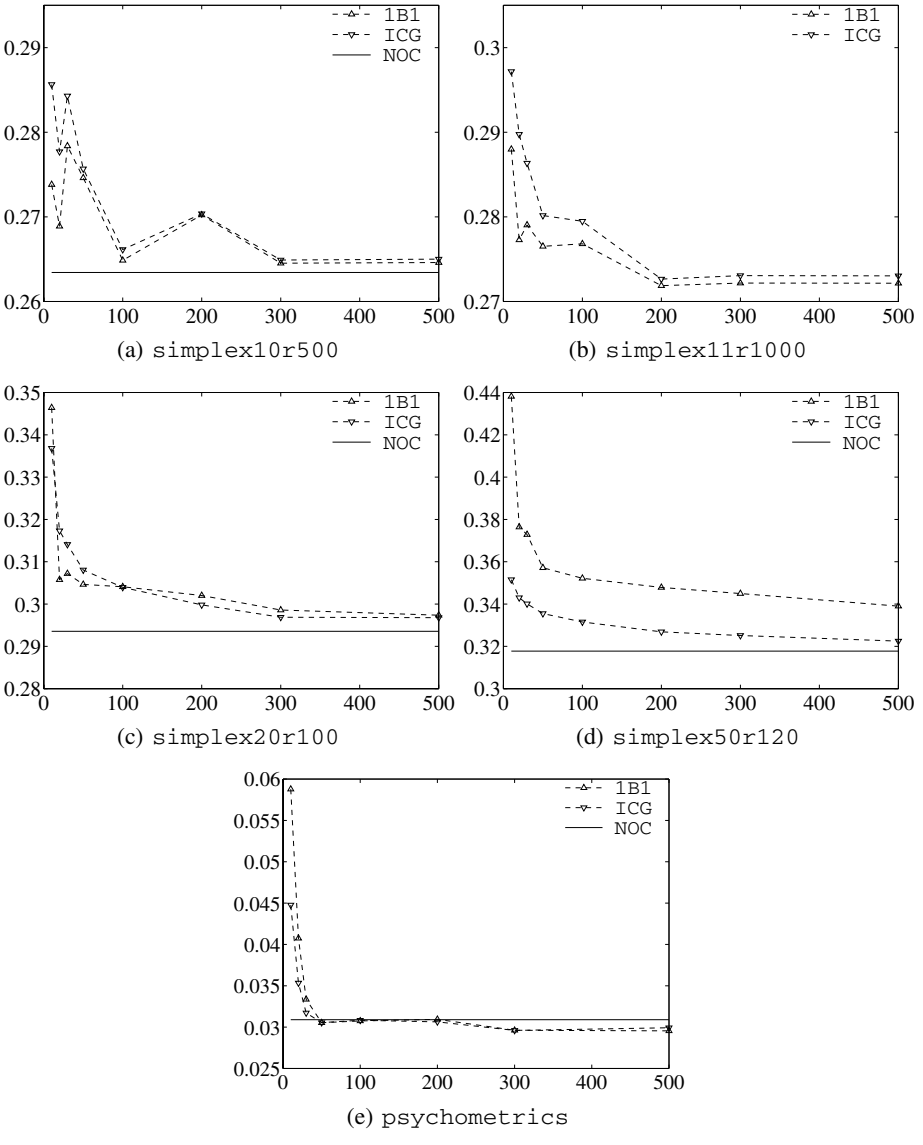


Fig. 1. Datasets 1-5: Final Stress values for varying Basis sizes, obtained by 3 different mapping schemes. Final Stresses reached by ICG variant are lower than with 1B1 variant for each dataset.

not start at zero, in order to enlarge Stress differences between the 3 mapping schemes. Consequently, the observed differences are not similar from one sub-figure to another and should not be compared. As could be expected, final Stresses decrease when N_B increases, due to a growing number of Basis reference points allowing more precise locations of added data. The continuous line (NOC, for no clustering) shows the minimum reached by Full MDS mapping of the entire dataset, when memory requirement made

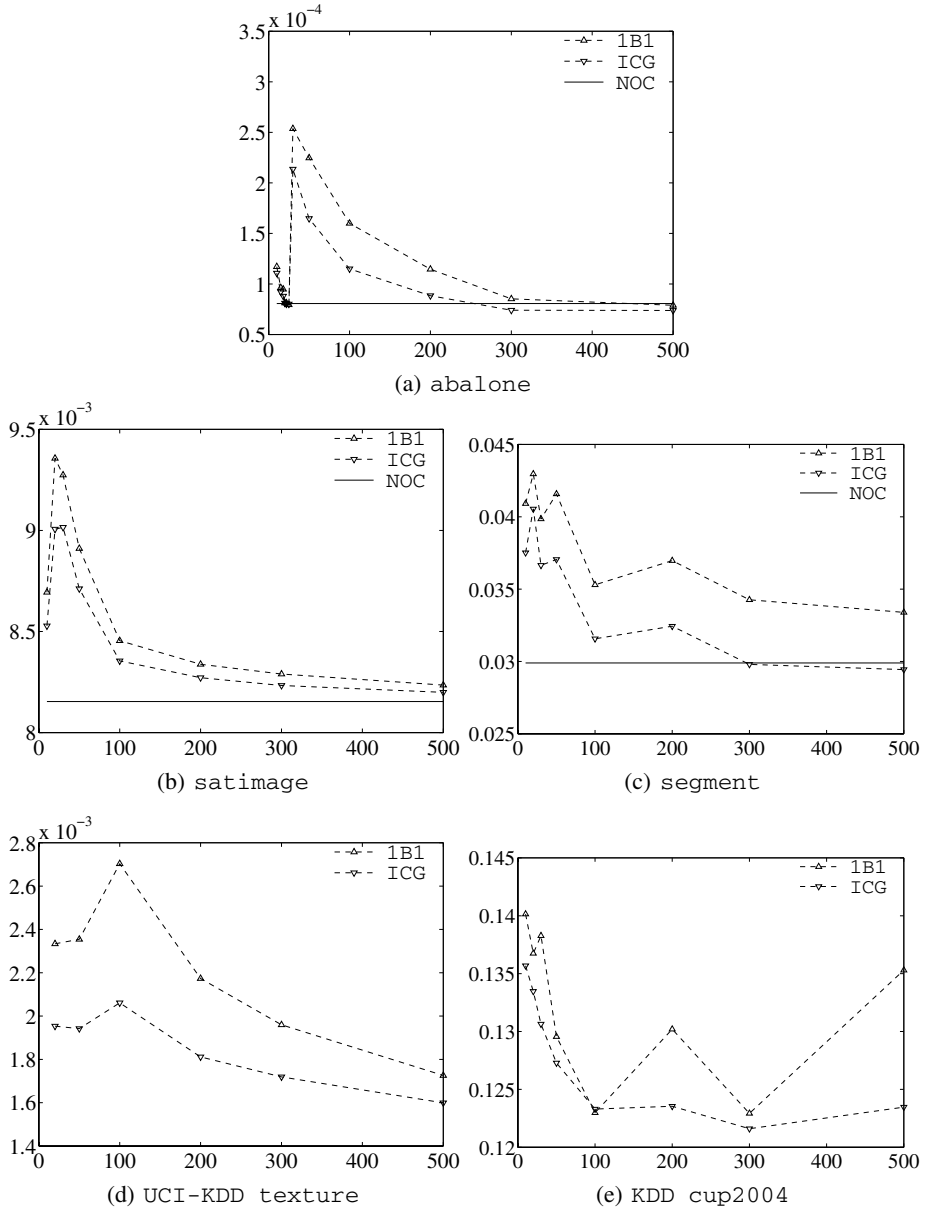
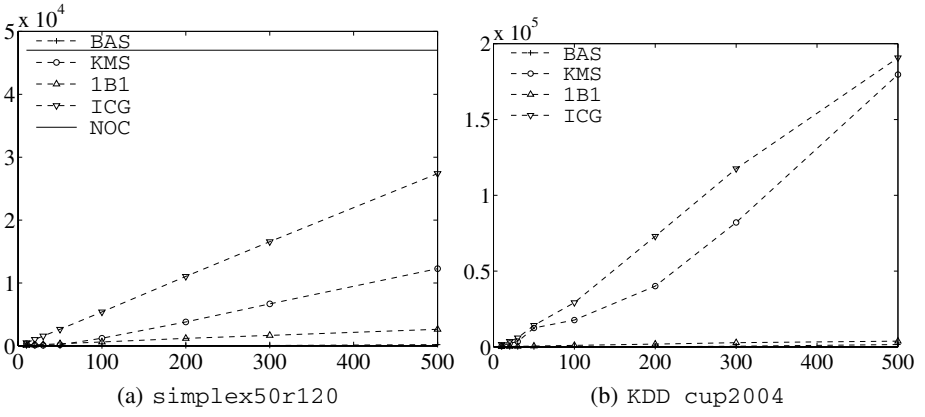


Fig. 2. Datasets 6-10: Final Stress values for varying Basis sizes. As in Fig. 1, final Stresses reached by ICG variant are lower than with 1B1 variant for each dataset.

this possible. The dotted lines show mappings based on k-means combined to relative MDS, through addition of data one by one (1B1 symbols) or through addition in inter-cluster groups (ICG symbols). Two layouts obtained with a Basis size $N_B = 100$ are presented in Fig. 4.



(a) `simplex50r120` (b) `KDD cup2004`

Fig. 3. Execution times as a function of Basis size N_B, for the different steps of the 3 mapping schemes. All execution times grow linearly with N_B. The inter-cluster groups variant is noticeably more time consuming than the one by one variant.

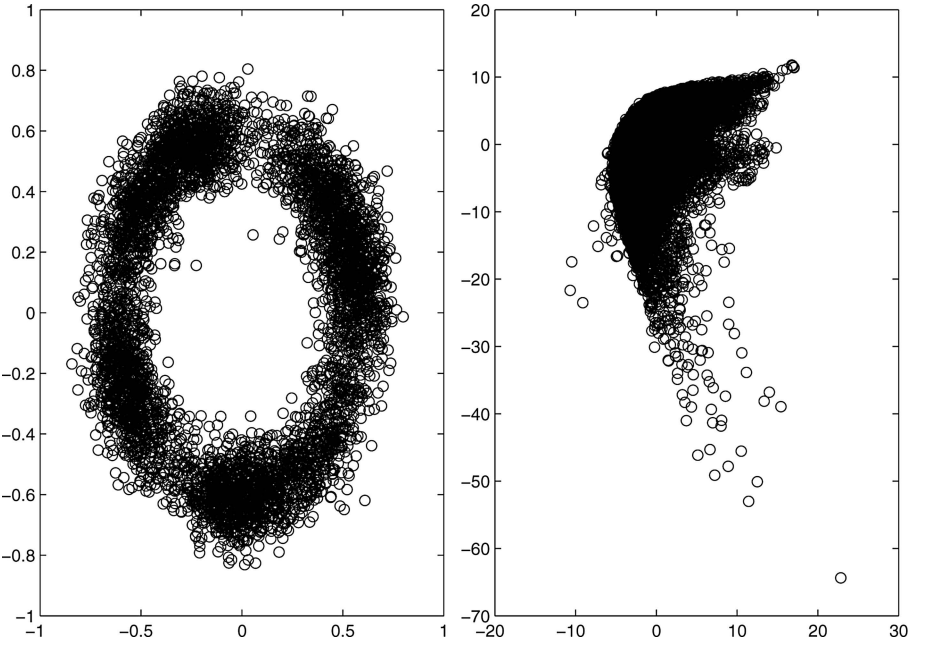


Fig. 4. Layouts of `simplex10r500` (left) and `UCI-KDD texture` (right) for $N_B = 100$

For smaller N_B values, important Stress variations are observed. These are probably related to the *curse of dimensionality* occurring as the number of objects is too law w.r.t. the number of dimensions of the input data. Nevertheless, important local minima can be observed for `simplex10r500` and `abalone` datasets (at $N_B \approx 20$), these effects

remain to be analyzed in greater details. The bottom solid lines represent minimal Stress values reached using Full MDS mapping on whole datasets. It is surprising that final Stress reached by Full MDS is in three cases outperformed by relative MDS (for $N_B > 200$ in `abalone` and `psychometrics`, and for $N_B > 300$ in `segment`). This result shows that combining k-means to relative MDS does not only allow to obtain mappings for large datasets faster than standard MDS does, it can also provide with more optimal solutions.

4 Conclusion

A new way to combine k-means clustering and multidimensional scaling, as an alternative to other approaches aimed at reducing the computational complexity of multidimensional scaling has been presented. The proposed association of relative MDS scaling allowed to obtain accurate layouts of datasets of size up to 68000 items in two hours. The computational complexity of the designed process is reduced by combining MDS to a naive iterative k-means clustering. The resulting solutions present very good Stress performances, sometimes even outperforming the results of full MDS solutions. From the computation times measurements, it appears that the bottleneck of the whole scheme is the k-means clustering stage. In order to speed up this stage, more efficient clustering techniques must be used instead, such as the ones proposed in [1] [9] [15]. The proposed scheme, applied here to data visualization, can be also useful everywhere a dimensionality reduction of data is needed, for instance as a preprocessing stage in pattern recognition applications.

Acknowledgement

This work was partially supported by the Polish Committee for Scientific Research, research grant 2005-2007.

References

1. K. Alsabti, S. Ranka, and V. Singh. “An efficient k-means clustering algorithm,” *Proceedings of the IPPS/SPDP Workshop on High Performance Data Mining*, 1998.
2. W. Basalaj “Incremental multidimensional scaling method for database visualization,” *Proceedings of the Visual Data Exploration and Analysis VI*, SPIE, vol. 3643, pp. 149–158, 1999.
3. C.M. Bishop, J.F.M. Svensen and C.K.I. Williams “GTM: The Generative Topographic Mapping,” *Neural Computation* vol. 10(1), pp. 215–234, Jan. 1998.
4. C.L Blake and C.J. Merz, “UCI Repository of machine learning databases,” Irvine, CA: University of California, Department of Information and Computer Science, 1998.
5. Brodbeck, D., L. Girardin, “Combining Topological Clustering and Multidimensional Scaling for Visualising Large Data Sets”, Unpublished paper (accepted for, but not published in *Proceedings of the IEEE Information Visualization 1998*)
6. M. Chalmers “A linear iteration time layout algorithm for visualising high-dimensional data”, *Proceedings of the IEEE Visualization’96*, San Francisco, pp. 127–132, 1996.

7. T. F. Cox and M. A.A. Cox “Multidimensional Scaling,”, Monographs on Statistics and Applied Probability, vol. 59, Chapman & Hall, 1994.
8. Christos Faloutsos and King-Ip Lin, “FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets,” *Proceedings of the SIGMOD Conference*, pp. 163-174, 1995.
9. Tapas Kanungo and David M. Mount and Nathan S. Netanyahu and Christine D. Piatko and Ruth Silverman and Angela Y. Wu “An Efficient k-Means Clustering Algorithm: Analysis and Implementation,” in IEEE Trans. PAMI, 24 (7), pp. 881-892, 2002.
10. T. Kohonen *Self-Organizing Maps* Heidelberg, Berlin: Springer-Verlag, 1995.
11. Morrison A., Ross G., Chalmers M. “Fast multidimensional scaling through sampling, springs and interpolation”, *Proceedings of the Information Visualization 2*, 1, pp. 68 - 77, 2003.
12. Morrison A., Chalmers M. “Improving hybrid MDS with pivot-based searching”, *Proceedings of the Information Visualization 4*, 2, pp. 109Ű-122, 2005.
13. A. Naud and W. Duch “Visualization of large datasets using MDS combined with LVQ” *Proceedings of the Sixth International Conference on Neural Networks and Soft Computing*, Zakopane 2002, pp. 632–637, L. Rutkowski and J. Kacprzyk eds.
14. A. Naud “Visualization of high-dimensional data using an association of multidimensional scaling to clustering” *Proceedings of the 2004 IEEE Cybernetics and Intelligent Systems*, Singapore 2004.
15. D. Pelleg and A. Moore, “Accelerating Exact k -means Algorithms with Geometric Reasoning,” in "Knowledge Discovery and Data Mining", pp. 277–281, 1999.
16. L. K. Saul and S. T. Roweis “Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds,” Journal of Machine Learning Research, 4, pp. 119-155, 2003.
17. B. Schölkopf, A. Smola and K.-R. Müller “Nonlinear Component analysis as a Kernel Eigenvalue Problem,” *Neural Computation*, vol. 10(5), pp. 1299–1319, July ‘1998.
18. F. Schwenker, H. Kestler and G. Palm “Algorithms for the visualization of large and multivariate datasets,” in *Self-organizing neural networks* U. Seiffet and L. C. Jain eds, chap. 8 pp. 165–183, Physica-Verlag, Heidelberg, 2002.
19. M. Williams and T. Munzner “Steerable, Progressive Multidimensional Scaling,” *Proceedings of the InfoVis 2004*, pp. 57-64, 2004.

The Multi-Agent System for Prediction of Financial Time Series

Šarūnas Raudys[1] and Indre Zliobaite[2]

[1] Institute of Mathematics and Informatics
Dept. of Informatics, MIF, Vilnius University,
[2] Naugarduko st. 24, Vilnius 03225, Lithuania
raudys@ktl.mii.lt, indre.zliobaite@mif.vu.lt

Abstract. To take into account different character of distinct segments of non-stationary financial time series the multi-agent system based forecasting algorithm is suggested. The primary goal of present paper is to introduce methodological findings that could help to reduce one step ahead forecasting error. In contrast to previous investigation [6], instead of single prediction rule we use a system of several adaptive forecasting agents. The agents evolve, compete among themselves. Final decision is made by a collective of the most successive agents and present time moment. New multi-agent forecasting system allows utilizing shorter training sequences and results in more accurate forecasts than employing single prediction algorithm.

Keywords: Classification, Forecasting, Sliding window, Training, Dimensionality.

1 Introduction

The Problem. Today much attention is drawn to analyzing processes, which are changing rapidly over the time. The changes could be of different nature, they can be *temporary* or *permanent*. Financial time series are examples of a dynamic system. Financial time series here are defined as the price of an asset over time, including stocks, commodities, bonds, price indices or exchange rates. In the financial forecasting task, the algorithms ought to include means to reflect the changes, to be able to adapt to sudden situational changes [1]. Financial time series might be affected by utilization of *diverse forecasting algorithms* employed in real market [2]. Thus, the foremost important properties of financial forecasting algorithm should have, are: quick *adaptability* to changing environments and utilization of *short* historical information. The primary goal of present paper is to introduce methodological findings which could help to reduce forecasting error for one step ahead financial time series forecasting task.

Research in the field. There are a lot of research papers in the financial time series forecasting field; however, the disclosure is limited due to profit opportunities involved. The repeatability of such experiments is limited, first of all, due to the same profit opportunity reason, but moreover, due to constant environmental changes. Financial time series forecasting algorithms should be readapting

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 653–662, 2006.
© Springer-Verlag Berlin Heidelberg 2006

to environmental (data) changes dynamically, otherwise they are not repeatable and they cannot be valuable. Many artificial neural networks (ANN) based approaches for solving financial time series forecasting tasks have been proposed. We would distinguish two main categories: *experimental design* orientated, which mostly assume neural networks as "black boxes" producing outputs from given inputs, and *methodological approaches*, analyzing what is inside the ANN, which usually do not have "plug-and-play" design to be used in real financial markets directly. We attribute our work to the second category.

Many existing financial time series forecasting algorithms employing ANN differ in formation of forecasting target (index), and in measuring the testing error. In this literature review we skip the experimental design oriented financial time series forecasting algorithms ("black box" approaches), and focus on the papers dealing with methodological issues, which to our mind have much more value added in the long term perspective.

One of the pioneering reviews was presented by Moody [3], where poor signal to noise ratio, non-stationarity and non-linearity of financial time series were addressed. The author pruned unnecessary MLP nodes, regularized MLP and used sliding window. He obtained minimum testing error at ~ 10 years training history. He used 1-12 months forecasting horizon (a number of steps ahead). Moody used rates of return as prediction index and market data ranging from 1950 to 1980 as inputs. It can be argued that market characteristics were different thirty years ago, at least due to an absence of powerful forecasting tools.

The reader is referred to an excellent review of foreign exchange rates forecasting by Huang *et al.* [4], where a number of forecasting methods using ANN were compared, to a large extent applicable to various financial time series. The authors aligned 11 papers comparing ANN forecasting with traditional financial time series forecasting methods, such as ARMA, ARIMA, GARCH and random walk. MLP was used in 10 out of the 11 papers. Although these papers highly differ in performance measures including several types of absolute and percentage errors as well as average relative variance, in 7 out of 11 papers, better results were achieved using the ANNs as compared to traditional statistical methods. In the remaining 4 papers mixed results were obtained.

Lendasse *et al.* [5] used radial basis function ANN for forecasting. They used stock market index data and obtained 42,8% testing error with 2100 days testing set. They reformulated the task into classification of increases and decreases of financial variables over time and used 500 days sliding window for training. The proposed approach to solve non stationarity problem was to disregard $t+1$ period forecasting if wrong predictions exceeded right in the last 5 days counting back from day t, obtaining 34,7% testing error in 1583 testing days.

Our previous research. Beforehand, we developed a method to increase the accuracy in situations when environments are changing permanently [6]. Training of forecasting rule was based on short data sequences what results to more accurate predictions as using lengthy historical data. Optimal training set size was determined both theoretically and experimentally. To reduce generalization error, the data dimensionality was reduced by mapping input vectors into low

dimensional space using MLP. Forecasting was performed by SLP based classifier. While training, the perceptron it was initialized with weight vector obtained after training with previous portion of the data sequence. To save useful information accumulated in financial time series data, the early stopping procedure was utilized. *For reader convenience let us call the former algorithm presented above* ***MSSE*** *(M-mapping, S-short history, S-save weights, E-early stopping).*

Thus, there is a common understanding in the field of financial time series forecasting, that in changing environments, it is necessary to reduce complexity, find ways to make use of the shortest possible historical data and the algorithm should have means to adapt to complex and continuously changing environments. Inspired by the nature, we claim that successive adaptation is possible employing a number of diverse competing agents. An entity of the agents allows creating distinct forecasting "styles" to learn rapidly different environmental changes.

In present paper, we expand our method proposed in [6] from the single forecast to multi-agent system (MAS). While the MSSE contributed towards solving a problem of *permanently* changing environment, the integration of MSSE algorithm to the MAS contributes to solving *temporary* changing environment problems. Forecasting algorithm ought to adapt to the changes quickly and start to predict accurately. Adaptation speed depends on accuracy of determination of initial weight vector. Since a character of multidimensional time series is changing often, for a single forecasting rule it is difficult to adapt rapidly. We suggest using several distinct forecasting algorithms (SLPs) capable to adapt to diverse changes. In our paper, at first we analyze a behavior of such system while dealing with artificial time series, then test our algorithm with real data. We developed a simplified prototype of such system where only a part of our ideas were realized.

2 Proposed Method

We formulate financial time series forecasting task as pattern classification problem [6], defining the classes as *increase*, *decrease* and *insignificant change* of a chosen financial variable at time $t + 1$ as compared to time t. We use forecasting/prediction terminology when referring to the problem itself and classification terminology, when referring to the proposed methodology. Forecasting procedure is based on *sliding window* approach, which is often used in financial time series forecasting domain: the system is trained on a particular segment of the multivariate time series historical data and the performance of the trained algorithm is tested on subsequent segment. After recording testing results, the testing set becomes a part of training set, the oldest training data are left over. Then we consider a new testing segment, etc.

The first stage of MSSE algorithm [6] was data preparation and dimensionality reduction, where data was mapped into low dimensional space by wrapper approach based neuro-linear dimensionality reduction [7]. *The second stage* was derivation of polynomial features and single SLP training - testing using the sliding window approach. Here we extend the final stage of the algorithm. In order to make forecasting system more robust to changes, we use MAS for final

forecast. In the new approach, r distinct adaptive forecasting agents are represented by r diverse SLPs. Each time we select r_{best} most successful agents for final decision making, see Stage II b in Table 1. For reader's convenience now on we call MSSE algorithm MSSE-1, having in mind single agent forecasting. The new algorithm based on MAS will be called ***MAFS*** (Multi Agent Forecasting System) or MAFS-r, when specific number of agents r will be used.

Table 1. The steps of MAFS algorithm

Stage I	Step 1 Data preparation (training TR and testing TE data blocks)
	Step 2 Neuro-linear dimensionality reduction (TR, TE $\rightarrow$ TR_3,TE_3)
Stage II a	Step 3 Derivation of polynomial features (TR_3, TE_3 $\rightarrow$ TR_9, TE_9)
	Step 4 Generating r agents with differentiated initial weights, using TR_9
Stage II b	Step 5 Each agent retraining; testing on TE_9 using "sliding window"
	Step 6 r_{best} best* agents (SLPs) vote to make final forecast
	Step 7 Retraining weights are saved only for r_w best* agents
* "best" in terms of factual testing error estimated in previous testing "window"	

Stage I: Data preparation and dimensionality reduction. This stage mainly deals with experimental design.

Step 1. After dividing the data into training (TR) and testing (TE) slots, we leave testing data aside and use training data slot for determination of model parameters. We divide training days into three non-overlapping classes: C_1 – index "ups" (25% of highest increases of forecasting index in two consecutive days), C_2 – index "downs" (25% of highest decreases) and C_{middle} – the "middle" class (the remaining 50% of training days) which we eliminate from the experiment to exclude insignificant changes and account for transaction costs.

Input data vectors X_t are formed using *four* consecutive days price history (from day $t-3$ to day t) of each of the *five* considered financial variables. In this way, we generate 20-dimensional vectors. The length of input has been determined experimentally and is a subject of user's choice. We apply the C_1 and C_2 input vectors transformation towards zero mean and unit variance.

Step 2. We apply the MLP classifier for dimensionality reduction from 20 down to 3 new features [7]. This simple feature extraction (FE) method performs linear FE with nonlinear performance criterion. The l *new features*: $Z = (z_1, z_2, \ldots, z_l)$, are linearly weighted sums, $z_s = \sum_{j=1}^{p} w_{sj} x_j$, $(s = 1, 2, \ldots, l)$ of p inputs $(l < p)$ calculated in l hidden neurons. The new extracted feature space depends on minimization criterion used in training, i.e. on complexity of decision boundary. In spite of simplicity, the neuro-linear FE method is very powerful tool, using information contained in all input features.

The parameters determined in Steps 1 and 2 using TE were applied to TR data set. For classification into 3 pattern classes we have chosen thresholds: $X_t \in C_1$, if $Y_t > Y_{\max}$, $X_t \in C_2$, if $Y_t < Y_{\min}$, where Y_t is the value of forecasted index at day t and $Y_{\max} = \max Y | Y \in C_1$, TR; $Y_{\min} = \min Y | Y \in C_2$, TR.

We did not recalculate Stage I after each training window, since the gain was negligible.

Stage II a: The MAFS algorithm architecture

Step 3. Financial data is complex. Thus, non linear decision boundary is needed. The MLP based classifier can be easily trapped into bad local minima. Therefore, we have chosen SLP classifier to work in the 2^{nd} order polynomial feature space derived from TR_3 and TE_3.

$Z = (z_1, z_2, z_3) \rightarrow (z_1, z_2, z_3, (z_1)^2, (z_2)^2, (z_3)^2,\ z_1z_2, z_1z_3, z_2z_3) = Q$, here $Q = (q_1, q_2, \ldots, q_9)$ leads to 9-dimensional input feature space (TR_9 and TE_9) [6].

Step 4. We generate r agents (SLPs) with identical architecture. In order to obtain diversity among the predictors, the perceptrons differ in their initial weight vectors $\mathbf{w}_{\mathrm{start}(i)}$, $j = 1, 2, \ldots, r$. At first, we divided training set TR_9 into $(r-5)/2$ non-intersecting segments. We utilized 60% of each segment for training and remaining 40% of vectors we used for validation. We had run trainings on each of those segments in order to determine starting weights of $(r-5)/2$ agents. The starting weights of other $(r-5)/2$ agents were determined by adding uniform distribution random components ξ, $\xi \in (-0,5; 0,5)$, to $(r-5)/2$ previously obtained weights. Remaining four agents got absolutely random weights $5^*\xi$. The very last agent got zero initial weights (see Table 2 for the agent selection).

Table 2. Initial agent selection for final decision making

Method for the agent selection	Number of agents
1 Initial weights obtained via training agents on the training block	$(r-5)/2$
2 Initial weights obtained via training agents on the training block + uniformly distributed random variable $\zeta \in (-0,5; 0,5)$	$(r-5)/2$
3 Uniformly distributed random weights $\mathrm{w}_{\mathrm{start}} \in (-2,5; 2,5)$	4 agents
4 Zero component initial weights	1 agent

We believe that for each segment of the data or short period of time we have different "styles" of financial data fluctuations, since non-stationarity assumption holds. Thus, *for each fluctuation style we do need different classification (forecasting) algorithms*. We train and test all r agents with the same portion of data, however, start training from the agent's individual weight vector.

Stage II b: Forecasting as the Multi-agent system. This stage gets into feedback loop, which steps through all the testing set TE_9.

Step 5. We retrain the system on subset $\mathrm{TR}'_9 = (Q_{t-k}, \ldots, Q_t)$ and test the performance on subset $\mathrm{TE}'_9 = (Q_{t+1}, \ldots, Q_{t+m})$, where k stands for training window size and m stands for system moving step, t stands for the time (today).

Step 6. The final forecasting decision for subset TE'_9 is made by r_{best} agents' majority voting procedure. The r_{best} agents are selected according to forecasting performance on subset $\mathrm{TE}_9^{-1} = (Q_{t-m+1}, \ldots, Q_t)$.

Step 7. If training of the jth agent was successful (it falls into a pool of the r_{w} best agents characterized by the smallest testing errors on TE'_9), its final weight vector was used as a starting weight in subsequent data segment training.

It is known that if the weights of the well trained non-linear SLP based classifier become large, they start slowing down further training process [8]. To overcome this complicatedness, and to save possibly useful information contained in starting 10-dimensional weight vector $\mathbf{w}_{\text{start}}$, each new training session was started from a *scaled* weight vector $\kappa \times w_{end}$, where $\mathbf{w}_{\text{end}}$ stands for weight vector obtained from previous training. Optimal value of parameter κ was determined from the minimum of the cost function calculated from the testing subset TR'_9 after recording current test results. In contrast, if previous training of the jth agent was unsuccessful, the initial weight vector for this agent training remains unchanged. In present version of the forecasting algorithm, a number of training epochs, n_{train}, was fixed *a priori*. It is a subject of future investigations.

3 Experiment Design

We used artificial data set for setting architecture, global parameters of MAFS system and investigation of its primary characteristics. Real world data was used to check usefulness of the MAFS and compare it with previous research.

Artificial data. Long-lasting non-stationary multidimensional data was used in the first part of the experiments. The data was generated by excitable media model composed of 250×250 cells in a hexagonal grid as described in [9]. Each cell mimics single financial market participant and affects six neighboring cells. Sums of the excitations in five non-overlapping areas served as five features that mimic random fluctuations of five financial variables. Random changes in strength and directions of wave propagation, refractory period, threshold value (when the cell is excited), were manually introduced in order to have *severe unexpected environmental changes*. The waves of cell excitation were assumed to represent information reaching some market participants and not known to the other ones. Such financial market model was not used by other researchers yet.

Artificial data used in the experiment consist of 6884 "days", 1884 from which were used for training and the rest of them – 5000 "days" for testing. A couple of excerpts from generated artificial time series are shown in Fig. 1a. In addition, in Fig. 2b real time series used in the experiment are pictured for comparison.

Real market data. As in [6], the real data was taken form commodity exchanges during period 1993-06-08 – 2005-10-27. It consisted of 3211 observation days. The data array was comprised of 5 time series: x_1 – Crude Oil-WTI Spot Cushing U$/BBL, x_2 – Cocoa-ICCO Daily Price US$/MT, x_3 – Corn No.2 Yellow Cents/Bushel, x_4 – Gasoline, Unld. Reg. Non-Oxy, NY, C/Gal and x_5 – LME-Copper, Grade A Cash U$/MT, the price index of latter security was predicted. We used 711 days for training and the remaining 2500 days for testing.

Experimental parameters. After several attempts we have chosen: $r = 41$, $r_w = 13$ and $r_{\text{best}} = 9$. The testing window size was chosen $|\text{TE}'_9| = m = 20$ days and TR'_9 was a variable in the range $|\text{TR}'_9| = k = 20...800$ days to come up to the optimal training window size with the smallest testing error. For *each* testing attempt with different training window length we repeatedly used

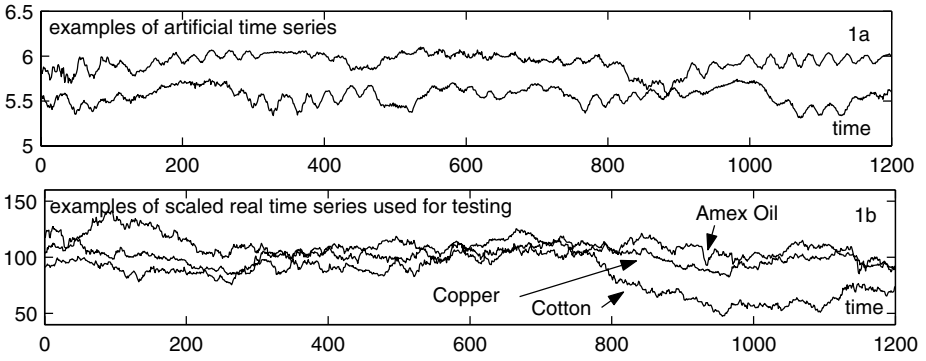


Fig. 1. (a) artificial time series; (b) real market time series (scaled to fit the grid)

$|\mathrm{TE}_9| = 5000$ testing "days" in the experiments with artificial data and $|\mathrm{TE}_9| = 2500$ testing days in the experiments with real data.

Index. We use rather simple measure of returns to construct less typical index, for labeling training and testing data. Our index is as follows:

$$\text{Index:}\quad Y_t = \log\big((B_{t+1} + B_t)/(B_t + B_{t-1})\big), \tag{1}$$

where B_t is the price vector of given security in time t.

This way we aim to forecast if the return tomorrow will be significantly higher (label 1) or significantly lower (label 0) than it was today. We know today's and historical prices. Therefore, we can calculate simplified returns. *It should be noted that the model is designed for testing of our methodology. The index is not a core aspect in this research. It might be changed by another one at user's convenience.*

4 Results

We compare the testing error results obtained using MAFS-41 with the results of forecast MSSE-1, proposed in previous paper. A number of differently generated artificial time series were used to develop new algorithm. In Fig. 2 we see forecasting examples where the previous algorithm and the new one are compared.

While considering artificial time series, we generated the series with discreet environmental changes of needed length. Our experiments show that the more non-stationary time series are, the more clear training window optimum we get and we have clearly convex pattern of experimental graph. Fig. 2a shows that MAFS is able to constantly outperform MSSE-1 during the *whole* training window spectrum. It is very positive and promising result. For the same set of model parameters, utilization of MLP gave almost constant 27% testing error rate, thus giving clear loose in accuracy as compared to both of our suggested strategies.

We validated the experiment with the real market data (Fig. 2b and Table 3). We observed notable gain in absolute testing error as well as we gained 170 days

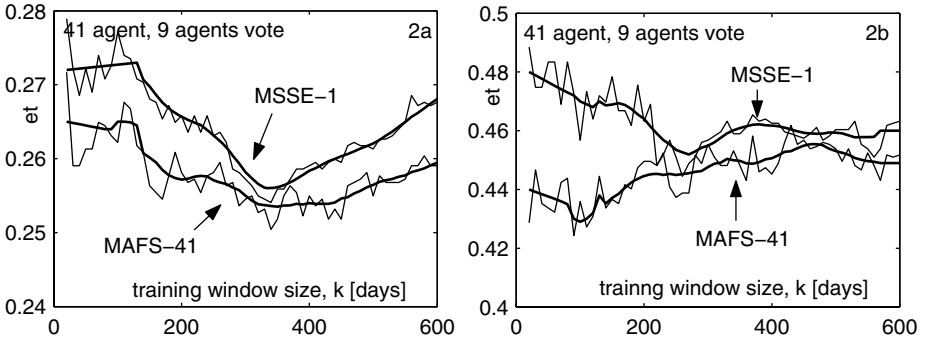


Fig. 2. Classification error e_t as a function of training window size k (in days): (a) artificial data, (b) real data. Bold lines – smoothed results, thin lines – original results.

in training window size. We achieved the gain *for all training window lengths, k.* (Fig. 2b). This aspect of the MAFS is essential as the optimal training window can hardly be determined in advance in applications for real time systems. In terms of testing error, we can argue that new MAS based forecasting algorithm is capable reacting to abrupt market changes more successfully.

Having the same time horizon and other experimental parameters we tested with other commodity prices, the results of three of which are provided in Table 3. We achieved gains in testing error and mixed results in terms of training history, gaining in accuracy in several experiments, but losing in the other ones.

5 Implementation and Limitations

A contribution of present paper is the suggestion to utilize adaptive multi-agent system for collective decision making in the final stage of MSSE algorithm. Such approach enables accumulation of prior information about different "styles" of the time series and makes adaptation to changing environments easier. Employing the MAFS for forecasting of financial variables allows creating diverse adaptation and forecasting "styles" that are good for distinct segments of time series. Use of MAFS improves the testing results of MSSE-1 and helps to reduce the length of training history. Moreover, while employing such forecasting system in practice, one does not know in advance, which agent is the best. Therefore, cooperation (voting) of the most successful agents also gives additional gain.

We suggest using several distinct forecasting algorithms (SLPs) capable to adapt to diverse changes rapidly. The algorithms should differ both in initial weight vectors and learning parameters (learning speed, the length of multidimensional segment of time series used for training, etc.). We set these parameters to be dissimilar to distinct predictors, organize them into adaptive MAFS, capable to change its structure and the global parameters interactively.

We let our MAFS work in real changing environments and then select these initial weight that have been successful at least in one time segment. Then we

Table 3. Results achieved using different real commodities time series starting with different initial conditions

	Testing error, e_t	Optimal "window", k	Testing error, e_t	Optimal "window", k
Index commodity	**MSSE-1 (1 agent)**		**MAFS-41 (MAS)**	
Artificially generated data	25,60%	340	25,35%	350
Copper	45,19%	270	40,29%	100
Amex Oil price index	39,73%	300	37,23%	190
Cotton	42,12%	170	41,57%	260

perform cluster analysis of a set of these initial weight vectors and use cluster centers as the initial weights for the agents of MAFS in subsequent work.

Our experimental design might be questioned by a practitioner as we repeated the experiment several times and used averages from test data sequences to determine mean generalization error, e_t. From a point of view of theoretician, we behaved correctly, since we compared different algorithms in identical conditions and showed principal way how to predict prices in unavoidably changing environments, making use of: (1) short training sequences, (2) dimensionality reduction, (3) early stopping to save previous information (4) faster adaptation to sudden severe changes by utilizing MAFS.

The key goal of the research was to introduce the pool of diverse prediction agents which could help to reduce forecasting error in one step ahead of markedly non-stationary financial time series forecasting task. We did not aim to design ready made system for trading in real market. Our analysis demonstrated usefulness of application of MAFS and indicated that much wider experimentation is necessary in order to determine global parameters of the decision making algorithm, which here were determined from several empiric experiments.

Unfortunately, we cannot compare our method with many of the methods proposed in the field in quantitative terms before rearranging and repeating the experiment due to lack of common problem formulation standards and wide variety of experimental designs in the field. The repetition of published experiments is often impossible due to lack of details provided. However, our paper deals with several important problems often addressed in this field, the most important of which is complexity of influencers and changing environments. Therefore, qualitative results here we believe more important than quantitative gains.

6 Conclusion

We expanded MSSE algorithm from single forecast to the MAFS approach. The MSSE-1 contributed towards solving a problem of *permanently* changing environments. The MAFS additionally contributes to solving *temporary* changing environments problem. It is done through utilization of a great number of diverse prediction rules that learn and adapt to environmental changes differently.

We achieved gains in testing error, while changes in training window size were controversial: for artificial time series minimum smoothed testing error was reduced from 25,60% to 25,35% by 0,25% (MLP gave 26,97% error). Training window size, however, was slightly increased from 340 to 350 days. For the real market Copper price time series minimum smoothed testing error was reduced from 45,19% to 40,29% and training window size decreased from 270 to 100. At 100 days training window MSSE-1 gave 46,9%. Therefore, at that point the gain in accuracy was even larger.

Positive effect tendencies of MAFS as compared to MSSE-1 can be clearly seen in the graphs. Although absolute gain is not large, we got promising principal results in artificial, as well as real financial variables testing. The most important result of our gain is that the improvement was achieved over all lengths of the training window sizes. The integration of MSSE algorithm to MAFS in addition to contribution towards solving *permanently* changing environment problem contributes to solving *temporary* changing environment problem.

Acknowledgements

The authors thank Dr. Aistis Raudys and Simas Tamošiūnas for useful and challenging discussions, sharing the Matlab codes ant the data sets.

References

1. Raudys S.: Survival of intelligent agents in changing environments. Lecture Notes in Artificial Intelligence, Vol 3070. Springer-Verlag, Berlin Heidelberg (2004) 109–117.
2. Fama, E.F.: Efficient capital markets: A review of theory and empirical work. Journal of Finance, Vol. 25. Blackwell Publishing Malden USA (1970) 383–417.
3. Moody J.: Economic forecasting: challenges and neural network solutions. In Proceedings of the International Symposium on Artificial Neural Networks, Hsinchu, Taiwan, 1995.
4. Huang W., Lai K.K., Nakamori Y. and Wang S.: Forecasting foreign exchange rates with artificial neural networks: a review. International Journal of Information Technology & Decision Making, Vol. 3(1). World Scientific Publishing, (2004) 145–165.
5. Lendasse, A., De Bodt, E., Wertz, V., Verleysen, M.: European Journal of Economic and Social Systems, Vol. 14(1). EDP Sciences Les Ulis Cedex, France (2000) 81–92.
6. Raudys S., Zliobaite I.: Prediction of commodity prices in rapidly changing environments. Lecture Notes in Computer Science, Vol. 3686. Springer-Verlag, Berlin Heidelberg (2005) 154–163.
7. Raudys A., Long J.A.: MLP based linear feature extraction for nonlinearly separable data. Pattern Analysis and Applications, Vol. 4(4). Springer London (2001) 227–234.
8. Raudys S.: An adaptation model for simulation of aging process. International Journal of Modern Physics, C. Vol. 13(8).World Scientific (2002) 1075–1086.
9. Raudys S.: Information transmission concept based model of wave propagation in discrete excitable media. Nonlinear Analysis: Modeling and Control, Vol. 9(3). IMI Vilnius (2004) 271–289.

Visualization of Single Clusters

Frank Rehm[1], Frank Klawonn[2], and Rudolf Kruse[3]

[1] German Aerospace Center, Braunschweig, Germany
frank.rehm@dlr.de
[2] University of Applied Sciences Braunschweig/Wolfenbüttel, Germany
f.klawonn@fh-wolfenbuettel.de
[3] Otto-von-Guericke-University of Magdeburg, Germany
kruse@iws.cs.uni-magdeburg.de

Abstract. Evaluation of clustering partitions is a crucial step in data processing. A multitude of measures exists, which - unfortunately - give for one data set various results. In this paper we present a visualization technique to visualize single clusters of high-dimensional data. Our method maps single clusters to the plane trying to preserve membership degrees that describe a data point's gradual membership to a certain cluster. The resulting scatter plot illustrates separation of the respecting cluster and the need of additional prototypes as well. Since clusters will be visualized individually, additional prototypes can be added locally where they are needed.

Keywords: Clustering, Visualization, Cluster Validity.

1 Introduction

Partitioning data sets is an important task in many domains such as costumer segmentation, organizing textual information or gene expression analysis. The agenda behind this process is knowledge discovery via abstraction over an appropriate data representation. However, despite the availability of powerful analytical methods, the evaluation of resulting models turns out to be non-trivial.

Common prototype-based clustering algorithms, such as k-means or fuzzy c-means, minimize an objective function [3]. As a matter of fact, clustering algorithms always fit the clusters to the data, even if the cluster structure is not adequate for the problem. Thus, the quality of a partition cannot be verified meaningfully by the value of the objective function. Therefore, many validity measures are developed to analyze the adequateness of clustering results [4,5,12,14,16,17].

Most of these measures evaluate the partitioning by means of analyzing the fuzzy partition matrix (for fuzzy clustering) or analyzing compactness and separation of clusters considering variance, dispersion, homogeneity or other derivatives drawn from the partitions resulting in a single value, which is of course associated with some loss of information. Primarily, global validity measures cannot give hints which part of the data should be explored more in detail.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 663–671, 2006.
© Springer-Verlag Berlin Heidelberg 2006

VAT, Visual Assessment of Cluster Tendency, is a tool to visualize pairwise dissimilarity information of objects $X = \{x_1, \ldots, x_n\}$ as a square image with n^2 pixels. VAT reorders the data objects so that the image highlights potential cluster structures. As a modification of this, bigVAT allows the visualization for larger data sets [13]. VCV, Visual Cluster Validity, is related to VAT, but takes the inter-datum distances into account that come from partitioning the data set [11].

FUZZSAM, an approach to visualize fuzzy partitions based on Sammon's Mapping is presented in [1]. The proposed tool maps the cluster centres and the data on an arbitrary low dimensional feature space such that the distances between the clusters and the data points will be preserved.

Recently, two visualization methods using the information of a fuzzy clustering partition were presented in [14]. One method arranges the membership degrees to the respective cluster over the distances to the according prototype vector. Each cluster is represented in a single plot. The other method represents a whole fuzzy partition by plotting the highest membership degree to each feature vector over the corresponding second highest membership degree. Such a plot gives an overall impression of a partition.

We propose in this paper a visualization technique to visualize single clusters of high-dimensional data. Our method maps a single cluster to the plane trying to preserve the fuzzy membership degrees that are directly obtained from fuzzy clustering or subsequently derived from crisp partitions. In the following section we briefly recall fuzzy clustering. Section 3 describes the visualization technique. In section 4 we will give some practical details to the implementation. In section 5, we illustrate our method on an artificial data set and on a benchmark data set as well. Finally, we conclude with section 6.

2 Fuzzy Clustering

Generally, fuzzy clustering algorithms partition a data set into several clusters as minimizing an objective function J that describes the sum of weighted distances d_{ij} between c prototypes vectors v_i and n feature vectors x_j of the feature space $\mathbb{R}^p$

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}. \tag{1}$$

Prototype vectors represent the respecting clusters by their location in the cluster's centre. By means of the fuzzifier $m \in (1, \infty]$ one can control how much the clusters overlap. Widely overlapping clusters, which can be obtained with high values for m, will be reflected by almost equal membership degrees u_{ij} to every cluster. Rather crisp partitions can be found with small values for m. Usually, the fuzzifier is set to $m = 2$. In order to avoid the trivial solution assigning no data to any cluster by setting all u_{ij} to zero and avoiding empty clusters, the following constraints are required:

$$u_{ij} \in [0,1] \qquad 1 \le i \le c, \;\; 1 \le j \le n \tag{2}$$

$$\sum_{i=1}^{c} u_{ij} = 1 \qquad 1 \le j \le n \tag{3}$$

$$0 < \sum_{j=1}^{n} u_{ij} < n \qquad 1 \le i \le c. \tag{4}$$

When the Euclidian norm

$$d_{ij} = d^2(\boldsymbol{v}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_j - \boldsymbol{v}_i)^T (\boldsymbol{x}_j - \boldsymbol{v}_i)$$

is used as distance measure for distances between prototype vectors v_i and feature vectors x_j, the fuzzy clustering algorithm is called fuzzy c-means algorithm. Other distance measures can by applied resulting in clustering techniques which can adopt different cluster shapes [9,10]. With the Euclidian distance measure the fuzzy c-means algorithm finds approximately equally sized spherical clusters.

The minimization of the functional (1) represents a nonlinear optimization problem that is usually solved by means of Lagrange multipliers, applying an alternating optimization scheme [2]. This optimization scheme considers alternatingly one of the parameter sets, either the membership degrees

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{1}{m-1}}} \tag{5}$$

or the prototype parameters

$$\boldsymbol{v}_i = \frac{\sum_{j=1}^{n} (u_{ij})^m \boldsymbol{x}_j}{\sum_{j=1}^{n} (u_{ij})^m} \tag{6}$$

as fixed, while the other parameter set is optimized according to equations (5) and (6), respectively, until the algorithm finally converges.

3 Visualizing Single Clusters

We propose in this paper to visualize single clusters by projection of the data points onto the plane under the constraint that the membership degrees to clusters are preserved. Note, membership degrees can be obtained directly when using a fuzzy clustering algorithm (e.g. fuzzy c-means), but also when calculating membership degrees after the partitioning, which can be done for any prototype-based clustering algorithm. To achieve the objective of membership preservation, we adopt the noise distance aspect of the noise clustering technique [6].

Noise clustering is based on the introduction of an additional noise cluster that is supposed to contain all feature vectors that are about a certain distance, the noise distance δ, away from all other prototype vectors. This means that the prototype v_c for the noise cluster c has no parameters. The clustering scheme

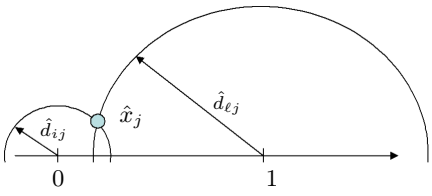


Fig. 1. Placement of $\hat{x}_j$ in the plane

differs only in one point from k-means or fuzzy c-means. When calculating the membership degrees the distance of the feature vector x_j to the noise cluster v_c is the fixed constant value $d_{cj} = \delta^2$. The proper specification of δ is discussed in [6,15].

With the objective to place the cluster in the plane, we need for each data point two coordinates. Note, that the constraint for the projection is not to preserve the distances d_{ij} but the membership degrees u_{ij}. The idea for our visualization is to compute the distances to the cluster prototypes by means of the membership degrees. To achieve this we consider the usual computation of membership degrees as mentioned in equation (5). This provides a very simple connection between membership degrees and distances

$$\frac{u_{ij}}{u_{\ell j}} = \frac{\frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{1}{m-1}}}}{\frac{1}{\sum_{k=1}^{c}\left(\frac{d_{\ell j}}{d_{kj}}\right)^{\frac{1}{m-1}}}} = \left(\frac{d_{\ell j}}{d_{ij}}\right)^{\frac{1}{m-1}}. \tag{7}$$

For the purpose of visualization we propose to place the cluster i to be visualized at $(0,0)$ and to choose a second cluster ℓ at $(1,0)$. The cluster at $(1,0)$ is a virtual cluster that contains all feature vectors with the highest membership degree apart from u_{ij}. The intention of this cluster is to collect all feature vectors that are assigned to another cluster than the one we want to visualize. Let us denote the membership degree to the most competing cluster by $u_{\ell j}$. Furthermore, we introduce a noise cluster to cover the clusters apart from i and ℓ. According to the distance of the two chosen cluster prototypes at $(0,0)$ and $(1,0)$, we define the noise distance $\delta = 1$. This means we have $u_{noisej} = 1 - u_{ij} - u_{\ell j}$. According to equation (7) this leads to

$$\frac{u_{ij}}{u_{noisej}} = \left(\frac{1}{\hat{d}_{ij}}\right)^{\frac{1}{m-1}}. \tag{8}$$

We denote the distance between cluster i and ℓ on the plane by $\hat{d}_{ij}$ to emphasize the fact that we do not deal with original distances any more but with representative distances with respect to the according membership degrees. Solving equation (8) for $\hat{d}_{ij}$ we obtain

$$\hat{d}_{ij} = \left(\frac{u_{noisej}}{u_{ij}}\right)^{m-1}. \tag{9}$$

Analogously, we obtain for the second cluster ℓ

$$\hat{d}_{\ell j} = \left(\frac{u_{noise j}}{u_{\ell j}} \right)^{m-1} . \tag{10}$$

This approach enables us to visualize some useful aspects: 1. which feature vectors can be assigned clearly to the cluster i of intrest, 2. if a feature vector cannot be assigned to i, is there another cluster ℓ, where the vector can be assigned to, 3. which feature vectors are near to one or more prototypes apart from i and ℓ, 4. are there feature vectors that cannot be assigned to any cluster clearly.

With equation (9) one can compute the distance of each feature vector x_j to the cluster i, so that it is possible to draw a circle around $(0, 0)$ as one hint for the feature vector's position in the plane. With the distance to the other cluster $(1, 0)$, one could draw another circle around the cluster centre. The intersection point of these two circles would be the position of the new feature vector in the plane.

Figure 1 illustrates this approach. The small circle that represents the potential coordinates of $\hat{x}_j$, can be drawn with distance $\hat{d}_{ij}$ obtained from equation (9). Analogous, the bigger circle can be drawn with $\hat{d}_{\ell j}$ that we get with equation (10). The intersection point of these two circles represents the feature vector $\hat{x}_j$ in the plane.

4 Implementation Aspects

Apart from the clustering itself, which leads to the membership degrees u_{ij} another parameter affects the transformation, namely m (see equations (8, 9, 10)). A priori, one would set m the same value as for the clustering. But it can be also useful to modify this parameter. Practical tests have shown that in some cases, i.e. when a feature vector is very close to a prototype vector, no intersection point can be obtained in the plane and consequently the membership degrees to the respecting feature vector cannot be preserved exactly.

The rules shown in algorithm (1) handle such cases while trying to preserve membership degrees approximately. Let us denote the transformed data set $\hat{X}$. With $\hat{x}_{2j} = 0$ we define for a feature vector that is very close to one certain prototype vector a position on the x-axis on the plane. The rest of the rule tries to find the proper position for $\hat{x}_j$ on the x-axis balancing the distances to cluster $(0, 0)$ and cluster $(1, 0)$. If the distance to both clusters is relatively small, say $\max(\hat{d}_{ij}, \hat{d}_{\ell j}) < 1$, then we compute a position between both clusters in relation to $\hat{d}_{ij}$ and $\hat{d}_{\ell j}$. Otherwise, which means one or both clusters are about a distance of 1 or further away from the feature vector we distinguish whether cluster $(0, 0)$ or cluster $(1, 0)$ is nearer. If the distance of x_j to cluster $(0, 0)$ is higher than the distance to cluster $(1, 0)$ then $\hat{x}_j$ will be placed to the right of cluster $(1, 0)$ at $\hat{x}_j = (1 + \hat{d}_{1j}, 0)$. If the distance $\hat{d}_{ij}$ to cluster $(0, 0)$ is smaller than the distance $\hat{d}_{\ell j}$ to cluster $(1, 0)$ then $\hat{x}_j$ will be placed to the left of cluster $(0, 0)$ at $\hat{x}_j = (-\hat{d}_{ij}, 0)$. This concept enables us to place the data point at least

Algorithm 1. Placement of $\hat{x}_j$ on the x-axis

```
if no intersection point then
  x̂_{2j} = 0
  if max(d_{ij}, d_{ℓj}) < 1 then
    x̂_{1j} = d_{ij} / (d_{ij} + d_{ℓj})
  else
    if d_{ij} > d_{ℓj} then
      x̂_{1j} = 1 + d_{ℓj}
    else
      x̂_{1j} = −d_{ij}
    end if
  end if
end if
```

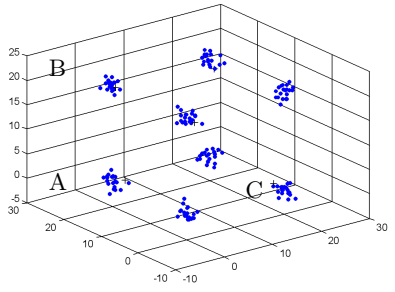
accurate relative to the nearest cluster. However, it is not very interesting to know exactly how far away the feature vector is, since the distance is quite large in fact.

With these rules the membership degrees cannot be preserved exactly, but approximated intuitively. Alternatively, one can avoid this kind of approximation by modifying parameter m for the transformation process. Small values $m \to 1$ prevent that no intersection point can be met. Otherwise, one can set higher values for m to force placements on the x-axis. Such transformations may be not that differentiated, but information can be reduced to some essential facts if needed. Generally, data points situated left from 0.5 on the x-axis can be assigned to cluster $(0, 0)$, while data points on the other side belong to another cluster.

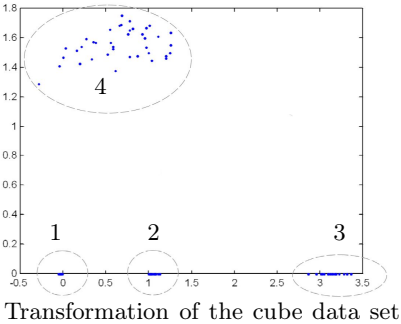
5 Results

Let us first apply our visualization method to an artificial data set. The cube data set (see figure 2(a)) consists of eight well separated clusters, which are in the corners of an imaginary 3-dimensional cube. A fuzzy c-means partition of the data set with five prototypes is shown in the figure. Of course, eight prototypes would be the best choice to partition the cube data set with. Thus, we can illustrate with this partition which information one can get from the visualization tool.

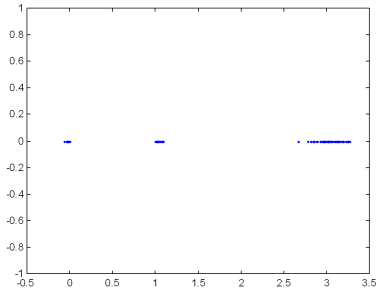
Figure 2(b) shows the transformation of the cube data set from the perspective of prototype A. Clearly four groups of data points can by observed (circled with a dashed line). The data points in group 1 are those, which can by clearly assigned to prototype A. Data points that are located in group 2 are those, which are not assigned to prototype A at all, but to another prototype. Note, a partition that only consists of these both groups is ideal. Group 3 stands for feature vectors, which are not assigned to prototype A and not to any other prototype. Instead, the data points have approximately the same membership degree to two or more prototype vectors (but not to prototype A). Group 4 represents feature



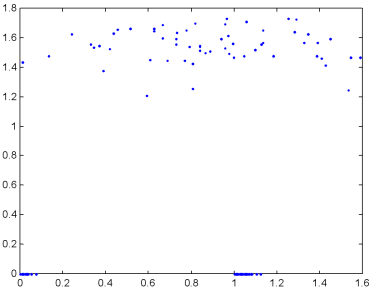
(a) Clustering of the cube data set with 5 prototypes



(b) Transformation of the cube data set from the perspective of prototype A



(c) Transformation of the cube data set from the perspective of prototype B



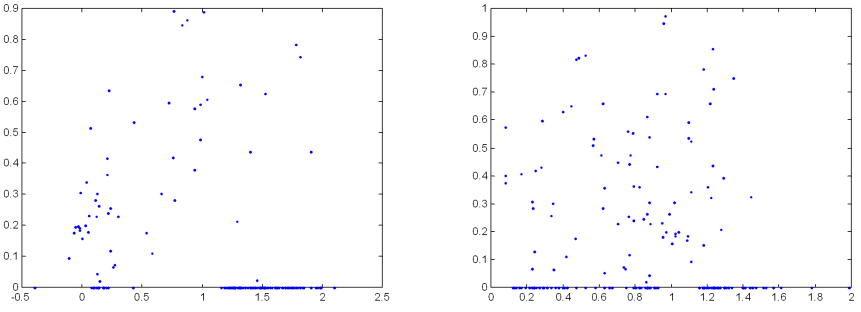
(d) Transformation of the cube data set from the perspective of prototype C

Fig. 2. An illustrative example

vectors that have approximately the same membership degree to prototype A and another prototype.

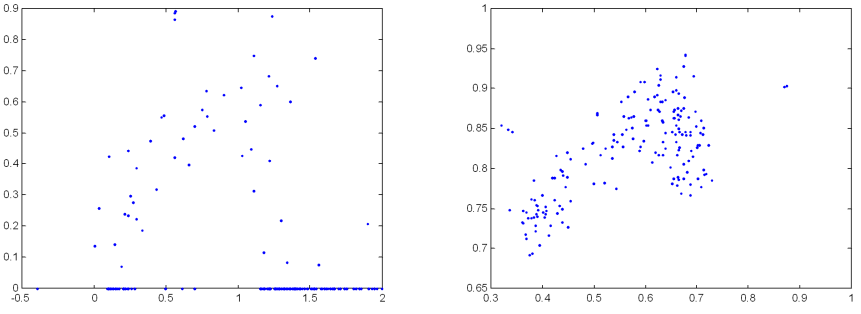
Figure 2(c) shows the transformation of the cube data set from the perspective of prototype B. At first sight one can notice that group 4 is absent. That means in fact that no other prototype than prototype B has high membership degrees to the data points in group 1. A closer look reveals that the distance of prototype B to some misrepresented data is higher comparing to other prototypes, such as prototype A and C. All other data points that could contribute to group 4 are clearly represented by some prototypes. The transformation of the cube data set from the perspective of prototype C is shown in figure 2(d). Now group 3 is missing in the plot. This becomes evident, because all data points that are underrepresented are directly between prototype C and at least one other prototype. As we have discussed above, group 3 only occurs when data points have low membership degrees to the regarding prototype and approximately equal membership degrees to two or more other prototypes. Since prototype C is at least as near as other prototypes, group 3 cannot be formed.

Figure 3 shows some results on the well known wine data set. The figure shows exemplarily two clusters of a partitioning with four prototypes. The left one is a visualization of a quite compact cluster. Data points left from 0.5 on the x-axis



(a) (b)

Fig. 3. Some transformations of clusters of the wine data set



(a) Transformation with $m = 1.2$ (b) Transformation with $m = 1.02$

Fig. 4. The effect of parameter m

whose component on the y-axis is greater than zero have only small membership degrees to the cluster $(1, 0)$ even if their distance to cluster $(0, 0)$ seams to be far. This is due to the relatively small fuzzifier. The cluster shown in figure 3(b) is much more overlapping other clusters as the points on the x-axis, fairly in the middle between both clusters, indicate. As mentioned above, using small values for m leads to rather sensitive transformations. Even a relative small membership degree to a certain cluster attracts the data point in the transformation. To smooth this effect it is advisable to decrease m for the transformation or increase m for the clustering if possible.

The effect of decreasing m for the transformation is shown in figure 4. While figure 4(a) shows the transformation of a cluster of the wine data set with $m = 1.2$, the figure 4(b) shows the same cluster transformed with $m = 1.02$. The changeover from cluster $(0, 0)$ to cluster $(1, 0)$, which is the imaginary line at 0.5 through the x-axis, is rather sparse. This fact indicates a compact cluster with only few feature vectors which cannot be assigned that clear to any cluster.

6 Conclusion

We presented in this paper a new method to visualize fuzzy clustering partitions on the plane. The visualization reveals whether a cluster is compact and if there is some data from the perspective of the respective cluster that is not well represented. Our results on two data sets are promising. Subject of future work will be the development of an appropriate evaluation method.

References

1. Abonyi, J., Babuska, R.: FUZZSAM - visualization of fuzzy clustering results by modified Sammon mapping. Proceedings of the IEEE International Conference on Fuzzy Systems pp. 365–370, 2004.
2. Bezdek, J.C.: A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2, pp. 1–8, 1980.
3. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, 1981.
4. Davies, D.L., Bouldin, W.: A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1, pp. 224–227, 1979.
5. Dunn, J.C.: Well separated clusters and optimal fuzzy partitions. Journal of Cybernetics, 4, pp. 95–104, 1974.
6. Dave, R.N.: Characterization and detection of noise in clustering, Pattern Recognition Letters, 12, pp. 657–664, 1991.
7. Dave, R.N., Krishnapuram, R.: Robust Clustering Methods: A Unified View, IEEE Transactions on Fuzzy Systems, 5, pp. 270–293, 1997.
8. Dave, R.N., Sumit, S.: On Generalizing the Noise Clustering Algorithms. In Proceedings of the 7th Fuzzy Systems Association World Congress (IFSA'97), 3, pp. 205–210, 1997.
9. Gath, I., Geva, A.B.: Unsupervised optimal fuzzy clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11, pp. 773–781, 1989.
10. Gustafson, D.E., Kessel, W.C.: Fuzzy Clustering with a Fuzzy Covariance Matrix. Proceedings of the IEEE Conference on Decision and Control, San Diego, pp. 761–766, 1979.
11. Hathaway, R.J., Bezdek, J.C,: Visual cluster validity for prototype generator clustering models. Pattern Recognition Letters, 24(9–10), pp. 1563-1569, 2003.
12. Höppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis, John Wiley & Sons, Chichester, 1999.
13. Huband, J.M., Bezdek, J.C., Hathaway, R.J.: bigVAT:Visual assessment of cluster tendency for large data sets. Pattern Recognition Letters, 38, pp. 1875–1886, 2005.
14. Klawonn, F., Chekhtman, V., Janz, E.: Visual Inspection of Fuzzy Clustering Results. In: Benitez, J., Cordon, O., Hoffmann, F., Roy, R. (eds.): Advances in Soft Computing: Engineering Design and Manufacturing. Springer, London , pp. 65–76, 2003.
15. Rehm, F., Klawonn, F., Kruse, R.: New approaches to noise clustering for detecting outliers. Soft Computing Journal, Springer, (to appear).
16. Rubens, M.: Fuzzy clustering algorithms and their cluster validity. European Journal of Operational Research, 10, pp. 294–301, 1992.
17. Windham, M.P.: Cluster validity for fuzzy clustering algorithms. Fuzzy Sets and Systems, 5, pp. 177–185, 1981.

Dynamic Data Condensation for Classification

Dymitr Ruta

British Telecommunications Group (BT),
Chief Technology Office, Research & Venturing,
Adastral Park, Orion MLB 1, PP12,
Ipswich IP53RE, UK
dymitr.ruta@bt.com

Abstract. Despite increasing amounts of data and rapidly increasing computational power current state-of-the-art pattern recognition models still can not handle massive and noisy corporate data warehouses, that have become the reality of today's businesses. Moreover, real-time and adaptive systems often require frequent model retraining which further hinders their use. The necessity is therefore to build the classification model on a much smaller representative subset of the original dataset. Various condensation methods ranging from data sampling up to density retention models attempt to capture the summarised data structure, yet they either do not account for labelled data or degrade the classification performance of the model trained on the condensed dataset. The proposed family of models called Dynamic Data Condensation (DDC) combine dynamic condensation, data editing and noise filtering in an attempt to maximally reduce labelled dataset yet with no harm on the performance of a classifier trained on the reduced set. The condensation is achieved by data merging and repositioning imposed by an electrostatic-type field applied to the data which attracts data from the same class but repels data from different classes, thereby trying to improve class separability. Initial experiments demonstrate that DDC outperforms competitive data condensation methods in terms of both data reduction and the classification performance, and is therefore considered better preprocessing step for classification.

1 Introduction

Rapid increase of cheap storage space and computation power led to the massive expansion of terabyte data-warehouses and boosted the demand for accessing and processing these massive information sources. Data analytics and mining continuously gain in importance for virtually any business today [2]. On the one hand increasing number of automated intelligent data-driven processes require information processed in real-time, on the other hand complex analytical and predictive models are being deployed for a variety of on-demand services. All these technologies attempt to use the most advanced data mining and machine learning models, yet the disparity between the requirements and model capabilities is worryingly growing due to inability to process vast amounts of data in real time. Classification methods are at the very heart of pattern recognition and machine learning yet the most advanced models like neural networks, support vector machines or density based classifiers [1] are of high - at least quadratic - complexity with respect to the number of samples. At such complexity classification models struggle with the data sizes in the order of tens of thousands yet face now the data sources

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 672–681, 2006.
© Springer-Verlag Berlin Heidelberg 2006

with millions or even billions or records. To deal with this problem the data needs to be reduced to the manageable sizes, yet the challenge is to do that with minimum information loss and no harm on the subsequent classification performance.

The simplest and possibly the most common condensation methods are random sampling techniques [3]. They are suitable for large datasets with simple structure, yet fail on smaller noisy data particularly with large density or class imbalances as they tend to ignore less populated data subspaces. It is believed that the proper condensation method should make use of all of the original data points [1]. The state-of-the-art data condensation methods in its majority try to select subset of data that maximally retains the original data density or other characteristics like quantisation error etc [4]. A number of methods in this group work around the principle of removing samples from the less dense regions in favour of strengthening the evidence in the denser regions such that the deviation from the original data density is minimum possible [4], [6]. Other methods use multi-resolution spatial analysis to split data into partitions or clusters [7] and use centres of these clusters as new condensed dataset. Classification-based condensation methods are relatively new and serve directly the purpose of retaining or improving classification performance given reduced training set. The typical examples of such classification focussed condensation are the reduced nearest neighbour rule, iterative condensation algorithm [5] or locally variable condensation models based on neural networks [8]. In all these efforts none of the methods try to actively change the data from its original position. It is reasonable to assume that releasing data from their original positions could further improve classification performance or at least allow for further condensation given similar classification performance. In [6] Girolami and He obtained improvement of the Parzen density fit of the reduced set by releasing and finding optimal data weights on the course of constrained optimisation process. A natural extension of such model would be to include data vectors themselves into the variables to be optimised yet this would undoubtedly trap the process into large number of local optima building up on the excess of the degrees of freedom for this ultra-high dimensional search space. In a response to this challenge a new condensation model is here proposed which applies electrostatic-like data field to condense and transform labelled dataset in order to retain or boost the performance of a classifier trained on such dataset. A set of 5 model variations is presented and tested in a form of dynamic iterative simulations carried out on standard dataset used for classification benchmarking.

The remainder of the paper is organised as follows. Section 2 introduces the concept of potential fields used in classification. The following section provides detailed description of the dynamic data condensation presented in 5 different variations. Section 4 shows the results of some comparative experiments demonstrating condensation and classification performance of the proposed model. Finally conclusions and further research are briefly drawn in the closing Section 5.

2 Data Fields for Classification

The concept of a field in classification is not new and in fact is related to the kernel methods [9]. The rationale behind using the field concept is to ensure that every data sample is actively contributing in the formation of final classification decision. By an

analogy with the particles in the physical world one can consider the data as charged particles each being the source of a central field affecting other samples in the input space. All the characteristics of such field are the results of the definition of potential and can be absolutely arbitrarily chosen depending on various priorities. For classification purposes the idea is to assign the class label to a previously unseen sample based on the class spatial topology learnt from the training data. This goal is achievable within the data field framework if we assume testing samples to be mobile and forced by the field to move towards affixed training data to share their label. The overall field measured in a particular point of the input space is a result of a superposition of the local fields coming from all the sources. Thus the positions of the training data uniquely determine the field in the whole input space and thereby determine trajectories of the testing data during classification process. If the field is designed in such a way that all possible trajectories end up in one of the sources then the whole input space can be partitioned into regions representing distinct classes. The boundaries between these regions form the ultimate class boundaries, completing the classifier design process.

2.1 Attracting Field Model

Given the training data acting as field sources, every point of the input space can be uniquely described by the field properties measured as a superposition of the influences from all field sources. Let us make an initial assumption that the field sources are fixed to their initial positions i.e. let us have a static glimpse of the field with all dynamic aspects imposed by the field ignored. Given a training set of n data points: $X = (\mathbf{x_1}, .., \mathbf{x_n})$ let each sample be the source of a field defined by a potential $U_j = -cs_i f(\overrightarrow{r_{ij}})$ where c represents the field constant, s_i stands for the source charge of $\mathbf{x_i}$, and $f(\overrightarrow{r_{ij}})$ is a certain non-negative function decreasing with an increasing distance $|\overrightarrow{r_{ij}}| = |\mathbf{r_{ij}}|$ from the source $\mathbf{x_i}$ to the point $\mathbf{y_j}$ in the input space. In the attracting field equivalent to the gravity field we simply have $f(\mathbf{r_{ij}}) = 1/|\mathbf{r_{ij}}|$. Overall the potential U_j and field interaction energy E_j in a certain point $\mathbf{y_j}$ of the input space is a superposition of the potentials coming from all the sources:

$$U_j = -c \sum_{i=1}^{n} \frac{s_i}{|\mathbf{r_{ij}}|} \qquad E_j = -cs_j \sum_{i=1}^{n} \frac{s_i}{|\mathbf{r_{ij}}|} \tag{1}$$

We can simplify model further by assuming that all data points are equally important and have the same charge equal to the unit: $s_i = 1$ thus eliminating it from the equations 1. Another crucial field property is its intensity E_j, which is simply a gradient of the potential and its solution leads to the following:

$$\overrightarrow{E_j} = \mathbf{E_j} = -\overrightarrow{\nabla} U_j = -\left(\frac{\partial U_j}{\partial y_{j1}}, ..., \frac{\partial U_j}{\partial y_{jm}} \right) = -c \sum_{i=1}^{n} \frac{\mathbf{y_j} - \mathbf{x_i}}{|\mathbf{r_{ij}}|^3} \tag{2}$$

A field vector shows the direction and the magnitude of the maximum decrease in field potential. By further analogy to gravitational field, the charged data point is affected by the field in the form of force attempting to move the sample towards lowest energy levels. As the charge has been assumed uniformly of unit value and excluded from the equations the force vector becomes identical to field intensity vector: $\mathbf{F_j} = s_j \mathbf{E_j} = \mathbf{E_j}$.

2.2 Attracting-Repelling Field Model

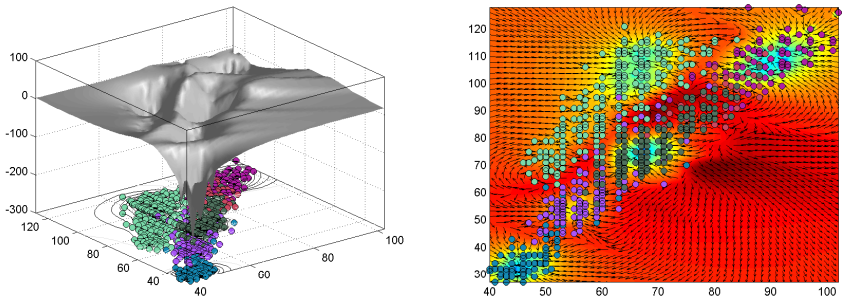
So far data have been only attracting each other which was a consequence of the same charge and as a consequence negative potential definition resulting in the energy wells intercepting samples found in the neighbourhood. Such field does not use the information about the class labels as all the samples are considered to hold the same charge. Ideally, the attracting force should be acting only upon the data from the same class. At the same time the samples from different classes should be repelled from each other to stimulate increased separability between classes. Again nature offers a perfect guide in a form of electrostatic field, where opposite charges attract each other and charges of the same sign repel from each other. To adopt this rule to the labelled data, the samples from the same class should interact with negative potential as in previous case, whereas samples from different classes should generate the positive potential of the same absolute value triggering repelling force. The major problem with electrostatic data field is that testing samples do not have labels and cannot straightforwardly interact with labelled training samples. Estimating the label of the testing sample means that classification is accomplished. To avoid this trivial situation we assume that each testing sample is decomposed into fractional subsamples with the charges proportional to different class potentials normalised to sum up to a unit.

Given the training set $X = (\mathbf{x_1}, .., \mathbf{x_n})$ with labels $L_S = (l_1, .., l_n)$ where $l_i \in (1, .., C)$, labels partition matrix $P^{N \times C}$ for the testing set $Y = (\mathbf{y_1}, .., \mathbf{y_N})$ can be simply obtained by $p_{jk} = |U_j^k| / \sum_{i=1}^{C} |U_j^i|$ where U_a^b stands for potential generated by samples $\mathbf{x_i}$ coming from class indexed by b measured in point $\mathbf{y_a}$. Given this label partition matrix the new definition of potential and field vector take the following form:

$$U_j = \sum_{i=1}^{n} \left(\underbrace{\frac{\sum_{k \neq l_i} p_{jk}}{|\mathbf{r_{ij}}|}}_{repelling} - \underbrace{\frac{p_{jl_i}}{|\mathbf{r_{ij}}|}}_{attracting} \right) = \sum_{i=1}^{n} \frac{1 - 2p_{jl_i}}{|\mathbf{r_{ij}}|} \quad \mathbf{E_j} = \sum_{i=1}^{n} \left[(1 - 2p_{jl_i}) \frac{\mathbf{y_j} - \mathbf{x_i}}{|\mathbf{r_{ij}}|^3} \right] \quad (3)$$

The numerator of the potential definition (3) can be both positive and negative depending on the class partial memberships. In the presence of many classes, regardless of their topology, the absolute values of the partition matrix P will naturally decrease to share the evidence among many classes. Effectively the potential would grow positive with the repelling force dominating the field landscape. In our model the data still has to slide down the potential towards the source samples. To satisfy this requirement, it is sufficient to normalise the field such that the overall potential of the field should not be larger than zero. Taking into account the fact that the field is substantially negative in the close neighbourhoods around the training samples, it is sufficient to satisfy the condition of $\sum_{j=1}^{N} U_j = 0$. To achieve this goal potential definition has to be parameterised and solved with respect to the regularisation coefficient q as in the following:

$$\sum_{j=1}^{N} U_j = \sum_{j=1}^{N} \sum_{i-1}^{n} \frac{1 - q p_{jl_i}}{|\mathbf{r_{ij}}|} = 0 \quad (4)$$



(a) 3D potential plot (b) Vector field plot

Fig. 1. Visualisation of the attracting-repelling field applied to the Land Satellite dataset with 2000 examples, 4 features and 6-classes. 1(a) Plot of the potential created by the samples. 1(b) Vector plot of the resulting field intensities.

In the model we use bisection method to find numerical estimation of the parameter q. Note that parameter q has a meaningful interpretation as the value $1 - q$ says in general how many times should the attracting interaction be stronger to compensate the excess of the repelling interaction coming from the multitude of different classes. Figure 1 demonstrates the examples of both field models applied to the Land Satellite dataset.

3 Dynamic Data Condensation

The data fields presented above will be used to condense the labelled data in the process of dynamic field interaction with its sources. For this purpose the data, are released from their initial positions and are moved by the field forces towards decreasing potential in the multidimensional input space. Whenever two or more data points meet, as a result of such repositioning, they instantly become a single data point with the summed charge strengthening the field around the new point yet increasing also its *mass* which reduces relative ability to shift. The whole process becomes in fact a simulation in which data-points move towards each other and gradually merge, thereby performing the act of condensation. Due to the fact that the introduced potential fields retain the total energy, the undesirable kinetic energy that would normally be gained as a result of such simulation is cancelled out after each simulation step.

The dynamic condensation process is defined as a sequence of simulation steps which start from the original data locations and finish when no more changes in the data positions is recorded after each step or it can be stopped arbitrarily by the user at desired level of data condensation. At each step the field and corresponding forces vectors are recalculated at all existing data point locations. Using the force vectors the points are then shifted by a small step d in the directions determined by the force vectors. After the shifting phase all the data are tested for mergers which are assumed to happen when the distance between two points is lower than an arbitrarily small merger range for simplicity set to d.

Computationally the critical process is the calculation of the distances among all the sources. Using matrix formulation of the problem and the appropriate mathematical software, this task can be obtained rapidly even for large datasets. Denoting by $X^{[n\times m]}$ the matrix of n m-dimensional data points, the task is to obtain the matrix D of the distances between each pair of of the considered data points. Introducing "$\circ$" as element-wise matrix multiplication and $\mathbf{1}^{[n\times m]}$ as an n by m matrix with all unit elements, the distance matrix can be calculated instantly by:

$$D^2 = X \circ X \bullet \mathbf{1}^{[m\times n]} - 2 \bullet X \bullet X^T + \mathbf{1}^{[n\times m]} \bullet X^T \circ X^T \tag{5}$$

Given the distance matrix the process of data evolution is very straightforward. The data points are simply sliding down the potential wells they have created to meet one or more of the neighbouring field sources. Ignoring the dynamics of sliding data i.e. removing the kinetic energy they gain during the drift towards each other each data point is moved by $\triangle X^{[n\times m]}$ again efficiently calculated using matrix formulation by:

$$\triangle X = \frac{d \bullet \mathbf{F}}{\sqrt{(F \circ F) \bullet \mathbf{1}^{[m\times m]}}} \tag{6}$$

where d is an arbitrarily small step. Negative potential definition appearing in both fields ensures that the data is trapped to its original data space and will not escape as a result of the simulation. To avoid numerical problems the data should be normalised within the same limits in each dimension and distances limited from the bottom by a small interception threshold comparable to d which prevents division by zero and "overshooting" the field sources during simulation.

3.1 Crisp Dynamic Data Condensation (DDC_C)

Let us first assume that individual data point is an indivisible fault free unit of evidence. During the condensation process these data points are let free and as they move and merge the field changes according to their new locations. Depending on how the class labels of the data are used the crisp electrostatic condensation models can be further subdivided into unlabelled and labelled which differ fundamentally from each other.

Let us first consider Crisp Unlabelled DDC (DDC_{CU}) model which uses attracting-only data field as shown in (1). To be able to effectively use such field for the condensation purposes the dynamic simulation model has to be applied to all classes in isolation as otherwise all the data merge to the centre-mass point and completely destroy the class structure. The simulation can be carried out simultaneously for all the classes yet in that case the intra-class interactions have to be insensitive to the other classes interactions. In this model data points of each class are collapsing gradually up to a single data point per class which terminates with the weights proportional to the counters of original points in the condensed data.

In the Crisp Labelled Electrostatic Condensation (DDC_{CL}) the labels of the training data are used to determine the sign of the potential. Negative potential is generated by the data from the same class and positive for the data from different class such that the data points from the same class are attracting each other and data from different classes repel from each other. Note that during condensation process all the labels of the condensing dataset are known hence the definition of potential becomes a special case of

(3) where all the charge partitions p_{ji} are crisp i.e. are either 1 or 0. An advantage of this approach for classification purposes is during such process the data from different classes would try to separate out from each other. However, in case of elongated class shapes some data tails or other isolated data points could be pushed away by the dense neighbouring regions from different classes. Another problem that emerges in the labelled electrostatic model is collisions of data from different classes. Such collisions could happen if the attracting forces generated by large data concentrations from the same class override weaker repelling actions from the samples of different classes.

3.2 Soft Dynamic Data Condensation (DDC_S)

It is reasonable to assume that the labelled classification data is noisy and often faulty and hence to treat it in the soft probabilistic terms. The data labels is summarised in the form of soft class membership or partition values. These class partitions are obtained from the original class densities obtained via Parzen window density estimator and then mapped onto the probabilities. According to Parzen-window approach [2] an estimate of the data density in point x_j can be obtained by calculating:

$$p(\mathbf{x_j}) = \mathbf{p_j} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{\mathbf{x_j}-\mathbf{x_i}}{h_n}\right) \tag{7}$$

where V_n is a window volume, φ is a specific window function and h_n is a smoothing parameter. This model uses the common Gaussian window function defined as follows:

$$\varphi(u) = \frac{1}{2\pi}e^{-u^2/2} \tag{8}$$

The leave-one-out maximum likelihood estimation is applied to find the optimal smoothing parameter h_n. Let p_k denote a vector of Parzen density estimates measured on dataset X but generated only from k^{th} class of the dataset X, $k \in (1, .., C)$. Class partitions are obtained from Parzen densities using typical transformation mapping used in classification and scaled up to sum up to a unit:

$$p_k^N = \frac{1}{1+e^{-p_k}} \qquad p_k^S = \frac{p_k^N}{\sum_{i=1}^{C} p_i^N} \tag{9}$$

Given the matrix of partitions as defined above, one could easily apply electrostatic potential (3) yet there is still a freedom in deciding how to control the variability of such soft labels during the condensation process. The most conservative model called Soft Fixed-Field Condensation (DDC_{SFF}) assumes fixed field built on the original data which is kept fixed during the condensation process. Such approach would try to maximally preserve the original Parzen density structure even if it requires continuous relabelling of moving samples. Here the merging process is free of any conflicts as the class partitions of the merging samples are simply adding up to the new "heavier" sample which stays partitioned after the merger.

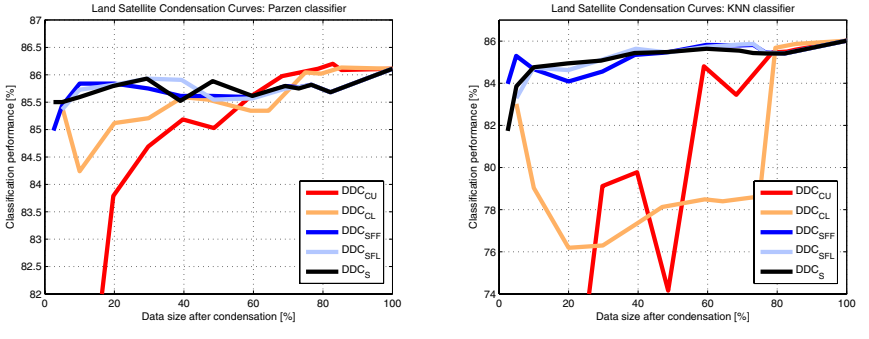
The Soft Fixed-Labels Condensation (DDC_{SFL}) keeps label partitions fixed during the condensation, once they are calculated before the simulation. The merging process remains additive as in the previous case.

Finally the least constrained Soft Dynamic Data Condensation (DDC_S) model releases all the constraints and lets both the field and data partitions change as the data samples evolve during condensation process. As in previous soft models colliding data merge into a single point with summed class label partitions.

For all soft condensation models the field is continuously renormalised using a regularisation parameter calculated as shown in (4). Both soft and crisp dynamic condensation processes finish when the sum of data shifts at a single step is smaller than an arbitrarily small distance, for simplicity chosen to be equal to d used in (6).

4 Experiments

The presented electrostatic condensation models have been evaluated in terms of classification performance obtained at different levels of data reduction and then compared with the performance obtained on the original data. All the models have been applied to the well-known Land Satellite Image dataset from UCI Repository [1]. The training set consisted of 2000 data points and the left-out testing set was sized 4435. All the dynamic condensation models have been applied to the multiple training sets collected during the condensation at increasing level of dataset reduction. These training sets were used to train the two selected benchmark classifiers: Parzen density classifier (PDC) and k nearest neighbour (KNN). The trained classifiers were then tested on the testing set to produce the performance condensation profiles as shown in Figure 2.

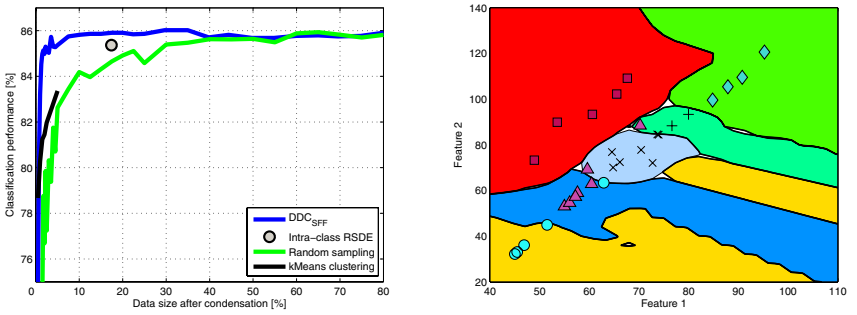


(a) Parzen density classifier (b) K nearest neighbour classifier

Fig. 2. Visualisation of the classification performance obtained on increasingly condensed Land Satellite datasets at different condensation levels for 2(a) Parzen density classifier 2(b) K nearest neighbour classifier

The soft condensation methods clearly outperformed the crisp methods with the constrained versions of DDC_{SFF} and DDC_{SFL} consistently taking the leading positions. In the following experiment the best dynamic condensation method i.e. DDC_SFF

[1] University of California Repository of Machine Learning Databases and Domain Theories, available free at: ftp.ics.uci.edu/pub/machine-learning-databases.



(a) Condensation profiles comparison

(b) Landsat data after 99% condensation using DDC_{SFF}

Fig. 3. Visualisation of the Parzen density classifier performance obtained on increasingly condensed Land Satellite datasets using DDC_{SFF}, intra-class RSDE method, random sampling and k-means clustering. 3(a) Condensation profiles comparison. 3(b) Visualisation of the LandSat dataset reduced by 99% using DDC_{SFF}, overlayed on the class boundary map of a Parzen density-based classifier trained on the full training set.

was compared with the typical random sampling, condensation based on k-means intra-class clustering, and the state-of-the-art RSDE density estimation method described in [6]. Both k-means and RSDE have been applied to reduce each class separately yet for RSDE it was only possible to obtain a single point of the condensation profile i.e. for the condensation level at which the RSDE method converged. For k-means clustering, the condensation level was controlled by selecting the numbers of cluster centres for each class that were proportional to the prior class probability. The random sampling was applied to the incrementally reduced training sets preserving the prior class probabilities. The results of this experiment are shown in Figure 3.

Clearly DDC_{SFF} outperformed the other methods and even showed higher testing performance than the model trained on the data reduced by the top-in-the-field RSDE method [6]. The top-ranked condensation algorithm managed to retain the 99% of the original classification performance for almost 99% reduced training set. An interesting observation comes our off the reduced set visualisation showed in Figure 3(b) which shows that some classes were partitioned into isolated clusters as a result of the repelling inter-class forces. The class remainders retained their identity outside of the main class mass yet they found the positions which do not disturb the purity of the main class densities as opposed to the fixed-position condensation methods which would most likely ignore the class remainders treating them as outliers. This property could be the key to retaining maximum class representativeness at high data condensation levels.

5 Conclusions

This work promotes a new type of data condensation which dynamically merges and repositions the data as a result of the electrostatic-like field generated by and acting upon

the labelled data in the input space. The electrostatic metaphor of charge-dependent attracting or repelling field was adopted in the classification context to generate intra-class attraction and inter-class repulsion to further encourage class separability. To boost the retention of the original class density, soft versions of the proposed dynamic condensation were developed which allow for mergers of data from different classes yet retaining soft/fuzzy class partitions. Morover in DDC_S and the most successful DDC_{SFF} the field is directly guided by the Parzen estimates of the original class densities which continuously redistribute class partitions as the data marge and move during the condensation process. On the course of comparative experiments carried out with typical classifiers on Land Satellite dataset the soft versions of the presented condensation algorithms outperformed comparable state-of-the-art algorithms in terms of classification performance on maximally reduced training sets. The best CCD_{SFF} model managed to retain 99% of the original classification performance on 99% reduced training set. Future advancements will include attempts to incorporate categorical data and provide automated field tuning mechanisms along with an attempt to make the presented algorithms more scalable and numerically stable.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. John Wiley & Sons (2001)
2. Morgan, M.: Unearthing the customer: data mining is no longer the preserve of mathematical statisticians. Marketeers can also make a real, practical use of it (Revenue-Generating Networks). Telecommunications International, May (2003)
3. Weiss, S.M., Kulikowski, C.A.: Computer systems that learn. Morgan Kaufmann Publishers, San Mateo (1991)
4. Mitra, P., Murthy, T.R., Pal, S.K.: Density-based multiscale data condensation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(6) (2002) 734-747
5. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-basd learning algorithms. Machine Learning 38(3) (2000) 257-286
6. Girolami, M., He. C.: Probability density estimation from optimally condensed data samples. IEEE Trans. on Pattern Analysis and Machine Intelligence 25(10) (2003) 1253-1264
7. Leung, Y., Zhang, J.-S., Xu, Z.-B.: Clustering by scale-space filtering. IEEE Trans. on Pattern Analysis and Machine Intelligence 22 (2000) 1396-1410
8. Plutowski, M., White, H.: Selecting concise training sets from clean data. IEEE Trans. on Neural Networks 4(2) (1993) 305-318
9. Shawe-Taylor, J., Cristianini, N.: Kernel methods for pattern analysis. Cambridge University Press, Cambridge (2004)

Handwriting Recognition Accuracy Improvement by Author Identification

Jerzy Sas

Wroclaw University of Technology, Institute of Applied Informatics, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
jerzy.sas@pwr.wroc.pl

Abstract. In this paper, two level handwriting recognition concept is presented, where writer identification is used in order to increase handwriting recognition accuracy. On the upper level, author identification is performed. Lower level consists of a classifiers set trained on samples coming from individual writers. Recognition from upper level is used on the lower level for selecting or combining classifiers trained for identified writers. The feature set used on the upper level contains directional features as well as the features characteristic for general writing style as line spacing, tendency to line skewing and proportions of text line elements, which are usually lost in typical process of handwritten text normalization. The proposed method can be used in applications, where texts subject to recognizing come form relatively small set of known writers.

1 Introduction

Automatic handwriting recognition is one of the most intensively explored area of artificial intelligence. Despite many years of research, there are still no mature methods allowing for handwritten text recognition with acceptably low error rate. One of the main reasons of the problem difficulty seems to be the great diversity in handwriting styles. Different writers write the same character in different manners. Examples presented on Fig. 1 show digit 4 and digit 9 written by different persons. The shape of digit 4 by writer A is very similar to digit 9 written by writer B, while digits 4 and 9 written by B are easy distinguishable. This misleading observation may cause degradation of character or word classifier accuracy, if it is trained using common learning set consisting of character or script samples coming from great number of writers.

In case of automatic handwriting recognition applications, where number of writers is known and relatively small, better results can be expected if separate classifiers are trained for each writer individually, using only text samples coming from single person. In this way, the classifier can better fit individual author writing style. Additionally, if particular writer style is difficult to analyze (i.e. different characters are written in similar way), then separation of classifiers by authors prevents deterioration of overall classification performance by limiting uncertainty in recognition to texts written by "careless" writer. The concept of individual writer classifiers can be applied both to character recognition, in

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 682–691, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Fig. 1. Digits written by two writers A and B a) digit 4 by A, b) digit 9 by B c) digit 4 by B

case where characters can be easily isolated (e.g. in case of forms recognition as described in [15]) and to cursive script recognition, where holistic approach to complete words recognition is used. In remaining part of this article we will focus on isolated characters recognition.

Applying classifiers trained for particular writers requires the ability to reliably identify the text author before individual classifier is applied. The problem of writer identification is being investigated by many researchers ([2], [8], [9], [12], [13]). In most of publications however, writer identification is the ultimate aim (as e.g. in forensic document analysis, forgery detection, signature verification etc). Here we will adopt writer identification methods as a stage of more complex text recognition problem. For our purposes the assessment of writer identification stage is different than in cases where exact writer identification is an ultimate goal. Here we rather expect that writer identification results in finding such writer which writing style is similar to the style of actual writer, so as to maximize the performance of character recognition on the lower level. For this reason it is more convenient to use soft classification paradigm ([5]) instead of typical crisp classification.

The main aim the work being described here is to examine how writer identification influences the final character recognition performance. Secondly, we want to investigate methods of using soft writer recognition results in combining character classifiers trained for individual authors and recommend the one for practical applications. We do not consider here in detail the problem of feature selection for writer identification. The features recommended by other authors ([8], [13]) extended with some features describing text layout, specific for individuals are used.

2 Two-Level Classifier with Writer Identification

Let us consider a problem of handwritten character recognition. The text containing characters to be recognized is written in upper case isolated letters, so precise segmentation of words into characters is not a problem. In practice, we usually deal with such kind of text in case of form recognition, where the text layout is strictly determined by the form structure. In majority of cases the aim of text recognition consists in correct recognition of words or complete sentences. Because low error rate of character classifier is crucial for lexical or semantic level text recognition accuracy, here we will restrict our considerations to isolated characters classification.

Two-level classification scheme is proposed. On the upper (text) level we deal with writer identification problem. Text being recognized comes from finite and relatively small set of M writers. The writer is being identified using writing style features $v^F = (v_1^F, v_2^F, ..., v_L^F)$ extracted from the text image F. Writer is identified by soft classifier $\Psi_R(v^F)$, which assigns the support value $d_i(v^F)$ to each of writers $i = 1, ..., M$:

$$\Psi_R(v^F) = [d_1(v), d_2(v^F), ..., d_M(v^F)]^T. \tag{1}$$

The support value $d_i(v^F)$ can be interpreted as the classifier confidence that text comes from i-th writer. For computational purposes we request that the support vector is normalized, i.e. its components are non-negative and they all sum up to 1.0.

On the lower (character) level we have the set of character classifiers $\Psi_C^r(x), r = 1, ...M$. The input of the classifier is the vector of features x extracted from isolated and appropriately preprocessed character image. The set of classes in the classification problem on the lower level corresponds to the alphabet containing L admissible characters. Each classifier $\Psi_C^r(x)$ is trained using the learning data coming from the r-th writer, so it is expected to perform best when classifying characters written by this individual. Soft classification is also applied here, i.e. each soft character classifier provides the support vector:

$$\Psi_C^r(x) = [d_1^r(x), d_2^r(x), ..., d_L^r(x)]^T, \tag{2}$$

where $d_i(x)$ is the support value for i-th character from the alphabet.

The basic problem of two-level classifier construction is how to utilize the soft writer classification result on the lower level. We deal here with rather typical classifier combination problem. There are great number of methods for classifiers combination ([5], [6], [10], [11]). We have selected frequently used weighted voting scheme. Because soft classification paradigm is applied on the lower level, the final result of recognition can be obtained as weighted combination of component classifiers outputs. Let $\Psi_C(x)$ denote the combined soft classifier on the character level producing the final support vector $[d_1(x), d_2(x), ..., d_L(x)]^T$. The support value $d_i(x)$ for i-th character is calculated as:

$$d_i(x) = \sum_{r=1}^{M} \beta_r d_i^r(x), \tag{3}$$

where β_r are weight factors dependent on the support values $d_r(v^F)$ fetched by the writer soft classifier on the upper level. In our experiments we tested two strategies of β_r calculation:

- single r^*-th classifier is used, for which $d_{r^*}(v^F)$ is maximal; so β_{r^*} is set to 1.0 while remaining β_i factors are set to 0.0,
- weighting factors are calculated according to the formula:

$$\beta_r = \alpha d_r(v^F) + (1 - \alpha) p_r, \tag{4}$$

where p_r is prior probability of appearance of text written by r-th writer and $\alpha \in [0, 1]$ depends on average assessment of writer identifier. In particular, α can be the estimated probability of correct writer identification converted from soft to crisp classification rule, i.e. this writer is recognized, which has the greatest support factor in (1).

3 Classifier Selection and Feature Extraction

The general two-level classification scheme described in the previous section can be applied with any soft classifiers on both levels. Because feature selection and analysis of its impact on writer identification was not our main aim, then we just borrowed concepts of writer identifiers and features sets from other works ([2], [9], [12]). It should be pointed out, that results reported by other researchers concern the problem of writer identification based mainly on cursive script sample, while our problem concerns upper case block character recognition. It seems that there are more measurable features characterizing writing style in cursive script than in block characters, so writer identification in our case may be more difficult.

On upper level we have considered two concepts of feature extraction for writer identification: texture features extracted from text image using a bank of Gabor filters ([2]) and directional features derived from character contour lines ([9], [13]). Writer identification accuracy achieved on our test data set using texture features and Gabor filters was much worse than reported in [2]. Much better results were obtained with directional contour features extracted according to procedures described in [13]. The features sets, which seem to be the most appropriate for isolated characters were used: edge direction distribution and edge-hinge distribution. Edge direction distribution informs about distribution of stroke directions in character images. It is strictly related to writing slant angle, very characteristic for a writer. The second feature group describes distribution of two directions d_1 and d_2 of contour segments incident to a point on the contour, as depicted on Fig 2. From perceptual point of view, it characterizes the writer tendency to write angular or smooth rounded allographs. Directions are quantized to 12 direction zones. In result, 12 features are obtained for edge distribution and 252 features for edge-hinge distribution. Details of feature extraction procedure are given in [9]. This features set containing 264 elements will be referenced as set A.

Directional feature set is complemented by a few additional features characteristic for handwriting and easy observable by human: average character width, standard deviation of characters width, average width to height character aspect

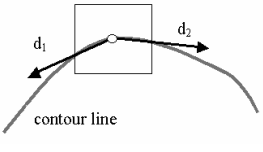


Fig. 2. Incident directions extraction for edge-hinge features set

ratio, standard deviation of characters height, average distance between characters within a word and average distance between words. This feature set will be referenced as set B.

Multi Layer Perceptron (MLP) is applied as the recognition method on the upper level. Due to different nature of feature sets A and B we decided to use two isolated MLPs for two sets of features. Outputs of MLPs for sets A and B are inputs of another MLP, which provides final soft writer recognition. In fact, upper level is also subdivided into two sublevels. MLP for set A has 264 inputs, one hidden layer with 20 nodes and M outputs (M is the count of writers). MLP for set B is a neural net with 6 inputs, 10 nodes in hidden layer and M outputs. MLP in upper sublevel has $2M$ inputs and M outputs. All MLPs were trained independently, in "1 of M" manner, i.e. only one of outputs corresponding to actual writer is expected to have value 1.0 while all others are expected to be 0.0.

Soft classification vector (1) can be easily obtained from upper sublevel MLP by appropriately normalizing its output. First, the output values are clipped to $[0, 1]$ range and then the vector of clipped values is normalized to sum up to unity.

On the lower level, there are M independently trained character classifiers. Again, MLP is used as the classification method. Although other researchers report that slightly better accuracy of "crisp" character classification can be obtained using SVM classifier, we selected MLP. This is because SVM superiority was not confirmed in case of soft classification of characters set gathered for our experiments consisting of 35 Polish letters. Also the training time, much shorter for MLP than in case of SVM, is an important factor as far as frequent adaptation to changing writers set is considered.

Gray level gradient features proposed by Liu in [7] were used as inputs to MLP. Gradient features extracted according to the algorithm proposed by Liu proved to be superior, as far as discriminative power is considered, and were successfully applied in our previous works and practical applications ([14], [15]). In recognizing Polish character set, the classifier achieved 90.1% of recognition accuracy. Feature vector consists of values describing distribution of stroke directions in 25 evenly spaced regions of the character image. The input for feature extraction procedure is a gray scale image of isolated character. The features are extracted in four steps:

1. Character image is clipped to smallest rectangle covering all character stroke pixels. Next, obtained image height is normalized to standard value. Image is scaled along x axis according to formula proposed by Liu in ([1]):

$$x_{out} = \sqrt{\sin(\frac{\mathit{II}}{2} x_{in})}, \tag{5}$$

   where x_{in} is clipped image width and x_{out} is the width of resultant image. The method produces images of similar width but prevents unnatural widening of characters which are inherently narrow (e.g 'i', 'l', '1').
2. Sobel operator is applied to grayscale character image resulting in calculating gradient of brightness at each pixel.

3. Obtained gradient direction is assigned to one of 8 regions (0-45 deg, 45-90 deg, 90-135 deg etc.). For each region there is corresponding directional subimage. Assigning direction to the region results in increasing the value accumulated in the corresponding sub-image pixel.
4. All directional sub-images are sampled at evenly spaced 25 points using Gaussian filter. By calculating 25 samples on all 8 directional sub-images 200 values are obtained, which are then used as feature vector.

The method produces 200 elements feature vectors which are inputs to three-layer MLP classifier with 50 nodes in hidden layer. Output layer contains as many nodes as is the count of characters in the alphabet. Soft classification vector (2) is derived from MLP output in the same way as described for upper level. Character classifiers are trained and their outputs interpreted in "1 of N" manner. Final character soft classification vector is calculated according to the formula (3).

4 Experimental Results

The aim of experiments was to assess the improvement of character recognition accuracy resulting from writer identification.

4.1 Data

The method described here was elaborated for applications, where the set of writers is relatively small. The set of texts used in our experiments come from 25 writers. Because the method is a part of wider project related to automatic recognition of handwritten medical documents collected on the hospital ward in defined period of time, this number seems to approximately correspond to typical number of physicians writing patient records.

Writers participating in the experiment were requested to write five samples of texts. The samples contents were taken from real medical documentation, mainly from treatment summary section of patient records. Each sample text contained 200-250 letters. Writers were asked to write the text in isolated capital letters. The forms used in the experiment have printed doted line borders, distant 2.5 cm each from other, so character sizes were not forced by the form layout.

Forms gathered from writers were automatically processed. Texts were automatically segmented into lines, words and characters, but automatic segmentation was verified by a human. In this way, the problems with incorrect segmentation into words and characters was avoided. Correctly segmented text images were next used for feature extraction for both levels.

Form images were divided into training and testing sets in this way, that single sample from each writer was assigned to the testing set while all remaining four samples were assigned to the training set. For better accuracy of results, leave-one-out method was applied. We repeated complete experiment five times, selecting each time another sample as an element of testing set for each writer. All results presented here are averaged values obtained in five iterations of the experiments.

4.2 Numerical Results

At the first stage of our experiment we tested writer identification accuracy on the upper level. MLPs on both sublevels were trained using 100 training objects in each iteration of leave-one-out procedure. Table 1 shows identification accuracy obtained for separate features sets A and B on lower sublevels of writer identification classifier and final identification accuracy on upper sublevel (set $A \cup B$). Because of soft writer recognition results application method on the lower level (in particular if the formula (4) is used as a character classifier combination rule) it is not essentially important that the actual writer is assigned the highest support value in the vector (1). Rather, it is expected that the actual class is in the small subset of classes with highest support factors. Therefore rows of the table contain recognition accuracy, where the recognition is considered as correct if actual writer is among k writers with highest support values $d_r(v^F)$ for $k = 1, 3, 5$.

Table 1. Writer identification accuracy

Criterion (k)	set A	set B	set $A \cup B$
k = 1	72%	31%	78%
k = 3	78%	37%	82%
k = 5	81%	41%	84%

Results obtained using only features set A are worse than results reported by Bulacu in [9] obtained when identifying writers of cursive scripts, despite the same fetaures set is used. We obtained 72% of correct identification for 25 writers while Bulacu reports 75% of correct recognition for combined features set and 250 writers. It should be noted however, that cursive script brings more information about writing style than text consisting of isolated upper case letters, in particular as far as stroke directional distribution is being concerned.

Using only features set B results in poor identification quality, but complementing features set A with set B causes noticeable increase of identification accuracy.

At the second stage of the experiment final character recognition performance was assessed. The following recognition methods were compared:

- **SGL** - single MLP trained with all available text samples without writer identification,
- **PWI** - artificial perfect writer identification (this experiment determines upper bound of accuracy that we are able to achieve with writer identification),
- **SCC** - single character MLP is used, trained with samples from the writer having highest support value in vector (1) on the upper level,
- **LCC** - linear combination of character classifiers is used according to (4).

In Table 2, accuracies achieved by compared methods are presented. Here again three correctness criterions are applied, where the recognition is considered

Table 2. Final character classification accuracy

Criterion (k)	SGL	PWI	SCC	LCC
k = 1	91.0%	95.6%	92.1%	94.7%
k = 3	94.3%	96.5%	94.9%	96.0%
k = 5	95.1%	97.2%	95.7%	96.8%

as successful if actual class is among k=1,3 and 5 characters with highest support factors. The motivation of such assessment method is similar as in case of upper level. The soft character recognition results can be used on higher levels of text recognition system (not discussed in this article), where complete words are recognized. The results of word recognition can be in turn used to complete sentence recognition using methods of natural language processing. Majority of methods used there assume that word recognizer provides the rank of words for each isolated word. The score for the word $w = (c_1, c_2, ...c_k)$ consisting of k characters can be obtained by multiplying the support factors $d^i_{c_i}$ evaluated for character c_i by the classifier applied to the character image on i-th position in the word. To obtain reliable words rank, it is necessary that the most likely characters have high support values. This goal will be achieved if the actual character is in the small set of characters with highest support factors. In result most likely words have high support factors and the whole sentence can be relatively reliably recognized.

General observation is that character recognition accuracy with writer identification is significantly higher that in case of single classifier trained on whole samples set from all writers (SGL vs. LCC). LLC method evidently outperforms SCC. In case of $k = 1$ and LLC method, the error rate was reduced from 9.0% to 5.3%. Interesting observation is that LLC method gives the accuracy close to the upper bound defined by PWI method, where the text is always recognized by character classifier trained for actual text writer, despite writer identification accuracy is below 80%. This is probably because writer recognizer, even if fails to recognize actual writer, assigns high support values to writers having similar writing style. It appears to be sufficient to correctly recognize characters on the lower level.

5 Conclusions

In the paper, results of using writer identification in order to improve the accuracy of handwritten text are presented. Described experiments confirm that the recognition accuracy can be significantly increased by applying personalized character classifiers trained for individual writers. The concept can be applied in cases, where all writers are known and their count is relatively small. Such situation appears e.g. in medical information systems, where some documents are primarily prepared in paper form and then must be automatically entered to the system database.

The work described here is a preliminary stage of larger project related to multilevel handwriting recognition algorithms and their application to medical texts. Our ultimate goal is to apply the technique described here to cursive script recognition. This is more complicated task than isolated character recognition, but on the other hand, writer identification algorithms perform better for cursive scripts. Hence, we believe that application of presented concept to cursive handwriting will result in even higher boost of words recognition accuracy.

Writer identification can be useful also on higher levels of text recognition system, where words and complete sentences are being recognized. Having sufficiently large corpus of texts written by particular writers individual writing style features can be extracted for each writer. In particular, probabilistic lexicons (unigram models) and language models (e.g. n-grams) can be build for each writer ([15]). Rough analysis of text corpora extracted from hospital information system database used in our project indicates that there are significant differences in word probability distribution between writers. Experiments with probabilistic lexicon application to words recognition proved that taking into account word probability distribution greatly improves the word recognition accuracy. Using probabilistic lexicons and language models adapted to actual writer will probably result in further improvement of words and sentence recognition accuracy.

Acknowledgement. This work was financed from Polish Ministry of Education and Science resources in 2005 - 2007 years as a research project No 3 T11E 005 28.

References

1. Liu C., Koga M., Sako H., Fujisawa H.: Aspect ratio adaptive normalization for handwritten character recognition, Advances in Multimodal Interfaces - ICMI 2000, LNCS, Springer Verlag (2000) 418-425
2. Said H.E.S., Tan T.N., Baker K.D.: Personal identification based on handwriting. Pattern Recognition Vol. 33. (2000) 149-160
3. Cha S.H., Srihari S.N.: Multiple Feature Integration for Writer Integration. Proc. of Seventh International Workshop on Frontiers in Handwriting Recognition, (2000) 333-342
4. Duda R., Hart P., Stork D.: Pattern Classification. John Wiley and Sons (2001)
5. Kuncheva L.: Combining classifiers: soft computing solutions. In: Pal S., Pal A. (eds.): Pattern Recognition: from Classical to Modern Approaches. World Scientific (2001) 427-451
6. Verma B., Gader P., Chen W.: Fusion of multiple handwritten word recognition techniques. Pattern recognition Letters, Vol. 22. (2001) 991-998
7. Liu C., Nakashima K., Sako H., Fujisawa H.: Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern Recogn., Vol. 36. (2003) 2271-2285
8. Schomaker L., Bulacu M., van Erp M.: Sparse-parametric writer identification using heterogeneous feature groups. Proc. of Int. Conf. on Image Processing ICIP, Vol. 1. (2003) 545-548
9. Bulacu M., Schomaker L., Vuurpijl V.: Writer identification using edge-based directional features, Proc. of Seventh International Conference on Document Analysis and Recognition, Vol. 2. (2003) 937-941

10. Gunes V., Menard M., Loonis P.: Combination, cooperation and selection of classifiers: a state of art. Int. Journal of Pattern Recognition and Artificial Intelligence, Vol. 17 No. 8 (2003) 1303-1324
11. Rahman A.F.R., Fairhurst M.C.: Multiple classifier decision combination strategies for character recognition: a review. Int. Journal on Document Analysis and Recognition, No. 5. (2003) 166-194
12. Schlapbach A., Bunke H.: Off-line handwriting identification using HMM based recognizers. Proc. 17th Int. Conf. on Pattern Recognition, Vol. 2. (2004) 654-658
13. Schomaker L., Bulacu M., Franke K.: Automatic writer identification using fragmented connected-component contours. Proc. of 9th IWFHR, Japan, Los Alamitos: IEEE Computer Society, (2004) 185-190.
14. Sas J., Luzyna M.: Combining character classifier using member classifiers assessment, Proc. of 5th Int. Conf. on Intelligent Systems Design and Applications, ISDA 2005, IEEE Press (2005) 400-405
15. Kurzynski M., Sas J.: Combining character level classifier and probabilistic lexicons in handprinted word recognition - comparative analysis of methods. In: Proc. XI Int. Conference on Computer Analysis and Image Processing, LNCS Springer Verlag (2005) 330-337

Adaptive Potential Active Hypercontours

Arkadiusz Tomczyk[1] and Piotr S. Szczepaniak[1,2]

[1] Institute of Computer Science, Technical University of Lodz
Wolczanska 215, 93-005, Lodz, Poland
tomczyk@ics.p.lodz.pl
[2] Systems Research Institute, Polish Academy of Sciences
Newelska 6, 01-447 Warsaw, Poland

Abstract. In this paper, the idea of *adaptive potential active hypercontours* (APAH) as a new method of construction of an optimal *classifier* is presented. The idea of *active hypercontours* generalizes the traditional *active contour* methods, which are extensively developed in image analysis, and allows the application of their concepts in other *classification* tasks. In the presented implementation of APAH the evolution of the *potential hypercontour* is controlled by *simulated annealing* algorithm (SA). The method has been evaluated on the IRIS and MNIST databases and compared with traditional *classification* techniques.

1 Introduction

The concept of *active hypercontours* (AH) was first introduced in [17] as a generalization of the *active contour* (AC) techniques ([1,2,3,4,5]) which are used in the image analysis. As shown in [16], *active contour* techniques can be in fact considered as search methods of an optimal *classifier* of pixels (it is usually a *contextual classifier*). The main advantage of *active contours* methods, in comparison with traditional *segmentation* techniques, is the possibility of an arbitrary choice of *energy function* which makes those techniques much more intuitive and allows an easier use of experts' knowledge than other *classification* methods (e.g. k-NN, *neural networks* NN ([6,7,10,13]), etc.). The concept of AH was proposed to enable an exchange of experience between those so far separately developed methodologies. In this article, a practical realization of AH is presented.

The paper is organized as follows: in section 2 the basic concepts of *active hypercontours* are described and their relationship with traditional *classifiers* is revealed, section 3 focuses on the description of the proposed APAH algorithm, section 4 presents obtained results comparing them to results achieved by means of traditional *classification* methods and finally the last section is devoted to conclusions and main ideas for future research directions.

2 Active Hypercontours and Classifiers

2.1 Hypercontours and Classifiers

In AC methods the *contour* of the object is sought in an optimization process of *energy* function $E : \mathcal{C} \rightarrow \mathbf{R}$, where $\mathcal{C}$ is the space of acceptable contours. The

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 692–701, 2006.
© Springer-Verlag Berlin Heidelberg 2006

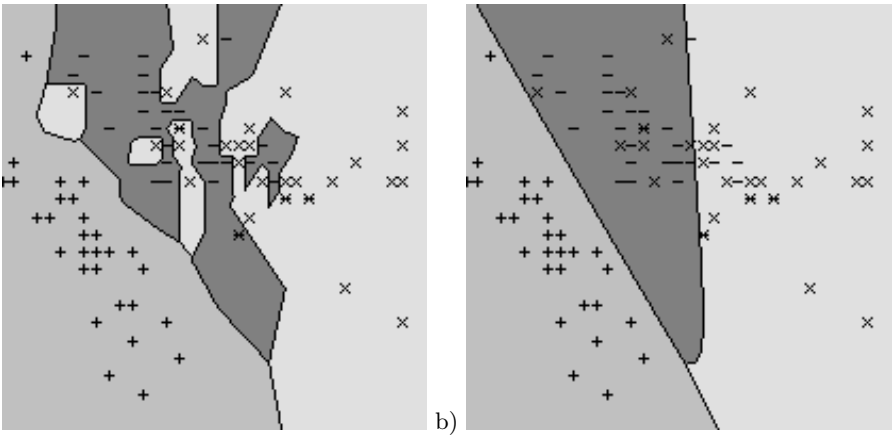


Fig. 1. Sample *hypercontours* generated by traditional *classifiers* (IRIS database, *features* $1-2$): (a) - k-NN method with Euclidean metric and $k=1$ (70), (b) NN (MLP) with 10 neurons in one hidden layer (74)

construction of a *classifier* is very similar. To find an optimal *classifier* every method bases on some *a priori* knowledge (e.g. on a training set of correctly labeled objects). That knowledge can be expressed in the form of *performance index* $Q : \mathcal{K} \to \mathbf{R}$ capable of the evaluation of the usefulness of each function $k \in \mathcal{K}$ (where $\mathcal{K}$ represents the space of all admissible *classifiers*). It is further assumed that $k : \mathcal{X} \to \mathcal{L}$ where $\mathcal{X} \subseteq \mathbf{R}^n$ denotes a *feature space* and $\mathcal{L} = \{1, \ldots, L\}$ denotes the set of labels. The *performance index* plays here a similar role to the *energy function*.

To exchange the experience between those groups of methods, the relationship between them was presented in [16] and further developed to the general definition of *hypercontour* in [17]. The *hypercontour* (*contour* is a special case of *hypercontour* for $n=2$ and $L=2$) can be defined in the following way:

Definition 1. *Let ρ denote any metric (e.g. Euclidean metric) in $\mathbf{R}^n$, $\mathcal{L} = \{1, \ldots, L\}$ denote the set of labels and let $K(\mathbf{x_0}, \varepsilon) = \{\mathbf{x} \in \mathbf{R}^n : \rho(\mathbf{x_0}, \mathbf{x}) < \varepsilon\}$ denote the sphere with center $\mathbf{x_0} \in \mathbf{R}^n$ and radius $\varepsilon > 0$. The set $h \subseteq \mathbf{R}^n$ together with information about labels of regions it surrounds is called a hypercontour if and only if there exists a function $f : \mathbf{R}^n \to \mathbf{R}$ and $p_0 = -\infty, p_1 \in \mathbf{R}, \ldots, p_{L-1} \in \mathbf{R}, p_L = \infty$ $(p_1 < p_2 < \ldots < p_{L-1})$ such that:*

$$h = \{\mathbf{x} \in \mathbf{R}^n : \exists_{l_1, l_2 \in \mathcal{L}, l_1 \neq l_2} \ \forall_{\varepsilon > 0} \ \exists_{\mathbf{x_1}, \mathbf{x_2} \in K(\mathbf{x}, \varepsilon)} \ p_{l_1 - 1} \leq f(\mathbf{x_1}) < p_{l_1} \wedge p_{l_2 - 1} \leq f(\mathbf{x_2}) < p_{l_2}\} \tag{1}$$

and the region $\{\mathbf{x} \in \mathbf{R}^n : p_{l-1} \leq f(\mathbf{x_1}) < p_l\}$ represents class $l \in \mathcal{L}$.

In [17] it has been proved that *hypercontours* are equivalent to *classifiers*. Each *classifier* generates a *hypercontour* in every *feature space* which has a sufficient discriminative power to distinguish *classified* objects:

$$p_0 = -\infty, \forall_{l\in\{1,\ldots,L-1\}}\ p_l = l + \frac{1}{2}, p_L = \infty$$
$$\forall_{\mathbf{x}\in\mathbf{R}^n}\ f(\mathbf{x}) = k(x) \tag{2}$$

Similarly, each *hypercontour* unambiguously generates the corresponding *classification* function:

$$\forall_{\mathbf{x}\in\mathbf{R}^n}\ k(\mathbf{x}) = l \text{ if } p_{l-1} \leq f(\mathbf{x}) < p_l \tag{3}$$

The name *hypercontour* is used to emphasize the relationship of the proposed technique with *active contour* methods.

Having the *hypercontour* defined as a generalization of *contour*, it is easy to generalize the AC technique for other *classification* problems than *image segmentation*. Only the *energy function* $E : \mathcal{H} \to \mathbf{R}$ must be properly defined to evaluate the usefulness of *hypercontours* $h \in \mathcal{H}$ (where $\mathcal{H}$ is the space of all the available *hypercontours*) and the optimization technique must be chosen to be able to find an optimal *classifier*. That leads to the formulation of the AH technique and thanks to that all the advantages of AC can be used not only in image analysis tasks but also in other *classification* problems.

2.2 Traditional Classifiers as Hypercontours

As stated in the previous section, each *classifier* generates a corresponding *hypercontour* in the proper *feature space*. Two typical *classifiers*: k-NN and NN (*multilayer perceptron* MLP) can be considered as examples of that fact. Each classifier assigns labels to vectors from the *feature space* and divides it into L regions of different topology. The boundaries of those regions are in fact a visual representation of *hypercontour*. In Fig. 1, sample *hypercontours* for the above mentioned two types of *classifiers* are presented.

In the case of NN, the similarity to AH is even more evident. The process of *neural network* learning is in fact a search for the optimal *classifier* basing on knowledge contained in the training set. Each iteration of the *back propagation* algorithm actually creates a new *classifier* (encoded in weights of *neural network*) and, in consequence, a new *hypercontour*. Thus, the adaptation of weights of neurons is in fact an evolution process controlled by the objective function (*performance index*, *energy*) which evaluates the progress of learning.

3 Potential Active Hypercontours

3.1 Potential Hypercontour

In this article *potential hypercontour* as a method of practical realization of *hypercontour* is introduced. For given values of n (the number of *features*) and L (the number of *classes*), the *potential hypercontour* is defined by means of a set of labeled control points: $D^c = \{\langle \mathbf{x}_1^c, l_1^c\rangle, \ldots, \langle \mathbf{x}_{N^c}^c, l_{N^c}^c\rangle\}$ where $\mathbf{x}_i^c \in \mathcal{X} \subseteq \mathbf{R}^n$ and $l_i^c \in \mathcal{L}$ for $i = 1, \ldots, N^c$. Each point is a source of potential the value of which decreases with the increase of distance from the source point (that concept

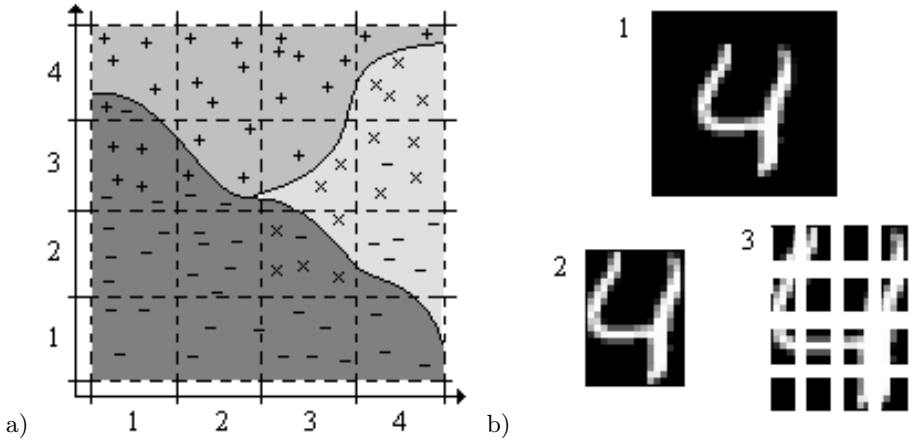


Fig. 2. (a) The method of *potential hypercontour* adaptation ($M = 4$). In the presented situation ($n = 2$, $L = 3$) new control points should be added in centers of regions $(1, 3)$, $(3, 2)$, $(4, 2)$. (b) The method of the extraction of *features* for MNIST database: 1 - sample object from MNIST database, 2 - the smallest region containing the whole digit, 3 - subregions where the number of pixels with intensity above given threshold is calculated.

is similar to the electric potential field). The *classifier* (and consequently the corresponding *hypercontour* (2)) is defined as follows:

$$\forall_{\mathbf{x}\in\mathcal{X}}\ k(\mathbf{x}) = \arg\max_{\mathbf{l}\in\mathcal{L}} \sum_{i=1}^{N^c} P_{Q_i\mu_i}(\mathbf{x}_i^c, \mathbf{x})\delta(l_i^c, l) \tag{4}$$

where $\delta : \mathcal{L} \times \mathcal{L} \to \{0, 1\}$, $l_1 \neq l_2 \Rightarrow \delta(l_1, l_2) = 0$, $l_1 = l_2 \Rightarrow \delta(l_1, l_2) = 1$ and $P : \mathbf{R}^n \times \mathbf{R}^n \to \mathbf{R}$ is a *potential function* e.g. *exponential potential* function:

$$P_{Q\mu}(\mathbf{x}_0, \mathbf{x}) = Qe^{-\mu\rho^2(\mathbf{x}_0,\mathbf{x})} \tag{5}$$

or *inverse potential* function:

$$P_{Q\mu}(\mathbf{x}_0, \mathbf{x}) = \frac{Q}{1+\mu\rho^2(\mathbf{x}_0,\mathbf{x})} \tag{6}$$

In both cases Q and μ are parameters characterizing the potential field and ρ denotes a metric (e.g. Euclidean metric) ([6]). The *potential hypercontour* defined in this way is able to describe almost each division of *feature space* (its shape depends both on the distribution of control points and on parameters of *potential* functions) and at the same time is very simple and intuitive in use.

3.2 Evolution of Potential Hypercontour

The shape of the *potential hypercontour* and its *classification* ability depend on the position of control points in the *feature space* and on the parameters characterizing *potential* functions. The search for the optimal distribution of control

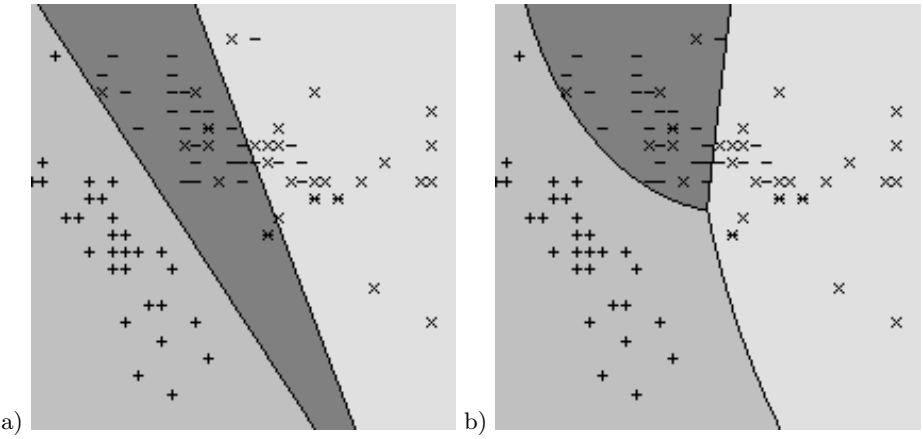


Fig. 3. The influence of the parameter Q of *potential* functions on the results of PAH algorithm (IRIS database, *inverse potential*, $\mu = 1$, *features* $1-2$): (a) - the parameter Q was not modified during optimization (76), (b) - the parameter Q was modified by *simulated annealing* algorithm (the search space contains more elements - the *hypercontours* are more flexible) (80)

points only appeared, after first experiments, to be sometimes unsatisfactory (the *hypecontours* are less flexible and its harder to achieve a desired shape especially if the initial configuration is far from the optimal one) (Fig. 3). Because of that reason also the optimal value of the parameter Q for each source of potential is sought (parameter μ is assumed to be constant during the optimization). In the proposed implementation of *potential active hypercontour* PAH the *simulated annealing* (SA), as an optimization algorithm, was used ([18]). That method on the one hand does not require any gradient information about *objective function* (only its values) and, on the other hand, it allows avoiding the local minima. Any other optimization techniques (e.g. *genetic algorithm* etc.) can also be used here.

3.3 Energy of Hypercontour

The main advantage of AH is its ability to define *energy* (*objective function*) in an almost arbitrary way. In this paper the *a priori* knowledge about the problem is hidden in a training set. In general, however, the *energy* can use any other information obtained from an expert as well as it can put any arbitrary chosen constraints on the shape of the desired *hypercontour*.

Let $D^{tr} = \{\langle \mathbf{x}_1^{tr}, l_1^{tr} \rangle, \ldots, \langle \mathbf{x}_{N^{tr}}^{tr}, l_{N^{tr}}^{tr} \rangle\}$ where $\mathbf{x}_i^{tr} \in \mathcal{X} \subseteq \mathbf{R}^n$ and $l_i^{tr} \in \mathcal{L}$ for $i = 1, \ldots, N^{tr}$ denote a sample training set of correctly labeled vectors. The *energy* of *potential hypercontour* h described by means of control points D^c can be defined as:

$$\forall_{h \in \mathcal{H}}\ E(h) = \sum_{i=1}^{N^{tr}} (1 - \delta(l_i^{tr}, \arg\max_{\mathbf{l} \in \mathcal{L}} \sum_{i=1}^{N^c}\ P_{Q_i \mu_i}(\mathbf{x}_i^c, \mathbf{x}_i^{tr}) \delta(l_i^c, l))) \qquad (7)$$

3.4 Adaptive Potential Active Hypercontour

Experimental results revealed that in some situations the initial large number of random control points can cause some problems during the optimization process i.e. it can be hard to get out of the local minima. (especially when there is no *a priori* information about some fragments of *feature space*). Moreover, at the beginning of the *hypercontour* evolution it is usually not known how many control points are needed for a satisfactory description of a desired *classifier*. Too many control points can reduce its generalizing abilities. Due to those reasons and to improve the performance of the proposed algorithm, an adaptation mechanism can be added to PAH. This adaptation allows in APAH to start the optimization phase several times. After each phase additional control points can be added to D^c in those areas of *features space* where the number of incorrect *classifications* is the largest. Thus, the evolution can begin with a smaller number of control points and can be finished when the *classification* results are satisfactory.

In the presented implementation each adaptation step adds either one or L control points, one for each class. To choose the points that should be added, the smallest n-dimensional cube containing all the points from the *training set* is considered. After each optimization phase that cube is divided into M^n identical but smaller n-dimensional cubes (in every dimension the smallest interval containing all the possible values of the *feature* is divided into M equal subintervals) (Fig. 2). Next, in all of those cubes the number of incorrect *classifications* of objects from D^{tr} is calculated and a new control point, for a given *class* $l \in \mathcal{L}$, is placed in the center of that cube where the highest number of wrong *classifications* of objects from that *class* was observed.

4 Results

The method was tested on the IRIS and MNIST databases. In both cases the *training set* D^{tr} and *test set* $D^{te} = \{\langle \mathbf{x}_1^{te}, l_1^{te} \rangle, \ldots, \langle \mathbf{x}_{N^{te}}^{te}, l_{N^{te}}^{te} \rangle\} \subseteq \mathbf{R}^n \times \mathcal{L}$ were considered. The latter was used to evaluate the results of *classification* (the percent of correct *classifications* in that set was used as a measure of the quality of the *classifier*).

The first data set contains $L = 3$ classes referring to 3 types of iris plants (*iris setosa*, *iris versicolour* and *iris virginica*) ([19]). Each class is represented by 50 objects and each object is described using $n = 4$ *features* (*sepal length*, *sepal width*, *petal length* and *petal width*). For evaluation purposes the whole set was randomly divided into *training set* D^{tr} (100 instances) and *test set* D^{te} (50 instances). The achieved results of *classification* for traditional methods as well as for PAH and APAH are presented in Table 1 and in Fig. 3, Fig. 4.

The second database contains a set of images with handwritten digits ($L = 10$) ([20]). One image contains one digit only. This set was divided into *training set* D^{tr} and *test set* D^{te} with 6000 and 1000 instances respectively. In this case to conduct experiments, the *features* had to be first extracted. The method of extraction proposed here first finds the smallest region containing the whole digit and then divides it into a given number (here 16) of identical subregions

Table 1. Sample *classification* results (percent of correctly labeled objects from D^{te}) for each combination of *features* (IRIS database). Each column corresponds to one classifier: (a), (b) k-NN method with Euclidean metric and $k = 1$ and $k = 7$, respectively; (c), (d) - NN with one hidden layer containing 5 and 10 neurons, respectively; (e), (f), (g) - PAH method with *inverse potential* where $\mu = 1$ and the number of control points in each class is equal to 1, 2, 10 respectively; (h), (i) - APAH with *inverse potential*, $\mu = 1$, $M = 8$ and 1, 9 adaptation steps respectively, (j), (k), (l), (m), (n) - the same as (e), (f), (g), (h), (i) but with $\mu = 15$.

Features	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	**Best**
1	68	68	74	68	72	74	72	72	72	76	72	72	72	72	PAH
2	46	50	50	50	52	52	52	52	52	52	52	52	52	54	APAH
3	92	94	92	92	92	92	92	92	92	92	92	92	92	94	k-NN, APAH
4	92	92	94	92	94	92	92	92	92	92	96	96	92	96	APAH
$1-2$	70	76	74	82	76	76	74	78	80	78	76	78	78	84	APAH
$1-3$	94	92	92	94	94	94	94	94	94	94	94	94	94	96	APAH
$1-4$	94	96	94	94	94	94	94	94	94	94	94	96	94	94	k-NN, PAH
$2-3$	90	94	92	92	90	92	92	92	92	92	92	92	92	92	k-NN
$2-4$	92	92	94	92	92	94	92	94	94	92	92	92	90	94	NN, APAH
$3-4$	96	96	96	96	96	96	96	96	96	96	96	96	96	96	ALL
$2-3-4$	98	94	94	94	94	94	96	94	96	94	94	94	94	98	k-NN, APAH
$1-3-4$	96	98	96	94	96	96	96	98	96	94	94	98	96	98	k-NN, APAH
$1-2-4$	94	98	92	94	92	90	94	94	96	94	94	94	98	98	k-NN, APAH
$1-2-3$	92	90	94	96	94	96	94	94	94	94	94	96	94	96	NN, APAH
$1-2-3-4$	96	96	94	94	94	96	94	96	98	96	96	96	94	96	APAH

Table 2. The influence of the adaptation process on the *classification*. Comparison of the achieved results (percent of correctly labeled objects from D^{te}) with results obtained by means of traditional techniques (MNIST database). Each column corresponds to one classifier: (a), (b), (c) - k-NN method with Euclidean metric and $k = 1$ and $k = 7$, respectively; (d), (e), (f), (g) - NN with one hidden layer containing 10, 20, 50 and 100 neurons, respectively; (h), (i), (j), (k), (l) - APAH method with *inverse potential*, $\mu = 1$ and $M = 2$ after 0, 4, 8, 12, 16 adaptation steps, respectively.

(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)
78.1	80.5	78.9	65.0	72.8	76.4	79.2	19.2	48.4	62.2	66.8	75.2

(Fig. 2). The *feature vector* is composed of the ratios of pixels with intensity above the given threshold (e.g. 128) to the whole number of pixels in every subregion (consequently $n = 16$). In spite of the fact that in the literature the better *features* can be found, those proposed here are sufficient for comparison of *classifiers*. The obtained results are gathered in Table 2.

It is worth mentioning that the choice of a *potential function* and its parameters affects the character of the *potential hypercontour* and consequently the *classification* results. For the *potential functions* considered here, the parameter

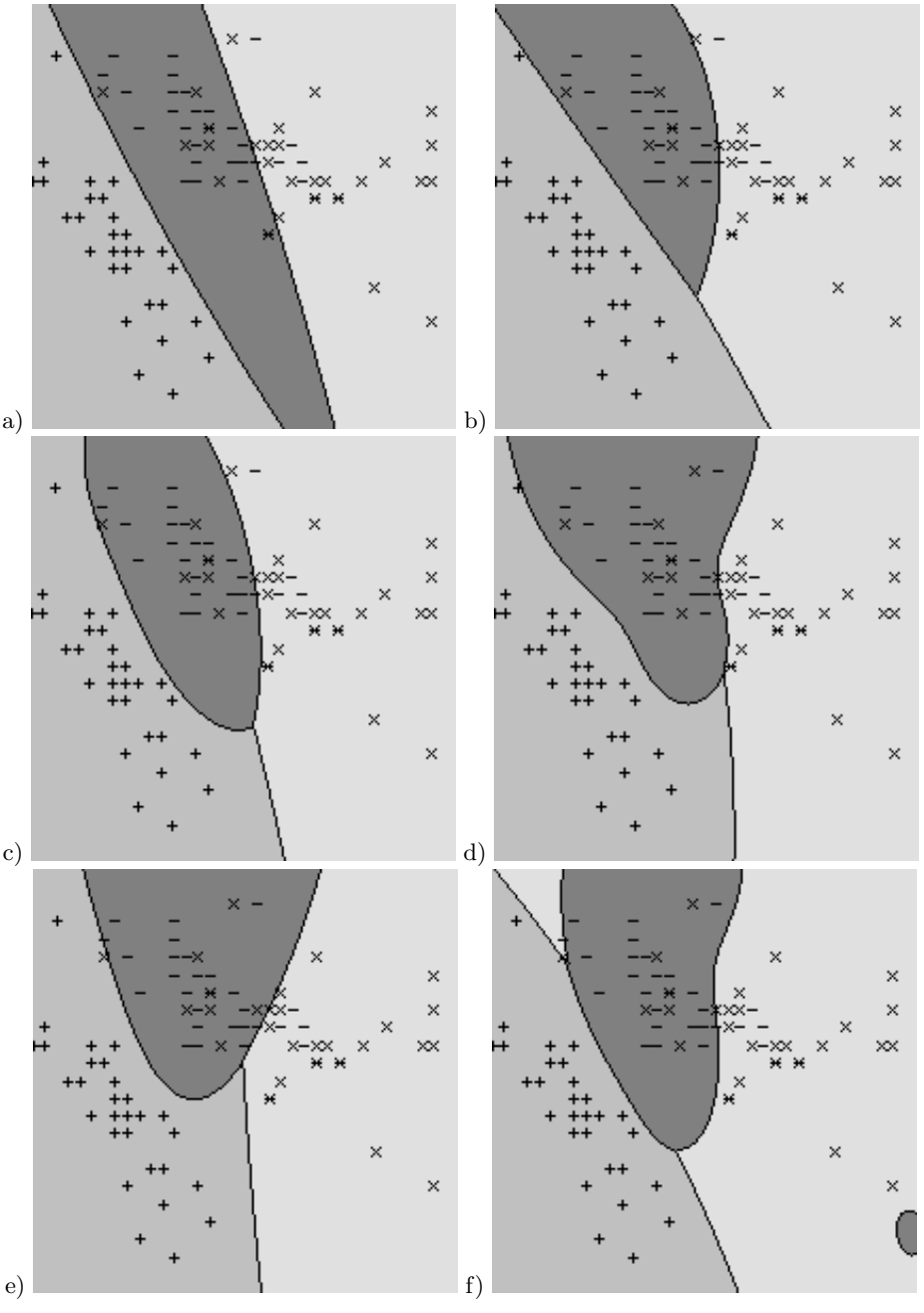


Fig. 4. The influence of parameters of APAH (IRIS database, *features* $1-2$, $M=8$): (a), (c), (e) - $\mu=1$, results after 0 (76), 4 (76), 7 (78) adaptations steps, respectively; (b), (d), (f) - $\mu=15$, results after 0 (78), 4 (80), 7 (82) adaptations steps, respectively

μ affects the flexibility of the APAH (Fig. 4). Thus μ can be used to control the ability of a *classifier* to generalize the available knowledge.

5 Conclusions

The presented APAH method is a new approach to an optimal *classifier* construction. It allows (even in its basic form presented here) to obtain *classifiers* which give similar and in some situations even better results than traditional methods. The main advantage of APAH, however, is its ability to straightforwardly use any experts' knowledge. Here, the knowledge was gathered in the form of a *training set*, but sometimes experts, basing on their experience, can add a heuristic information which can improve the *classification* significantly (e.g. *fuzzy* information). This method allows also to take into account any additional constraints that can be put on the shape of desired *hypercontour* (as for example in *support vector machines* SVM) which is not always possible in traditional techniques. In the APAH approach all of that can be simply encoded in the *energy* function. Moreover, it is also possible to consider the *potential active contour* (PAC) method as a new method for image segmentation. It is worth mentioning that there are some analogies between this approach and other methods known from *pattern recognition* to call RBF *neural networks* as an example. All these aspects are presently under further, practical investigation.

References

1. Kass M., Witkin W., Terzopoulos D., (1988) *Snakes: Active Contour Models*, International Journal of Computer Vision, 321–331
2. Caselles V., Kimmel R., Sapiro G., (1997) *Geodesic Active Contours*, International Journal of Computer Vision 22(1) 61–79
3. Xu Ch., Yezzi A., Prince J., (2000) *On the Relationship between Parametric and Geometric Active Contours*, in Proc. of 34th Asilomar Conference on Signals, Systems and Computers 483–489
4. Cootes T., Taylor C., Cooper D., Graham J., (1994) *Active Shape Model - Their Training and Application*, CVGIP Image Understanding, 61(1) 38–59 Janvier
5. Grzeszczuk R., Levin D., (1997) *Brownian Strings: Segmenting Images with Stochastically Deformable Models*, IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 19 no. 10 1100-1013
6. Tadeusiewicz R., Flasinski M., (1991) *Pattern Recognition*, PWN, Warsaw (in Polish)
7. Kwiatkowski W., (2001) *Methods of Automatic Pattern Recognition*, WAT, Warsaw (in Polish)
8. Sobczak W., Malina W., (1985) *Methods of Information Selection and Reduction*, WNT, Warsaw (in Polish)
9. Nikolaidis N., Pitas I., (2001) *3-D Image Processing Algorithms*, John Wiley and Sons Inc., New York
10. Bishop Ch., (1993) *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford

11. Pal S., Mitra S., (1999) *Neuro-fuzzy Pattern Pecognition, Methods in Soft Computing*, John Wiley and Sons Inc., New York
12. Bennamoun M., Mamic G., (2002) *Object Recognition, Fundamental and Case Studies*, Springer-Verlag, London
13. Looney C., (1997) *Pattern Recognition Using Neural Networks*, Theory and Algorithms for Engineers and Scientists, Oxford University Press, New York
14. Sonka M., Hlavec V., Boyle R., (1994) *Image Processing, Analysis and Machine Vision*, Chapman and Hall, Cambridge
15. Gonzalez R., Woods R., (2002) *Digital Image Processing*, Prentice-Hall Inc., New Jersey
16. Tomczyk A., Szczepaniak P. S., (2005) *On the Relationship between Active Contours and Contextual Classification*, 4th International Conference on Computer Recognition Systems (CORES), Rydzyna (near to Leszno), Poland, Springer, 303-311.
17. Tomczyk A., (2005) *Active Hypercontours and Contextual Classification*, 5th International Conference on Inteligent Systems Design and Applications (ISDA), Wroclaw, Polska, IEEE Computer Society Press, 256-261.
18. Kirkpatrick S., Gerlatt C. D. Jr., Vecchi M.P., (1983) *Optimization by Simulated Annealing*, Science 220, 671-680.
19. IRIS, `http://www.ics.uci.edu/~mlearn`.
20. MNIST, `http://yann.lecun.com/exdb/mnist/`.

KIDBSCAN: A New Efficient Data Clustering Algorithm

Cheng-Fa Tsai and Chih-Wei Liu

Department of Management Information Systems
National Pingtung University of Science and Technology, Pingtung, Taiwan, 91201
Telephone: 886-8-7703202 ext. 7906
Fax: 886-8-7740306
cftsai@mail.npust.edu.tw

Abstract. Spatial data clustering plays an important role in numerous fields. Data clustering algorithms have been developed in recent years. K-means is fast, easily implemented and finds most local optima. IDBSCAN is more efficient than DBSCAN. IDBSCAN can also find arbitrary shapes and detect noisy points for data clustering. This investigation presents a new technique based on the concept of IDBSCAN, in which K-means is used to find the high-density center points and then IDBSCAN is used to expand clusters from these high-density center points. IDBSCAN has a lower execution time because it reduces the execution time by selecting representative points in seeds. The simulation indicates that the proposed KIDBSCAN yields more accurate clustering results. Additionally, this new approach reduces the I/O cost. KIDBSCAN outperforms DBSCAN and IDBSCAN.

1 Introduction

Larger amounts of data are produced and stored in databases, explaining why approaches for analyzing effectively and efficiently these data are in great demand. The need to extract useful implicit information has made data-mining increasingly popular over recent years [1].

As a method of spatial data analysis, data clustering classifies objects into groups. Objects in the same group are highly similar. Conversely, those objects in different groups are quite dissimilar. Based on a general definition, data clustering algorithms can be classified into four categories; (1) partitioning, (2) hierarchical, (3) density-based and (4) grid-based. However, some algorithms may fall into more than one category.

Partitioning clustering is the conventionally adopted approach in this field, and most such algorithms identify the center of a cluster. The most well-known partitioning algorithm is K-means [2]. K-means is fast, easily implemented and finds most local optima for data clustering. However, the crucial shortcoming of K-means is the difficultly of recognizing arbitrary shapes. Hierarchical clustering employs a hierarchical data structure and constructs a tree of clusters to sort data. Hierarchical clustering includes two sub-categories - agglomerate and divisive. Agglomerative clustering, which involves BIRCH [3], CURE [4] and ROCK [5], is a

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 702–711, 2006.
© Springer-Verlag Berlin Heidelberg 2006

bottom-up technique that organizes similar objects from one point to a cluster. In contrast, divisive clustering, such as the CHAMELEON [6], is a top-down approach which decomposes one cluster into numerous similar clusters.

Hierarchical clustering compares all objects before they are merged. This comparison takes much time. Density-based clustering emphasizes dense areas, which are grouped into a single cluster. Restated, the density in the cluster should exceed that out of the cluster. Density-based clustering algorithms can handle arbitrarily shaped clusters and detect noise. DBSCAN [7] and IDBSCAN [8] are two density-based clustering algorithms. Grid-based clustering segments the data space into pieces like grid-cells. Grid-based clustering algorithms treat these grid-cells as data points, so grid-based clustering is more computationally efficient than other forms of clustering. Typical examples are STING [9] and STING+ [10].

This work presents a new algorithm that combines K-means and IDBSCAN. K-means yield the core points of clusters, regardless of whether data sets are regular and converge rapidly. Then, clusters are expanded from these core points by executing IDBSCAN. IDBSCAN has the advantages of density-based clustering and employs sampling techniques, to reduce the execution time below that of DBSCAN. The synergy provided by merging K-means and IDBSCAN is an effective way to reduce the I/O cost. The algorithm herein can be classified as a partition- and density-based clustering algorithm.

The rest of this paper is organized as follows. Section 2 discusses several clustering algorithms related to our work. Section 3 then introduces- the proposed algorithm KIDBSCAN. Next, Section 4 summarizes the experimental and analysis. Conclusions are finally drawn in Section 5.

2 Related Works

This section discusses various data clustering algorithms. Merits and limitations of these data clustering algorithms are described below.

2.1 K-Means

In 1967, McQueen presented K-means as the first clustering algorithm. Simple rules for data clustering are required to reduce the execution time. K-means involves the following iteration. (1) Randomly select K cluster centers from data set. (2) Assign each object to its closest cluster center. (3) Recalculate the cluster centers until convergence. K-means is almost always converges to local optima. However, its greatest weakness is that it tends to form in round clusters it is weak at detecting noise. Fig. 1 shows the K-means concept for data clustering.

2.2 DBSCAN

DBSCAN is based on the concept of dense areas to form data clustering. The distribution of objects in the cluster should be denser than that outside of the

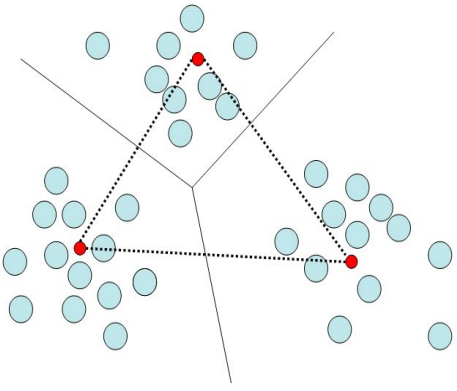


Fig. 1. K-means concept for data clustering

cluster. DBSCAN is the first algorithm that clusters using density. It depends on two parameters to specify the dense area:ε and MinPts. ε represents the radius of the circle, and MinPts denotes the number of minimal points in a circle.

DBSCAN seeks clusters by examining the ε-neighborhood of each object in the database. If the ε-neighborhood of an object Q has at least MinPts points, then Q is called a core point and a cluster is formed. However, the cluster from an object P is extended, then the number of points in its ε-neighborhood under

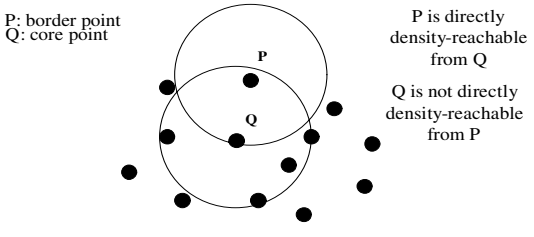


Fig. 2. Directly density reachable [7]

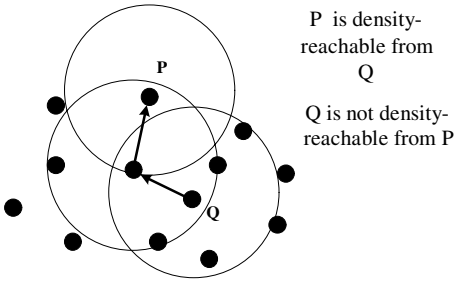


Fig. 3. Density reachable [7]

MinPts. P labels the border points in Fig. 2. Fig. 2 demonstrates that these core points are called directly density reachable. Fig. 3 depicts some points in the cluster extended from the border points. These are regarded as density reachable. During continuous expansion, the clusters take shape in the end.

DBSCAN can detect arbitrary shapes and separate noises. However, the two parameters are difficult to arrange and severely affect the clustering results and the quality of clustering. They can be determined from a series of observations.

2.3 IDBSCAN

DBSCAN expands clusters by adding ε-neighborhood points to seeds. These seeds will also query their ε-neighborhood points to expand clusters iteratively. This is a time-consuming process that must be repeated the data sets are enormous. Accordingly, some representative points, rather than all of the points in ε-neighborhood, should be sampled.

The method that selects representative points illustrates in the two-dimension database. Eight distinct points are selected as Marked Boundary Objects (MBO), displayed in Fig.4. For each of these MBOs, the closest object in the ε-neighborhood of an object P is found and selected as a seed. If the same point is identified as the nearest point for more than one MBO, then this point must be regarded only once as a seed. Therefore, at most eight points are seeds at an one-time. This number is lower than the corresponding number in DBSCAN. IDBSCAN yields the same quality of DBSCAN but is more efficient.

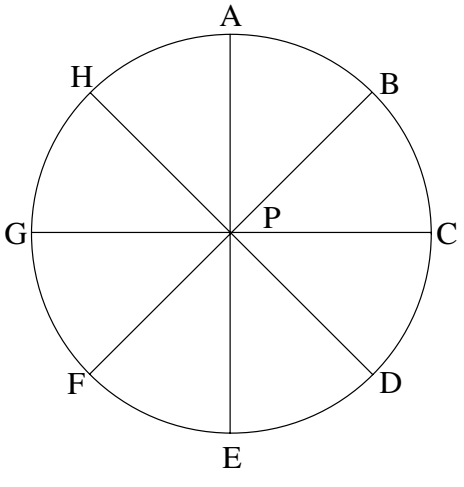


Fig. 4. Circle with eight distinct points [8]

3 KIDBSCAN

The sampling of MBOs in IDBSCAN significantly affects execution time. Truly assigning unnecessary points to seeds causes the execution to be redundant. Hence, a much more effective way to reduce dispensable seeds is sought.

According to shown in Fig. 5, when IDBSCAN is executed according to the sequence of data points, so more time must be taken to find dispensable, low-density point or border points from which the expansion of clusters begins. Accordingly, an effective means to solve this problem is sought.

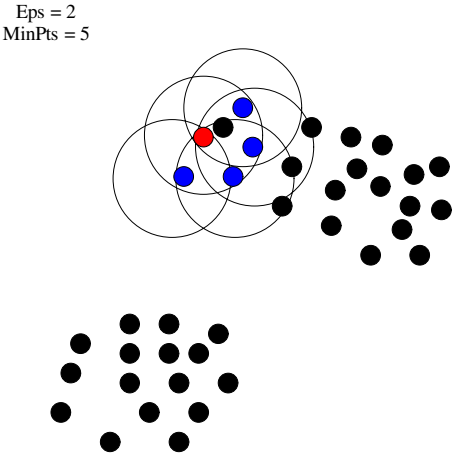


Fig. 5. Expansion from low-density point

The irregular order of data sets is such that nothing is known about them a priori. In this situation, clusters are expanded from high-density comes out. As presented in Fig. 6, clusters are expanded from high-density points. Consequently, K-means is used to overcome the difficulty. K-means' rapid location of high-density center points is exploited. Then, the expansion of clusters from high-density center points reduces the number of redundant steps and increases efficiency.

KIDBSCAN is developed here to merge K-means and IDBSCAN. KIDBSCAN has three parameters. They are cluster,ε and Minpts . KIDBSCAN consists of the following three stages.

1. Initial stage: initialize variables required by K-means and IDBSCAN, and included in the input data set.
2. K-means stage: a parameter K is input. K-means is applied to the input data set to yield K high-density center points. Find the points in the data set that are closest to the center points. These points are moved to the front of data.
3. IDBSCAN stage: the two parameters are ε and MinPts. IDBSCAN is first executed on an adjusted data set determined by K-means. Finally, the clustering result is generated.

These center points are exploited in high-density areas obtained from K-means. They help IDBSCAN to expand clusters. Additionally, IDBSCAN yields

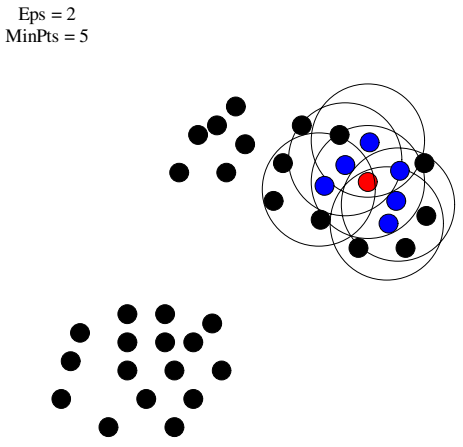


Fig. 6. Expansion from high-density point

superior clusters and handles noise more effectively. The synergy of K-means and IDBSCAN provides high quality data clustering, and substantially reduces I/O cost. The KIDBSCAN algorithm is shown in Fig. 7.

4 Experiment and Analysis

This section describes numerous experiments conducted to confirm the proposed algorithm. Four data sets are used to check the accuracy of clustering. Both regular and arbitrary shapes are used. Second, large data sets are used to compare the performance of DBSCAN, IDBSCAN and the proposed algorithm.

In the simulations, these algorithms are implemented in Java-based programs. All results were obtained using a desktop computer with 1G MB of RAM, an Intel 1.3 GHz CPU. The results of the experiments are presented below.

4.1 Correctness Experiments on KIDBSCAN

Four data sets were used as samples to measure the accuracy of KIDBSCAN in two-dimensional space. DS1 comprises three rectangular clusters. DS2 includes three clusters - two rectangular and one triangular. DS3 follows a Gaussian distribution comprises four circular clusters. DS4 is constructed from clusters with four arbitrary shapes and with noise, to demonstrate the sensitivity to noise. Figs. 8-11 depict the original data sets of DS1-DS4 and their clustering results.

Accordingly, after the parameters were adjusted, the experiments revealed that KIDBSCAN can clearly identify clusters, even though noise exists.

4.2 Speed Experiments on KIDBSCAN

Artificial data sets of seven sizes were examined to measure the speed of the proposed algorithm. These data sets contain three clusters. The experimental

```
Input : Data{Xi}, K,  ε , MinPts
Output : Clusters
/* Utilize K-means to find high-density cluster centers and adjust the order of dataset*/
K-means(Data, K)
   Centers := Random_generator();
   while (old cluster centers != new cluster centers)
       Cluster();
       Compute centers;
   End while;
   AdjustData();
   Return Adjusted_Data;
End;

/* Execute IDBSCAN to expand clusters */
IDBSCAN(Adjusted_Data,  ε , MinPts)
   ClusterID := nextID(First);
   For i := 1 to N {
          Point = i;
          If Point is Unclassified Then
               If ExpendCluster(Adjusted_Data, Point, ClusterID,  ε , MinPts) Then
                    ClusterID = nextID(ClusterID)
               END If
          End If
   End For
End;

ExpendCluster(Adjusted_Data, Point, ClusterID,  ε , MinPts)
   seed_count :=find_neighbors(Point,  ε )
   If seed_count < MinPts Then
      changeClusterID(seed, Noise)
      Return false
   Else
      changeClusterID(seed, ClusterID)
      Add_MBO(Point,  ε )
      While seeds != Empty
          CurrentP := seed.first()
          result_count := region_query(CurrentP,  ε )
          Add_MBO(CurrentP,  ε )
          If result_count >= MinPts Then
               For i := 1 to result_count
                    resultP := result.get(i);
                    If resultP.ClusterID := Noise || resultP.ClusterID :=Unclassified Then
                         changeClusterID(result, ClusterID)
                    End If
               End For
          End If
          seeds.delete(currentP)
      End While
      Return True
   End If
End
```

Fig. 7. KIDBSCAN Algorithm

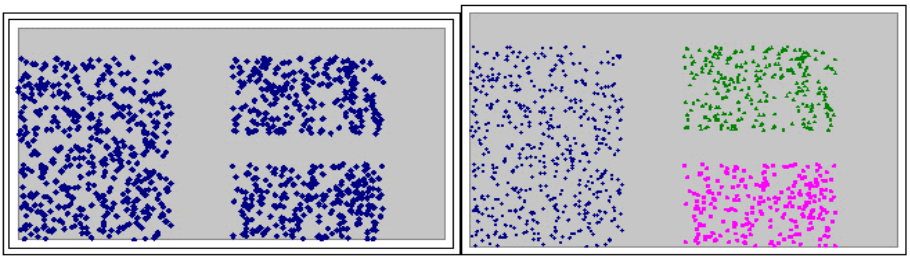


Fig. 8. The original data set of DS1 and its clustering result

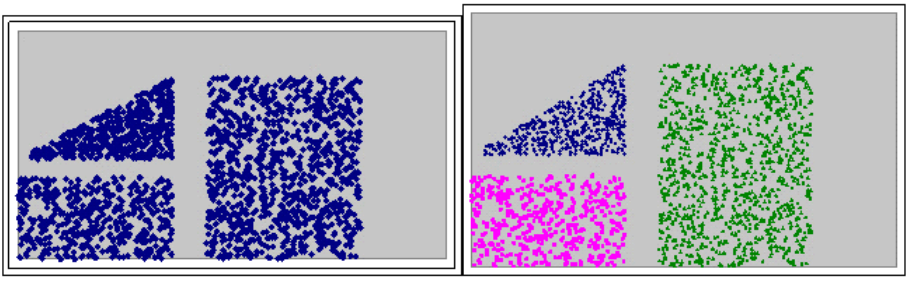


Fig. 9. The original data set of DS2 and its clustering result

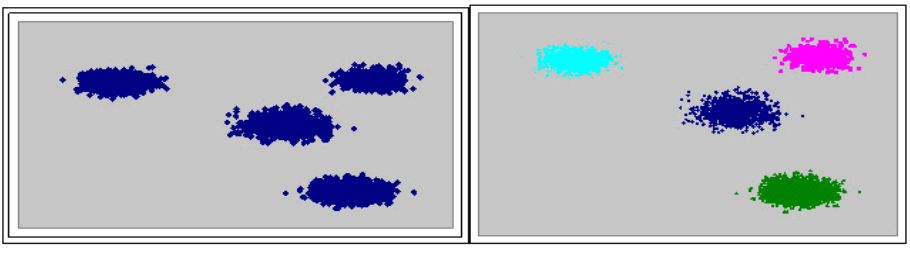


Fig. 10. The original data set of DS3 and its clustering result

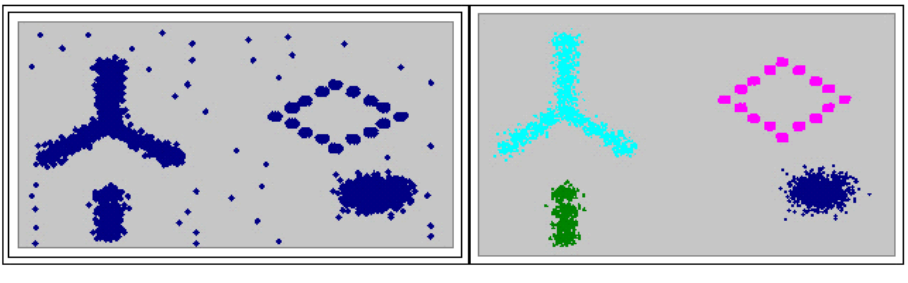


Fig. 11. The original data set of DS4 and its clustering result

Table 1. Comparisons with data sets (seconds)

Data Size	DBSCAN	IDBSCAN	**KIDBSCAN**
1000	0.2	0.1	**0.1**
20000	121	9	**9**
30000	276	16	**15**
40000	457	41	**35**
60000	896	66	**52**
80000	2419	109	**85**
100000	3699	131	**98**

data are in seconds. In order to be unaffected by parameters, these data sets are in the same shape.

Table 1 indicates that as the size of the data sets increases, KIDBSCAN still outperforms than DBSCAN and IDBSCAN. The expansion from high-density center points improves efficiency.

5 Conclusion

This work presents a new clustering algorithm to perform data clustering efficiently. Two clustering algorithms are merged to refine each shortcoming. The main advantages of the synergy are as follows. (1) The computational time does not increase with the number of data points. (2) It performs excellently for arbitrary shapes. (3) It is not limited by memory when dealing with large data sets. KIDBSCAN improves the accuracy of the clustering result. Furthermore, the proposed technique can reduce the I/O cost. KIDBSCAN outperforms DBSCAN and IDBSCAN.

References

1. R. Xu, D. Wunsch, Survey of Clustering Algorithm, IEEE Transactions on Neural Networks, Vol. 16, No. 3, pp. 645 - 678, 2005.
2. J.B. McQueen, Some Methods of Classification and Analysis of multivariate Observations, Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297, 1967.
3. T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: An efficient Data Clustering Method for Very Large Data Bases, Proceedings of the ACM SIGMOD International Conference on Management of Data, Vol. 25, No. 2, pp. 103-114, 1996.
4. S. Guha, R. Rastogi, K.Shim, CURE: An Efficient Clustering Algorithm for Large Data Bases, Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, Vol. 27, No. 2, pp. 73-84, 1998.
5. S. Guha, R. Rastogi, K. Shim, ROCK: A Robust Clustering Algorithm for Categorical Attributes, Proceedings of 15th International Conference on Data Engineering, pp. 512-521, 1999.
6. G. Karypis, E.H. Han, V Kumar, CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling, IEEE Computers, Vol. 32, No. 8, pp. 68-75, 1999.

7. M. Ester, H. P. Kriegel, J. Sander, X. Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, Proceedings of International Conference on Knowledge Discovery and Data Mining, pp. 226-231, 1996.
8. B. Borah, and D.K. Bhattacharyya, An Improved Sampling-Based DBSCAN for Large Spatial Databases, Proceedings of International Conference on Intelligent Sensing and Information, pp. 92-96, 2004.
9. W. Wang, J. Yang, and R. Muntz, STING: A Statistical Information Grid Approach to Spatial Data Mining, proceedings of 23rd International Conference on Very Large Data Bases, pp. 186-195, 1997.
10. W. Wang, J. Yang, and R. Muntz, STING+: An approach to Active Spatial Data Mining, Technical report, UCLA CSD, No. 980031, 1998.

Localization and Extraction of the Optic Disc Using the Fuzzy Circular Hough Transform

M. Blanco[1], M.G. Penedo[1], N. Barreira[1], M. Penas[1], and M.J. Carreira[2]

[1] Computer Science Department, University of A Coruña, Spain
{mblancom, mgpenedo, nbarreira, mpenas}@udc.es
[2] Electronics and Computer Science Department, University of Santiago de Compostela, Spain
mjose@dec.usc.es

Abstract. This paper presents an algorithm for automatic extraction of the optic disc in retinal images. The developed system consists of two main parts. Firstly, the localization of the region containing the optic disc is performed by means of a clustering algorithm. Then, in order to extract the optic disc, the fuzzy circular Hough transform is applied to the edges of the region. The optic disc might not be extracted since there are vessels in the inside of the optic disc. To avoid this, a crease extraction algorithm is applied to the retinal image. The vessels are extracted and the vessel edge points contained in the edge image are removed. The final system was tested by ophthalmologists. The localization of the region of interest is correct in 100% of the cases and the extraction of the optic disc is obtained in 98% of the cases.

1 Introduction

The retinal fundus photographs are widely utilized in the diagnosis of eye diseases. Processing automatically a large number of retinal images can help ophthalmologists increase the efficiency in medical environment. The optic disc is the brightest area in images that have not large areas of axudates and it is a slightly oval disc. It is the entrance region of vessels and its detection is very important since it works as a landmark for the other features in the retinal image.

There are many previous works on optic disc localization. Lalonde et al. [1] extract the optic disc using Hausdorff–based template matching and pyramidal decomposition. It is neither sufficiently sensitive nor specific enough for clinical application.

On the other hand, strategies based on active contours [2, 3, 4] are used to detect the optic disc boundary in retinal images. These techniques are very robust against noise but their main disadvantage is their high computational cost.

In this paper, a new methodology to extract the optic disc is proposed. Firstly, the localization of the region containing the optic disc is performed. Then, the fuzzy circular Hough transform is applied to the edges of the region in order to extract the optic disc. The fuzzy circular Hough transform might not extract

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 712–721, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the disc due to the vessels in the inside of the optic disc. In order to avoid it, an automatic extraction of vessels is done by means of a crease extraction algorithm. The proposed methodology is shown in figure 1.

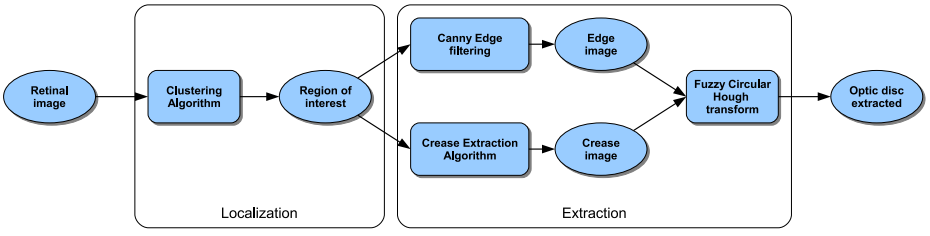


Fig. 1. Main parts of the proposed methodology

This paper is organized as follows. In Section 2 the localization of the region of interest is discussed. Section 3 explains the fuzzy circular Hough transform and the crease extraction algorithm. Section 4 shows the main results. Finally, in Section 5 the conclusions are presented.

2 Localization of the Region of Interest

Since the intensity of the optic disc is much higher than the retinal background, a posible method in order to localize the optic disc is to find the largest clusters of pixels with the highest gray levels. Therefore, the pixels with the highest 1% gray levels are selected. After this, a clustering algorithm groups the nearby pixels into clusters. Iniatilly, each point is a cluster and its own centroid. If the euclidean distance between two centroids is less than a specified threshold ϵ, these clusters are combined to one cluster. The new centroid (c_x, c_y) is computed as

$$c_x = \sum_{i=0}^{n} \frac{x_i}{n} \qquad (1) \qquad\qquad c_y = \sum_{i=0}^{n} \frac{y_i}{n} \qquad (2)$$

where (x_i, y_i) is each cluster point and n is the number of points of the cluster.

After the combination process, the cluster with the maximum number of points is selected. The points of this cluster correspond with the points of the optic disc since the utilized images have not large area of exudates. The region of interest is defined as $n \times m$ rectangle whose center is the centroid of this cluster. The rectangle size depends on the image resolution. Figure 2 shows the computed regions of interest in two retinal images.

3 Extraction of the Optic Disc

Once the region containing the optic disc is computed, the extraction of the optic disc is performed. Since the optic disc has a circular structure, the extraction

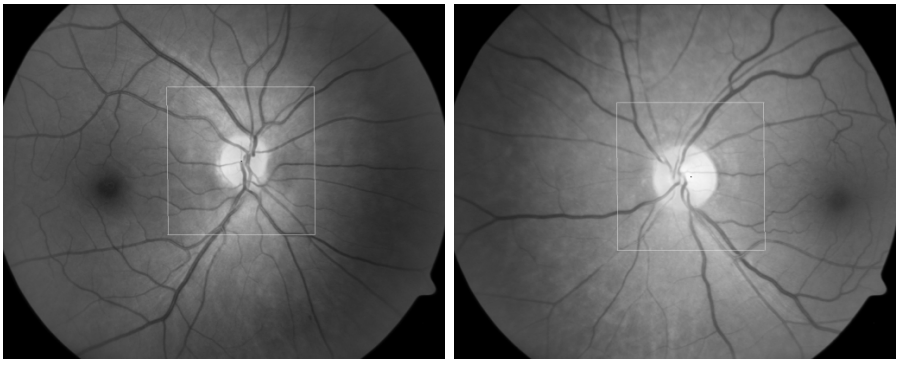


Fig. 2. Regions of interest in two retinal angiographies

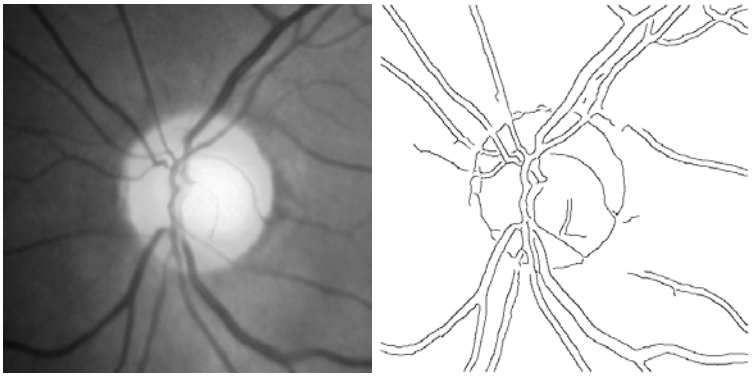


Fig. 3. The computed region of interest and the edges of this region

process consists of searching for circular shapes. To this end, the fuzzy circular Hough transform is applied to the edges of the region of interest. These edges are computed by means of the Canny filter [6]. Figure 3 shows the computed region and the edges of this region.

3.1 The Fuzzy Circular Hough Transform

The Hough transform [5] is widely used in Computer Vision and Pattern Recognition for the detection of geometrical shapes that can be defined through parametric equations. This paper describes the fuzzy circular Hough transform based on the edge images obtained from previous process.

The Hough transform for the detection of circles is based on the parametric equation of the circle, defined as:

$$(x_i - a)^2 + (y_i - b)^2 = r^2 \tag{3}$$

where (a, b) are the coordinates of the circle center and r is the radius. The implementation of the Hough transform in a digital computer requires the quantisation of the continuous $a - b - r$ space into suitable sized x cubes and the association of each of these cubes with a cell of a $3D$ accumulator array A of x size.

In order to reduce the computational cost and using the known characteristics of the optic disc, the circle (a, b, r) must satisfy two restrictions. First, the radius r must be between the minimum radius r_{min} and the maximum radius r_{max}. Then, the center (a, b) must be in a $c_a \times c_b$ window centered at the center point of the region of interest. The optic disc size and the image resolution are considered to set these values.

In order to generate the accumulator array, the contribution of each edge pixel $p = (x_i, y_i)$ to the accumulator array is computed. Firstly, the angle θ_p of the pixel is determined from the Sobel operator [7]. Then, the voting space of the pixel is constructed. In the fuzzy Hough transform, each pixel votes for the set of centers and corresponding radios contained in the gray area depicted in figure 4. This area is delimited by the two lines that pass through p with slopes $\theta_p + \frac{\pi}{12}$ and $\theta_p - \frac{\pi}{12}$. Moreover, the centers and the radius contained in this area must satisfy the previous restrictions in order to belong to the voting space.

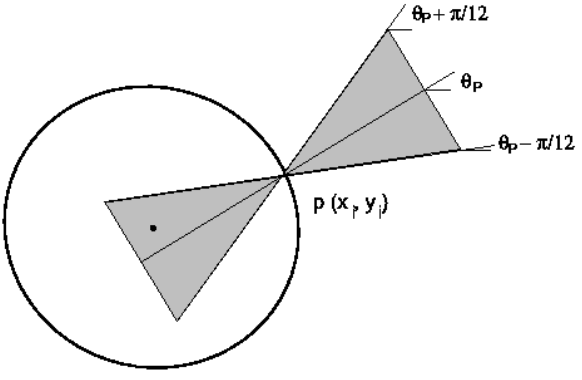


Fig. 4. Voting space of pixel p in the fuzzy circular Hough transform

The contribution of the pixel p to the accumulator array is not homogeneous for all the (a_j, b_j, r_j) in figure 4. The contribution to the accumulator array is maximum over the line that passes through p with slope θ_p and must rapidly decrease with the orientation difference, falling to the minimum when $d(\theta_p, \theta_j) = \frac{\pi}{12}$. θ_j is the slope of the line that joins (a_j, b_j) and p. Specifically, the contribution is defined through the following Gaussian function:

$$A(a_j, b_j, r_j) = e^{-\beta \cdot d(\theta_p, \theta_j)} \tag{4}$$

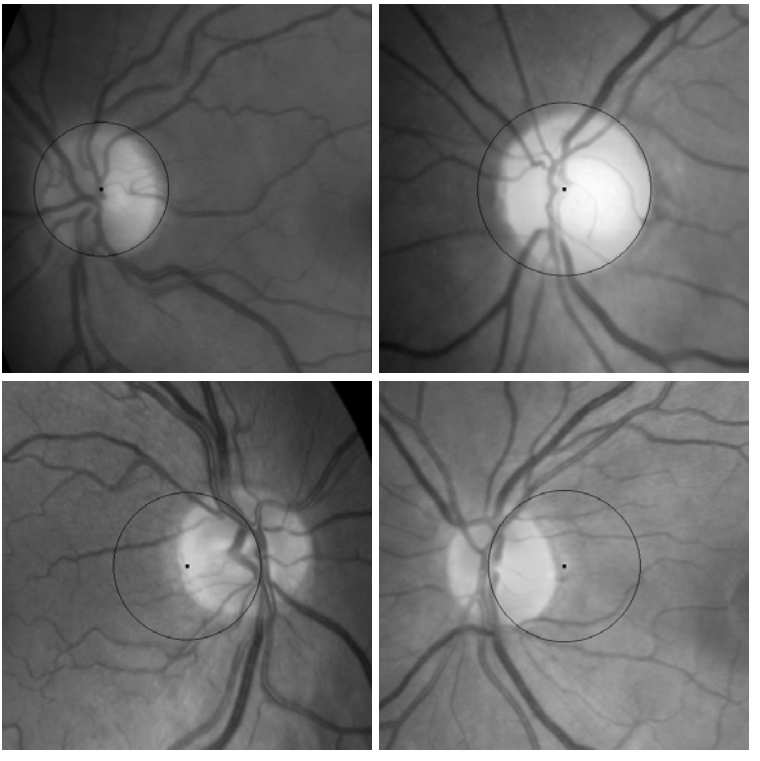


Fig. 5. In the first two images, the optic disc is extracted while in the following ones the two optic discs are not extracted due to the vessel edges

where β is the parameter that defines the decay of the Gaussian function.

After processing all edge points, the maximum value of the accumulator array (a, b, r) should correspond with the optic disc. Due to the vessel edges in the inside of the optic disc, the fuzzy circular Hough transform might not extract the optic disc.

Four different results are shown in figure 5 . The first two images show that the optic discs are extracted while the following ones show that the discs are not extracted due to the vessel edges.

3.2 Automatic Extraction of Vessels

In order to eliminate circles which belong to vessel edges, the vessel edge points are removed. To this end, an automatic extraction of vessels is done by means of the following crease extraction algorithm.

Vessels are reliable landmarks in retinal images because they are almost rigid structures and they appear in all modalities. Moreover, they can be thought of as creases (ridges or valleys) when images are seen as landscapes.

Amongst the many definitions of crease, the one based on level set extrinsic curvature, LSEC (5), has useful invariance properties. Given a function $L : \mathbb{R}^d \rightarrow \mathbb{R}$, the level set for a constant l consists of the set of points $\{\mathbf{x}|L(\mathbf{x}) = l\}$. For $2D$ images, L can be considered as a topographic relief or landscape and the level sets are its level curves. Negative minima of the level curve curvature κ, level by level, form valley curves, and positive maxima ridge curves.

$$\kappa = (2L_xL_yL_{xy} - L_y^2L_{xx} - L_x^2L_{yy})(L_x^2 + L_y^2)^{-\frac{3}{2}} \tag{5}$$

However, the usual discretization of LSEC is ill–defined in a number of cases, giving rise to unexpected discontinuities at the center of elongated objects. Instead, we have employed the $MLSEC-ST$ operator, as defined in [8] and [9] for the case of 3–D landmark extraction of CT and MRI volumes. This alternative definition is based on the divergence of the normalized vector field $\bar{\mathbf{w}}$:

$$\kappa = -\mathrm{div}(\bar{\mathbf{w}}) \tag{6}$$

Although (5) and (6) are equivalent in the continuous domain, in the discrete domain, when the derivatives are approximated by finite centered differences of the Gaussian–smoothed image, (6) provides much better results.

The creaseness measure κ can still be improved by pre–filtering the image gradient vector field in order to increase the degree of attraction/repulsion at ridge–like/valley–like creases, which is what κ is actually measuring. This can be done by the structure tensor analysis:

1. Compute the gradient vector field $\mathbf{w}$ and the structure tensor field $\mathbf{M}$

$$\mathbf{M}(\mathbf{x}; \sigma_{\mathrm{I}}) = G(\mathbf{x}; \sigma_{\mathrm{I}}) * (\mathbf{w}(\mathbf{x}) \cdot \mathbf{w}(\mathbf{x})^{\mathrm{t}}) \tag{7}$$

   being $*$ the element–wise convolution of matrix $\mathbf{w}(\mathbf{x}) \cdot \mathbf{w}(\mathbf{x})^{\mathrm{t}}$ with the Gaussian window $G(\mathbf{x}; \sigma_{\mathrm{I}})$.
2. Perform the eigenvalue analysis of $\mathbf{M}$. The normalized eigenvector $\mathbf{w}'$ corresponding to the highest eigenvalue gives the predominant gradient orientation. In the structure tensor analysis, opposite directions are equally treated. Thus, to recover the direction we put $\mathbf{w}'$ in the same quadrant in 2–d, or octant in 3–d, as $\mathbf{w}$. Then, we obtain the new vector field $\tilde{\mathbf{w}}$ and the creaseness measure $\tilde{\kappa}_d$:

$$\tilde{\mathbf{w}} = \mathrm{sign}(\mathbf{w}'^{\mathrm{t}}\mathbf{w})\mathbf{w}' \tag{8}$$

$$\tilde{\kappa} = -\mathrm{div}(\tilde{\mathbf{w}}) \tag{9}$$

For full details and further refinements regarding this operator please refer to [8, 9].

Once the crease image in the region of interest is computed by means of the previous process, the vessel edge points are removed. It is checked if a edge point (x_i, y_i) is a part of a vessel. A $w_a \times w_b$ neighbourhood window centered at the crease point is considered in the edge image. If the direction of an edge point of

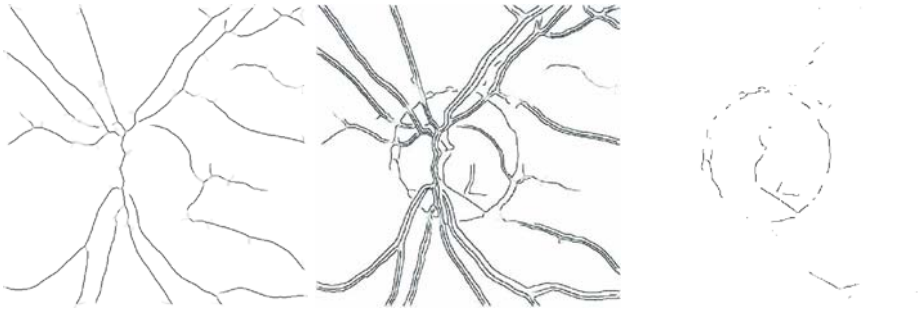


Fig. 6. The crease image (left), the edge image of figure 3 plus the crease image (center) and the final edges after removing the vessel points (right)

the window is the same as the direction of the crease point, this edge point is removed.

Figure 6 shows the crease image of the region of figure 3, this crease image plus the edge image of figure 3 and the final edge points after removing the vessel points.

4 Results

In this work, 1024×1024 gray level images are used. The optic disc is the brightest area in these images, since these images have not large area of axudates.

In the localization of the optic disc, a value of 350 pixels was used for the parameter ϵ in the clustering algorithm. The rectangle size which defines the region of interest $n \times m$ is set to 350×350.

In the Canny filter, a value of 2 was used for the parameter σ. The low threshold of edge strength t_{low} was set to 0.5 and the high threshold t_{high} was set to 0.8.

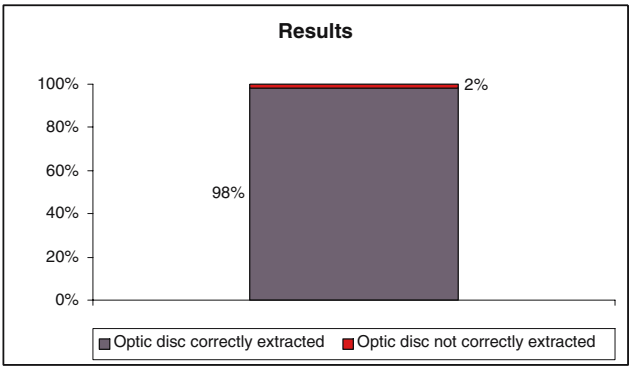


Fig. 7. The optic disc is correctly extracted in 98% of the cases

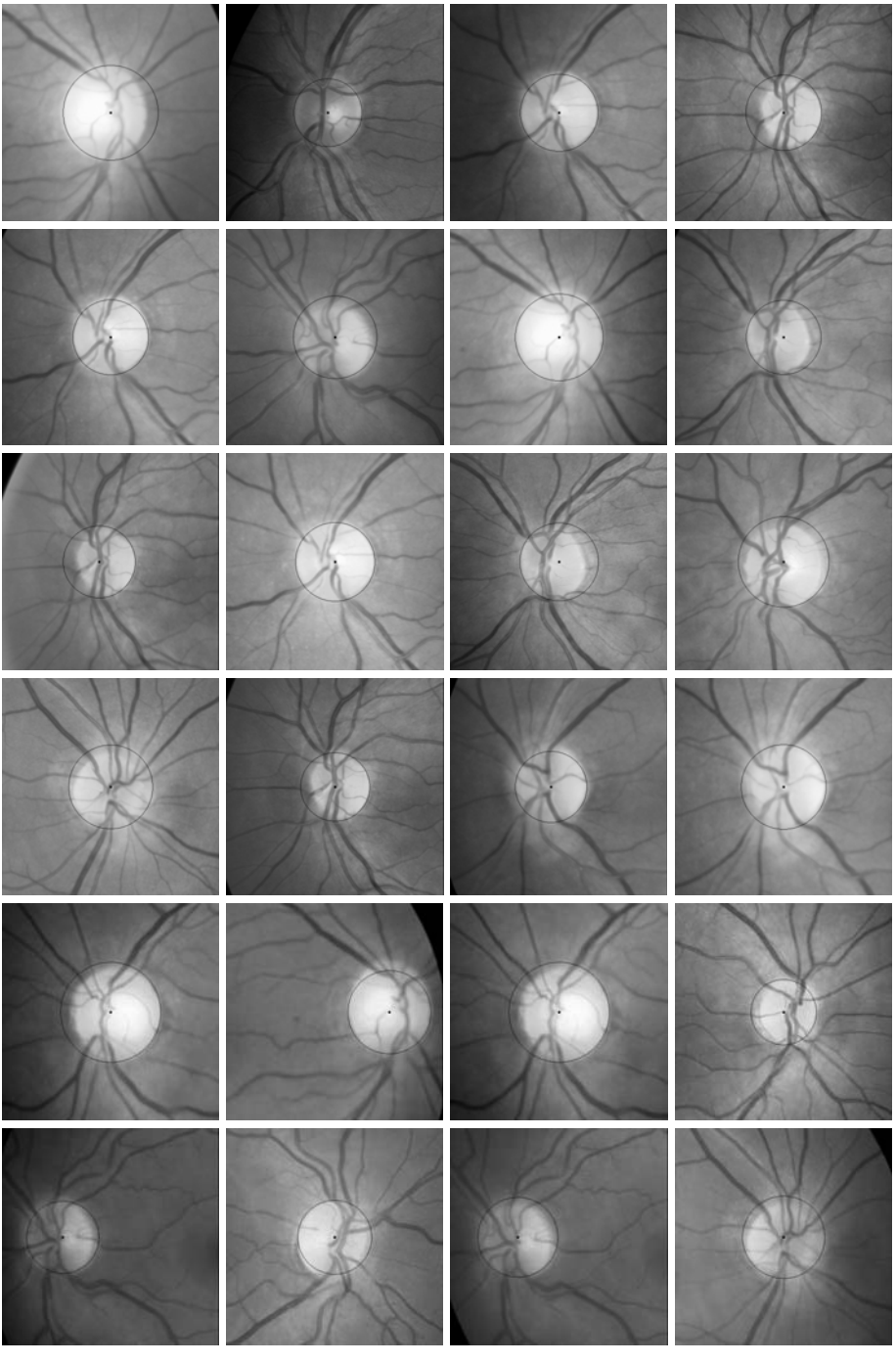


Fig. 8. Obtained results in different retinal images

In the fuzzy circular Hough transform, the parameter of the maximum radius r_{max} was set to 125 and 75 was used for the minimum radius r_{min}. The window size $c_a \times c_b$ where the center must be is set to 80×80. The parameter β was set to 10 in the gaussian function. Finally, the neighbourhood window size $w_a \times w_b$ was 6×6 in the process in order to remove vessel edges.

The system was tested on a set of 100 images given by the University Hospital of Santiago de Compostela (CHUS). The final results were evaluated by several ophthalmologists. The proposed system localizes the region of interest in 100% of the cases and extracts the optic disc in 98% of the cases. Figure 7 shows this statistic.

Figure 8 shows the obtained results in retinal images with different contrast and different optic disc sizes.

5 Conclusions

A new method to extract the optic disc has been proposed. Two main parts have been described. On one hand, a clustering algorithm is used in order to localize the region where the optic disc is. This region is compute in 100% of the cases since the utilized images have not large area of exudates.

On the other hand, the fuzzy circular Hough transform is applied to the edges points of this region in order to extract the optic disc. The optic disc might not be extracted since there are vessels in the inside of the optic disc. For this reason, a crease extraction algorithm is applied to the retinal image. The vessels are extracted automatically and the vessel edge points are removed in the edge image. The final technique extracts the optic disc in 98% of the cases.

References

1. Lalonde, M. and Beaulieu M. Gagnon, L.: Fast and robust optic disk detection using pyramidal decomposition and Hausdorff–based template matching. IEEE Transactions on Medical Imaging **20** (2001) 1193–1200
2. Mendels, F. and Heneghan C. and Thiran J.P.: Identification of the optic disk boundary in retinal images using active contours. Proceedings of the Irish Machine Vision and Image Processing Conference (1999) 103–115
3. Lowell, J. and Hunter, A. and Steel, D. and Basu, A. and Ryder, R. and Fletcher, E. and Kennedy, L.: Optic nerve head segmentation. IEEE Transactions on medical Imaging **23** (2004) 256–264
4. Chanwimluang, T. and Fan, G.: An efficient algorithm for extraction of anatomical structures in retinal images. IEEE International Conference on Image Processing (2004) 1093–1096
5. Hough, P.V.C.: Method and means for recognizing complex pattern. U.S. Pattern 06954 (1962)
6. Canny, J.: A computational aproach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **8** (1986) 679–698

7. Pratt, W.: Digital Image Processing. New York: Wiley (1978)
8. López, A. and Lloret, D. and Serrat, J. and Villanueva, J.J.: Multilocal creaseness based on the level set extrinsic curvature. Computer Vision and Image Understanding **77** (2000) 111–144
9. López Peña, A. and Lumbreras, F. and Serrat, J. and Villanueva, J.J.: Evaluation of methods for ridge and valley detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999) 327–335

Object Recognition for Obstacle Avoidance in Mobile Robots

José M. Bolanos, Wilfredis Medina Meléndez, Leonardo Fermín, José Cappelletto, Gerardo Fernández-López, and Juan C. Grieco

Simon Bolivar University,
Mechatronics Group, ELE-302
Sartenejas 1080-A Miranda, Venezuela
wmedina@usb.ve

Abstract. In this paper is shown an obstacle avoidance strategy based on object recognition using an artificial vision application. Related works focus on the implementation of efficient algorithms for image processing. This work emphasizes in using minimum information from an image in order to generate free obstacles trajectories. The algorithm used is based on Pattern Matching for detection of the robot and Classification for the rest of objects. Each form of detection has its particular algorithm: Cross Correlation for Pattern matching and Nearest Neighbor for Classification. The objective pursued is to demonstrate that, with a very simple system, precise information can be provided to a navigation system in order to find free obstacle paths.

1 Introduction

Nowadays traditional robotics sensors have been shifted by artificial vision systems. Several works on this area have been published and the interest of most researchers is to develop faster systems trying to do real-time processing [1] [2]. This work presents an artificial vision system where the objective is to extract the minimum information in order to detect objects and supply their position, size and orientation. This data will be useful to guarantee the generation of free-obstacle trajectory in future works, instead of efficiency improvements.

The application developed takes pictures and, using techniques of pattern recognition and classification, is able to provide the position of obstacles, and models last ones as simple geometric forms containing the real obstacle. The system is also able to detect a mobile robot and return its position and orientation. Tests have been done and are shown to demonstrate the effectiveness of the application.

The work is presented in the following way: In Sect. 2 the methodology used to develop the application as well as the concepts involved are described. Section 3 is devoted to explain tests made to validate the system. An analysis is included. Paper ends with Sect. 4 where conclusions are shown.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 722–730, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Implementation

2.1 The Application

The main sensor used is a wireless camera providing 30 frames per seconds of 640x480 pixels. The image acquired is cropped in order to provide 480x480 pixels, covering an area of $6.25\,m^2$. All the experiments were done with 8 bits grayscale images, and the workspace, the surface where the mobile robot moves, was chosen black in order to solve luminosity variability. The platform employed for development was Labview v7.1, where a Dynamic Link Library (DLL) was created in order to improve the processing speed of the application. The DLL is called from a program that is being developed in Matlab to generate free-obstacles paths.

The processes described in Sect. 2.2 and Sect. 2.3 are executed and the system provides enough information to a velocity field generator. The global objective is to generate dynamic velocity references for a controller based on this Vision System.

2.2 Mobile Robot Detection

A pattern matching algorithm is used to detect the mobile robot. It consists in the localization of regions that match with a known reference pattern on a grayscale image. The reference pattern is also known as template, and contains information related to edge pixels and region pixels, removing redundant information in a regular image. In this way, the matching process is done in a faster and more accurate manner. In the case where a pattern appearing rotated in the image is expected, it is possible to store pixels specially chosen, whose values reflect the pattern rotation.

Robot detection process is divided into two stages: Learning and matching.

Pattern Learning Algorithm. It can be summarized as follows [3]:

- **Pseudo-random sub-sampling of image**, which allows an improvement in the sampling uniformity through the template without using a predefined grid. If an uniform grid is used, information related to horizontal or vertical edges could be lost in the process of sampling. In the other hand, if a random sampling is used then it may produce clusters in the same area, or open non-sampled areas which contain vital information.
- **Stability analysis**, where the pseudo-random pixels are analyzed to check their stability in their neighborhood. Based on this information every pixel is classified according to the size of its stable neighborhood. Doing this, a reduction in the number of comparisons in the matching phase is achieved.
- **Features identification**, which consists in an edge detection process, storing information about their position.
- **Rotation-Invariant Analysis**, in which a circular intensity profile is identified in the template and is used to recognize a version of the template rotated having the same profile.

Process Description. The detailed description of the pattern learning algorithm implementation is as follows [4]:

- **Initialization:** An 8-bit blank image is created.
- **Image file load:** Over the blank image, the image file containing the pattern desired for learning is loaded. It must be an 8-bit image.
- **Image conversion:** The image is converted to a grayscale image so it can be processed by the learning module.
- **Pattern Selection:** The region of interest is selected and it will be the pattern to match. Figure 1 shows the selection process.

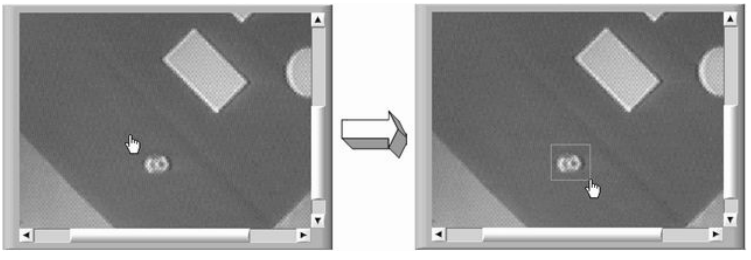


Fig. 1. Initialization - Pattern Selection

- **Learning module setup:** The learning module is configured to generate information for rotation-invariant pattern matching, that means, for detection of the mobile robot pattern regardless of its rotation.
- **Learning:** the learning process is done according to the configuration above described and resulting data is stored in a PNG file.

The image generated in the process above is employed by the pattern matching sub-system described next.

Pattern Matching Algorithm. It can be divided into two main processes [3]:

1. A circular intensity profile, obtained in the learning phase, is used to locate shifted versions of itself through the image.
2. Pseudo-random sampled pixels are used in a correlation process between candidates identified in the previous process, generating a score for each one employed later to determine if it matches or not.
   Correlation process is based in the calculus of the squared Euclidean distance [5][6]:

$$d^2_{f,t}(u,v) = \sum_{x,y} \left[f(x,y) - t(x-u, y-v) \right]^2 , \tag{1}$$

   where f is the image, and the sum is over (x, y), in the window containing the sub-image t located at u, v. Expanding d^2, it results:

$$d^2_{f,t}(u,v) = \sum_{x,y} \left[f^2(x,y) - 2f(x,y)t(x-u, y-v) + t^2(x-u, y-v) \right] , \tag{2}$$

where the term $\sum_{x,y} t^2(x-u, y-v)$ is constant. If $\sum_{x,y} f^2(x,y)$ is nearly a constant, the remaining term of the cross correlation

$$C(u,v) = \sum_{x,y} f(x,y)t(x-u, y-v) \tag{3}$$

is the similarity or matching measure between image and template. Due to sensibility of this term to changes in the image amplitude, the correlation coefficient is normalized [7]:

$$R(u,v) = \frac{\sum_{x,y} \left(f(x,y) - \overline{f_{u,v}}\right)\left(t(x-u, y-v) - \overline{t}\right)}{\left[\sum_{x,y} \left(f(x,y) - \overline{f_{u,v}}\right)^2 \sum_{x,y} \left(t(x-u, y-v) - \overline{t}\right)^2\right]^{\frac{1}{2}}}, \tag{4}$$

where $\overline{t}$ is the mean of the intensities of pixels in the image and $\overline{f_{u,v}}$ is the mean of $f(x,y)$ inside the template.

Process Description. The detailed description of the pattern matching algorithm implementation is as follows [4]:

- **Initialization:** Two 8-bit blank images are generated. One will be used for the video capture, and the other one will be used to load the image containing the desired pattern. Video acquisition in the NI IMAQ 1407 is initialized.
- **Image capture:** A real grayscale image from the workspace is captured.
- **Cropping:** To simplify position calculations and to achieve an increase in the detection speed, the captured image is cropped to a 480x480 pixels (from 640x480 pixels). This image size allows the visualization of a workspace of $6.25\,\mathrm{m}^2$.
- **Information load:** The information (related to pattern learning) contained in the PNG image stored in the Learning process is loaded.
- **Pattern matching module setup:** The pattern matching module is set to rotation-invariant mode so it can detect the desired pattern regardless of its rotation.
- **Matching:** The matching process is done according to the configuration above described between the captured image and the loaded image (with the information from the learning process). If the desired pattern is located, the result will be its position within the image and its orientation.

2.3 Obstacle Detection

The obstacle detection process has two main phases: learning process, and object detection and classification [3].

The learning process or training consists in the collection of a set of samples of images emulating the possible obstacles that will be encountered and captured by the camera, here limited to regular geometric forms. From these samples a set of features are extracted and different classes are created: “circle”, “square”

and "rectangle". Once classification is finished, next step is to calculate the features value, in this case, elongation and circularity. With these values, a classifier session file containing them is generated, which will be employed later for comparisons in order to achieve the classification of the unknown figures captured by the camera.

More complex objects will be considered as rectangles. Once an object of any shape is detected, the application will provide information corresponding to the detection of a rotated rectangle whose area is the minimum needed to surround the whole object.

The object detection and classification phase involves input image preprocessing, feature extraction and classification. This part of the system was implemented in the same DLL where mobile robot detection stage was done, giving the facility of sharing the first three steps between the two stages of the application. The only difference is in the number of blank images generated. For this stage, four 8-bit blank images are generated.

The input parameters are adjusted in a way that the system "can see" the objects within the visual field. The detailed description of the obstacle detection process implementation is as follows [4]:

Image Preprocessing. The input image is passed to a particle analyzer, which will convert the grayscale image to a binary image through a thresholding process. The resulting image will contain the particles within the threshold range. Then, this image is filtered with a morphological process of erosion, rejecting small or insignificant particles.

Particle Detection. The filtered image is passed to a classification particle analyzer. This will return the center of masses of particles and the coordinates of the rectangles that enclose them (top-left and bottom-right pixels).

Classification. With the position of the particles obtained in the particle detection step and the classifier session already created, the particles in the binary image are classified by feature extraction. Through this, image data is reduced because the whole process will only consider the values of the features that distinguish each one of the different classes. In addition, those features are invariant to changes in the scale, rotation and mirror symmetry, making possible to classify objects correctly regardless of the rotation or scale they have within the image.

Last step is to classify objects captured in the images using the extracted features. The basic problem in classification is to assign n items in terms of k, based on the attributes of the item [8] [9]. The algorithm used is Nearest Neighbor (NN) [10], chosen due its calculus simplicity and effectiveness in situations where the number of features involved is low. The metric used is taxicab metric and was chosen to reduce the number of calculations.

Under this classification algorithm, the distance of a set of input features X of an unknown class to a class C_j is defined as the distance to the closest sample which is used to represent the class:

$$D(X, C_j) = \min_i d(X, X_i^j) \ , \tag{5}$$

where $d(X, X_i^j)$ is the taxicab distance [11] between X and X_i^j.

Then applying NN algorithm results in the following rule:

$$X \in \text{class } C_j \text{ if } D(X, C_j) = \min_i D(X, C_i) \ . \tag{6}$$

Obstacle Parameters Measurement. Once the objects are classified, the size of the smallest diagonal of the rectangle that enclose each one is obtained (this rectangle has 0° of rotation). When rectangle and squares are detected, the rotation of this objects are determined. For this task, the Rotation Detect function of the IMAQ Vision package is used, which obtains the object rotation comparing its image with a reference (previously stored) image containing the same figure with null rotation. Figure 2 shows the representation of the angle measured for each object detected.

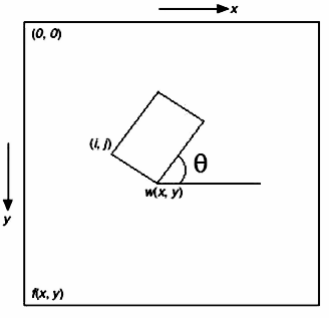


Fig. 2. Representation of the rotation angle of a detected object

Results. The values obtained in the processes above are arranged in arrays for each kind of object, which means that there are three arrays for position, three arrays for rotation angles and three else for diagonal size, that is, one array for each type of obstacle detected.

3 Experimental Tests and Results

To adjust the system a program under LabVIEW 7.1 was made. This program integrates the pattern matching and the obstacle detection systems described previously. In this program the values related to rotation and position of the mobile robot can be observed, as the values of orientation, size and position of the detected obstacles.

3.1 Pattern Matching System Results

For the rotation detection tests, the following procedure was done [4]:

– Initially, the mobile robot was placed over a small black plane of 100x50 cms. with a metallic rule aligned with it.

– The robot was oriented 0° according to the reference established by the image taken by the camera over the workspace.
– With the aid of a protractor, the black plane was rotated seven times in steps of 45° until complete a revolution, and measurement was done. Figure 3 shows images adquired and Table 1 the summary of results.

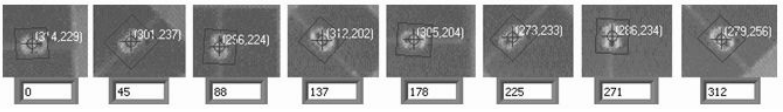


Fig. 3. Pattern Detection Results

Table 1. Mobile detection test results

	#1	#2	#3	#4	#5	#6	#7	#8
Manually Mesuared Angle	0°	45°	90°	135°	180°	225°	270°	315°
Obtained Angle	0°	45°	88°	137°	178°	225°	270°	312°

Table 1 shows that there is a slight difference between some values obtained with the application and the manually measured values, being ±3° the maximum variation obtained. Besides this variation, the detection system showed random fluctuations in the pattern rotation value, but within the maximum variation value.

3.2 Obstacles Detection Results

Figure 4 shows the set of templates used for the obstacle classification training stage. Two different tests were done. Figure 5 shows the obtained results for classification. In Table 2 a summary of results for position and orientation is shown.

Figure 5 shows the correct classification of objects in images. Table 2 shows a comparison between real position and orientation of objects values measured by the application. In the two tests shown, the error encountered was always less than 2 pixels or 1 cm. Comparing this error with the dimensions of the objects involved, it resulted in insignificant differences.

For orientation, worst error found was for the square object: in one case 3° and in the other or 4°, while for the rectangle was 2°. Although this errors could be considered high and not desirables, they appear on a figure that is modeled as a circle to simplify calculations. So, the error doesn't have a practical meaning. It is important to note that the orientation of a circle is not measurable.

At this point it is important to remark that results provided by the application are always integers.

Additional tests were applied to the system using a mobile robot under velocity control, and real-time execution of path tracking was achieved. Details are beyond from the scope of this paper.

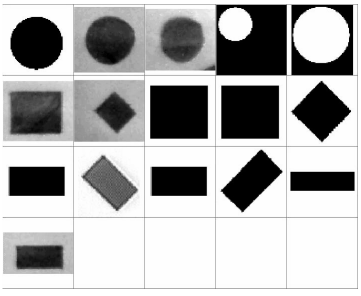


Fig. 4. Set of templates used for classification training

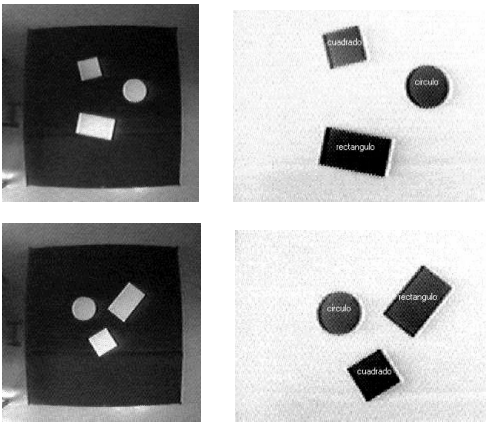


Fig. 5. Experimental results for classification in test 1 and test 2

Table 2. Summary of tests results

	Test # 1					
	Rectangle		Circle		Square	
	Reference	Obtained	Reference	Obtained	Reference	Obtained
X	214	214	322	320	229	229
Y	160	159	211	212	296	296
θ	-32°	-30°	N/A	N/A	25°	28°
	Test # 2					
X	300	299	198	198	243	243
Y	194	194	209	209	290	291
θ	72°	70°	N/A	N/A	59°	55°

4 Conclusions

A system for identification of a mobile robot and classification of obstacles was shown. After the proper processing of an image taken by a camera it returns the

position of every object in the scene as well as their orientation and size. The system was tested using a common environment and results obtained suggest that the system is very suitable for robotics applications.

The application developed and here presented is very light, computationally speaking; the amount of info used is too small such as objects position and orientation. The algorithm has been proved in real time applications where the mobile robot avoids perfectly the obstacles in the path. With only the small amount of info supplied a dynamic velocity field is created and modified for the obstacles in the navigation path. To present time, only simple geometric obstacles have been considered.

The behavior of the algorithm is achieved thanks to the DLL developed in order to connect the VI running in the Labview environment with the Matlab package.

Although satisfactory results were achieved, they can be improved using a better acquisition system which allows the use of more complex images, for instance, RGB, HSL, 32 bits. Besides, a more complex classification system can be used too. However, between complexity and velocity there is a tradeoff that has to be kept on mind. Future works point to include the classification of more than simple geometric forms to constitute a complete Vision System for the generation of general dynamic velocity fields for mobile robotics.

References

1. Hel-Or, Y., Hel-Or, H.: Real Time Pattern Matching Using Projection Kernels. IEEE Trans. on Patt. Anal. and Mach. Int. **27** (2005) 1430–1445.
2. Uenohara, M. and Kanade, T.: Use of Fourier and Karhunen-Loeve decomposition for fast pattern matching with a large set of templates. IEEE Trans. on Patt. Anal. and Mach. Int. **19** (1997) 891–898.
3. National Instruments: IMAQ Vision Concepts Manual, (2005) 12-1–12-8, 16-1–16-21.
4. Bolanos, J. M.: Embedded Control System Implementation for a Differencial Drive Vehicle. BSc. Thesis. Simon Bolivar University. (2006) 42–53, 69–76.
5. Richards, J. A., and Xiuping Jia; Remote Sensing Digital Image Analysis. Springer-Verlang, Third revised and enlarged edition. (1999) 124–132.
6. Lewis, J. P.: Fast Normalized Cross-Correlation. Industrial Light & Magic. Available in <http://www.idiom.com/˜zilla/index.html# Publications>
7. Gonzales, Rafael C., and Woods, Richard E.; Digital Image Processing. Addison-Wesley Publishing Company, Inc. (1992).
8. Tou, J. T. and Gonzalez, R. C.: Pattern Recognition Principles. Addison-Wesley Publishing Company, Inc. (1974).
9. Sing-Tze, Bow.: Pattern Recognition: Application to Large Data-Set Problems. Marcel-Dekker. (1984).
10. Mitchell, Tom.: Machine Learning. McGraw-Hill Science/Engineering/Math. (1997) 230–235.
11. Willard, S.: General Topology. Addison-Wesley. (1970) 16.

Gait Synthesis and Modulation for Quadruped Robot Locomotion Using a Simple Feed-Forward Network

Jose Cappelletto, Pablo Estevez, Wilfredis Medina, Leonardo Fermin, Juan M. Bogado, Juan C. Grieco, and Gerardo Fernandez-Lopez

University Simón Bolívar, Mechatronics Group, LabC-302
1080-A Sartenejas, Miranda. Venezuela
cappelletto@usb.ve

Abstract. This paper describes a technique for statically stable gait synthesis for a quadruped robot using a simple Feed Forward Neural Networks (FFNN). A common approach for gait synthesis based on neural networks, is to use an implementation with Continuous Time Recurrent Neural Network (CTRNN) of arbitrary complex architecture as pattern generator for rhythmic limb motion. The preferred training method is implemented using genetic algorithms (GAs). However, to achieve the desired trajectory becomes an obstacle during the training process. This paper presents a much more simpler process converting a statically stable gait into actuator's space via inverse kinematics; the training of the network is done with those references. By doing so, the training problem becomes a spatio-temporal machine learning problem. It is described a solution for trajectory generation combining a simple oscillator model with a Multilayer Feedforward Neural Network (MFNN) to generate the desired trajectory.

1 Introduction

Several works on legged robots have used biological principles as a source for solutions to common problems with biomechanical systems. At present time, it's widely accepted that motion control process in animals takes place in the spinal chord by the Central Pattern Generator (CPG). It's there where reflex signals, and high level brain and cerebellum signals are combined in order to produce coordinated excitation of neuromuscular system. Several authors have tried to model neuromotor system using Continuous Time Recurrent Neural Networks (CTRNN), as a consequence of rythmic and dynamical behaviour of CPG. This has proven be a good choice because of CTRNNs ability to model dynamical systems. However, in order to synthesize the $N^2 + 2.N$ parameters required to describe an N-neuron recurrent network, well known backpropagation training methods cannot be applied in a direct way. Different techniques for training of recurrent networks to model dynamical systems and other tasks have been proposed as those described by Tsung[12], Molter[5] and Nishii[10]. However,

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 731–739, 2006.
© Springer-Verlag Berlin Heidelberg 2006

coordination of legs movement given by phase relation between them remains unsolved with those training approachs. The most common training technique for CPG-based locomotion model, is to train a fixed-size recurrent network using Genetic Algorithms (GAs). R. Beer [3] uses a simple leg model and dynamical analysis to evaluate CPG performance, M. Lewis [9] based his research in waveform analysis of CPG output and a simple body model, or by heuristical methods as Cohen and Fukuoka[7, 8]. Other authors have used different CPG models, as the Amari-Hopfield neural oscillator[1] or a simple Amplitud Controlled Phase Oscillator (ACPO)[2].

This paper describes the use of a simple structure for the neural network based on biological models[6]: a main oscillator that acts as *pacemaker* and a multilayer feedforward network, to model a CPG that can generate different trajectories for quadruped robot legs. The conditions for stable walking and leg kinematic are described in Sect. 2. The complete neural network structure is shown and explained in Sect. 3. The training process of the neural network is described in Sect. 4, and finally Sect. 5 describes the experimental results obtained of the training process of the neural network for gait synthesis.

2 Static Walking in Quadruped Robot

In legged robot walking, the gait system generates the trajectory for the terminal element of the leg. Usually that trajectory is provided as a triplet of values in the $\mathbb{R}^3$ space for position, and converted to actuator's space via inverse kinematics. There is a model described in [11] that synthesize a wide variety of gaits like pace, trot, gallop and crawl. In that model there are three important parameters: Stride length ($\boldsymbol{\lambda}$) that represents the length of the leg step for a complete cycle, Period($\boldsymbol{\tau}$) it's the time measure for a step cycle and the Duty cycle($\boldsymbol{\beta}$) or the quotient between the support phase of the leg, when it is in contact with ground and $\boldsymbol{\tau}$. Modifying these parameters and the phase relations between each leg, it can be obtained different gait patterns.

For a legged platform the condition for stability in a static walking pattern is that vertical projection of weight vector onto the support surface must be inside the polygon described by the legs in contact with surface. In the particular case of a quadruped robot, that polygon can have either three or four sides. As mechanical platform it was used the Quadruped-3 of Lynxmotion. The legs have the typical reptile configuration with 3 Degrees of Freedom (DOF). Each DOF is directly actuated by a Hitec HS-475HB servomotor and named q_1, q_2 and q_3 respectively. Leg segments lengths are: $L_1 = 33\,mm$, $L_2 = 70\,mm$ and $L_3 = 113\,mm$.

Using Denavit-Hartenberg convention for selection of leg reference system, it was obtained the forward and inverse kinematics equations. Also, based on stable gait model we generated three different trajectories for leg's terminal element: triangle, rectangle, and rounded rectangle. Using the inverse kinematic it was obtained a set of waveforms for each leg's actuator. In the Fig. 1 are shown the waveforms for the different desired trajectories.

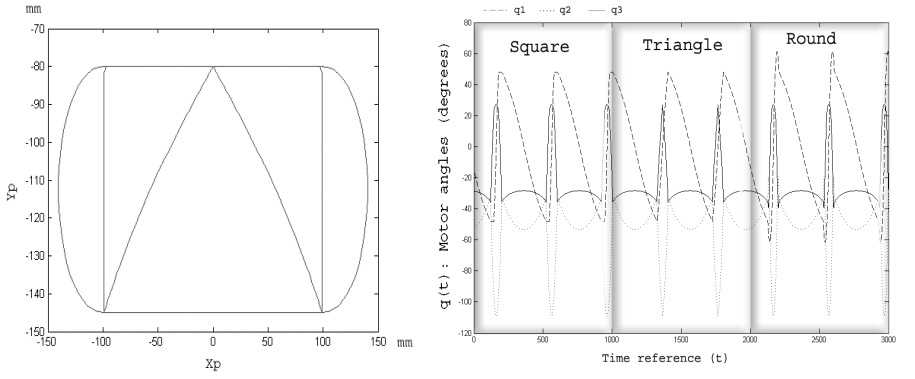


Fig. 1. Leg References: $\mathbb{R}^3$ space, and actuator's space

For all of the trajectories the gait parameters were $\beta = 0.8$, $\lambda = 100\,mm$ and $\tau = 7\,seg$, and leg phase relation was set to obtain the *crawling* gait. The condition for stability in a static walking pattern was met.

3 Neural Network Architecture

The whole pattern generator system is separated into two subsystems: a pacemaker subsystem and the MFNN. At the input level, the pacemaker subsystem generates a temporal reference output vector with dimension M. The following subsystem, implemented with a MFNN, performs a nonlinear space transformation $M \rightarrow N$, where N is the dimension of the output of the complete system; in this case $N = 3$ given by the number of actuators per leg. This network architecture allows extra parameters inclusion at Feed-Forward network input that can be used to modulate temporal reference output. It can be done without increasing complexity in training process as long as the number of neurons in the hidden layers is choosen so that Vapnik Chervonenkis (VC) dimension of the neural network is greater than problem's one.

3.1 Temporal Reference Subsystem

Three different models for temporal reference were evaluated. All of them had the same period T_t. The first one was an 1-D vector with a normalized ramp signal between 0 and 1. It was also used a 2-D vector described in (1).

$$\begin{cases} U = A \cdot \sin(\omega_s t + \phi_s) \\ V = B \cdot \cos(\omega_c t + \phi_c) \end{cases} \tag{1}$$

With this reference model it's possible to map any (U, V) point in $\mathbb{R}^2$, while modulating modulating A, B, ϕ_s and ϕ_c. Also modifying ω_s and ω_c it's possible to obtain different Lissajous figures for the (U, V) output, but for the model here described it's enough to stablish a fixed frequency value for both U and V.

The third temporal signal was a simple 2-neuron CTRNN. The recurrent neural network was based on the neuron model described by R. Beer[3]:

$$\tau_i \dot{y}_i = -y_i + \sum_{j=i}^{M} w_{ji}\sigma(y_j + \theta_j) + I_i, \quad i = 1, \cdots, M \tag{2}$$

where y_i is the output of the i^{th} neuron, τ is the neuron's membrane time constant, the conections weights are represented by w_{ij}, and the bias term θ_j for each neuron. The term I_i is an extern input that is held constant at zero, in order have the natural response of the system. The threshold function is the standard logistic sigmoid:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

3.2 Feed Forward Neural Network

For the space transformation subsystem it was used a simple multilayer neural network, with one hidden layer and the output layer. For the hidden layer it was used $K = 18$ standard neurons with sigmoidal transfer function (4), and $N = 3$ neurons with linear transfer function for the output layer (5).

$$y_i = \sigma\left(\sum_{j=1}^{M} w_j \cdot I_j + \theta_j\right), \quad i = 1, \cdots, K \tag{4}$$

$$y_i = \sum_{j=1}^{K} w_j \cdot I_j + \theta_j, \quad i = 1, \cdots, N \tag{5}$$

At hidden layer level, it was included an aditional *mode* input to the MFNN, in order to synthesize any of the three desired space trajectories with the same network. This was made to allow a soft transition between all training trajectories, instead the standard approach in geometrical models of robot locomotion, that performs abrupt changes between different leg trajectories. The resulting feedforward network structure is shown in Fig. 2:

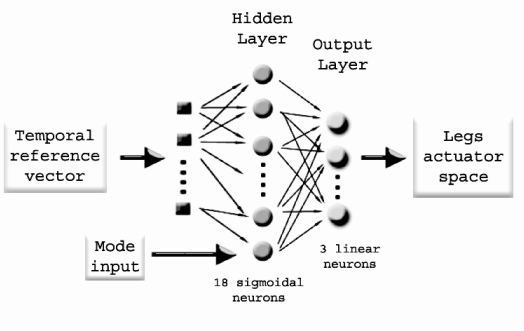


Fig. 2. Feedfoward network architecture

4 Network Training

The network was trained with standard backpropagation algorithm for feedforward networks. In this work, the feedforward network was used to synthesize a desired output trajectory. It was used Least Mean Square (LMS) error in order to compute the waveform aproximation error. The network training was perfomed using waveforms shown in Fig. 1 as target outputs and the different temporal references cases as inputs, giving three kinds of neural networks.

It was used the batch training method with a 100 points vector per batch. The training was applied 15 times to each network, starting from different random seeds, during 500 epochs each. Normal gradient descent approach was employed, without special modificiations added as momentum. This was done because this works points toward the simplest training method, instead the use of faster modified methods. For the whole training process it was used the neural network tool NNTOOL provided by Matlab®. Figure 3 shows the training scheme for the whole system.

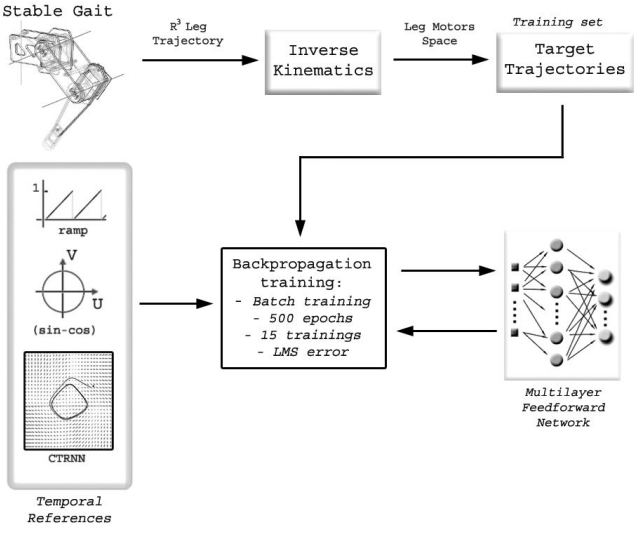


Fig. 3. MFNN training scheme

The parameters of the two-neuron recurrent network (U, V) used as pacemaker were synthesized through a simple genetic algorithm (GA). The GA used binary encoding for τ, θ and w_{ij} parameters. Those variables were coded with 16 bits with an absolute value lower than 20. As fitness function, it was used the Relative Frequency (RF) value of CTRNN 2-D output vector to evaluate its performance. RF was computed as the quotient between the total numer of oscillations and the total number of points of CTRNN output vector. This approach is faster and simpler than those described in [3] and [9] because there are less parameteres

involved in fitness function, and no special output waveform is requiered as long as is mets RF target. The cross-over subsystem was feeded with individuals obtained through the tournament selection algorithm. It was used the mutation operator with a mutation rate of 2%. The stop conditions for the training process with GA were the number of epochs (1000) or the percentual error for the RF fitness function ($\leq 1\%$).

5 Experimental Results

In the Fig. 4 is shown the temporal evolution of the MFNN network training process with the three different temporal references.

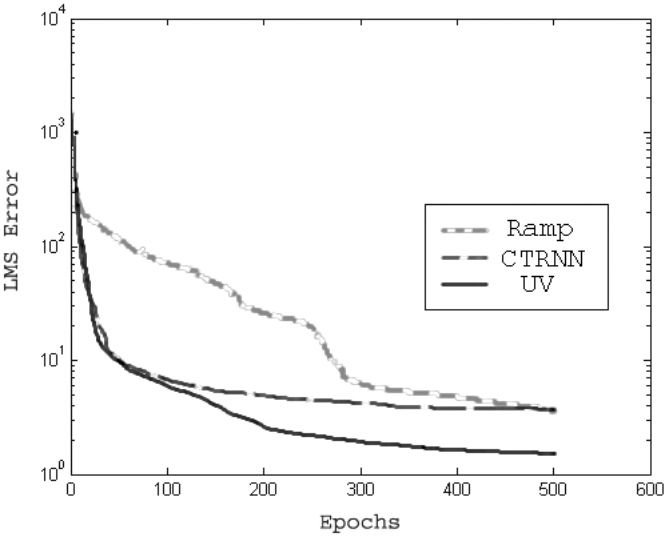


Fig. 4. MFN backpropagation training for 3 temporal reference inputs

It can be seen that backpropagation method eventually converges to solutions with low mean square error for waveform aproximation task, as it should be expected because of FFNN ability to perfom function aproximation with an arbitrary low output error. The best performance in training speed and output error value was obtained for the UV temporal input. For the *ramp* input the convergence was lower than the other two input cases. It can be explained if we see at desired output vector that describes a closed cycle trajectory as UV and $CTRNN$ output does too, helping both type of temporal references to converge to a suitable solution for the FFNN, this behaviour is not present for the *ramp* temporal reference.

The angle outputs for the networks trained with the three temporal reference modes are shown in Fig. (5) for the *ramp* input, Fig. (6) for (U, V) input and Fig. (7) for the CTRNN case.

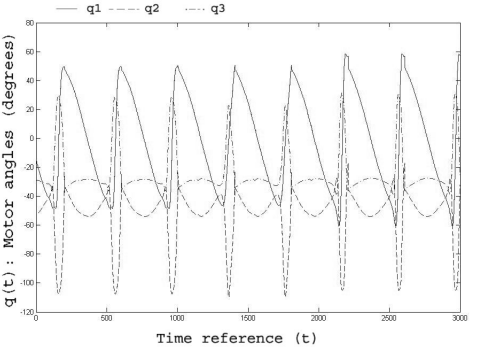


Fig. 5. Angles outputs for *RAMP* input case

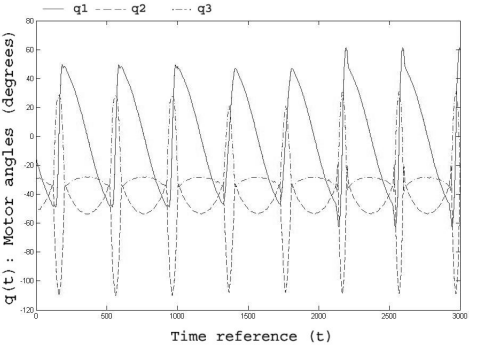


Fig. 6. Angles outputs for *UV* input case

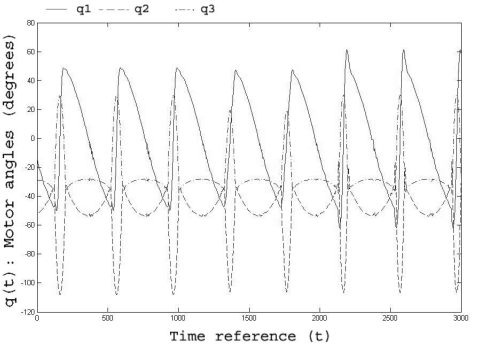


Fig. 7. Angles outputs for *CTRNN* input case

It was also observed that overfitting occured in the training process. This problem appeared because no special considerations to avoid it were taken during the training process. It would be enough to create a separate validation set, and include the stop condition for the backpropagation training when overfitting is detected. The overfitting itself represents an important problem for the design here described. It is desired to perform soft transition between the different leg trajectories shown in Fig. 1, but when overfitting occurs the neural network ability to modulate between those trajectories in a soft way is degraded. This is because it tries to match perfectly the training examples and also because no intermediate solutions were provided during training process.

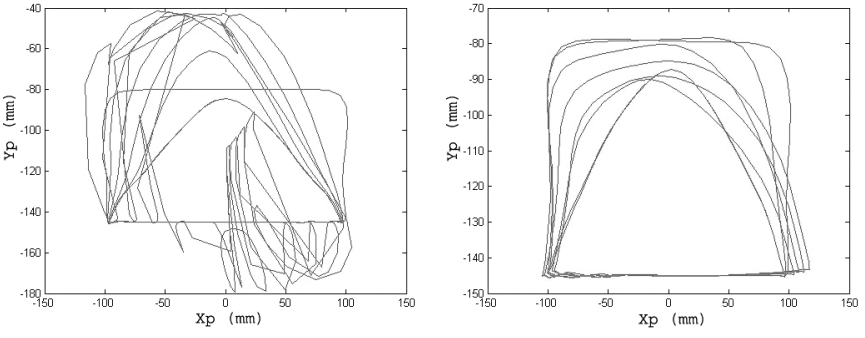


Fig. 8. Outputs transition for UV input with MSE = 0.131 and MSE = 0.417

The Fig.8 shows two MFNN solutions illustrating the negative incidence of overfitting in the trajectory modulation. It can be seen that the network with lower error match almost perfectly the training examples (triangle and rectangle), but intermediate outputs for *mode* varying from one trajectory to the other one exhibits abrupt changes. However the other MFNN solution, with a higher error for the examples, performs a better trajectory transition.

In further implementations, it can be used the same sigmoidal neuron model for output layer instead of linear neurons, because the output vector (servomotor angles) is a bounded space by mechanical limitations allowing values that range between -90 and 90 degrees. This only requires a linear transformation as $y = mx + b$ for the feedforward network outputs.

6 Conclusions

In this paper it has been shown that it is possible to synthesize the desired trajectories for 3DOF quadruped legs using simple Feed Forward neural networks. The simple idea, coming from biological systems, is to use oscillatory patterns designed to assure stable gaits to train the FFNN. The method presented improves previous methodologies using Recurrent Neural Networks, because the proposed subsystems, a pacemaker subsystem and a MFNN, can be implemented with the

well known backpropagation algorithm. In this way the use of GAs is limited to synthesize a simple 2-neuron oscillator for test without special amplitud requirement. Also it was shown that the use of the simple $\sin - \cos$ temporal reference has a better performance for trajectories generation. Several experimental results have been shown using the proposed method and the gait was sinthesized assuring the gait stability. The gait was tested on a small quadruped showing stable behaviour.

In the near future some different approaches are going to be tested, as combination of gait synthesis using the FFNN with strategies of position-force control on the quadruped leg. Also, it will be tested the inclusion of more gait parameters at the multilayer network input level, in order to sinthesize a wider range of leg trajectories allowing control of quadruped platform for irregular terrain locomotion.

References

1. S. Amari: Characteristic of the Random Nets of Analog Neuron-like Elements IEEE trans. on System, Man and Cybernetics, SMC-2, pp.643–657, 1972.
2. Buchli J. and Auke Jan Ijspeert: Distributed Central Pattern Generator Model for Robotics Application Based on Phase Sensitivity Analysis In Proc. of 1st. Intl. Workshop Bio-ADIT. 2004
3. Hillel J. Chiel, Randall D. Beer, John C. Gallagher: Evolution and Analysis of Model CPGs for Walking: I. Dynamical Modules Journal of Computational Neuroscience **7**, pp 99-118 (1999)
4. Hillel J. Chiel, Randall D. Beer, John C. Gallagher: Evolution and Analysis of Model CPGs for Walking: II. General Principles and Individual Variability Journal of Computational Neuroscience **7**, pp 99-118 (1999)
5. Molter, Colin: Chaos in small Recurrent Neural Networks : theoretical and practical studies Spec. Report, Univ. Libre de Bruxelles. 2004
6. Gerstner and Kistler: Spiking Neuron Models. Single Neurons, Populations , Plasticity Cambridge University Press, 2002
7. Fukuoka Y., Kimura H. and A. H. Cohen: Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Based on Biological Concepts Int. Journ. of Robot. Res. Vol 22, No. 3–4. 2003, pp 187–202
8. Kimura H., Fukuoka Y., Hada Y. and K. Takase: Three-Dimensional Adaptive Dynamic Walking of a Quadruped - Rolling Motion Feedback to CPGs Controlling Pitching Motion Proc. of ICRA 2002. pp 2228–2233.
9. Lewis, M.A., Fagg A. and G. Bekey: Genetic Algorithms for Gait Synthesis in a Hexapod Robot Recent Trends in Mobile Robots, pp 317–331, World Scientific, New Jersey, 1994.
10. Nishii J. and Susuki R.: Oscillatory network model which learns a rhytmic pattern of an external signal In Proc. of IFAC Symposium, pp 501–502. 1994.
11. D. J. Todd: Walking Machines. An Introduction To Legged Robotics Kogan Page Ltd. 1985
12. F-S Tsung: Modeling Dynamical Systems with Recurrent Neural Networks PhD Dissertation. Univ. of California, San Diego. 1994
13. Xu Guan, Haojun Zheng and Xiuli Zhang: Biologically Inspired Quadruped Robot BIOSBOT: Modeling, Simulation and Experiment 2nd Intl. Conf. on Autonomous Robots and Agents. New Zealand, 2004.

A Two-Stage Fuzzy Filtering Method to Restore Images Contaminated by Mixed Impulse and Gaussian Noises

Jyh-Yeong Chang and Shih-Mao Lu

Department of Electrical and Control Engineering
National Chiao-Tung University, Taiwan 300, R.O.C.
jychang@mail.nctu.edu.tw

Abstract. In this paper, we propose a two-stage fuzzy filtering method to sequentially remove the mixed noises of images corrupted with nonlinear impulse and linear Gaussian noises as well. In the first stage, a new decision-based method, called nonlinear fuzzy K-nearest neighbor (FK-NN) filter, detect and replace the outlier pixels, based on local processing window, to remove the nonlinear impulse noise. Then we derive a linear modified fuzzy rule-based (MFRB) filter to remove the linear type Gaussian noise while preserving the image edges and details as much as possible. For practical consideration, we design several sets of universal MFRB filters in correspondence to the estimated values of contaminated Gaussian noise variance in the image. The correspondent MFRB filter closest to the estimated Gaussian noise level will be selected to remove the Gaussian noise of the processed image. According to the experiment results, the proposed method is superior, both quantitatively and visually, compared to several other techniques.

1 Introduction

In the real-life, images are often contaminated by mixture of impulse and Gaussian noises of varying noise intensities due to the imperfection of sensors and communication channels when transmitted. The objectives of image noise removal are to remove the mixed noise and to retain the edges or other salient structures in the original image. Noise smoothing and edge enhancement are inherently conflicting processes, since smoothing a region will destroy an edge and sharpening edges might lead to enhance the unnecessary noise. Thus it is a difficult work for a universal algorithm [1] that can remove different kinds and intensities of noise from images and preserve their sharpness and details.

There are mainly three kinds of fuzzy approaches used in mixed noise removal of an image. The first kind is the fuzzy weighted average filter [2] and fuzzy weighted median filter [3]. Peng [4] proposed a multi-level adaptive fuzzy (MLAF) filter, which uses fuzzy sets to adaptively combine simple linear and nonlinear filters to remove varying mixed noise with different levels. The second kind is the fuzzy logic filter [5]-[7], which suggests that individual pixels should

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 740–749, 2006.
© Springer-Verlag Berlin Heidelberg 2006

not be uniformly fired by each of the fuzzy rules. Choi et al. [5] derived three different filters for each of the three objectives using the fuzzy weighted least squares (FWLS) method. Criteria are defined to select one of the three filters based on the local context using the fuzzy rules. Taguchi [6] proposed the modified fuzzy filter (MFF) with new local characteristic calculated with fuzzy rules by using multiple difference values between arbitrary pixels in the filter window. Farbiz et al. [7] proposed the fuzzy logic filter (FLF), which adopted the general structure of fuzzy if-then-else rules mechanism. The S-type fuzzy function enables the non-uniform firing of the basic fuzzy rules. For the third kind, Russo [8] used fuzzy reasoning for noise removal, which is embedded into the neural network structure through genetic learning algorithm. It is able to adapt the filtering action to different distributions of mixed noise.

The hybrid fuzzy filters are difficult to remove the mixed noises without blurring the edge and details information. The two tasks, involving suppressing the impulse noise and removing the Gaussian noise in an image, are very different in characteristics because each of them respectively facilitates the nonlinear and linear filtering operations. Besides, the presence of impulse noise can seriously degrade the performance of a restoration technique that is designed mainly to remove Gaussian type noise. Consequently, we propose in this paper a two-stage filtering technique to remove the nonlinear impulse and linear Gaussian noises sequentially. In the first stage, a decision-based method, fuzzy K-nearest neighbor filter (FK-NN), is proposed to detect and replace the outlier pixels detected. Based on local processing window, this method can remove the nonlinear impulse noise very efficiently and almost ignore the Gaussian noise which is usually not observed to be impulsive. Then we derive a linear modified fuzzy rule-based (MFRB) filter to remove the linear type Gaussian noise while best preserving image details. The noise in this stage is almost Gaussian because the image contains the original Gaussian noise and the small fraction residual impulse noise not filtered by the FK-NN scheme.

2 Fuzzy K-Nearest Neighbor Filter

The design rationale behind generalized median-type approaches is the order statistic theory used for impulse noise removal. Along this line of reasoning, the proposed fuzzy K-NN filter is a decision-based median-type filter, and it is augmented with a classifier to detect the impulsive pixel. For impulse noise removal of images, the fuzzy K-NN decision rule [9] is introduced to determine whether the central operating pixel of the sliding window belongs to the majority class or not. If the operating pixel is in the majority class, then it is left unchanged because it is probably a noise-free pixel. On the other hand, it is likely an outlier, i.e., an impulse corrupted pixel. Then the operating pixel is replaced by the median of the majority class. With this filter, the image structures and details can be best retained because the new median filter only modifies the outlier detected by the fuzzy K-NN classifier. The details of the fuzzy K-NN filter will be omitted here; interested reader may refer to [9] for more details.

3 Modified Fuzzy Rule-Based Filter

To remove the Gaussian noise artifacts in an image, we wish to design a group of adaptive filters, whose weighted value combinations constitute the fuzzy rules for pixel value restoration. Fuzzy rules have been found useful to reconstruct the pixel value from the pixel itself and its neighboring pixels as well. This is because that the local statistics and structural characteristics of the signals can be taken into consideration by the fuzzy rules. Along this line of reasoning, Arakawa [10] developed the Fuzzy Rule Based (FRB) filter that adapted the ambiguity of image signals caused from the following three parameters: gray level variations, signal spatial distribution, and the local structure of the pixels. The first parameter is the difference between the input pixels' gray values; indexed by j, the second is the time (or position) difference between pixel points; indexed by l, and the third is the pixel's local variance; indexed by m, in the sliding window. The filter coefficients are determined through learning from the difference between the noisy image and its original image. After training, the dominant roles of the signal distribution texture, spatial and edge structures, and local pixel statistics of the image will be learned by a set of weight parameters constituting the FRB filter. For best-fit and unbiased considerations, fuzzy rule based system through learning can achieve the best combinations of rules, and thus can usually produce a better off one in comparison with the filters constructed from the knowledge or experience of a domain expert. Therefore, the proposed MFRB filters scheme adopts a training method to learn suitable sets of weight parameters for image Gaussian noise removal. Namely, the weights of MFRB filters are obtained by minimizing the mean square output error between the noisy and original image data. As a result, pixel x_n can be restored, to remove the contaminated Gaussian noise, by MFRB filter using the pixels x_{n-k} inside the sliding window centered at x_n by $\hat{x}_n = \sum \mu_{jlm}[k] \cdot x_{n-k}$. MFRB filter [11] can adapt reasonably well from smooth regions to edge areas.

3.1 The Proposed Modified FRB Filter

In this section, we will propose the modified fuzzy rule based filters for image Gaussian noise removal. A set of filter coefficients of MFRB denoising filter can be determined through the training over Gaussian corrupted images. In the denoising phase, any corrupted images can be processed to reduce the noise employing the set of weighting coefficients we have learned. In an image, the pixel's gray levels, statistical variances, and positional distances and orientations are the three most important factors to be considered altogether. For the MFRB design below, we will not only better determine the gray-level difference interval $[\varepsilon_{j-1}, \varepsilon_j)$ and variance interval $[\delta_{m-1}, \delta_m)$ but also propose a new clustering scheme of pixels inside the working window, taking both pixels' positional difference and spatial correlation into consideration.

The details regarding the design of these three parameter spaces will be illustrated below. (1) to better design the error difference interval $[\varepsilon_{j-1}, \varepsilon_j)$ for the first parameter, the gray level difference or variable [a]: We propose that

the size of gray-level difference interval be chosen inversely proportional to the probability of gray level difference, to reach the spirit of A-law. That is, the smaller the gray-level magnitude difference is; i.e., the more frequent occurrence is, the smaller difference interval will be chosen for this gray level difference dimension, and vice versa. (2) to choose the cluster the pixel should belong to, for the second parameter, the spatiality or variable [b]: In our experiment, the size of the processing window of the MFRB filter is chosen to be 5×5. In the variable [b], it is reasonable to take both the position distance and orientation relation between pixel and pixel into consideration. With this concept in mind, we can divide the window pixels, except the central, into 12 clusters. Each cluster, indexed by l, contains two pixels and the line connecting them will have the same orientation and positional distance from the center pixel. (3) to better determine variance interval $[\delta_{m-1}, \delta_m)$ for the third parameter, the gray level local variance or variable [c]. The design of the third parameter, variance, is similar to the way we design variable [a], by dividing the variance interval $[\varepsilon_{j-1}, \varepsilon_j)$ being approximated inversely proportional to the relative frequency of local variance.

To design the MFRB filter, it follows from the LMS learning algorithm that μ_{jlm} can be obtained iteratively as follows:

$$\mu_{jlm}(T+1) = \mu_{jlm}(T) + \alpha \cdot t_{jlm} \cdot (x_{n-k} - y_n)(d_n - y_n) / \sum_{k=-N}^{N} \mu_{jlm}, \qquad (1)$$

where $\mu_{jlm}(T)$ is the value of μ_{jlm} at learning iteration T, α is the learning factor, and t_{jlm} equals 1 when belongs to the specified j, l, and m intervals and equals to zero otherwise. So in the training procedure, signal points are trained to update its involving interval weighting coefficient μ_{jlm} until μ_{jlm} converges. Note that the concept of the fuzzy rule based filter is named in a broad sense. There are no conventional fuzzy rules in the FRB filter because each 3-index weighting coefficients, i.e., the case differentiated by the partition in the 3-D parameter space, will be used to predict a distorted or noise-corrupted value by weighted-sum filtering procedure. Each μ_{jlm} coefficient corresponding to the specified j, l, and m intervals is considered as if it were a rule, and the set of all possible j, l, and m combinations constitutes the whole rule set. Moreover, the weighting coefficients versus each index dimension, i.e., j, l, or m, play the similar role of the membership function defined for that dimension. In summary, the size of the membership function matrix in each noise level is chosen to be a look-up table of size $64 \times 13 \times 100$, designed with the spirits of efficient partitioned intervals and adaptive edge-directed spatial relations being proposed.

4 Two-Stage Universal MFRB Filter for Mixed Noise Removal

For practical applications, the percentage of impulse noise and the variance of Gaussian noise are unknown in a noise-corrupted image. It is demonstrated that the fuzzy K-NN filter can effectively remove the impulse noise in spite of the different impulse noise corruption percentage [9]. So the noise remained after the

first stage, fuzzy K-NN filter stage, is mostly the Gaussian noise original in the image and a small fraction impulse noise not removed or imprecisely restored impulsive pixels.

Though the MFRB noise removal filter is most robust to the level of Gaussian noise corrupted, it is impractical and too hardware consuming to train the look-up table of each noise level. Since the MFRB filter works effectively in a certain noise level range, a universal MFRB filter can thus be obtained through training over several typical images having similar levels of Gaussian noise contamination. For cost-effectiveness consideration, we designed three MFRB filters at σ =10, 20, and 30 to constitute a universal MFRB filter. From our experimental experiences, these three MFRB filters can produce good enough filtering result and they do not cost too much in hardware and memory requirements. Accordingly, each MFRB filter of a certain noise level, i. e., σ =10, 20, and 30, can be trained by some ensemble images that are corrupted with the same Gaussian noise level. As usual, a train loop covering all training image set once is called an epoch, and let μ_{jlm}^{p} denote the membership function after learning p epochs. When the variation of the membership function in the training process is smaller than a predefined threshold, we will stop the training process. In our experiment, if the condition satisfies

$$\frac{\parallel \mu_{jlm}^{d} \parallel}{\parallel \mu_{jlm}^{p} \parallel} \leq \theta, \tag{2}$$

where $\mu_{jlm}^{d} = \mu_{jlm}^{p} - \mu_{jlm}^{p-1}$, and $\theta = 0.01$, we will terminate the training process and obtain the universal MFRB filter μ_{jlm} for this Gaussian noise variance. In this way, we will have a universal MFRB Gaussian noise removal filter in the sense that it is image independent and contains three MFRB de-noise filters with each constituted μ_{jlm} being designed for a Gaussian noise level σ =10, 20, or 30, respectively.

Since the particular MFRB filter to be employed from the universal MFRB filter depends on the Gaussian noise level in an image, the noise level estimation is necessary for efficient noise removal. It is well known that the noise variance of a local area can be estimated better by the local variance of a flat area to get rid of excessive offset by the edge and minute details of images. Accordingly, a method has been devised to estimate the Gaussian noise level of an image by averaging the smaller half variances estimated on the overlapping 7×7 sliding window over the whole image. After the Gaussian noise level estimation routine is activated to find the Gaussian noise level of an image, the nearest corresponding filter in the universal MFRB filter is selected to remove the remained noise of the image processed by the first, FK-NN, stage.

5 Experimental Results

For performance comparison, the proposed technique is evaluated and compared with other post-processing schemes, which include 3×3 and 5×5 Adaptive Wiener filter, FWLS by Choi *et al.* [5], MMF by Taguchi [6], FLF by Farbiz

et al. [7], and PBM filter by Chen *et al.* [1]. The simulations are based on a set of 256 level grayscale images of size pixels. For performance evaluation, the peak signal-to-noise ratio (PSNR) is used as a quantitative indication. The proposed algorithm has been extensively tested on various simulated images. In each performance comparison table, we rank the top three best-performed algorithms by superscript for clarity. We divide our experiments into three parts to demonstrate the capabilities of our proposed algorithm. First, we test the effectiveness of our second stage filter, the MFRB filter, to remove the Gaussian noise in images. The powerfulness of our first-stage FK-NN scheme to remove impulsive noise in an image is demonstrated in [9], and hence it is omitted here. Secondly, images contaminated with mixed noise, assume the Gaussian noise level is known, are processed to verify the proposed two-stage MFRB filtering method. At last, images corrupted with real-world mixed noise, assume unknown noise levels, are processed by the proposed two-stage universal MFRB scheme.

A. Gaussian Noise (Only) Case: Table 1 shows the different filter scheme results for four images corrupted with Gaussian noise of $\sigma = 20$. From Table 1, the proposed MFRB algorithm performs the best among the four test images. Our proposed MFRB filter also performs excellently both in removing the Gaussian noise and in preserving the edge and details in the image.

Table 1. Comparative Results of PSNR in the Cases of Corruption by Gaussian Noise ($\sigma = 20$)

	Win 3	Win 5	FWLS [5]	MFF [6]	FLF [7]	PBM [1]	MFRB
Boats	28.62	28.97^{2}	26.59	28.95^{3}	28.87	27.20	30.13^{1}
Bridge	26.25^{2}	25.04	24.99	25.04	25.48^{3}	25.46	26.40^{1}
Goldhill	28.21	28.51^{2}	26.46	28.47^{3}	28.44	26.97	29.45^{1}
Lena	28.91	29.95^{3}	26.88	30.12^{2}	29.63	27.69	30.98^{1}

B. Mixed Noise Case of Known Gaussian Noise Intensities: For general assessment, we experimented the proposed algorithm with various combinations of impulse noise densities ($p = 10\%$, 20%, and 30%) and Gaussian noises ($\sigma = 10$, 15, and 20). We individually apply the fuzzy K-NN filter to remove the impulse noise and then apply the MFRB filter with corresponding noise level to remove the Gaussian noise. To be brief, Fig. 1 shows the processed image of "Lena" having 20% impulse and Gaussian of σ=20. And we only show the results of the two-stage filter schemes in Table 2, the case of largest image corrupted noise. Wiener filters are also combined with the FK-NN to remove the impulsive noise in the first stage. FWLS [5] and FLF [7] decay abruptly in PSNR metric in the cases of high impulse rate noisy images. On the contrary, MFF [6], PBM [1] and our proposed algorithm are constantly stable in all the noise level combination. Our proposed two-stage MFRB filters can sequentially remove the impulse and Gaussian noises well and perform the best among them.

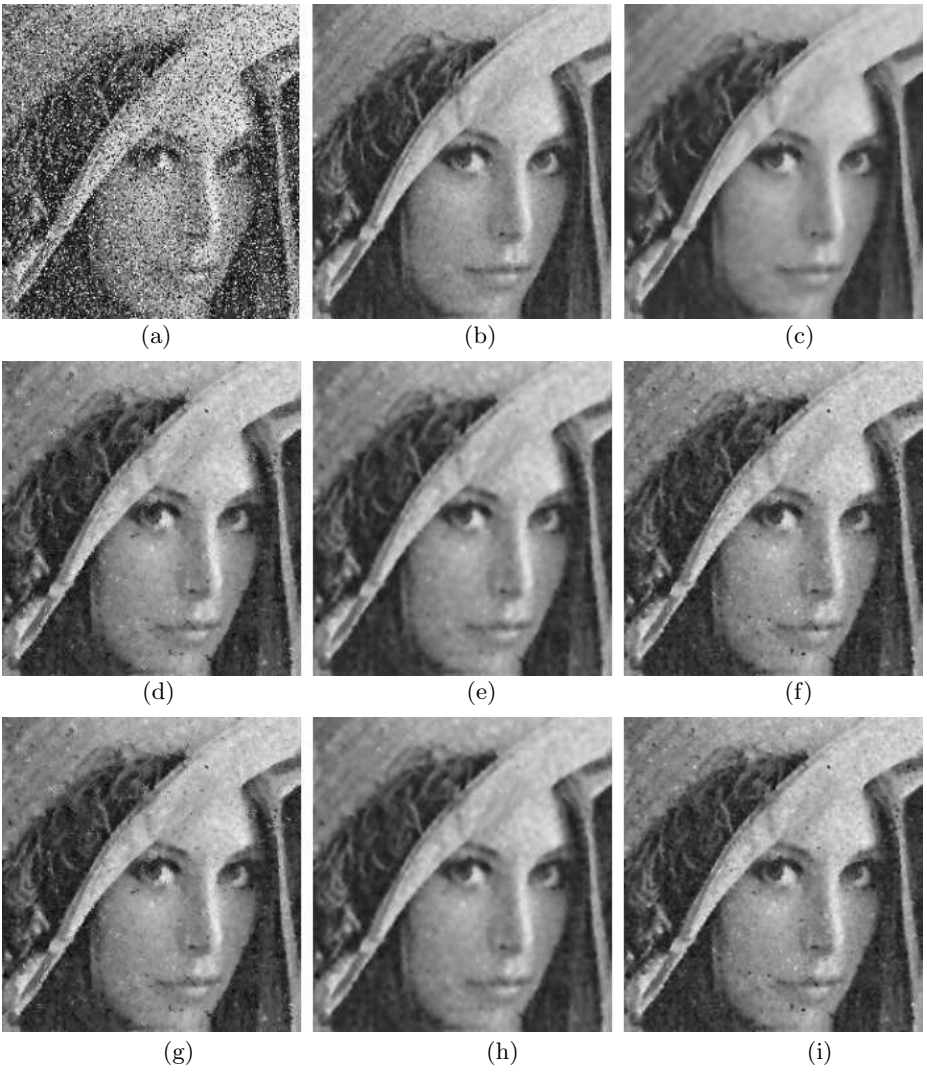


Fig. 1. The corrupted images (a) "Lena" with 20% impulses and Gaussian noise of $\sigma=$ 20; (b) FK-NN with Wiener filter; (c) FK-NN with Wiener filter; (d) FWLS [5]; (e) MFF [6]; (f) FLF [7]; (g) PBM [1]; (h) FK-NN; and (i) Two-Stage method

In view of visual perception from Fig. 1, both FWLS [5] and the FLF [7] method cannot remove the impulse noise well, which result in a large decay in the PSNR measurement. The other algorithms, MFF [6] and PBM [1], are incapable of removing the Gaussian noise well. The proposed nonlinear FK-NN filter can detect and remove the nonlinear impulse noise well in the first stage. Then the second-stage MFRB filter removes the residual Gaussian noise efficiently and preserves the image edges and details well.

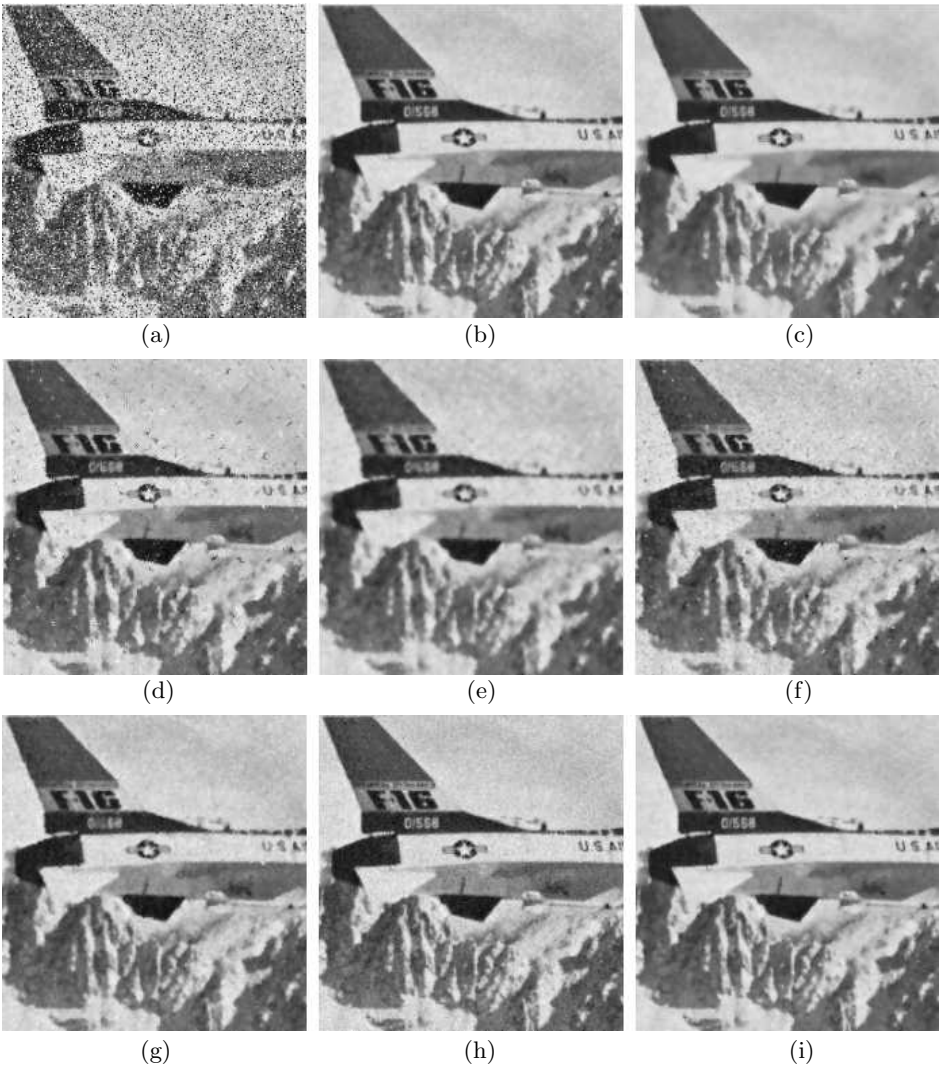


Fig. 2. The corrupted image (a) "Airplane" with 20% impulses and Gaussian noise of σ= 20; (b) FK-NN with Wiener filter; (c) FK-NN with Wiener filter; (d) FWLS [5]; (e) MFF [6]; (f) FLF [7]; (g) PBM [1]; (h) FK-NN; and (i) Two-Stage method

C. Real-World Mixed Noise Case: We have tested and proved good of our Gaussian noise standard deviation estimation routine by several typical images, each contaminated with various Gaussian noises including σ=10, 20, and 30. To demonstrate the effectiveness of the proposed algorithm in the real-world situation, Fig. 2 shows one of resulting images for brevity. Table 3 summarizes the filtered results by various post-processing schemes, for four images corrupted with unknown impulse noise rates and Gaussian levels as the case of practical

Table 2. Comparative Results of PSNR in the Cases of Corruption by Mixed Gaussian ($\sigma = 20$ known) and Impulse Noise ($p = 30\%$)

	FK-NN +Win 3	FK-NN +Win 5	FWLS [5]	MFF [6]	FLF [7]	PBM [1]	FK-NN [11]	FK-NN +MFRB
Boats	27.83^{3}	27.92^{2}	20.87	24.99	21.52	25.46	23.08	28.96^{1}
Bridge	24.51^{2}	23.76^{3}	17.86	22.44	20.23	22.83	22.06	24.29^{1}
Goldhill	27.47^{3}	27.64^{2}	22.02	26.89	23.88	20.89	29.61	28.11^{1}
Lena	28.32^{3}	28.93^{2}	20.08	25.66	21.97	25.97	23.13	29.76^{1}

Table 3. Comparative Results of PSNR in the Cases of Random Mixed Noise

	FK-NN +Win 3	FK-NN +Win 5	FWLS [5]	MFF [6]	FLF [7]	PBM [1]	FK-NN [11]	FK-NN+Universal MFRB
Airplane	27.83^{3}	27.92^{2}	23.74	27.74	24.95	27.78	26.29	28.96^{1}
Pepper	24.51	23.76	23.49	26.75^{2}	24.55	25.38^{3}	20.13	27.36^{1}
Lena	27.47^{3}	27.64^{2}	22.02	26.89	22.88	21.09	29.61	28.11^{1}
House	28.32^{3}	28.93^{2}	20.67	23.35	21.05	20.32	22.03	29.76^{1}

"Airplane" corrupted with Gaussian ($\sigma = 13$) and Impulse Noise ($p = 20\%$)
"Peppers" corrupted with Gaussian ($\sigma = 27$) and Impulse Noise ($p = 15\%$)
"Lena" corrupted with Gaussian ($\sigma = 8$) and Impulse Noise ($p = 30\%$)
"House" corrupted with Gaussian ($\sigma = 33$) and Impulse Noise ($p = 25\%$)

application. Our two-stage universal MFRB with Gaussian noise intensity estimation routine still performs the best and obtains pleased resulting images. From these figures and table, we can see that the proposed two-stage universal MFRB filter approach provides a practicable way to remove the mixed noise and produces very good restoration results, in comparison to other methods both in metric measurement and visual quality perception.

6 Conclusion

A new two-stage mixed noise removal scheme for images is proposed in this paper. In the first stage, the fuzzy K-NN filter is employed to detect and replace the impulse corrupted pixels in the image. The fuzzy K-NN filter can remove the impulse noise and preserve the image details better than the order statistic filters. In the second stage, the MFRB filter is validated to suppress the Gaussian noise and preserve the image details and structures very well. For practical application, we combine the fuzzy K-NN filter and universal MFRB filter, together with the Gaussian noise level estimation routine, to sequentially remove the mixed noise in an image. The proposed two-stage filtering scheme has demonstrated the effectiveness and robustness, in comparison with other filters in mixed noise removal of images.

Acknowledgement

This research was supported in part by the MOE under grant EX-91-E-FA06-4-4, by the MOEA under grant 94-EC-17-A-02-S1-032, and by the National Science Council under grant NSC 94-2213-E-009-097, Taiwan, R.O.C.

References

[1] Chen, T. and Wu, H. R.: Application of partition-based median type filters for suppressing noise in images. IEEE Trans. Image Processing, Vol. 10 (2001) 829-836
[2] C. Jing, Y. Jinsheng, and D. Runtao, Fuzzy weighted average filter, Proc. of ICSP 2000, (2000) 525-528
[3] A. Taguchi, A design method of fuzzy weighted median filters, Proc. third IEEE Int. Conf. Image Processing, Vol. 1 (1996) 423-426
[4] Peng, S. and Lucke, L.: Multi-level adaptive fuzzy filter for mixed noise removal. IEEE International Symposium on Circuits and Systems, Vol. 2 (1995) 1524-1527
[5] Choi. Y. and Krishnapuram, R.: A robust approach to image enhancement based on fuzzy logic. IEEE Trans. Image Processing, Vol. 6 (1997) 808-825
[6] Taguchi, A.: Removal of mixed noise by using fuzzy rules. Second International Conference Proceedings on Knowledge-Based Intelligent Electronic Systems, Vol. 1, (1998) 176-179
[7] Farbiz, F., Menhaj, M. B., Motamedi, S. A., and Hagan, M. T.: A new fuzzy logic filter for image enhancement. IEEE Trans. Syst. Man, and Cybern. Part B, Vol. 30 (2000) 110-119
[8] Russo, F.: Noise removal from image data using recursive neurofuzzy filters. IEEE Trans on Instrumentation and Measurement, Vol. 49 (2000) 307-314
[9] Chang, J. Y., Chen, J. L.: Classifier-augmented median filters for image restoration. IEEE Trans. on Instrumentation and Measurement, Vol. 53 (2004)
[10] Arakawa, K.: Fuzzy rule-based signal processing and its application to image restoration. IEEE Journal on Selected Areas in Communications, Vol. 12 (1994) 1495-1502
[11] Chang, J. Y., Lu. S. M.: Image blocking artifact suppression by the modified fuzzy rule based filter. International Journal of Fuzzy Systems, Vol. 6 (2004) 81-89

Determination of the Optimal Seam-Lines in Image Mosaicking with the Dynamic Programming (DP) on the Converted Cost Space

Jaechoon Chon[1] and Hyongsuk Kim[2]

[1] Center for Spatial Information Science at the University of Tokyo, Japan
jcchon@iis.u-tokyo.ac.jp
[2] Division of Electronics and Information Engineering, Chonbuk National University, Republic of Korea
hskim@chonbuk.ac.kr

Abstract. A novel algorithm to determine the optimal seam-line in image mosacking is proposed to establish the best borderline among neighboring images. The algorithm is based on the utilization of the dynamic programming algorithm on the converted cost field. The converting function is created adaptively based on the distribution of the pixel difference values. The path found with this algorithm runs along the smaller cost regions regardless its length.

1 Introduction

Image mosaicking is a technique of constructing a large image by the combination of smaller image patches and is often used for large and expanded high-resolution satellite images, realistic 3D image modeling, continuous image reconstruction in virtual reality, and video compression. Misalignment or different characteristics between two neighboring images will cause discontinuity between images at the image border area. The most important requirement in the mosaicking process is reducing such discontinuity. The first step of the mosaicking process is with the alignment of two neighboring images as precisely as possible. The next step is with the determination of the least noticeable border line on the overlapped area. Rather straightforward algorithms for this task are the histogram matching and the image blending. Typical use of the histogram matching algorithm involves making the histograms of two images to be as close as possible using lookup tables [1]. The blending algorithm reduces the difference of two neighboring images using weights as a function of distance in the overlapping [2],[3].

Martin et al. proposed a method called the "snake technique" to determine an optimal seam-line [4]. Their algorithm starts with a line, called a "snake", on the overlapped area of two neighboring images. The sum of the mismatching values on the line is considered to be energy, and the curve line with the smaller energy is determined to be the optimal seam-line. One problem with this technique is that

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 750–757, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the line of the "snake" gets easily stuck to local minima. An improved algorithm called the "twin snake" algorithm has also been proposed. The two lines of "twin snake" starting from opposite borders on the overlapped area evolve, while they are forced to be attracted to each other. The optimal seam-line is determined if two lines are merged into one. However, this algorithm cannot overcome the local minima problem completely, and it requires a high computation load. In this paper, a novel algorithm that employs Dynamic Programming (DP) on the converted cost space is proposed. The optimal seam-line is found by means of the optimality power of the dynamic programming.

2 Similarity of the Optimal Seam-Line with Water Way

The proposed seam-line determination is based on the minimum cost-path finding capability of the DP [5]. One problem associated with the DP-based approach is that the minimum cost path is likely to be the shorter path, because the cost increases as the path is longer. However, the human visual system is more sensitive to the higher pixel difference, regardless of the length of the seam-line. The proposed optimal seam-line algorithm is with the utilization of the DP after the cost conversion.

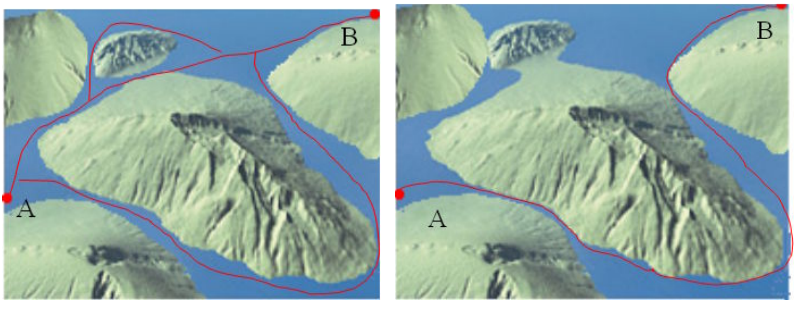
Let the pixel value difference between two images on the overlapped region be the cost. Since such cost is a quantity for each pixel point, the cost field can be expressed in 3D. Using the similarity of the 3D cost field with the terrain structure, the proposed seamline algorithm can be explained with the water way appeared on the 3D terrain structure when water is filling. Imagine that water is filled on the terrain structure and its level is high as in the Fig. 1(a), where many possible ways between two points A and B through water appear. If the water level is lowered slowly, water paths are thinned and the paths formed along the lower places of the cost field are obtained. The paths satisfy the global requirement for the optimal seam-line; The paths run along the smaller costs regardless the length of the path. One problem is that such water connections are often composed of areas instead of lines. Therefore, the local algorithm to obtain finer seam-line is also required.

2.1 The Simulated Water Level

To simulate the water level, the *thresholdedcost* concept is introduced. The thresholded cost space is the simplified cost field on which the cost values are replaced with 0 for the values below a given threshold value as shown in (1).

$$tcost(i,j) = \begin{cases} cost(i,j) & : \quad if cost(i,j) \geq threshold \\ 0 & : \quad otherwis \end{cases} \tag{1}$$

The water level is equivalent to the threshold level in this expression. If the threshold is high, larger area which has lower cost than the threshold is replaced with 0. In this arrangement, the paths filled with 0 cost are corresponding to the water ways. Finding the water ways at some water level is equivalent to the finding paths of 0 cost.



(a) (b)

Fig. 1. The proposed seam-line algorithm can be explained with the water way on the terrain structure. (a) When the water level is high, many possible paths between two points A and B appear. (b) With the lowered water level, the finer water ways which satisfies the requirements of the optimal seam-line are obtained. The path runs along smaller cost regions regardless of its length.

2.2 Adaptive Cost Conversion

For easiness of explanation, observe the 1D cost function along a path as in Fig.1 where different kinds of slopes at different locations are shown. It is easy to know that the total length of path between A and B increases more at the slower slope than steeper one as in the figure when the threshold is lowered by $\triangle$c. So does the sum of costs. Let the increment rate of the length of the path about $\triangle$c be $\triangle l/\triangle c$. Then, the spatial slope which is the slope along x axis is

$$\frac{\triangle c}{\triangle x} = \frac{\triangle c}{\sqrt{\triangle l^2 - \triangle c^2}} = \frac{\triangle c}{\sqrt{(\triangle l/\triangle c)^2 - 1}} \tag{2}$$

where $\triangle l/\triangle c$ can be obtained by the $DPwiththethresholdedcost$ which is discussed below. The $\triangle c/\triangle x$ is the spatial slope at the cost c. Therefore, it is proportional to the $visualdiscomfort$ on the seamline since it is the pixel value difference. Let such visual discomfort at the cost c be $d(c)$.

$$d(c) \cong k/|\frac{\triangle l}{\triangle x}| \tag{3}$$

Finally, cost converting function, $\Psi(c)$, is obtain via the integration of such $visualdiscomfort$ about the cost axis c as in (5).

$$\Psi(c) = k\int_0^c 1/|\frac{\triangle l}{\triangle c}|dz \tag{4}$$

$Computationof|\frac{\triangle l}{\triangle c}|$: Let $\triangle l$ be the change of the sum of path-cost obtained when the threshold is changed by $\triangle c$. The sum of the path-cost at each threshold level is obtained utilizing DP on the thresholded cost field. We call such DP

operation the $preliminaryDP$ which is different from the $finalDP$ operation which will be performed on the converted cost field.

Let the sum of path-cost obtained with the preliminary DP at the threshold th be $\phi(the)$ and the DP operation be O_{Dp} . Then, the path length obtained with the DP at the threshold th is

$$\phi(th) = O_{DP}(tcost(th)) \tag{5}$$

Since $\triangle l$ is the difference of the sum of the path-cost at the different thresholds, $|\frac{\triangle l}{\triangle c}|$ is computed as

$$|\frac{\triangle l}{\triangle c}| = \phi(th) - \phi(th - \triangle c) = O_{DP}(tcost(th)) - O_{DP}(tcost(th - \triangle c)) \tag{6}$$

In the example of Fig. 2, the sum of the path-cost, $\phi(c)$, which is obtained from the cost function of Fig. 2(a) is as shown in Fig. 2(b). Note that the horizontal axis is the cost differently from the Fig. 2(a). $|\frac{\triangle l}{\triangle c}|$ which is obtained by differentiating Fig. 2 (b) is shown with the function of c as in Fig. 2 (c). Also, the visual discomfort function d as the function of c is illustrated in the Fig. 2(d). In the figure, d is small if $|\frac{\triangle l}{\triangle c}|$ is big and vice versa. Fig. 2 (e) is the cost converting function obtained with the integration of Fig. 2(d). As seen in the illustration of Fig. 2, costs are converted adaptively according to the shape of the cost function along the path. After the cost conversion, the seamline is determined by the final DP operation on the converted cost field.

3 Visual Discordance of the Seam-Line

The Visual Discordance (VD) is also proposed as a measure of the discontinuity on the seam-lines. Let the array of the original costs (before the cost conversion) on the path be **COST**, the function to be sorted in descending order be $SORT_{DEC}$, and its results be . Then,

$$\mathbf{COST} = SORT_{DEC}\,(\mathbf{COST}) \tag{7}$$

The measure of visual discordance (VD) is defined as the sum of the predetermined number of biggest costs. Therefore, the VD is

$$VD = \sum_{k=1}^{FN} cost(k) \tag{8}$$

where FN is a fixed, predetermined number.

4 Experiments

The experiment of the seam-line determination has been done on an Ortho-Satellite image, as shown in Fig. 3. The lower part of the left image in Fig. 3(a) is overlapped with the left part of the right image in Fig. 3(b). Mosaicked images

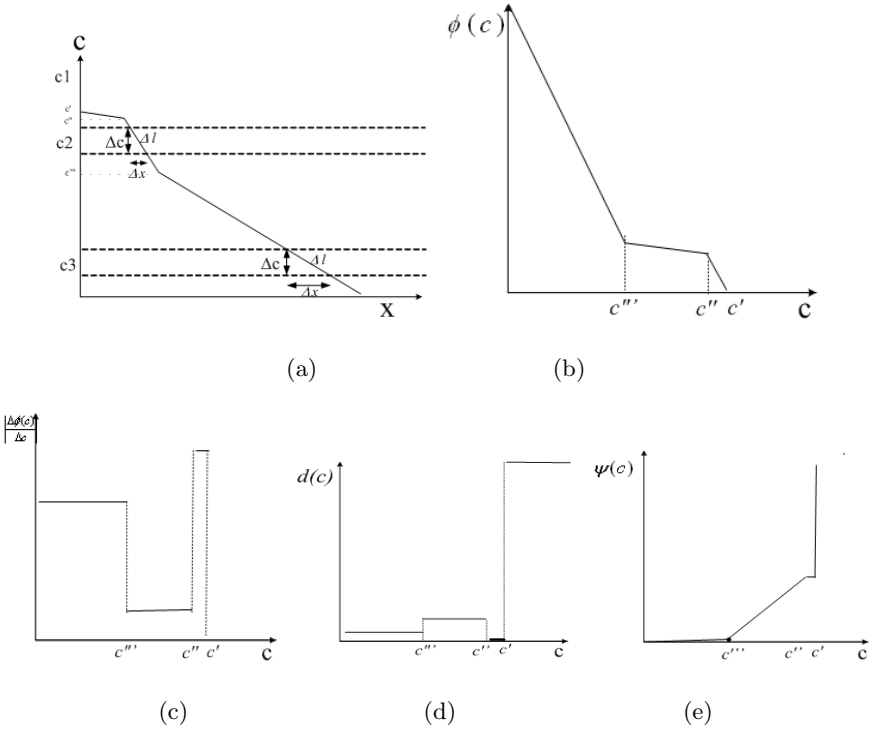
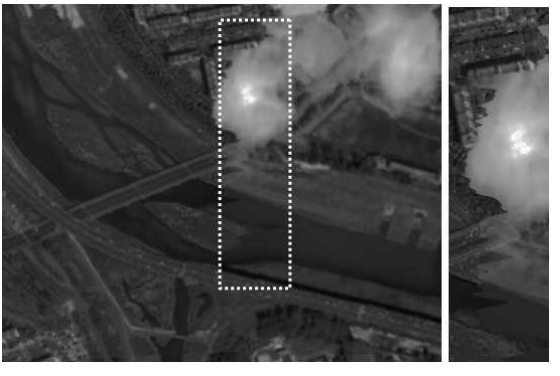


Fig. 2. An illustration of the construction procedure of the proposed adaptive cost converting function (a)1D cost function along the path, (b) sum of the path-cost function about c, (c) $|\frac{\triangle l}{\triangle c}|$, (d) the $visualdiscomfortfunctiond$ which is the inverse of (c), (e) the cost converting function which is obtained with the integration of (c)

(a)Left image (b)Right image

Fig. 3. Two images to be mosaicked (Ortho-Satellite images)

(a)



(b)

(c)

Fig. 4. Mosaicked images (a) with the original dynamic program algorithm, (b) with the twin snake algorithm, and (c) with the proposed algorithm. Significantly strong visual discontinuities appear in the image area with the original dynamic programming algorithm and with the twin snake algorithm, while no visual discontinuities are found with the proposed algorithm.

Table 1. Comparison of VD

Algorithm	*ProposedAlgorithm*	*OriginalDP*	*TwinSnake*
VD	5.6	12.2	16.47

with the aforementioned seam-lines are shown in Figure 4 where significant visual discontinuities appear in the dotted box area of images constructed with the original dynamic programming and with the twin snake algorithm. However, no visual discontinuity can be seen on the image constructed with the proposed algorithm.

A quantitative comparison has been done among the proposed, the original DP and the twin snake algorithms with the proposed Visual Discordance (VD) measure, as shown in table 1. The number of costs to be summed in computing the VD is 30. As seen in the table, the VD of the proposed algorithm is much smaller than that of either the original dynamic programming or the twin snake algorithms. The experimental results show that the proposed algorithm produces visually and quantitatively superior results, compared to the original dynamic programming algorithm and the twin snake algorithm.

5 Conclusion

We have proposed a cost-converting method that allows the dynamic programming to be applied directly to optimal seam-line determination. This cost-conversion function is obtained through the test of the thresholded cost field. The adequacy of the proposed algorithm has been explained analytically. Also, a figure of merit, which is the summation of a fixed number of biggest pixel differences (cost), is suggested as an evaluative measure of seam-lines.

The performance of the proposed algorithm has been tested, both quantitatively and visually, using various kinds of images that have been compared with those constructed with the original dynamic programming algorithm and the twin snake algorithm. The experimental results show that the proposed algorithm always produces visually and quantitatively superior results, compared to the original dynamic programming algorithm and the twin snake algorithm.

Acknowledgment

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment)" (IITA-2005-C1090-0502-0023).

References

1. Yong Du, Josef Cihlar, Jean Beaubien, and Rasim Latifovic: Radiometric Normalization, Compositing, and Quality Control for Satellite High Resolution Image Mosaics over Large Areas. IEEE Trans. on Geoscience and Remote Sensing,**39(3)** (2001) 623–634

2. Uyttendaele, M., Eden, A., Szeliski, R.:Eliminating ghosting and exposure artifacts in image mosaics. CVPR. II,(2001) 1144-1154.
3. Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J., M., O.J.: Pyramid method in image processing. RCA Engineer **29(5)** (1984) 33–41
4. Martin Kerschner:Seamline detection in colour orthoimage mosaicking by use of twin snakes. ISPRS Journal of Photogrammetry and Remote Sensing, **56** (2001) 53–64.
5. Bellman,R.:Dynamic Programming. Princeton, NJ: Princeton Univ. Press, (1957)

Symmetry-Based Salient Points Detection in Face Images

Michał Choraś and Tomasz Andrysiak

Image Processing Group, Institute of Telecommunications
University of Technology & Agriculture
S. Kaliskiego 7, 85-796 Bydgoszcz
{chorasm, andrys}@atr.bydgoszcz.pl

Abstract. In the article we propose the automatic method of symmetry-based salient points detection in face images. The proposed method is based on the modified Discrete Symmetry Transform. Firstly, multiresolution Gabor Wavelets filtration is performed. Secondly, DST is used in order to detect symmetry-based salient points in face images. Then, having automatically extracted a number of salient points, further processing including feature extraction for various applications is performed. Finally, on the basis of the extracted feature vectors, face recognition in a biometrics system can be performed.

1 Introduction and Motivation

Saliency definition and detection in images is a crucial issue in computer vision. There are many methods of salient (fiducial) points extraction in images. The basis for those methods depend on the class of considered images and applications. Symmetry-based salient points extraction is especially interesting and valid for face images. Even though faces are not ideally symmetrical and even the biometrics of asymmetrical face has been recently developed [9][10], the points of the highest symmetry are appropriate to define saliency within face images.

There are many applications, in which detection and extraction of salient points is needed for further processing including feature extraction and face recognition. The most important areas involved in implementing good solutions for that problem are: face biometrics, interpretation of face expression, human-computer interaction, face coding and face tracking.

Despite the huge interest and rapid growth in the mentioned fields of computer vision, nowadays still in many proposed methods and systems, extraction of the salient points is performed manually [15]. In contrast to such methods we propose to search for those points automatically on the basis of symmetry-based saliency definition and detection. Such approach is not well-researched yet, even though there have been some recent developments in this field [4][7].
The contribution of this paper is a novel approach to saliency definition and extraction in face images on the basis of Gabor filtration and the modified DST.

Hereby, we modify the Discrete Symmetry Transform proposed by DiGesu and Valenti [5][6].

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 758–767, 2006.
© Springer-Verlag Berlin Heidelberg 2006

We implement Gabor Wavelets in order to perform multiresolution face image filtration. Then we detect symmetry-based salient points within the image using the modified DST calculated on the basis of the filtered images. Moreover, Gabor filter responses calculated in the salient points may be later used as features representing the face images [1].

In section 2 face image filtration and the method of high symmetry points extraction are presented in detail. Moreover the algorithm of crucial symmetry points selection is described. In section 3 the calculated features are presented and application to face recognition is discussed. Conclusion and references are given next.

2 Symmetry-Based Salient Points Detection

We use Discrete Symmetry Transform for choosing the points with the highest symmetry value within the face image. In contrast to a proposed method of DST [5][6], we apply DST onto combined image of Gabor directional filtration images.

2.1 Image Filtration

The Gabor Wavelets are used for image analysis because of their biological relevance and computational properties. The Gabor filter kernels model similar shapes as the receptive field of simple cells in the primary visual cortex. Those are multi-scale and multi-orientation kernels and each kernel is a product of a Gaussian envelope and a complex plane wave. We use Gabor Wavelets to extract the facial features as the set of filter responses with determined scale and orientation values.

The responses image of the Gabor filter can be written as a convolution of the input image $I(\boldsymbol{x})$, with the Gabor kernel $\psi_{\mu,\nu}(\boldsymbol{x})$, such as:

$$R_{\mu,v}(\boldsymbol{x}) = I(\boldsymbol{x}_o) * \psi_{\mu,\nu}(\boldsymbol{x} - \boldsymbol{x}_o), \tag{1}$$

where vector coordinates $\boldsymbol{x}$ of the image $I(\boldsymbol{x})$ are equal to $\boldsymbol{x} = (x, y)$ and $*$ denotes the convolution operator.

The filtration result is a complex function consisting of the following two components:

$$R_{\mu,\nu}(\boldsymbol{x}) = R^e_{\mu,\nu}(\boldsymbol{x}) + iR^o_{\mu,\nu}(\boldsymbol{x}). \tag{2}$$

The even component is given by:

$$R^e_{\mu,\nu}(\boldsymbol{x}) = I(\boldsymbol{x}_0) * \psi^e_{\mu,\nu}(\boldsymbol{x} - \boldsymbol{x}_0) \tag{3}$$

and the odd component is:

$$R^o_{\mu,\nu}(\boldsymbol{x}) = I(\boldsymbol{x}_0) * \psi^o_{\mu,\nu}(\boldsymbol{x} - \boldsymbol{x}_0). \tag{4}$$

Some examples of Gabor kernels are presented in Figure 1.

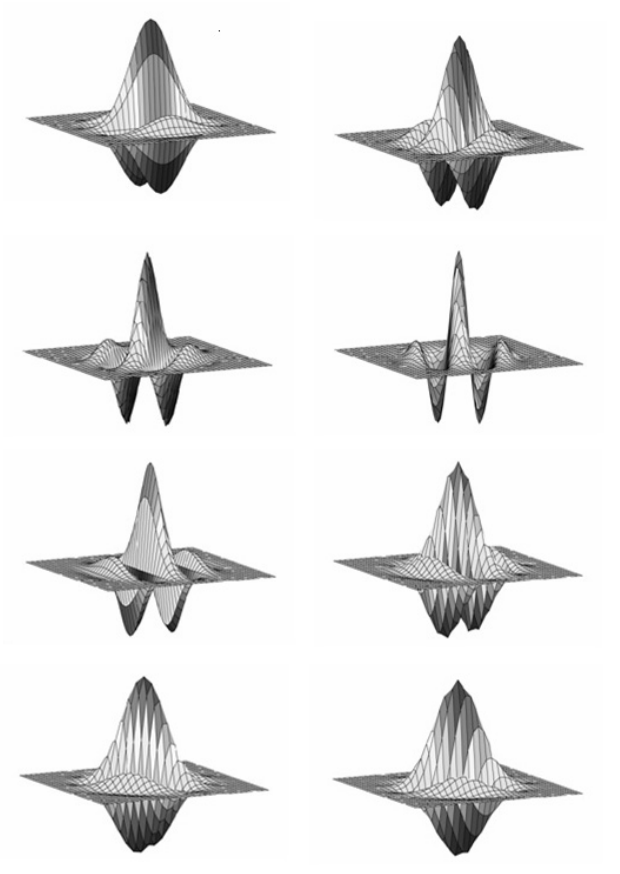


Fig. 1. Imaginary parts of Gabor kernels for different orientations $\mu = 0, 1, ..., 7$

The Gabor filters $\psi_{\mu,\nu}$ (kernels) can be formulated as:

$$\psi_{\mu,\nu}(\boldsymbol{x}) = \frac{\boldsymbol{k}_{\mu,\nu}^2}{\sigma^2} \exp\left(\frac{\boldsymbol{k}_{\mu,\nu}^2 \boldsymbol{x}^2}{2\sigma^2}\right) \left[\exp\left(i\boldsymbol{k}_{\mu,\nu}\boldsymbol{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right]. \tag{5}$$

It is a complex filter combined of the real (even) and imaginary (odd) parts, such as:

$$\psi_{\mu,\nu}^{e}(\boldsymbol{x}) = \frac{\boldsymbol{k}_{\mu,\nu}^2}{\sigma^2} \exp\left(\frac{\boldsymbol{k}_{\mu,\nu}^2 \boldsymbol{x}^2}{2\sigma^2}\right) \left[\cos\left(\boldsymbol{k}_{\mu,\nu}\boldsymbol{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right], \tag{6}$$

$$\psi_{\mu,\nu}^{o}(\boldsymbol{x}) = \frac{\boldsymbol{k}_{\mu,\nu}^2}{\sigma^2} \exp\left(\frac{\boldsymbol{k}_{\mu,\nu}^2 \boldsymbol{x}^2}{2\sigma^2}\right) \left[\sin\left(\boldsymbol{k}_{\mu,\nu}\boldsymbol{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right]. \tag{7}$$

The parameters μ and ν define the orientation and scale of the Gabor kernels and $\sigma = 2\pi$. The wave vector $\boldsymbol{k}_{\mu,\nu}$ is defined as follows:

$$\boldsymbol{k}_{\mu,\nu} = \begin{pmatrix} k_{x_{\mu,\nu}} \\ k_{y_{\mu,\nu}} \end{pmatrix} = \begin{pmatrix} k_\nu \cos\phi_\mu \\ k_\nu \sin\phi_\mu \end{pmatrix}, \tag{8}$$

where: $k_\nu = 2^{-\frac{\nu+2}{2}}\pi$ and $\phi_\mu = \frac{\pi}{8}\mu$ (where k_ν is the spacing factor between kernels in the frequency domain).

In most cases Gabor wavelets are used at five different scales and eight orientations [15]. Sometimes other configurations e.g. six orientations are also deployed [16]. Hereby, we use eight orientations and three scales as presented in Figure 2 (3 scales $\nu \in \{0, 1, 2\}$ are sufficient for reliable face representation; adding 2 more resolutions $\nu \in \{3, 4\}$ does not improve the method because the filter responses are small and not distinctive within various images).

In general, $\psi_{\mu,\nu}(\boldsymbol{x})$ is complex, however, in our approach, only the magnitudes are used since they vary slowly with the position while the phases are very sensitive.

Fig. 2. Filtration outputs of example image "FaDab157" [3] (8 orientations and 3 resolutions)

2.2 Extraction of the High Symmetry Points

The face salient points extraction is based on the Discrete Symmetry Transform as presented in [5]. Hereby we modify the known method of symmetry points detection by applying Gabor filtered images in the first step of the Discrete Symmetry Transformation.

Our application to detect points of symmetry is developed in two steps. The first step is the Gabor Wavelet filtration for proper values of orientations and resolution. Secondly for each point of the gradient image we compute the symmetry value.

The algorithm to compute the modified Discrete Symmetry Transform is following:

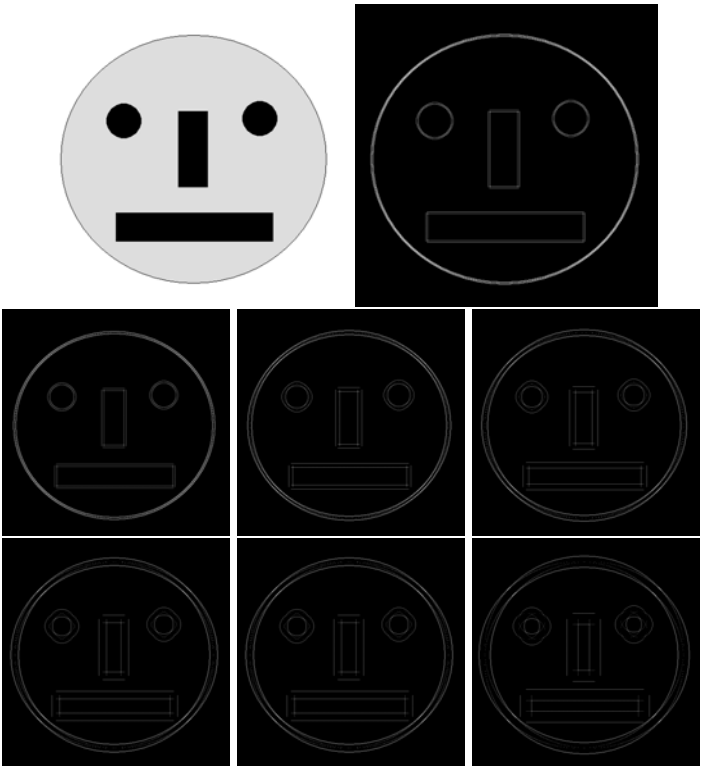


Fig. 3. Results of Discrete Symmetry Transform shown on the "ideal face image" for different radius $r = 1, 2, 4, 6, 8, 10, 12$ and $n = 2$, respectively

1. First we filter the image with the Gabor filters for all the orientations. Then we sum all these images and in result of such filtering we obtain the combined image $O(\boldsymbol{x})$, such as:

$$O\left(\boldsymbol{x}\right) = 1 - \sum_{\mu,v} R_{\mu,v}\left(\boldsymbol{x}\right). \tag{9}$$

   The image $O(\boldsymbol{x})$ is presented in the Figure 4 (left).
2. Computation of the DST. The Discrete Symmetry Transform is computed as the multiplication of the filter response image $O(\boldsymbol{x})$ with the image $M(\boldsymbol{x})$ (Figure 4 (right)) such as:

$$DST\left(I\left(\boldsymbol{x}\right)\right) = O\left(\boldsymbol{x}\right) \times M\left(\boldsymbol{x}\right), \tag{10}$$

   where:

$$M\left(\boldsymbol{x}\right) = \sqrt{\frac{1}{n}\sum_{k=0}^{n-1}\left(M_k(\boldsymbol{x}\right)^2 - \frac{1}{n^2}\left(\sum_{k=0}^{n-1}\left(M_k(\boldsymbol{x}\right)\right)^2}, \tag{11}$$

Fig. 4. Image $O(\boldsymbol{x})$ and the image $M(\boldsymbol{x})$

and:

$$M_k(\boldsymbol{x}) = \sum_{(p,q)\in \Pi_r} \left| (p-d)\sin\left(\frac{k\pi}{n}\right) - (q-e)\cos\left(\frac{k\pi}{n}\right) \right| \times I(\boldsymbol{x}), \qquad (12)$$

where:

- (p, q) are the coefficients of each calculated symmetry point,
- (d, e) are the coefficients of the point belonging to the circle Π_r with the distance r from the point (p, q),
- Π_r is the circle centred in (p, q),
- r limits the size of the neighbourhood of each point (p, q),
- n is the number of axial moments with the slope $k\pi/r$ with $k = 0, 1, ..., n$.

The final result of the modified DST computation is presented in Figure 5.

Fig. 5. Final image $DST(I(\boldsymbol{x}))$

2.3 Symmetry Points Selection

The computed $DST(I(\boldsymbol{x}))$ gives the number of extracted symmetry points, but not all the symmetry points become our salient points.

Fig. 6. Original test image "Fadab127" from $FaDab$ Database [3] and the corresponding resulting image $ThreshDST(I(\boldsymbol{x}))$ with the extracted salient points

Fig. 7. Original image "Fadab044" from $FaDab$ Database [3] and the corresponding resulting image $ThreshDST(I(\boldsymbol{x}))$ with the extracted salient points

Fig. 8. Original image from IMM database [14] and the corresponding resulting image $ThreshDST(I(\boldsymbol{x}))$ with the extracted salient points

In order to find the most significant of those points we perform the threshold operation according to the following rule [5]:

$$ThreshDST\left(I\left(\boldsymbol{x}\right)\right) = \begin{cases} 1 \; if \; DST\left(I\left(\boldsymbol{x}\right)\right) > mean + 3 * \mathrm{var} \\ 0 \; otherwise \end{cases} \tag{13}$$

where *mean* and *var* are the mean value and the standard deviation of $DST(I(\boldsymbol{x}))$, respectively.

The resulting image $ThreshDST(I(\boldsymbol{x}))$ for the test image "Fadab127" is presented in Figure 6.

Usually we obtain 30 - 50 points of the highest symmetry value. Then in those points we calculate face features based on Gabor Wavelets responses as presented in the next section.

More results of our method for images from FaDab and IMM databases are shown in Figures 7 and 8.

3 Feature Extraction and Application to Face Recognition

The filter responses that result from the application of filter bank of Gabor filters can be used directly as texture features. Three different preferred spatial frequencies and eight different preferred orientations were used, resulting in a bank of 24 Gabor filters. Such response-vectors corresponding to face salient points are often called Gabor Jets [15].

Moreover, other features such as Gabor energy, local frequency information, gradient properties and complex moments are computed in order to enhance the effectiveness of the known *EBGM* (Elastic Bunch Graph Matching) face recognition algorithm based on the Gabor Jets.

Gabor Energy is given by:

$$\mathrm{E}_{\mu,\nu}\left(\boldsymbol{x}\right) = \sqrt{\left(R^{o}_{\mu,\nu}\left(\boldsymbol{x}\right)\right)^{2} + \left(R^{e}_{\mu,\nu}\left(\boldsymbol{x}\right)\right)^{2}}. \tag{14}$$

where $R^{o}_{\mu,\nu}\left(\boldsymbol{x}\right)$ and $R^{e}_{\mu,\nu}\left(\boldsymbol{x}\right)$ are the responses of the linear even and odd Gabor filters, respectively.

Gabor filter response phase is calculated as:

$$\Phi_{\mu,\nu}\left(\boldsymbol{x}\right) = \arctan\left(\frac{R^{o}_{\mu,\nu}\left(\boldsymbol{x}\right)}{R^{e}_{\mu,\nu}\left(\boldsymbol{x}\right)}\right). \tag{15}$$

Local frequency information can be extracted from $\Phi_{\mu,\nu}$ as follows:

$$\Gamma_{\mu,\nu}\left(\boldsymbol{x}\right) = \arctan\left(\frac{\nabla_{y}\left(\Phi_{\mu,\nu}\left(\boldsymbol{x}\right)\right)}{\nabla_{x}\left(\Phi_{\mu,\nu}\left(\boldsymbol{x}\right)\right)}\right), \tag{16}$$

where ∇_y and ∇_x are gradient estimation functions.

The next calculated feature is the dominant gradient angle such as:

$$\theta_{\mu,\nu}(\boldsymbol{x}) = \begin{pmatrix} \Gamma_{\mu,\nu}(\boldsymbol{x}) & for\ |\phi_\mu - \Gamma_{\mu,\nu}(\boldsymbol{x})| \leq \frac{\pi}{2} \\ \Gamma_{\mu,\nu}(\boldsymbol{x}) + \pi & for\ |\phi_\mu - \Gamma_{\mu,\nu}(\boldsymbol{x})| > \frac{\pi}{2} \end{pmatrix} \tag{17}$$

and:

$$\mathrm{P}_{\mu,\nu}(\boldsymbol{x}) = \sqrt{\nabla_x^2(\Phi_{\mu,\nu}(\boldsymbol{x})) + \nabla_y^2(\Phi_{\mu,\nu}(\boldsymbol{x}))}, \tag{18}$$

where ϕ_μ is the orientation of the Gabor filter, $\Gamma_{\mu,\nu}$ is the direction of the gradient vector, and $\mathrm{P}_{\mu,\nu}$ is the module of the gradient vector.

Furthermore we calculate the spatially localized estimate of the frequency along the direction ϕ_μ and the direction of maximal phase change rate $\theta_{\mu,\nu}$ (i.e., highest local frequency). This value is expressed as:

$$\mathrm{H}_{\mu,\nu}(\boldsymbol{x}) = \mathrm{P}_{\mu,\nu}(\boldsymbol{x}) \cdot \cos(\phi_\mu - \theta_{\mu,\nu}(\boldsymbol{x})) \tag{19}$$

Finally the complex moments of the the gradient vector $H_{\mu,\nu}(\boldsymbol{x})$ are defined as follows:

$$C_{\mu,\nu}^{(m,n)}(\boldsymbol{x}) = \int\int (u - iv)^m (u - iv)^n \mathrm{H}_{\mu,\nu}(\boldsymbol{x})\, dudv. \tag{20}$$

Those features are translation invariant inside homogeneous texture regions and give information about the presence or absence of dominant orientations in the texture. The sum $m + n$ is called the order of the complex moment; it is related to the number of dominant orientations in the texture. In our application, the nonzero real and imaginary parts of the complex moments are used as features.

4 Conclusion

In the article we presented an efficient method of image filtering in order to extract symmetry-based salient points within face images. In our work we base on directional Gabor Wavelets and the modified Discrete Symmetry Transform for the extraction of the face saliency corresponding to the points of the highest symmetry within the face image.

Feature vectors representing face images are calculated only in the extracted salient points. We propose to use Gabor-based features such as filter responses, Gabor-energy and the complex moments. Furthermore local frequency information, dominant gradient angle and gradient vector are used as features.

Then the calculated feature vectors can be stored and compared in various face recognition systems. Some possible applications include face image retrieval from face databases, facial expression understanding and passive biometrics based on acquired face images.

References

1. Andrysiak T., Choraś M., *LPT and DST for Face Feature Extraction and Representation*, Proc. of Pattern Recognition and Information Processing PRIP, Minsk, 148-151, 2005.

2. Andrysiak T., Choraś M., *Face Feature Extraction and Representation,* Proc. of ACS-CISIM, vol. 1: Image Analysis, Computer Graphics, Security Systems and Artificial Intelligence Applications, 297-306, University of Finance and Management in Bialystok, 2005.
3. Bobulski J., *Face Identification Method Based on HMM, PhD Thesis (in polish)*, Czestochowa University of Technology, 2004.
4. Capodiferro L., Laurenti A., Rava P., Jacovitti G., *Salient Points Detection of Human Faces by a Circular Symmetry Index Based on Fisher's Information*, Proc. of COST 275 Workshop: Biometrics on the Internet, 15-18, Hatfield, UK, 2005.
5. Gesu Di V., Valenti C., *The Discrete Symmetry Transform in Computer Vision*, Technical Report DMA-011 95, DMA Univ. Palermo, 1995.
6. Gesu Di V., Valenti C., *Symmetry Operators in Computer Vision*, Proc. CCMA Workshop on Vision Modeling and Information Coding, Nice, 1995.
7. Gonzalez-Jimenez D., Alba-Castro J.L., Argones-Rua E., Kittler J., *A Two-stage Approach to the Discriminative Point Selection in Face Images*, Proc. of COST 275 Workshop: Biometrics on the Internet, 19-22, Hatfield, UK, 2005.
8. Howell A.J., *Introduction to Face Recognition*, in *L.C. Jain et al. (Eds): Intelligent Biometric Techniques in Fingerprint and Face Recognition,* 217-283, CRC Press 1999.
9. Kompanets L., Kubanek M., Rydzek S., *Czestochowa-Faces and Biometrics of Asymmetrical Face*, in: L. Rutkowski et. al. (Eds.): Artificial Intelligence and Soft Computing - ICAISC 2004, LNAI 3070, 742-747, Springer, 2004.
10. Kompanets L., *Giving Birth to Biometrics of Asymmetrical Face*, Computing, Multimedia and Intelligent Techniques, vol. 1, no. 1, 19-35, 2005.
11. Liu C., Wechsler H., *A Gabor Feature Classifier for Face Recognition*, Proc. of IEEE Intl. Conf. on Computer Vision, Vancouver, Canada, 270-275, 2001.
12. Lee T., Ranganath S., Sanei S., *An Analytical Overview of New Trends in Face Recognition*, Proc. of IASTED Intl. Conf. on Signal Processing, Pattern Recognition and Applications, Greece, 202-206, 2002.
13. Reisfeld D., Yeshurun Y., *Robust Detection of Facial Features by Generalized Symmetry*, Proc. of Intl. Conf. on Pattern Recognition, 1:117-120, The Netherlands, 1992.
14. Stegmann M. B., Ersboll B. K., Larsen. R., *FAME - a flexible appearance modeling environment*, IEEE Trans. on Medical Imaging, vol. 22, no. 10, 1319-1331, 2003.
15. Wiskott L., Fellous J.M., Kruger N., Malsburg C.v.d., *Face Recognition by Elastic Bunch Graph Matching*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, 775-779, 1997.
16. Zhang Z., Lyons M., Schuster M., Akamatsu S., *Comparison Between Geometry-Based and Gabor-Wavelets-Based Facial Expression Recognition Using Multi-Layer Perceptron*, Proc. of Intl. Conf. on Automatic Face- and Gesture- Recognition, Nara, Japan, 1998.

Cellular Neural Networks and Dynamic Enhancement for Cephalometric Landmarks Detection

D. Giordano[1], R. Leonardi[2], F. Maiorana[1], and C. Spampinato[1]

[1] Dipartimento Ingegneria Informatica e delle Telecomunicazioni
University of Catania, Catania 95125, Italy,
{dgiordan, fmaioran, cspampin}@diit.unict.it
WWW home page: www.cdc.unict.it
[2] Clinica Odontoiatrica II - University of Catania,
Policlinico Cittá Universitaria, Catania 95125, Italy
rleonard@unict.it

Abstract. Cephalometric landmarks detection is a knowledge intensive activity to identify on X-rays of the skull key points to perform measurements needed for medical diagnosis and treatment. We have elsewhere proposed CNNs (Cellular Neural Networks) to achieve an accuracy in automated landmarks detection suitable for clinical practice, and have applied the method for 8 landmarks located on the bone profile. This paper proposes and evaluates a CNNs approach augmented by local image dynamic enhancemet for other 3 landmarks that are notoriously difficult to locate; the advantages of this method in the landmark detection problem are pointed out.

1 Introduction

Cephalometric landmark detection is a knowledge intensive activity to identify on standardized, lateral X-rays of the skull (cephalograms) key points to perform measurements needed for medical diagnosis, treatment planning and evaluation (for example, an orthodontic intervention or maxillo-facial surgery). A cephalometric analysis can include from 8 to 30 landmarks, and requires expert knowledge of anatomical structures. Automatic landmarks detection is needed because this is a very time consuming process (up to thirty minutes), and accuracy is affected by human measurements errors. Several automated approaches have been attempted, (e.g., [1],[2],[3]), but have been criticized because the accuracy they achieve is not suitable for standard clinical practice[4]. Measurements are considered precise when errors are within 1 mm; errors within 2 mm are considered acceptable, and are used as a reference to evaluate the recognition success rate. We have recently proposed a landmarking method based on CNNs (Cellular Neural Networks) and landmark specific knowledge-based algorithms [5]. This approach has been validated by experimental evaluation on a set of landmarks located on bone profiles. This paper extends and evaluates the use of CNNs to locate 3 points that were not included in the former analysis (Sella,

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 768–777, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Gonion, Orbitale); these landmarks are located in regions that may be partially hidden or where there is ambiguity due to overlapping bone structures, and are not in edges or bone profiles. After surveying the state of art of cephalometric landmark detection in section 2 and outlining the functioning of CNNs in section 3, the paper turns to describing the method proposed to locate Sella, Gonion and Orbitale, this latter one by the joint application of CNNs and local dynamic enhancement. Section 5 presents and discusses the experimental evaluation of the proposed method.

2 Automated Cephalometric Landmarks Identification

In this section we review the state of art in cephalometric landmarks identification with respect with the techiniques employed. A detailed comparison with respect to the accuracy and number of detected landmarks is carried out in section 5, after our results have been illustrated. The first automatic landmark detection systems [6], [7], [8] were based on filters use to reduce noise and image enhancement followed by a knowledge based approach to locate landmarks in enhanced edges. In some cases the techinque of the resolution pyramid was also used to speed up the process to find first a region of interest and then to refine the search in the selected area. A knowledge-based system, based on a blackboard architecture was proposed in [9], although it has been criticized as having the major drawback of rule rigidity. Other approaches to landmark detection are based on the use of pattern-matching techniques. Mathematical morphology was used in [1] in conjunction with a search for several anatomical structures assured to maintain an almost exact geometrical relation to the landmark, in order to improve accuracy; however, these structures can only be determined for a small number of landmarks. Grau et. Al. [10] used a technique based on edge detection and pattern-matching. The system detects a set of reference lines used to determine the search area for the landmarks. Then a template matching technique using morphological operations is used similar to the one used in [1]. Spatial spectroscopy was used in [3], but this approach suffers of false detection that can result in large error since the pattern detection steps are not supported by other confirming techniques. Other approaches have used neural networks together with genetic algorithms [11]. A model of the gray-levels around each point was generated from a training set and matched to a new image in order to locate the points of interest. Accuracy of this method was assessed in [4]. Fuzzy neural networks have been used in [12] and in [13] (in conjunction with template matching), whereas [14] have used Pulse Coupled Neural Networks. Active shape models (ASMs) were used in [2]. The technique is based on a set of deformable templates of important structures that are deformed in order to match the deformable template to the structure in a particular image. This method can be used for a first landmark location estimate since the accuracy is not suited for exact identification (mean error above 2 mm.). Similarly, in [15] a technique based on statistical pattern recognition applied to points and shapes is used. The model is defined for ordered set of points that are approximated by Bézier

curves, the Active contours with a similarity function are used. The model suffers the presence of noise. El-Feghi et al. [16] proposed a method based on Partial Least Squares Regression. This model tries to establish a relationship between independent and dependent variable by the principal component analysis and by maximizing the correlation with the dependent variable. Giordano et al. [5] recently used CNNs templates and knowledge based algorithm for landamarks identification. The approach proved general and flexible enough to deal with variability in the anatomical morphologies and achieve the needed accuracy for 8 landmarks loocated in the bone profiles. The method proposed in this paper extends this work by using CNNs and dynamic thresholding, as descrived in the next section.

3 Cellular Neural Networks

CNN is a powerful computational model equivalent to a Turing machine [17] implemented on chip known as CNN-UM (CNN Universal Machine). CNNs consist of computational units (cells) arranged in matrix forms (2D or 3D). 2D matrix arrangement are suitable to process images [18], [19]. Each cell is a dynamic unit with an input, an output and one state. Each cell is influenced by the input and the output of all the n neighboring cells. A cell in a matrix of MxN is indicated by C(i,j). The state of each cell (i, j) depends on the states of the neighbouring cells from i - n to i + n and from j - n to j + n, and on their gray level values. If a cell (i, j) depends on the cells from i - 1 to i + 1 and from j - 1 to j + 1, then the CNN consists of the 3 x 3 array shown in fig.1.

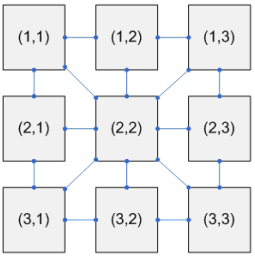


Fig. 1. 3x3 CNN

CNNs dynamics are determined by the equations (1) and (2), where x is the state, y the output, u the input, x(i,j) is the generic cell belonging to the matrix; I(i,j) is the activating treshhold for each cell, Nr (i,j) is the neighborhood radius of the interacting cells. A is known as feedback template and B is known as control template. By setting the proper initial conditions for the state variables and for the input values several processing tasks can be accomplished. Changing template parameters allow CNN programming [20].

$$\dot{x_{i,j}} = -x_{i,j} + \sum_{C(k,n)\in N_r(i,j)} A(i,j,k,h)y_{kh}(t) + \sum_{C(k,n)\in N_r(i,j)} B(i,j,k,h)u_{kh}(t) + I \tag{1}$$

$$y_{i,j} = \frac{1}{2}(|x_{i,j}+1| + |x_{i,j}-1|) \tag{2}$$

CNNs are intrinsically parallel and analogic structures and can be used as fast elaboration tools for systems with a matrix organization, as images. Moreover, they don't need training and can be easily implemented in VLSI, since interactions are localized in a limited neighborhoods.

4 Cephalometric Landmarks Identification by CNNs

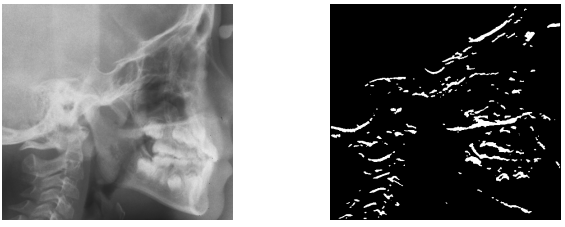
The tool developed for automated landmarking of cephalograms is based on a software simulator of a CNN of 512 x 480 cells able to map an image of 512 x 480 pixels, with 256 grey levels [5]. Each point is located by employing different algorithms, after a pre-processing stage consisting of three steps: 1) CNN processing, 2)Thresholding, 3) Noise removal by a Median filter. Each landmark needs specific CNN templates and also different thresholding values defined by the parameters T_{low} and T_{up}. All the pixel values greater than T_{up} and smaller than T_{low} will be changed to 0 (i.e., black), whereas the other pixels will be set to the max value in the pixels variation range. For some points dynamic thresholding will be used. Let's analyze the algorithms for each point.

4.1 Sella

The Sella is the depression of the superior face of sphenoid (hypophysial fossa). The landmark Sella is a constructed radiological point defined as the central point of the region Sella. In this case pre-processing involves: CNN elaboration by the templates (see 3) with 5000 cycles and 0.1 as integration step; thresholding with $T_{up} = 255$ and $T_{low} = 87$; application of a median filter.

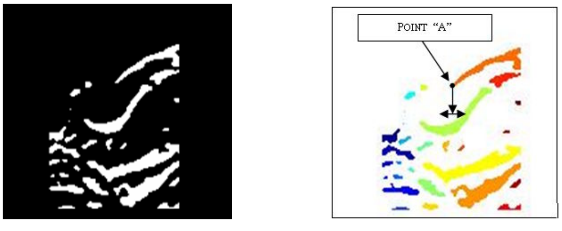
$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}; B = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 2 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}; I = 0.5; \tag{3}$$

Fig.2(a) and 2(b) illustrate the original and the output binary image. Then the approach is to search for Sella in a region of interest (ROI) that is likely to contain it, in order to reduce search time and the error probability, since the smaller the region the less the number of objects to analyze. This ROI has been empirically defined as comprised within rows 9-160 and columns 160-262 (fig. 3(a)); the comprised objects are identified by applying the connected components algorithm (fig. 3(b)). A property of this ROI is that the Sella can be identified by a search starting from the furthest right and up connected element.



(a) Original X-Ray (b) Processed Image

Fig. 2. Processing using CNNs



(a) ROI in the image (b) Connected components in ROI

Fig. 3. Connected components algorithm in the ROI defined for Sella

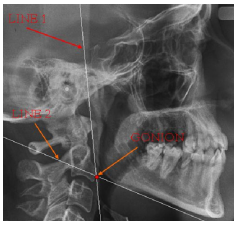
The search algorithm starts from the point A, defined as the one belonging to the furthest right and up connected element and with minimum coordinates. From point A, a new point *Camp* is derived. Its coordinates are *Camp (i) = A (i)* and *Camp (j) = A(j) + 18*. Starting from *Camp*, a 20 pixels wide horizontal segment is analyzed. When a white pixel is found, its coordinates are assigned to *Camp*. The new point might belong to Sella, this is verified by the *single evaluation* procedure, that checks if the points belongs to a connected element wider than 20 pixels; if yes, the Sella is found. The landmark is then computed as the intersection of two segments: one linking the borders of the Sella and the line that divides in a half the Sella.

4.2 Gonion

The Gonion is a constructed point defined as the intersection of the lines tangent to the more distal border of the ascending ramo and to the mandibular plane (line 1 and line 2, respectively, in fig.4(a)). The X-ray is processed as follows: first a gradient filter 5X5 in south-west direction is applied by means of the CNN templates shown in (4); then the image is binarized by thresholding with $T_{up} = 255$ and $T_{low} = 100$; and finally a median filter is applied. To compute line 1 in the output image, a ROI is chosen. In this case ROI is defined vertically by the parameters 160 and 220, and horizontally by 220 and 365. As in the previous case, the connected component algorithm is applied, but recognition is performed simply based on dimension. In particular, the longest object will

be the one being sought (i.e., the ascending ramo). The tangent line is found with good approximation by the least squares method. An analogous approach is used to compute the second line. The only difference is in the parameters that identify the ROI to carry out the search. In this case these are 365 and 452 horizontally, and 250 and 370 vertically.

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}; B = \begin{pmatrix} 1 & 0.5 & -1 & -1 & -1 \\ 0.5 & 1 & 1 & -1 & -1 \\ -1 & 1 & 5 & 1 & -1 \\ -1 & -1 & 1 & 1 & 0.5 \\ -1 & -1 & -1 & 0.5 & 1 \end{pmatrix}; I = -13; \quad (4)$$



(a) Gonion in the original image

(b) Filtered Image

Fig. 4. Processed Image for Gonion Detection

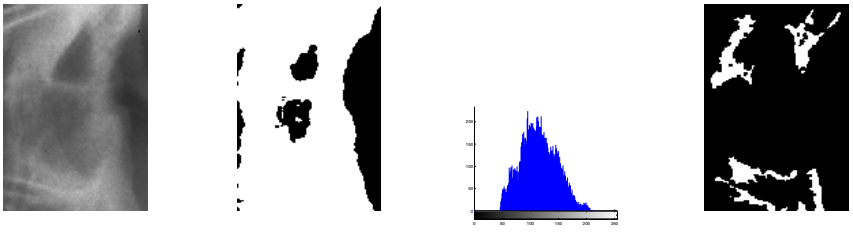
4.3 Orbitale

The Orbitale is the lowermost point of the orbit in the radiograph. One of the key difficulties in locating this landmark is that sometimes the relevant anatomical structures are hardly visible with respect to other zones; this has been solved by applying dynamic thresholding in the relevant ROI. The processing methods discussed in the previous sections are global, because that pixels are modified by a transformation function based on the gray-level content of an entire image. Although this global approach is suitable for overall image enhancement, there are cases in which it is necessary to enhance details over small areas in an image, like in the case of Orbitale. The features of the pixels in these areas may have negligible influence on the computation of a global transformation whose application does not necessarily guarantee the desired local enhancement. The applied solution is to devise transformation functions based on the gray-level distribution in the desired ROI. The detection algorithm is as follows. First, a 5X5 gradient filter in north-east direction is applied by the CNN template shown in (3). For orbitale a suitable ROI is delimited vertically by parameters 120 and 210, and horizontally by 370 and 450. Local enhancement to pinpoint the anatomical structures of interest is perfomed by dynamic thresholding, i.e., T_{up} and T_{low} are defined respectively as μ and $\mu + \sigma$ where μ is the mean of the histograms in the ROI(*Region Of Interest*), and σ is the standard deviation, defined as follows:

$$\mu = \sum_{(s,t)\in ROI} r_{s,t} \cdot p(r_{s,t}) \tag{5}$$

$$\sigma = \sum_{(s,t)\in ROI} [r_{s,t} - \mu] \cdot p(r_{s,t}) \tag{6}$$

where $r_{s,t}$ is the gray level at coordinates (s,t) in the ROI, and $p(r_{s,t})$, is the normalized histogram component corresponding to the considered value of gray level. The effect of local enhancement is shown in figure 5, where using the T_{up} and T_{low} of the Sella case it is impossible to detect some shapes for the Orbitale(see fig.5(d)), whereas using the μ and σ of the histogram, the new values of T_{up} and T_{low} are respectively 165, 115, fig.5(c), the anatomical structures for the detection are highlighted, fig.5(d).



(a) ROI for Orbitale Detection

(b) Image with Static Enhancement

(c) Histogram of the selected ROI

(d) Image with Local Enhancement

Fig. 5. Dynamic Local Enhancement

The binary image is then denoised by the median filter. Filtering of the objects in the ROI proceeds by taking into account their area. In particular, the objects with area greater than the mean area are filtered. Among these objects, those whose width is greater than average are selected, and the 2nd order derivative is computed. If this value is positive (concavity up) the object might be the lower part of the orbit. Since two objects might be present due to the overlapping of two symmetrical structure, the midpoint is selected, as in standard clinical practice.

5 Experimental Evaluation

Performance of the developed algorithms for these 3 landmarks was assessed on a sample of 19 cases at 300 dpi, by computing the Euclidean distance from the same point marked by an expert orthodontist. Points with a distance less than 2mm were considered successfully located. Table 1 reports the findings.

The overall mean success rate is 93%. Best detection performances were on Sella, with 100% recognition rate and 89% of the found landmarks within 1 mm

Table 1. Experimental results of the proposed method

Land-mark	Mean Error (mm)	Standard deviation (mm)	≤1 mm	>1mm ≤2mm	> 2mm ≤3mm	>3mm	Success Rate
Sella	0.43	0.34	89%	11%	0	0	100%
Gonion	1.36	1.86	53%	37%	5%	5%	90%
Orbitale	1.10	0.62	42%	47%	11%	0	89%

precision. We can compare our findings to the performance of other automatic landmarking systems, with respect to recognition rate (RR) and mean error (m.e.) on the same points. By *morphological processing coupled with a statistical approach* [1] the RR for Sella was 53%; for Gonion 61%; and for Orbitale 40% (40 cases). By *edge detection and pattern matching* [10], the RR for Sella and Orbital was 65% with m.e. 1.92 mm, for Gonion RR was 85% with m.e. 1.11 mm (20 cases). By image spectroscopy [3] Sella was found with m.e. 5.06±3.37 mm, and Orbitale with m.e. 2.46±3.7 mm (14 cases). By *multilayer perceptron and genetic algorithms* [4] Sella was found with m.e. 0.94±0.54mm, Orbitale with 5.28±4.1mm and Gonion with 4.53±3.13 mm (38 cases). By *Active Shape Model* [2] Sella was found with m.e. 5.5±6.8mm, Orbitale with 5.5±3.4 mm and Gonion with 5.8±6.0mm (63 cases). By s*tatistical pattern recognition* [15]Sella was found with m.e. of 1.2 mm. By *Pulse Coupled Neural Networks* [14]the Sella RR was 36% (cases 109). By *a neurofuzzy system* [12] the Sella RR was 77%, Gonion 87% and Orbitale 74% (600 cases). By *Partial Least Squares Regression* [16] Sella was detected in 83% of the cases, Gonion in 71%, and Orbitale in 72% (100 cases). For Sella our method has the lowest mean error (comparable to the performance of the multilayer perceptron [4]) and achieves 100% RR. Concerning Gonion, in the literature RR varies from 61% to 87%; our RR and m.e. are similar to the best ones [10]. For Orbitale we obtain again the same mean error as [10] but our recognition rate is 84% vs. their 65%, and superior to the best recognition rate reported in the literature, i.e. 72%. Thus our approach, for each landmark outperforms or equates the performance of the other approaches, but is the only one able to achieve these levels in all the three landmarks. Also, the mean error is smaller than those reported in the other studies, and, consistently with our findings in [5], very close to the human mean error, which is 1.26 mm [4].

6 Concluding Remarks

In this paper we have illustrated a novel approach to detect relevant landmarks in cephalometric analysis by using Cellular Neural Networks combined, for the Orbitale point, with local dynamic thresholding. A skull X-ray contains severals anatomical structures, each with a different distribution of gray levels, thus making static thresholding inadequate to properly highlight each of them. To point out these different structures and related landmarks we have used, dynamic enhancement. This method facilitates the search of the landmarks because it uses

specific statistical parameters of the histogram in the chosen ROI, prior to the application of CNNs. CNNs effects on X-rays processing could also be obtained by spatial filtering by means of a gradient filter. For example, in the case of Sella, application of a north-south gradient filter produces the image shown in fig.2(b). Using a simple gradient filter would allow a computation speed one order of magnitude greater than a CNN software simulator. However, CNNs afford peculiar advantages over other filters. One is the possibility of visualizing the intermediate processing steps. This can be convenient if an overall integrated and adaptive landmarking system is devised, since our results exploit the transient solution provided by the CNNs, and in some cases it might be convenient for the user to adjust the n. of cycles of elaboration, or the integration step. Also, the hardware implementation in a chip ACE 16K [21] allows the parallel processing of 128x128 pixel, thus lowering significantly the processing time.

References

1. Cardillo, J., Sid-Ahmed, M.A.: An image processing system for locating craniofacial landmarks. IEEE Trans. On Medical Imaging 13 (1994) 275-289.
2. Hutton, T.J., Cunningham. S., Hammond, P.: An evaluation of active shape models for the automatic identification of cephalometric landmarks. European Journal of Orthodontics, 22 (2000), 499-508.
3. Rudolph, D.J., Sinclair, P.M., Coggins, J.M.: Automatic computerized radiograohic identification of cephalometric landmarks. American Journal of Orthodontics and Dentofacial Orthopedics 113 (1998) 173-179.
4. Liu, J., Chen, Y., Cheng, K.: Accuracy of computerized automatic identification of cephalometric landmarks. American Journal of Orthodontics and Dentofacial Orthopedics 118 (2000) 535-540.
5. Giordano, D., Leonardi, R., Maiorana, F. Cristaldi, G. and Distefano, M. (2005). Automatic landmarking of cephalograms by CNNS, In S. Miksch et al. (Eds.) Proc. AIME'05 Artificial Intelligence in Medicine, LNAI 3581, 342-352.
6. Levy-Mandel, A.D., Venetsamopolus, A.N., Tsosos, J.K.: Knowledge based landmarking of cephalograms. Computers and Biomedical Research 19, (1986) 282-309.
7. Parthasaraty, S., Nugent, S.T., Gregson, P.G., Fay, D.F.: Automatic landmarking of cephalograms. Computers and Biomedical research, 22 (1989), 248-269.
8. Tong, W., Nugent, S.T., Jensen, G.M., Fay, D.F.: An algorithm for locating landmarks on dental X-Rays. 11th IEEE Int. Conf. on Engineering in Medicine & Biology (1990).
9. Davis, D.N., Taylor, C.J.: A blackboard architecture for automating cephalometric analysis. Journal of Medical Informatics, 16 (1991) 137-149.
10. Grau, V., Alcaniz, M., Juan, M. C., Monserrat, C., Knoll, C.: Automatic localization of cephalometric landmarks. Journal of Biomedical Informatics 34 (2001) 146 - 156.
11. Chen, Y., Cheng, K., Liu, J.: Improving Cephalogram analysis through feature subimage extraction. IEEE Engineering in Medicine and Biology (1999) 25-31.
12. El-Feghi, I., Sid-Ahmed, M.A., Ahmadi, M.: Automatic localization of craniofacial landmarks for assisted cephalometry. Pattern Recognition 34 (2004) 609 - 621.
13. Sanei, S., Sanei, P., Zahabsaniesi, :Cephalograms analysis applying template matching and fuzzy logic. Image and Vision Computing 18 (1999) 39-48.

14. Innes, A., Ciesilski, V., Mamutil, J., Sabu, J. : Landmark detection for cephalometric radiology images using Pulse Coupled Neural Networks. In H. Arabnia and Y.Mun (eds.) Proc. Int. Conf. on Artificial Intelligence, 2 (2002) CSREA Press.
15. Romaniuk, B., Desvignes, M., Revenu, M. Deshayes, M.-J.: Shape variability and spatial relationships modeling in statistical pattern recognition. Pattern Recognition Letters 25 (2004) 239 - 247
16. El-Feghi, I., Sid-Ahmed, M.A., Ahmadi, M.: Craniofacial landmarks extraction by partial least squares regression. Proc. Of the 2004 International symposium on Circuits and Systems, 2004. (ISCAS 2004). Vol V 45 - 48
17. Chua, L.O., Roska, T.: The CNN paradigm. IEEE TCAS, I, 40 (1993), 147-156.
18. Szabo, T., Barsi, P., Szolgay, P. Application of analogic CNN algorithms in telemedical neuroradiology. Proc. 7th IEEE International Workshop on Cellular Neural Networks and Their Applications, 2002. (CNNA 2002). 579 - 586.
19. Aizemberg, I, Aizenberg, N., Hiltner, J., Moraga, C. Meyer zu Bexten, E.: Cellular neural networks and computational intelligence in medical image processing. Image and vision computing 19 (2001) 177-183.
20. Roska, T., Kek, L., Nemes, L., Zarandy, Szolgay, P.: CSL CNN Software Library (Templates and Algorithms). 1999 Budapest, Hungary.
21. G. Liñán, R. Domnguez-Castro, S. Espejo, A. Rodrguez-Vázquez, ACE16k: A Programmable Focal Plane Vision Processor with 128 x 128 Resolution. ECCTD '01-European Conference on Circuit Theory and Design, pp.: 345-348, August 28-31,2001, Espoo, Finland.

Adaptive Critic Neural Networks for Identification of Wheeled Mobile Robot

Zenon Hendzel

Rzeszow University of Technology,
Department of Applied Mechanics and Robotics,
Powstancow Warszwy 8, 35-959 Rzeszow, Poland
zenhen@prz.edu.pl

Abstract. A new applications of adaptive critic identifier for wheeled mobile robot is presented. In this approach the architecture of adaptive critic identifier contains a neural network (NN) based adaptive critic element (ACE) generating the reinforcement signal to tune the associative search element (ASE), which is applied to approximate nonlinear functions of the mobile robot. The proposed system identification that can guarantee tracking performance and stability is derived from the Lyapunov stability theory. Computer simulation have been conducted to illustrate the performance of the proposed solution by a series of experiments on the emulator of wheeled mobile robot Pioneer-2DX.

1 Introduction

The problem of identification consists of choosing an appropriate identification model and adjusting its parameters such that the response of the model to an input signal approximates the response of the plant under the same input [3,6,7]. The synthesis of the mobile wheeled-robot state identifier is a complex problem for these objects are not linear, non-holonomic but multidimensional systems. It would be desirable to use more advanced learning and intelligent features of NN in system identification design as reinforcement learning (RL)methods. RL had been applied to acquire a control system, identification for many fields, such as robotics, image processing, etc., as learning without any teacher [8]. The approach taken in this paper involves an actor-critic learning system for RL [1,5,8]. Actor-critic architectures differ from other methods in that separate data structures are used for the control policy (the actor) and the value function (system performance) (the critic). One advantage of the actor-critic framework is that action selection requires minimal computation.[1,8]. The proposed RL systems will be capable of avoiding too many trial and error learning process and guarantee the stability of the proposed identifier. The remainder of the paper is organized as follows. Chapter 2 displays a two layer neural network. Dynamic equations of the mobile 2-wheeled-robot and identification properties are included in Chapter 3. Chapter 4 includes results of the identification algorithm tests, obtained after numerical simulation. Chapter 5 resumes the results of the research.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 778–786, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Linear-in-the-Parameter Neural Nets

A two layer NN [2,4] is depicted in figure 1, where there are two layers of neurons, with one layer having m neurons feeding into a second layer having r neurons, with linear activation functions on the output layer. The output of the two-layer NN is given by the equation

$$y = W^T S\left(V^T x\right) \tag{1}$$

If the first-layer weights V are predetermined by some a priori method, then only the second-layer weights W are considered to define the NN. Define the fixed function $\phi(x) = S\left(V^T x\right)$ so that such a one-layer NN has the equation

$$y = W^T \phi(x) \tag{2}$$

This NN is linear in the NN parameters W so that it is easier to train the NN by tuning the weights. It is shown in [4] that, selecting the matrix V in (1) randomly the resulting function $\phi(x) = S\left(V^T x\right)$ is a basis, so that the linear in the NN parameters has the universal approximation property. In this approach, can be the standard sigmoid functions.

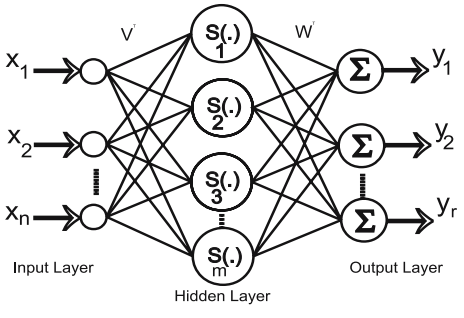


Fig. 1. Neural network scheme

3 Modeling and System Identification

The mechanical structure of the mobile robot, like Pioneer-2DX, is shown in figure 2. Presented robot has two degrees of freedom. In the word co-ordinates a posture is defined as $[x_A, y_A, \beta]^T$, where (x_A, y_A) is the position of the point A, and β is the heading angle of the robot with respect to absolute co-ordinates (x, y) . Using Maggi's formalism the dynamics of wheeled mobile robot can be written as [2]

$$\begin{bmatrix} a_1 + a_2 + a_3 & a_1 - a_2 \\ a_1 - a_2 & a_1 + a_2 + a_3 \end{bmatrix} \begin{bmatrix} \ddot{\alpha}_1 \\ \ddot{\alpha}_2 \end{bmatrix} + \begin{bmatrix} 0 & 2a_4\left(\dot{\alpha}_2 - \dot{\alpha}_1\right) \\ -2a_4\left(\dot{\alpha}_2 - \dot{\alpha}_1\right) & 0 \end{bmatrix} \begin{bmatrix} \dot{\alpha_1} \\ \dot{\alpha_2} \end{bmatrix} + \\ + \begin{bmatrix} a_5 sgn\dot{\alpha_1} \\ a_6 sgn\dot{\alpha_2} \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \tag{3}$$

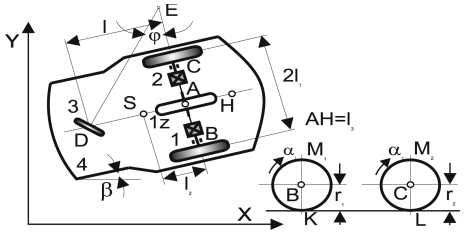


Fig. 2. The schematic of mobile robot

where **a** is a vector of the mobile robot parameters, which results from the system geometry, weights distribution and motions resistance, $u = [M_1, M_2]$ is a vector of the moments propelling driving wheels and $\alpha = [\alpha_1, \alpha_2]^T$ is a vector of turn angle of the driving wheels. The equation (3) can be written in the form

$$M(q)\ddot{q} + C_r(q, \dot{q})\dot{q} + F(q) = u \tag{4}$$

where $q = \alpha \in R^2$. Establishing $x_1 = q \in R^{n1}, x_2 = \dot{q} \in R^{n1}$ with n1=2 dynamic equations of motion (4) can be written in state space

$$\dot{x} = f(x, u) \tag{5}$$

where $f(.) \in R^{2n1x1}$ is a non-linear function vector. The identification model for the mobile robot (5) can by expressed by [2,6]

$$\dot{x} = Ax + G(x, u) \tag{6}$$

where $G(x, u) = f(x, u) - Ax$ and $A \in R^{2n1x2n1}$ is a Hurwitz matrix. Suppose that a neural network is used to approximate the non-linear vector $G(x, u) \in R^{2n1}$ according to

$$G(x, u) = W^T \phi(x) + \epsilon \tag{7}$$

with W the ideal approximating weights, $\phi(x)$ provides a suitable basis and ϵ is the approximation error with $\|\epsilon\| \leq z_\epsilon$. Then an estimate of $G(x, u)$ is given by

$$\hat{G}(x, u) = \hat{W}^T \phi(x, u) \tag{8}$$

This gives the following identification model

$$\dot{\hat{x}} = A\hat{x} + \hat{W}^T \phi(x, u) \tag{9}$$

where $\hat{x}$ denotes the state vector of the network model. Define the state error vector as

$$\tilde{x} = x - \hat{x} \tag{10}$$

so that the dynamical expression of the state error is given by

$$\dot{\tilde{x}} = A\tilde{x} + \tilde{W}^T \phi(x, u) + \epsilon \tag{11}$$

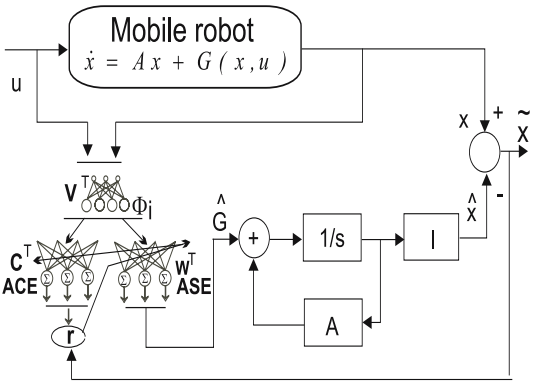


Fig. 3. Configuration for the identification framework

In RL context, the estimate of the function is often used in the actor-critic configuration as shown in figure 3. The estimate of $\hat{G}(x,u)$ is implemented by the ASE which the weights will be tuned by the signal from ACE. The outputs of ASE and ACE can by represented as $\hat{W}^T\phi$ and $\hat{C}^T\phi$ respectively. The ACE generates a reinforcement signal vector r to tune neural net weights $\hat{W}$. The first layer of the actor-critic system employs neural net with standard sigmoid functions and the matrix V is selected randomly to implement the decoder [1,8]. The proposed RL systems will be capable of avoiding too many trial and the error learning process and guarantee the stability of the proposed tracking controller. It means that the neural networks weights W and C are tuned on-line with no learning phase and all signals in the closed-loop system are bounded. Assume the input signal u be bounded and the norms of ideal weights $\|W\|$,$\|C\|$ are bounded by known positive values W_m ,C_m respectively. In order to calculate the reinforcement signal r and law of learning weights of the neural net actor-critic system, which is essential for the net's stability, we introduce the Lyapunov's function in the form of [4,5,6]:

$$V = 0.5\tilde{x}^T\tilde{x} + 0.5tr\tilde{W}^T F_w^{-1}\tilde{W} + 0.5tr\tilde{C}^T F_c^{-1}\tilde{C} \tag{12}$$

Making a differential calculus along the result of the system (11), and using the principle of learning weights of the net

$$\dot{\hat{W}} = F_w\phi r^T - \gamma F_w \|\tilde{x}\| \hat{W} \tag{13}$$

$$\dot{\hat{C}} = -F_c \|\tilde{x}\| \phi \left(\hat{W}^T\phi\right)^T - \gamma F_c \|\tilde{x}\| \hat{C} \tag{14}$$

with F_w ,F_c positive and diagonal constant matrices and using reinforcement signal equals to

$$r = \tilde{x} + \|\tilde{x}\| \hat{C}^T\phi \tag{15}$$

we get:

$$\dot{V} \leq -\tilde{x}^T A\tilde{x} + \tilde{x}^T\epsilon + \|\tilde{x}\| \operatorname{tr}[-\tilde{W}^T\phi\left(\hat{C}^T\phi\right)^T + \gamma\tilde{W}^T\hat{W} + \\ +\tilde{C}^T\phi\left(\hat{W}^T\phi\right)^T + \gamma\tilde{C}^T\hat{C}] \tag{16}$$

Since $\tilde{W}\hat{W} \leq \left\|\tilde{W}\right\| W_m - \left\|\tilde{W}\right\|^2$ and $\tilde{C}\hat{C} \leq \left\|\tilde{C}\right\| C_m - \left\|\tilde{C}\right\|^2$ and $\left\|\tilde{\phi}\right\|^2 \leq m$ there results

$$\dot{V} \leq -\lambda_{max}(A)\|\tilde{x}\|^2 - \\ -\gamma\|\tilde{x}\|\left\{\begin{array}{c} \left[\left\|\tilde{W}\right\| - (mC_m + \gamma W_m)/2\gamma\right]^2 + \\ +\left\|\tilde{C}\right\| - \left[(mW_m + \gamma C_m)/2\gamma\right]^2 - \left[(mC_m + \gamma W_m)/2\gamma\right]^2 - \\ -\left[(mW_m + \gamma C_m)/2\gamma\right]^2 - z_\epsilon/\gamma \end{array}\right\} \tag{17}$$

which is negative as long as

$$\|\tilde{x}\| > \gamma\left\{\left[(mC_m + \gamma W_m)/2\gamma\right]^2 + \left[(mW_m + \gamma C_m)/2\gamma\right]^2 + z_\epsilon/\gamma\right\}/ \\ /\lambda_{max}(A) = b_x \tag{18}$$

or

$$\left\|\tilde{W}\right\| > (mC_m + \gamma W_m)/2\gamma + \\ +\sqrt{\left\{\left[(mC_m + \gamma W_m)/2\gamma\right]^2 + \left[(mW_m + \gamma C_m)/2\gamma\right]^2 + z_\epsilon/\gamma\right\}} = b_w \tag{19}$$

or

$$\left\|\tilde{C}\right\| > (mW_m + \gamma C_m)/2\gamma + \\ +\sqrt{\left\{\left[(mC_m + \gamma W_m)/2\gamma\right]^2 + \left[(mW_m + \gamma C_m)/2\gamma\right]^2 + z_\varepsilon/\gamma\right\}} = b_c \tag{20}$$

where $\lambda_{max}(A)$ is the maximum eigenvalue of the matrix A. Thus $\dot{V}$ is negative outside a compact set. This guarantees that the error $\tilde{x}$,$\tilde{C}$ and $\tilde{W}$ are uniformly ultimately bounded [4,5].

4 Simulation Results

In this section, the computer simulation results of adaptive critic identifier for wheeled mobile robot are given to demonstrate the feasibility of the proposed

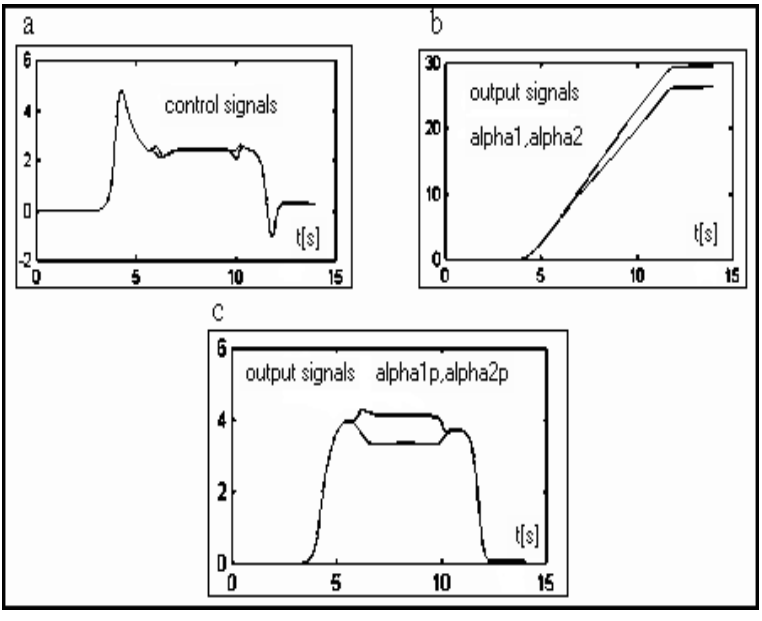


Fig. 4. Input signal u and generated the parameters of movement of mobile robot

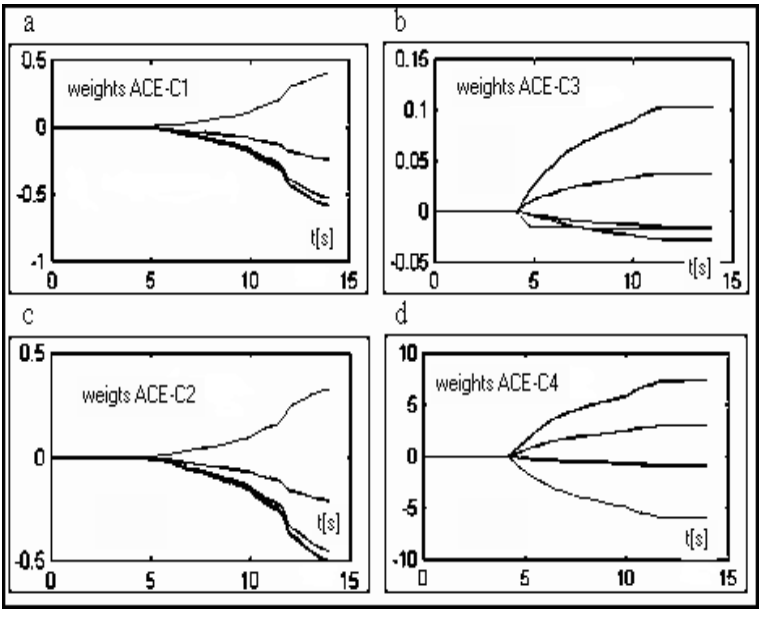


Fig. 5. The weights for neural critic element

solution. We have to transform the dynamical equation (3) into its state-variable description to the form (6). Let $X = [x_1, x_2, x_3, x_4]^T = [\alpha_1, \alpha_2, \dot{\alpha_1}, \dot{\alpha_2}]^T$ we get the identification model

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} -a_{x1} & 0 & 0 & 0 \\ 0 & -a_{x2} & 0 & 0 \\ 0 & 0 & -a_{x3} & 0 \\ 0 & 0 & 0 & -a_{x4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} x_3 + a_{x1}x_1 \\ x_4 + a_{x2}x_2 \\ f_3(X,u) + a_{x3}x_3 \\ f_4(X,u) + a_{x4}x_4 \end{bmatrix} \tag{21}$$

where a_{xi} are designed elements of a Hurwitz matrix A and $f_3(X,u), f_4(X,u)$ is a non-linear function elements, which results from the equation (3). Verification of the neural identifier suggested will be carried out for three periods of mobile robot motion, such as: starting period, travel at constant speed of point A, when point A is moving along circular trajectory, and stopping period [2]. The control signals u and the generated parameters of movement of mobile robot (real state) against time t are shown in figure 4a,b,c respectively. The elements of matrix A are chosen as $a_{x1} = 4.61, a_{x2} = 4.61, a_{x3} = 3.2, a_{x4} = 3.2$. Modeling the non-linear function vector $G(X,u) \in R^{4x1}$ in (21) with two layer NN, which is depicted in figure 1, and selecting the matrix V in (1) randomly using $m = 5$ neurons described by standard sigmoid functions for each element of vector G, gives the decoder for ASE and ACE elements. The parameter values used in this example are as follows: $\gamma = 0.01, F_w = diag(3), F_c = diag(0.01)$. The sampling time used in the simulation is 0.01 second. The rest of the data, and

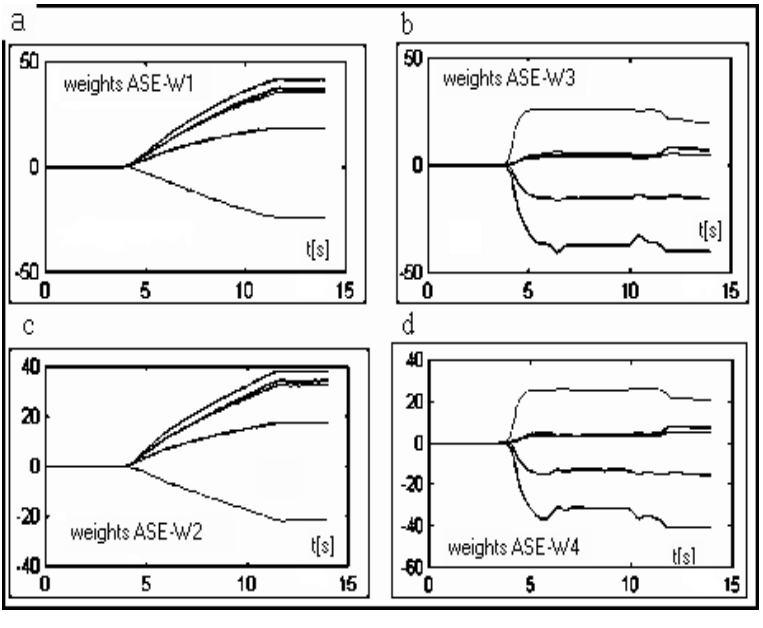


Fig. 6. The weights for neural actor element

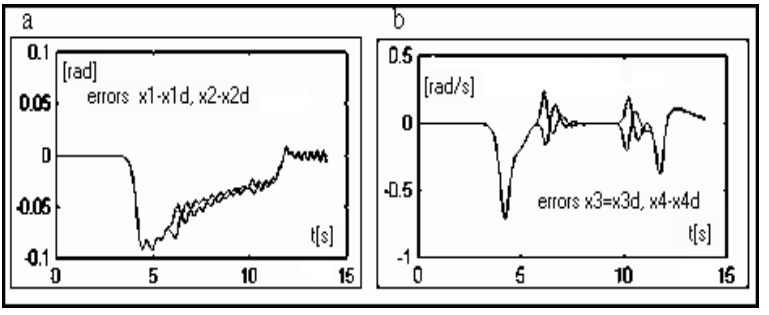


Fig. 7. State errors with time

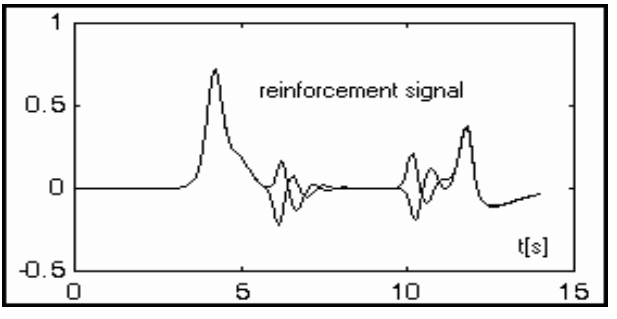


Fig. 8. Reinforcement signals with time

the average quantities of the parameters a in (3), which were assumed for the simulation, are given in [2]. The performance of the proposed RL systems are shown in figures 5-7. The weights for neural critic element and actor element are depicted in the figure 5,6 respectively. In the actor-critic NN identifier scheme derived in this paper there is no preliminary off-line learning phase. The weights are simply initialized at zero, for then figure 3 and equation (9) show that the identifier is just a linear with designed Hurwitz matrix A and gives bounded errors if $\lambda_{max}(A)$ is large enough. Therefore the closed-loop identifier remains stable until the actor-critic NN begins to learn as we can see in the picture 7. Moreover, it can be found that the weights shown in the figure 5,6 are bounded. The state errors $\tilde{x} = x - \hat{x} = x - xd$ of the mobile robot against time t are shown in figure 7a,b. It can be seen that the state errors are bounded. Moreover, it can be found that reinforcement signal shown in the figure 8 is bounded. Note that persistence of excitation (PE) is not needed to establish the bounds on $\tilde{C}, \tilde{W}$ with the modified weight tuning algorithm. The importance of the γ term added to the NN weight tuning algorithm (13),(14) is that it is possible to establish that is negative outside a compact set in the $\left(\|\tilde{x}\|, \left\|\tilde{C}\right\|, \left\|\tilde{W}\right\|\right)$ plane.

5 Conclusion

A sequential identification scheme for mobile robot with unknown nonlinearities using ASE-ACE reinforcement learning method has been developed. This work shows how to cope with nonlinearities through adaptive critics with no preliminary off-line learning required. The proposed scheme consists of a feedforward action generating NN that compensates for the unknown system nonlinearities. The learning of this NN is performed on-line based on a signal from a second higher level NN acting as a critic. The proposed identifier that can guarantee good performance and stability is derived from the Lyapunov stability theory. In this paper have been discuss relatively new developments in the general area of complex robotic systems. Basin on numerical analysis, it has been demonstrated that the tested identification algorithm is stable and simulated experiment conformed to theoretical expectations.

Acknowledgment. This research was supported by MEiN Grant No. 4 T07A 030 29.

References

1. Barto, A.G., Sutton R.S. , and Anderson W.: Neuron-like adaptive elements can solve difficult learning con-trol problems, IEEE Trans, on Systems, Man, and Cybernetics, Vol. 13, No. 5, 1983, 834-846.
2. Giergiel J., Hendzel Z., Zylski W., Modelling and Control of Wheeled Mobile Robots, WNT, Warsaw, (in Polish), 2002.
3. Hunt K.J., Sbarbaro D., Zbikowski R.,Gawthrop P.J.: Neural networks for control systems-A survey, Automatica, Vol. 28, No. 6, 1992, pp. 1083-1112.
4. Lewis F.L., Jagannathan S, Yesildirek A.: Neural network control of robot manipulators and nonlinear systems, Taylor and Francis, London, 1999.
5. Lin C-K.: A reinforcement learning adaptive fuzzy controller for robots, Fuzzy Sets and Systems, 137, 2003, 339-352.
6. Liu G.P.: Nonlinear identification and control, Advances in Industrial Control, Springer-Verlag, 2001.
7. Narendra K.S., Parthasrathy K.: Identification and control of dynamical systems using neural networks, IEEE Transaction on Neural Networks, vol. 1, no. 1, 1990, pp. 4-27.
8. Sutton R.S. and Barto A.G.: Reinforcement Learning, An Introduction, Cambridge, MA: MIT Press, 1999.

A New Chromatic Color Image Watermarking and Its PCA-Based Implementation

Thai Duy Hien[1], Zensho Nakao[1], Kazuyoshi Miyara[1], Yasunori Nagata[1], and Yen Wei Chen[2]

[1] University of the Ryukyus, Okinawa 903-0213, Japan
{tdhien, nakao, miyara}@augusta.eee.u-ryukyu.ac.jp
[2] Ritsumeikan University, Shiga 525
{chen}@is.ritsumei.ac.jp

Abstract. This paper presents a color image watermarking scheme based on spatial information analysis in spatio-chromatic space. Principal component analysis (PCA) is used as an appropriate tool for estimating information. We use RGB color images with a regular arrangement RGB channels into spatio-chromatic space, and PCA performs de-correlation of these signals. We show that spatio-chromatic principal components of images contain spatial information and color information, and that PCA spatio-chromatic analysis is able to help the reconstruction of full images. We propose an efficient watermarking method for color images based on the observation. Watermarks are embedded in some spatio-chromatic principal components of the image, which provide invisibility and robustness. The experimental results show that the proposed method possesses robustness against most common attacks, and image processing techniques.

1 Introduction

There have been proposed many watermarking methods, and they have mainly focused on grey scale image watermarking. Extension to the color case still presents one of the open issues in watermarking research. In field of image watermarking, research has been mainly focused on grayscale image watermarking, whereas the extension to the color case is usually accomplished by marking the image luminance, or by processing each color channel separately. There have been many watermarking algorithms developed in the image space, Fourier, DCT, Mellin-Fourier transforms and wavelet domains [1-4].

Cox *et al* [1] proposed a secure spread spectrum watermarking for multimedia in DCT domain. Application of the method to color images is straightforward. Color images are therefore converted in to a YIQ representation and the brightness component Y is then watermarked. The color image can then be converted to other formats but must be converted back to YIQ prior to extraction of the watermark. However, robustness against certain color image processing procedures should be investigated. Kutter *et al* [5, 6] proposed a color image watermarking scheme that embeds the watermark into the blue-channel of each pixel

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 787–795, 2006.
© Springer-Verlag Berlin Heidelberg 2006

by modifying its pixel value. The reason why the pixels in the blue channel are selected to embed the watermark is because the message in the blue channel is less sensitive to the human eyes.

In this work, we take a new approach to analysis and watermarking color images: consider color information in spatio-chromatic domain. We analyze spatial information in spatio-chromatic images. Principal component analysis (PCA) is used as an appropriate tool for estimating information. RGB color images are rearranged among color channels (Figure 1) into spatio-chromatic images and PCA perform de-correlation of those signals. We show that spatio-chromatic principal components of images contain spatial information and color information, and that PCA spatio-chromatic analysis is able to help the reconstruction of full spatio-chromatic images.

Based on the observation, watermarks are embedded in some spatio-chromatic principal components of images which possess invisibility and robustness. The experimental results show that the proposed method is robust against most common attacks, and image processing techniques.

2 Spatio-chromatic PCA of Color Images

Principal component analysis is a well-known statistical method for reducing the dimension of data sets. This method finds a new representation of the data set preserving only the most important information. It is based on spectral decomposition of the covariance matrix of the data set [7][8].

Step 1. Denote an RGB image of size MxNx3, by $\mathrm{I}_{\mathrm{R,G,B}}(m, n)$, where m and n take integer values from 0 through $M-1$ and $N-1$, respectively.

The image $\mathrm{I}_{\mathrm{R,G,B}}(m, n)$ is partitioned into sub-images of size KxK; we then construct a two dimensional matrix $\boldsymbol{X}$ which contains for each row a column vector x_j composed of spatial neighbors of size K for color bands (Figure 1). Thus, the size of matrix $\boldsymbol{X}$ is $(\mathrm{MxN}/K^2)\mathrm{x}(3K^2)$. This matrix, on which we apply PCA analysis, can be interpreted as containing on each row a representation of the spatio-chromatic random variables of a color image. We extract the principal components of sub-image color bands by finding the PCA transformation matrix (basis functions). Each sub-image color bands is then transformed by the PCA basis function.

The following steps are applied to the $\boldsymbol{X}$ to find the transformation matrix $[\Phi]$:

Step 2. Partition $\boldsymbol{X}$ into a number of sub-pixel KxK for convenience in numerical implementation. Sub-units are selected by random choosing KxK sub-pixel in R, G, and B at X separately, and then the size of sub-unit is Kx3K.

Consider each sub-unit as a vector (vector of pixels), and the data vectors can be written as: $\mathrm{X} = (\mathrm{x}_1,\mathrm{x}_2,\mathrm{x}_3,....,\mathrm{x}_\mathrm{m})^\mathrm{T}$
where vector x_i is the i^{th} sub-unit and T denotes the transpose matrix. Each sub-unit has Kx3K pixels, and each vector x_i has Kx3K dimensions.

Step 3. Calculate the covariance matrix $\boldsymbol{C}$ of $\boldsymbol{X}$,

$$\mathrm{C} = \mathrm{cov(X)} = (\mathrm{X} - \overline{\mathrm{X}})^{\mathrm{T}}(\mathrm{X} - \overline{\mathrm{X}}) \tag{1}$$

where $\overline{\mathrm{X}}$ is the mean vector of each sub-vector. The resulting matrix $\boldsymbol{C}$ of size $3K^2\mathrm{x}3K^2$ is then decomposed into eigenvalues $\boldsymbol{S}$ and corresponding eigenvectors $\boldsymbol{U}$, $C = USU^{-1}$ the columns of $\boldsymbol{U}$ are the eigenvectors and represent the basis functions of the transformation denoted by $[\Phi]$. We can represent these basis functions as spatio-chromatic samples and display their spatial and chromatic properties as color image. We have discovered that, the first principal components are mostly achromatic basis functions. If we had chosen a set of RGB images instead of one single image, we probably would have obtained a more accurate result.

Step 4. The matrix $[\Phi]$ represents a rotation matrix that is used to transform the original spatio-chromatic space into a new space where components are de-correlated. If we call $\boldsymbol{Y}$ the de-correlated matrix corresponding to $\boldsymbol{X}$, we have

$$\mathrm{Y} = \mathrm{X}\Phi \Rightarrow (\mathrm{Y} - \overline{\mathrm{Y}})^{\mathrm{T}}(\mathrm{Y} - \overline{\mathrm{Y}}) = \Phi^{\mathrm{T}}(\mathrm{X} - \overline{\mathrm{X}})^{\mathrm{T}}(\mathrm{X} - \overline{\mathrm{X}})\Phi \tag{2}$$

Equation 2 shows clearly that Y is a de-correlated data set.

Step 5. The reconstruction matrix X^* is obtained from X as follows :

$$X^* = X\Phi D\Phi^{-1} \tag{3}$$

where $\boldsymbol{D}$ is a diagonal matrix that contains zero or one depending on if the corresponding vector is used or not.

As shown in Figure 2(a), only the first principal components of spatio-chromatic samples give a good approximation of the image (PSNR=18.09). We showed that the achromatic basis functions are able to reconstruct accurately the luminance information and color information of the original images. Figures 2(b), 2(c) and 2(d) show the reconstructed images with first two (PSNR=21.5), first five (PSNR=25.36) and first ten (PSNR=27.79) basis functions respectively. Since the main information of the sub-image is concentrated in first several principal components, it is easy to choose the components for embedding the watermark. We can choose components to embed watermarks to adapt the invisibility and robustness.

3 Spatio-chromatic PCA Based Watermarking

3.1 Embedding

We suppose that the original image $\mathrm{I}_{\mathrm{R,G,B}}(m, n)$ is sub-divided into a number of sub-images. The matrix X can be obtained by step 1 and step 2 described earlier. We can get PCA chromatic basis function which is denoted by $[\Phi]$ (step 3). Principal components are computed on these sets of sub-images. The de-correlated PCA coefficients are calculated thought step 4. A set of coefficients

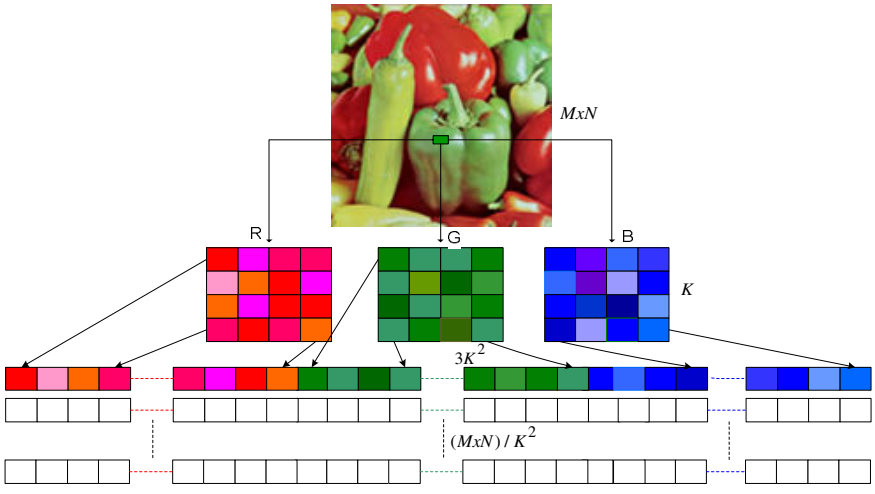


Fig. 1. Decomposition of a color image into spatio-chromatic images

Fig. 2. Reconstruction of the original RGB image, (a) using only the first principal component, (b) using first two principal components, (c) using first five principal components and (d) using first ten principal components

in each sub-unit is selected to embed the watermarks by modifying the PCA coefficients. Watermarks are added in perceptually insignificant components of sub-images.

In this proposal, the watermark consists of pseudo-random number sequence of length M with values w_i normally distributed, which can be written as $W = (w_1, w_2, ..., w_M)$ where w_i is random numbers. The method embeds the watermark into selected components of each PCA sub-unit uncorrelated coefficients which are obtained by step 4. Thus, we choose one watermark w_i and embed into y_i by the equation:

$$y^{'} = y + \alpha y_i w_i \tag{4}$$

where i = 1, 2. . .m is a scaling parameter to control strength of the watermark and $y^{'}$ is a watermarked coefficient. Finally the watermarked image $\mathrm{I}^{'}_{\mathrm{R,G,B}}(m, n)$ is obtained by applying the inverse PCA process (step5).

3.2 Detection

Watermark detection is applied to the watermarked image $\mathrm{I}'_{\mathrm{R,G,B}}(m,n)$ for copyright purposes. The procedure of watermark detection is as follows:

Partition the watermarked image into a number sub-image. The image data vectors can be written as: $X' = \left(x'_1, x'_2, ..., x'_M\right)^T$ where vector x'_i is the x'_{th} sub-units and T denotes the transpose matrix. (Step 1 and step2). Sub-unit coefficients of X' are computed by applying the PCA image basis function [Φ] (step 4). The PCA coefficients, which were embedded watermarks, are selected to generate watermarked coefficient vector: $C^* = (c_1^*, c_2^*, ..., c_M^*)$

The correlation value between the watermark W and possibly corrupted coefficient C^* is calculated to detect the mark:

$$Corr = \frac{WC^*}{M} = \frac{1}{M}\sum_{1}^{M} w_i c_i \tag{5}$$

By applying the correlation-based watermarking method, watermark correlations are calculated first for watermarks $W = (w_1, w_2, ..., w_M)$, and then for 1000 different random-watermarks $W_k = (k = 1 \sim 1000)$. The correlation can be used to determine whether a given mark is present or not. In order to detect the watermark, a threshold which is estimated on the watermarked image is applied to evaluate the detection system [9]. The threshold is defined by the equation:

$$T = \frac{\alpha}{3M}\sum_{i=1}^{M} y_i. \tag{6}$$

4 Computer Simulations

In this section, we demonstrated on inserting a watermark in color images based on the properties of spatio-chromatic PCA. Experiments are performed with some standard color images of size 512x512. Images are partitioned into sub-images of 8x8 pixels; RGB color channels are separated and the spatio-chromatic space is constructed. The PCA spatio-chromatic orthogonal basis functions are determined by those data and they are adaptive to data (Section 2, step 3). Furthermore, since the main information of the color image is concentrated in principal components, it is easy to select suitable components for embedding.

For each of block spatio-chromatic coefficients, sixteen different watermark random numbers are inserted into the sixteen coefficients. To carry out this process a watermark which has total length of M = 655361 is randomly generated with standard normal distribution. Signal to noise ratio (PSNR) or mean square error (MSE) is used to evaluate the quality of the watermarked image.

One important requirement of watermarking is to compromise between the invisibility and the robustness. First, the watermark should be embedded in an invisible way to avoid degrading of perceptual quality. Second, the watermark

should be robust against watermark attacks which are applied to image content. Those attacks include but not limited to lossy image compression, filtering, noise-adding, and geometrical modification.

To assess the validity of spatio-chromatic PCA watermarking in terms of watermark invisibility, we select a set of standard color images and we marked each image. It is noted that the robustness can be improved by increasing the strength of the embedded watermark, which affects perceptible degradation of the image. For the experiments we chose empirically $alpha = 0.4$. Figure 3 shows five original images and the corresponding watermarked images. Both original and watermarked images are evidently indistinguishable, and the watermarked images have corresponding detector response, and PSNR values in Table 1.

One important question in applying PCA is the computational cost for detection of the watermarks. The detection simulation took few second of computing time on an AMD 3000++ personal computer (PC). It is clearly possible to design watermarking application using computational method based on PCA.

Extensive testing was performed to assess the performance of the proposed watermarking scheme from the point of view robustness. Tests were carried out aiming at measuring the robustness of the watermark against standard image

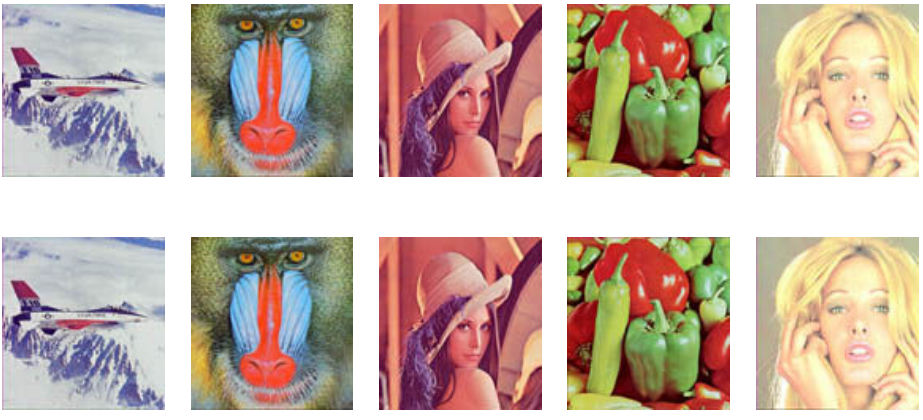
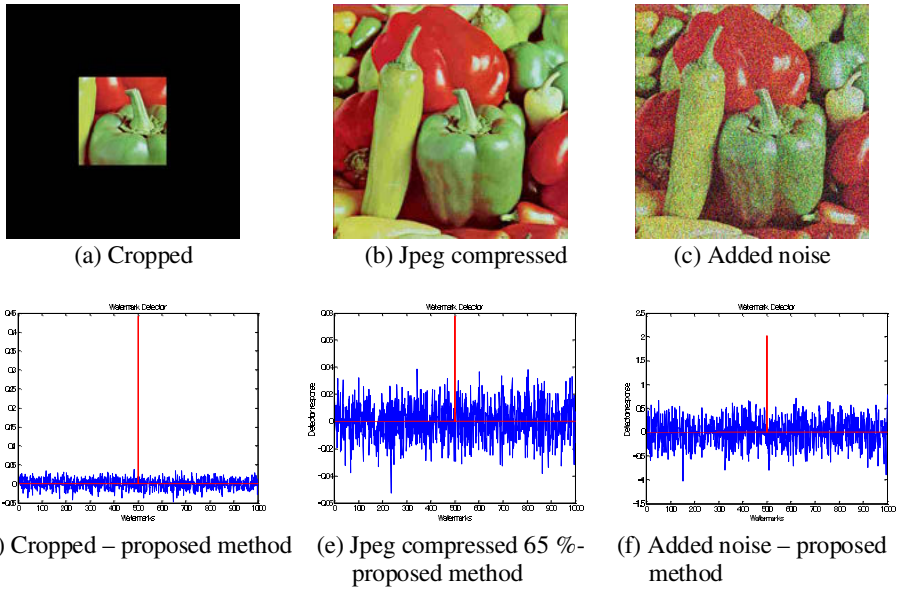


Fig. 3. The original and the corresponding watermarked images in experiments, with scaling parameter

Table 1. The detector response, Threshold, MSE, and PSNR: the proposed method

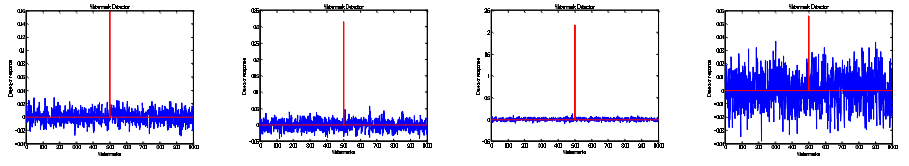
Images	Detector response	Threshold	MSE	PSNR
Airplane	0.616	0.285	0.134	56.9
Baboon	3.89	1.27	2.23	44.6
Lena	1.2	0.444	0.302	53.3
Peppers	1.69	0.596	0.528	50.9
Tiffany	1.24	0.481	0.346	52.7



(a) Cropped (b) Jpeg compressed (c) Added noise

(d) Cropped – proposed method (e) Jpeg compressed 65 %- proposed method (f) Added noise – proposed method

Fig. 4. Cropping, Jpeg compression and added noise attacks, and the corresponding detector responses



(a) Lowpass filtered (b) Median filtered (c) Resized (d) Jpeg2000 Compressed ratio 0.15

Fig. 5. Detector responses of filtering, Jpeg 2000 and resizing attacks

Table 2. The corresponding detector response attacks

Attacks	Airplane	Baboon	Lena	Peppers	Tiffany
Adding random noise (Power = 5000)	1.11	5.72	2.15	2.26	2.12
Low-pass filtered (5x5)	0.10	0.77	0.24	0.16	0.16
Median filtered (5x5)	0.23	1.04	0.42	0.32	0.24
Center crop (85%) 6	0.11	1.28	1.04	1.1	0.88
Jpeg compression (65%) 3	0.07	0.06	0.06	0.07	0.05
Jpeg2000 compression (rate =0.15)	0.06	0.07	0.07	0.05	0.05
Resized	1.04	5.19	1.77	2.17	1.76

processing, lossy compression and geometric manipulation. For each attack, the response of the detector for 1000 randomly generated watermarks, including the actually embedded one within image, was measured. The response relative to the true watermark and the highest response among those corresponding to the other watermarks are plotted. Figures 4 and Fig.5 show the detector response of attacks and Table 2 shows the corresponding detector response values. Experimental results show that watermarks by the proposed method are invisible as needed in most practical applications and robust enough against common attacks, and image processing techniques.

5 Conclusions

A long-standing problem in statistics and related areas is how to find a suitable representation of multivariate data. Representation here means that we show some how transform the data so that their essential structure is made more visible or accessible. It is important for subsequent analysis of the data, such as in de-noising, edge detection, pattern recognition, and watermarking, that the data is represented in a manner that facilitates the analysis. Several principles and methods have been developed to find a suitable linear representation. These include principal component analysis (PCA), factor analysis (FA), projection pursuit (PP), and independent component analysis (ICA). In this paper PCA is used as an appropriate tool for estimating color information in a spatial chromatic image domain, since the basis functions are determined by data and they are adaptive to the data.

A new color watermarking method is introduced in this paper. Our PCA spatio-chromatic analysis is able to fully reconstruct color images. We proposed a new color image watermarking based on spatial information analysis in spatio-chromatic images. Watermarks are embedded in some spatio-chromatic principal components of images which adapt for invisibility and robustness. The proposed method is closely related to DCT or wavelet based frequency domain method, but the orthogonal PCA spatio-chromatic basis functions are determined by data and they are adaptive to the data. Furthermore, since main information of the spatio-chromatic image is concentrated in the principal components, it is easy to choose the components for embedding. Our results show that the proposed method is robust against the most common attacks, and image processing techniques.

References

1. J.Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. IEEE Transactions on Image Processing, 6(12):1673-1687, December 1997
2. Christine I. Podilchuk, Wenjun Zeng, "Image-Adaptive Watermarking Using Visual Models," IEEE Journal Selected Areas of Communications, JSAC, vol.16, No.4,pp. 525-539, 1998

3. Thai D. Hien, Zensho Nakao, and Yen-Wei Chen, "ICA-based robust logo image watermarking," Proceedings of SPIE on Security, Steganography, and Watermarking of Multimedia Contents VI, January, 2004, San Jose, USA, vol. 5306
4. Thai D. Hien, Yen-Wei Chen, & Z. Nakao, "A Robust Digital watermarking technique Based on Principal Component Analysis," International Journal of Computational Intelligence and Applications, Vol.4, No.2, pp.138-192, 2004.
5. M. Kutter, SK Bhattacharjee and T. Ebrahimi, "Towards Second Generation Watermarking Schemes," Proc., of the IEEE Inter., Conf., on Image Processing, ICIP 99, Kobe, Japan, pp. 320-323, vol1, 1999
6. M. Kutter, F. Jordan, F. Bossen, 2Digital signature of color image using amplitude modulation," in: I.K. Sethi, R. Jain (Eds.), Storage and Retrieval for Image and Video Databases V, Vol. 3022, SPIE, San Jose, CA, February 1997, pp. 518–526
7. Xiang-Yan Zeng, Yen-Wei Chen, Zensho Nakao, and Hanqing Lu, "A new texture feature based on PCA pattern maps and its application to image retrieval," IEICE Trans. Information & Systems, Vol.E-86D, pp. 929-936, May 2003
8. P. J B Hancock, R. J Baddeley, and L. S Smith, "The principal components of natural images," Network, vol.3, pp.61-70, March 1992
9. M. Barni, F. Bartolini, V. Cappellini, A. Piva, "A DCT-domain system for robust image watermarking," Signal Processing, Special Issue in "Copyright Protection and Access Control for Multimedia Services," pp.357-372, 1998

Human Identification Based on Fingerprint Local Features

Maciej Hrebień and Józef Korbicz

Institute of Control and Computation Engineering
University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Góra
{m.hrebien, j.korbicz}@issi.uz.zgora.pl
http://www.issi.uz.zgora.pl

Abstract. To meet today's demands for constructing even better biometric systems of human identification, a few fingerprint matching techniques are presented in this paper. One can also find here a short description of fingerprint image pre-processing and the minutiae extraction scheme used in our research. Because there is still a need to find the best matching algorithm, preliminary research was conducted to compare quality differences and answer times between the analysed methods for a prepared *on-line* system.

1 Introduction

In the last decade we have observed an increased interest in biometric technologies [13], that is, human identification based on one's individual features. Most of this interest is due to security reasons – payment operations without cash, the secrecy of information stored in databases, restricted access to specific areas, etc. One can also notice that such systems if, for example, considered as a lock are more comfortable in use – we do not have to care any more about keys or ID cards which can be additionally easily lost, forgotten or even stolen.

Fingerprint identification is one of the oldest and still very important biometric technologies considered nowadays. A fingerprint itself can be defined as a structure of ridges and valleys. The shape and the way in which ridge continuities are disturbed is individual and unique for all human beings [4].

Fingerprint matching algorithms try to compare exclusive local characteristics of a finger and the relationships between them, and decide whether two fingerprint impressions delivered by a scanner belong to the same person [9][10][12]. Certainly, beside their effectiveness, their answer times are equally important in all real-time implementations – the use of any system based on these methods should be as convenient as possible if it acts, for example, as a door lock.

This paper discusses a few fingerprint matching techniques, that is, the Hough transform, the structural global star method and the speeded up correlation approach (Sect. 4). One can also find here a brief description of image enhancement (Sect. 2) and the minutiae extraction method applied in this work (Sect. 3). Preliminary experimental results are presented in Section 5.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 796–803, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Fingerprint Image Enhancement

A very common technique for reducing the quantity of information received from a fingerprint scanner in the form of a grayscale image is known in literature as the Gabor filtering [9]. The greatest advantage of the filter is its nearly binary output – the intensity histogram has a U-shaped form [5] so the fingerprint structure of ridges and valleys can be easily binarized. The filter is also capable of removing any impulsive or Gaussian noise in similar efficacy as, for example, the method proposed in [8] for cDNA images.

The Gabor filter is defined by

$$h(x, y, f, \theta) = exp\Big\{ -\frac{1}{2}\Big(\frac{x_\theta^2}{\delta_x^2} + \frac{y_\theta^2}{\delta_y^2}\Big)\Big\} \cos(2\pi f x_\theta), \tag{1}$$

where $x_\theta = x \sin\theta + y \cos\theta$, $y_\theta = x \cos\theta - y \sin\theta$ [1] and, in our case, θ is the local ridge orientation, f is the estimation of ridge frequency in the currently processed block of the image, δ_x and δ_y are space constants defining the stretch of the filter.

3 Minutiae Detection

The uniqueness of a fingerprint is exclusively determined by its local ridge characteristics called the minutiae points. In this paper only two fundamental types of minutiaes are considered, that is, ending and bifurcation. An ending point is the place where a ridge ends its flow, while a bifurcation is the place where a ridge forks into two parts.

To determine whether a pixel at the position (i, j) in the binary form of a fingerprint image is a minutiae point, we can use the image thinning technique [3] witch conjunction to the mask rules illustrated in Fig. 1a-d.

To define minutiae orientation, we can use a (7×7) mask technique (see, for instance, Fig. 1e-f) with angles quantized to 15° and the center placed at the minutiae point. The orientation of an ending point is equal to the point where a ridge is crossing through the mask. The orientation of a bifurcation point can be estimated with the same method but only the leading ridge is considered, that is, the one with a maximum sum of angles to the other two ridges of the bifurcation.

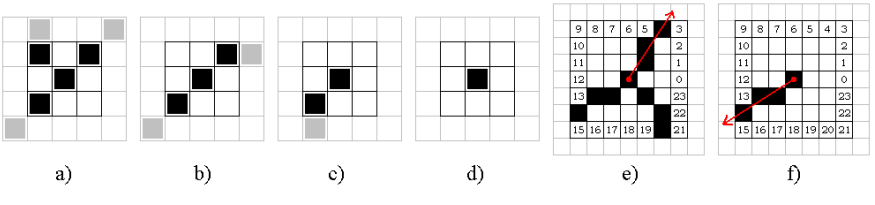


Fig. 1. Example of: 3×3 masks used to define: a) bifurcation, b) non-minutiae point, c) ending, d) noise; and masks used to estimate: e) bifurcation (60°) and f) ending (210°) point orientation

4 Minutiae Matching

4.1 Hough Transform

Let $M_A = \{m_1^A, m_2^A, \ldots, m_m^A\}$ and $M_B = \{m_1^B, m_2^B, \ldots, m_n^B\}$ denote minutiae sets determined from the images A and B. Each minutiae is defined by the image coordinates (x, y) and the orientation angle $\theta \in [0 \ldots 2\pi]$, that is, $m_i^A = \{x_i^A, y_i^A, \theta_i^A\}_{i=1\ldots m}$ and $m_j^B = \{x_j^B, y_j^B, \theta_j^B\}_{j=1\ldots n}$.

The Hough transform [10] can be performed to find the best alignment between the M_A and M_B sets including the possible scale, rotation and displacement between the images A and B. The transformation space is discretized – each parameter of the geometric transform $(\Delta x, \Delta y, \theta, s)$ comes from a finite set of values. A four dimensional accumulator A is used to accumulate the evidences of alignment between each considered pair of minutiaes. The best parameters of the geometric transform, that is, $(\Delta x^+, \Delta y^+, \theta^+, s^+)$ are arguments of the maximum value from the accumulator (see the procedure in Fig. 2).

$$
\begin{array}{l}
\forall_i \forall_j \forall_k \forall_l \;\; A(i,j,k,l) \leftarrow 0 \\
\\
FOR \; \{x_i^A, y_i^A, \theta_i^A\} \in M_A, \quad i = 1 \ldots m \\
\quad FOR \; \{x_j^B, y_j^B, \theta_j^B\} \in M_B, \quad j = 1 \ldots n \\
\qquad FOR \; \theta \in \{\theta_1, \theta_2, \ldots, \theta_k\}, \quad k = 1 \ldots K \\
\qquad\quad IF \;\; \min(|\theta_i^A + \theta - \theta_j^B|, 2\pi - |\theta_i^A + \theta - \theta_j^B|) < \theta_0 \\
\qquad\qquad FOR \; s \in \{s_1, s_2, \ldots, s_l\}, \quad l = 1 \ldots L \\
\qquad\qquad \{ \\
\qquad\qquad\quad \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \leftarrow \begin{bmatrix} x_i^A \\ y_i^A \end{bmatrix} - s \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_j^B \\ y_j^B \end{bmatrix} \\
\\
\qquad\qquad\quad \Delta x^\#, \Delta y^\#, \theta^\#, s^\# \leftarrow discretize(\Delta x, \Delta y, \theta, s) \\
\qquad\qquad\quad A(\Delta x^\#, \Delta y^\#, \theta^\#, s^\#) \leftarrow A(\Delta x^\#, \Delta y^\#, \theta^\#, s^\#) + 1 \\
\qquad\qquad \} \\
\\
\Delta x^+, \Delta y^+, \theta^+, s^+ \leftarrow \arg\max(A)
\end{array}
$$

Fig. 2. Hough transform routine

After performing the transformation, minutiae points have to be juxtaposed to calculate the matching score with respect to their type, orientation and distance tolerance.

An example result of the Hough transform is shown in Fig. 3.

4.2 Global Star Method

The global star method is based on a structural model of fingerprints. A star is a structure with the central point placed in one of minutiaes and arms directed to the remaining ones of the same type (Fig. 4). Assuming that $M_A = \{m_1^A, m_2^A, \ldots, m_m^A\}$ and $M_B = \{m_1^B, m_2^B, \ldots, m_n^B\}$ indicate the sets of minutiaes of one type, m stars for the image A and n stars for the image B can be

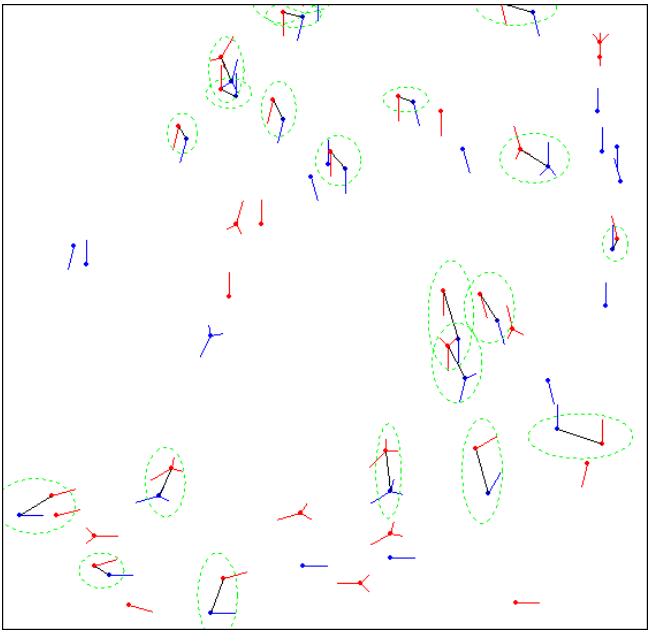


Fig. 3. Example result of the Hough transform (matched minutiae points are marked with an elipsis and connected with a straight line)

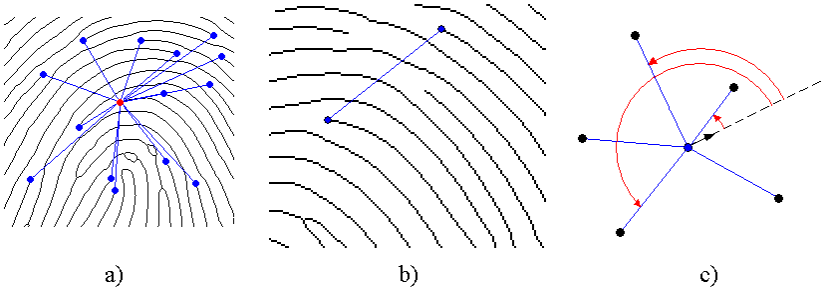


Fig. 4. General explanation of the star method, examples of: a) star created for fingerprint ending points, b) ridge counting (here equal to 5), c) relative angle determination between the central minutiae and the remaining ones

created: $S^A = \{S_1^A, S_2^A, \ldots, S_m^A\}$ and $S^B = \{S_1^B, S_2^B, \ldots, S_n^B\}$, where each star can be defined by $S_i^A = \{m_1^A, m_2^A, \ldots, m_m^A\}_{i=1\ldots m}$ with the center in m_i^A and $S_j^B = \{m_1^B, m_2^B, \ldots, m_n^B\}_{j=1\ldots n}$ with the center in m_j^B.

In opposition to the local methods [12], the voting technique for selecting the best aligned pair of stars $(S_w^A i, S_w^B j)$ can be performed (Fig. 5), including matching such features like the between-minutiae angle K and the ridge count D (Fig. 4). In the final decision also the orientation of minutiaes, corrected by

$\forall_i \forall_j \; A(i,j) \leftarrow 0$

$FOR \; S_i^A \in S^A, \quad i = 1 \ldots m$
  $FOR \; S_j^B \in S^B, \quad j = 1 \ldots n$
    $FOR \; m_k^A \in S_i^A - \{m_i^A\}$
      $assuming \; that: \; m_l^B \in S_j^B - \{m_j^B\}$
      $IF \; \exists_l (|D(m_j^B, m_l^B) - D(m_i^A, m_k^A)| < d_0 \; \& \; |K(m_j^B, m_l^B) - K(m_i^A, m_k^A)| < k_0)$
        $A(i,j) \leftarrow A(i,j) + 1$

$S_{wi}^A \leftarrow S^A(\arg_i(\max(A))$
$S_{wj}^B \leftarrow S^B(\arg_j(\max(A))$

Fig. 5. First stage of the global star matching algorithm

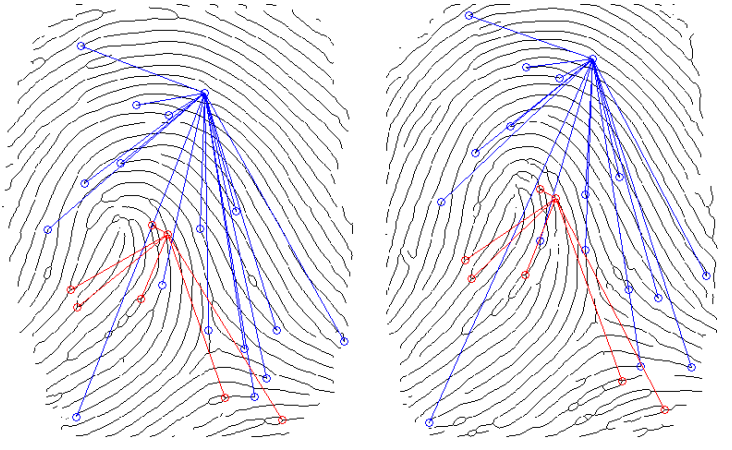


Fig. 6. Example result of the global star matching method

the angle of the star's central point orientation difference, have to be taken into account (with a given tolerance).

An example result of the global star method is shown in Fig. 6.

4.3 Correlation

Because of non-linear distortion, skin conditions or finger pressure that cause the varying of image brightness and contrast [9], correlation between fingerprint images cannot be applied directly. Moreover, taking into account the possible scale, rotation and displacement, an intuitive sum of squared differences is computationally very expensive. To eliminate or at least reduce some of the abovementioned problems, a binary representation of the fingerprint can be used. To

speed up the process of preliminary alignment, a segmentation mask can be used with conjunction to the center of gravity of binary images. Also, the quantization of geometric transform features can be applied, considering the scale and rotation only at the first stage (since displacement is the difference between the centers of gravity) minimizing the F_{seg} performance index:

$$F_{seg}(A,B) = \sum_{i=1}^{N}\sum_{j=1}^{N} \begin{cases} 1, \text{ where } A(i,j) \neq B(i,j) \\ 0, \text{ otherwise.} \end{cases} \tag{2}$$

After finding nearly the best alignment of segmentation masks, looking for the best correlation is limited to a much more reduced area. Including the rotation, vertical and horizontal displacement, stretch and arbitrarily selected granularity of these features, the best correlation can be found by searching for the maximum value of the F_{img} criteria:

$$F_{img}(A,B) = \sum_{i=1}^{N}\sum_{j=1}^{N} \begin{cases} 1, \text{ where } A(i,j) = B(i,j) = obj \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

where *obj* represents the object's (ridge) pixel.

Because fingerprint correlation does not inform us directly about minutiae matching, the thinning process with minutiae detection should be applied to both binary images from the best correlation. Then two sets of minutiaes can be compared to sum up the matching score.

An example result of the correlation method is shown in Fig. 7.

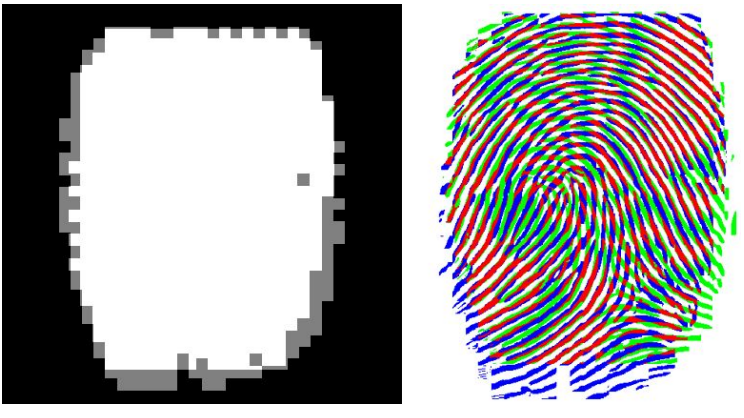


Fig. 7. Example result of the correlation method: best aligned segmentation masks (left) and best correlated binary forms of fingerprint images (right)

5 Experimental Results

The experiments were performed on a PC with a Digital Persona U.are.U 4000 fingerprint scanner. The database consists of 20 fingerprint images with 5 different impressions (plus one more for the registration phase).

There were three experiments. The first and the third one differ in parameter settings of each method. In the second experiment the image selected to the registration phase was chosen arbitrarily as the best one in the arbiter's opinion (in the first and the third experiment the registration image was the first fingerprint image acquired from the user).

All images were enhanced with the Gabor filter presented in Section 2 and matched using the algorithms described in Section 4. The summary of matching results for Polish regulations concerning fingerprint identification based on minutiaes [4] and time relations between each method are grouped in Tab. 5.

As one can easily notice, the Hough transform gave us the fastest response and the highest hit ratio from the methods considered. Moreover, it can be quite easily vectorized to perform more effectively with SIMD organized computers.

The global star method is scale and rotation independent but more expensive computationally because of star the creation process – determining the ridge count between the m_A and m_B minutiaes requires a time consuming iteration process. Moreover, possible filtering errors causing breaks in ridge continuity can disturb proper ridge count determination.

The analysis of an error set of the correlation method shows that it is the most sensitive one in the case of image selection for the registration phase from the group of the algorithms considered. Additionally, it is time consuming because of its complexity.

Table 1. Summary of the achieved results (HT – Hough Transform, GSM – Global Star Method, CRL – Correlation)

	Experiment	HT	GSM	CRL
Match percentage	1	85	45	37
	2	88	76	70
	3	82	80	61
Avg. count of endings / bifurcations in the best match	1	25/7	10/3	16/4
	2	25/7	13/4	19/5
	3	26/7	19/5	17/5
Number of images that did not cross match threshold	1	1	22	6
	2	1	10	1
	3	1	2	2
Time relation	1, 2, 3	1 HT	$\sim$ 6 HT	$\sim$ 14 HT

6 Conclusions

This paper reviews several selected fingerprint matching techniques. The preliminary experimental results show quality differences and time relations between the

algorithms considered. Additionally, the influence of selecting an image for the registration phase can be observed – the more representative image selected, the higher match percentage.

Because the selected parameters for each investigated method were suboptimal, it is still a challenge to use global optimization techniques for finding the best values on a much more extended database. Additionally, automatic image pre-selection (classification) [2][6][7] and/or hardware implementation [10] can speed up the whole matching process for large databases.

References

1. Andrysiak, T., Choraś, M: Image Retrieval Based on Hierarchical Gabor Filters, Int. J. of Appl. Math. and Comput. Sci., vol. 15, no. 4, pp. 471-480, 2005
2. Cappelli, R., Lumini, A., Maio, D., Maltoni, D.: Fingerprint Classification by Directional Image Partitioning, IEEE Trans. Pattern Anal. Mach. Intell., vol. 21, no. 5, pp. 402-421, 1999
3. Fisher, R., Walker, A., Perkins, S., Wolfart, E.: Hypermedia Image Processing Reference, John Wiley & Sons, 1996
4. Grzeszyk, C.: Dactyloscopy, PWN, Warszawa, 1992 (in Polish)
5. Hong, L., Wan, Y., Jain, A.: Fingerprint Image Enhancement: Algorithm and Performance Evaluation, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 777-789, 1998
6. Jain, A., Minut, S.: Hierarchical Kernel Fitting for Fingerprint Classification and Alignment, in Proc. Int. Conf. Pattern Rocog., vol. 2, pp. 469-473, 2002
7. Karu, K., Jain, A.: Fingerprint Classification, Pattern Recog., vol. 29, no. 3, pp. 389-404, 1996
8. Lukac, R., Smołka, B.: Application of the Adaptive Centerweighted Vector Median Framework for the Enhancement of cDNA Microarray Images, Int. J. of Appl. Math. and Comput. Sci., vol. 13, no. 3, pp. 369-383, 2003
9. Maltoni, D., Maio, D., Jain, A., Prabhakar, S.: Handbook of Fingerprint Recognition, Springer NY, 2003
10. Ratha, N., Karu, K., Chen, S., Jain, A.: A Real-time Matching System for Large Fingerprint Databases, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 28, no. 8, pp. 799-813, 1996
11. Stock, R., Swonger, C.: Devolopment and Evaluation of a Reader of Fingerprint Minutiae, Cornell Aeronautical Laboratory, Technical Report, 1969
12. Wahab, A., Chin, S., Tan, E.: Novel Approach to Automated Fingerprint Recognition, IEE Proc. in Vis. Image Signal Process., vol. 145, no. 3, 1998
13. Zhang, D., Campbell, P., Maltoni, D., Bolle, R. (Eds): Special Issue on Biometric Systems, IEEE Trans. on Systems, Man, and Cybernetics, vol. 35, no. 3, pp. 273-450, 2005

Genetic Graph Programming for Object Detection

Krzysztof Krawiec and Patryk Lijewski

Institute of Computing Science, Poznań University of Technology
Piotrowo 2, 60965 Poznań, Poland
krawiec@cs.put.poznan.pl

Abstract. In this paper, we present a novel approach to learning from visual information given in a form of raster images. The proposed learning method uses genetic programming to synthesize an image processing procedure that performs the desired vision task. The evolutionary algorithm maintains a population of individuals, initially populated with random solutions to the problem. Each individual encodes a directed acyclic graph, with graph nodes corresponding to elementary image processing operations (like image arithmetic, convolution filtering, morphological operations, etc.), and graph edges representing the data flow. Each graph contains a single input node to feed in the input image and an output node that yields the final processing result. This genetic learning process is driven by a fitness function that rewards individuals for producing output that conforms the task-specific objectives. This evaluation is carried out with respect to the training set of images. Thanks to such formulation, the fitness function is the only application-dependent component of our approach, which is thus applicable to a wide range of vision tasks (image enhancement, object detection, object tracking, etc.). The paper presents the approach in detail and describes the computational experiment concerning the task of object tracking in a real-world video sequence.

1 Introduction

Automated interpretation of visual information is in general difficult, mainly due to various forms of imperfectness of visual data, including, but not limited to, incompleteness (e.g., due to object occlusion), inconsistency (e.g., due to incoherent object labelling), and imprecision (e.g., due to spatial sampling). This calls for use of methods that autonomously learn from an interaction with the external world and acquire the knowledge required to solve the visual task (like recognition, location, navigation, tracking, etc.). Handcrafting of computer vision (CV) systems, though still prevailing and predominant for the nearest future, does not offer general and scalable solutions, and leads to over-specialization of particular CV disciplines. Motivated by these issues, in this paper we present a novel visual learning framework based on evolutionary design. Our approach uses a variant of genetic programming (GP, [4]) to search for an (sub)optimal program composed of elementary image processing and analysis steps.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 804–813, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Related Work and Contribution

Contemporary recognition systems are mostly open-loop and human expert's input is still predominant in their design. In most approaches reported in the literature, learning, if any, is limited to parameter optimization that usually concerns only a particular processing step, such as image segmentation, feature extraction, etc. Only a few contributions [1,2,3,12,14,11,8,9] attempt to close the feedback loop of the learning process at the highest (e.g., recognition) level, and test the proposed approach in a real-world setting. Moreover, some of the proposed methods make an intense use of domain-specific knowledge and are highly specialized towards a particular application.

In our former work on feature synthesis [1,5], we proposed a visual learning framework inspired by the linear genetic programming and tested it on different object recognition tasks. The main contribution of this paper is the elaboration and experimental evaluation of a novel model of visual information processing and learning. To represent procedures that reason from visual data, the proposed approach uses a variant of genetic programming called hereafter genetic graph programming. The automation of the learning process frees the system's designer from the tedious and time-consuming trial-and-error approach. In particular, using graphs rather than trees for representing image processing procedures reduces the risk of code bloat, simplifies the interpretation of the evolved solutions, and reduces the computational cost of individual evaluation.

3 Genetic Graph Programming

Genetic Programming (GP, [4]) is a variant of evolutionary computation (EC). In GP, individuals processed by the algorithm, rather than being *passive* solutions to the problem being solved (e.g., a sequence of towns in traveling salesperson problem), are *active programs* that process some external data. Usually, in GP a non-linear representation of solutions is preferred. In particular, most GP-based approaches assume that the individuals undergoing evolution have the form of Lisp-like expressions (expression trees). Such expressions expect input data in leaves, process it in inner tree nodes (operation nodes), and produce output at the tree root node. The evaluation of an individual consists in letting it process some external (training) data, and comparing the produced output with the expected value.

As an illustration, let us refer to the well-know application of GP to the task of *symbolic regression*. In symbolic regression, the task of the learner is to synthesize a symbolic (analytic) form of an unknown function f, given a limited (training) sample of its arguments and values (examples). In particular, each training example consists of a vector of values of independent variables $\langle x_1, x_2, \ldots, x_n \rangle$ and the corresponding function value $f(\langle x_1, x_2, \ldots, x_n \rangle)$. The GP process starts with a population of random initial solutions to the problem (individuals). Each individual encodes a symbolic description of some function in a form of a tree. The terminal tree nodes return the values of the independent

variables $x_1, x_2, \ldots, x_n$ and constants. The non-terminal nodes implement fundamental arithmetic operators (like addition, subtraction, multiplication, and division) and elementary functions (e.g., exponential, logarithm, trigonometric functions). The fitness of particular individual is estimated by feeding it with the values of independent variables $\langle x_1, x_2, \ldots, x_n \rangle$, computing the 'response' of the tree (at its root), and measuring the discrepancy between that output and the desired value $f(\langle x_1, x_2, \ldots, x_n \rangle)$. These errors are computed for all the training examples and aggregated by means of root mean square error. The evolution maintains a population of solutions evaluated in this way. As in standard genetic algorithm, the selection operator promotes solutions with high fitness, the mutation operator introduces small modifications of solutions (e.g., by replacing randomly selected tree fragment by a new randomly generated subtree), and the crossover operator exchanges some genetic material between solutions (e.g., by exchanging randomly selected tree fragments between parent solutions). In [4], Koza has proven the ability of GP to find symbolic definitions of different polynomials.

The non-linear and non-fixed in length representation of GP solutions is much more flexible than the fixed-length representation used in most other EC branches. It also relieves the experimenter from estimating *a priori* the expected complexity of the solution. As a consequence, the GP paradigm proved extremely successful in many real-world applications and, in some cases, produced results superior to human-handcrafted solutions (see Chapter 1 of [4]).

Nevertheless, one of deficiencies of GP is the inability of code (subexpression) re-usage within a particular solution. Whenever there is a need of multiple reference to the same subexpression within a GP solution, it cannot be implemented in other way than by duplicating the desired code fragment. This feature of GP contributes negatively to the performance of the approach, resulting, among others, in so-called code-bloat, i.e. the tendency to produce overgrown solutions.

This is why we use a variant of GP which is in following referred to as *Genetic Graph Programming* (GGP, [6]). GGP shares most of its features with GP, except for the fact that solutions (programs) are represented as directed acyclic graphs (DAGs) rather than trees. A solution may have an arbitrary number of input (starting) nodes, but exactly one output node (sink) where it produces its response.

Functionally, the DAG representation is equivalent to expression trees, as any such DAG may be univocally converted into tree. However, such representation enables the solutions to reuse code fragments and, thus, it prevents (or at least reduces) the so-called code bloat and decreases the computation time. This is especially advantageous when the processing performed in particular nodes is time-consuming (e.g., image processing considered in this paper). The only price paid for this convenience is the increased complexity of recombination procedures.

The evolutionary operators that create (breed) the initial solutions, mutate solutions, and cross them over, have been defined in the following way. The solution breeding starts with random selection of n (currently 10) GGP operations.

Table 1. The complete list of GGP operations

Group	Operators
Filters	Convolve, GradientMagnitude, Invert, MaxFilter, MedianFilter, MinFilter, UnsharpMask
Arithmetic	AddConst, Add, Divide, Max, Min, Multiply, Subtract, SubtractConst
Functions	Absolute, Binarize, Exp, Log
Logic	And, Not, Or, Xor

Next, the individual is being built starting from the sink node, by adding randomly selected operations and connecting them to the partially built individual. After all n nodes have been added, the 'hanging' inputs of the individual are connected to the source of the input data. The crossover operator works as follows: given two parent individuals S_1 and S_2, it selects at random a node in S_1. Then, starting from that node, it marks (selects) a connected subgraph s_1 of random size in S_1, that is to be exchanged with analogous subgraph in S_2. For this purpose, we search in S_2 for a subgraph s_2 with precisely the same numbers of inputs and outputs (arity). If such s_2 is found, we exchange it with s_1. Otherwise, we retry the whole process. If no appropriate s_2 is found in 10 such trials, the crossover is cancelled and the offspring are copied from parents.

For mutation, the following four operators have been implemented. First operator simply deletes a randomly chosen node in the individual undergoing mutation. As removing of one node may imply removal of other nodes connected to it (to keep the individual consistent), we allow to remove only the direct successors of input nodes. The second mutation operator inserts a randomly selected node into the individual at a randomly selected location, where the output of the inserted node is being connected to its 'parent'. The remaining two mutation operators do not change the structure of the graph. One of those operators replaces the operation realized by randomly selected node by a randomly selected operation (however, the replacing operation must have the same arity as the replaced one). The last mutation operator changes a randomly selected parameter of a single operation, performing in fact only very smooth modification in individual's chromosome.

4 The Experiment

The general outline of GGP is quite universal. To adapt it to a particular real-world problem (object detection in our case), one has to define (i) the representation of input and output data, and (ii) the set of elementary operators (nodes) to be used. As far as the former issue is concerned, we use raster images as the representation of visual data that is accepted as input, processed, and produced by GGP solution. Thus, in its current form, the individual's acting is limited to image processing only. For instance, an individual is unable to compute global image features.

Fig. 1. The sample of the image data used in experiment (the upper row shows the complete training set, the lower row shows the selected frames from the testing set)

Consequently, we need appropriate operators (nodes) that perform the actual image processing. Each such node has exactly one output and one or more inputs (depending on the number of input arguments used by a particular operation). The operators used currently in our approach may be grouped according to their interpretation in terms of computer vision. These grouping is shown in Table 1. For detailed description of operators, see [13].

For an experimental verification, we choose the task of object detection in a video sequence. In particular, we use the tennis video sequence obtained from the Signal Analysis and Machine Perception Laboratory at Ohio State University [10]. This sequence shows two men playing table tennis and is composed of 150 color frames, each with dimensions 352×240 pixels, and color depth 24 bits per pixel. To provide reliable verification of our approach, from this complete set we selected training and testing subset of frames. The four frames presented in upper row of Fig. 1 constitute the complete training set. The testing set is composed of other 15 randomly selected frames; the lower row of Fig. 1 depict four of them. Note that the images exhibit significant variability as far as zoom and background contents are concerned. In particular, the tennis ball to be tracked appears at different sizes and is blurred in some frames.

Our task is to evolve GGP expressions that are able to localize the tennis ball in an image. This requires a specific interpretation of localization. First of all, we need to define the way in which an individual selects the pixels. As the GGP expressions operate on and produce raster images, this decision must be defined in pixel-oriented way too. For the purpose of this experiment, we interpret the pixels with brightness greater than 127 (on [0,255] scale) on all channels (Red, Green, and Blue) as being 'lit', i.e., selected by the GGP expression.

Given this interpretation of pixel selection, also have to provide an appropriate fitness function f. That function should:

1. *reward* an individual for selecting pixels located close to the ball, and
2. *penalize* an individual for selecting pixels distant from the ball.

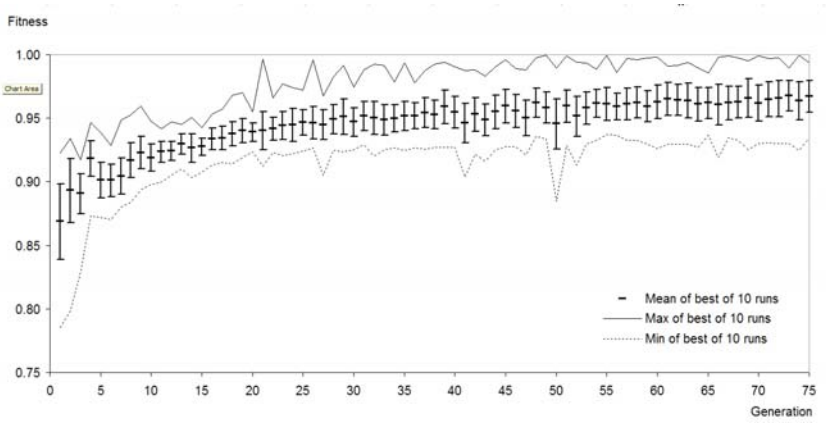


Fig. 2. Mean fitness graph of the evolutionary runs: mean, 0.95 confidence intervals (whiskers), minimum, and maximum

Note that these properties are neither equivalent nor complementary. A fitness function that realizes only rewarding would also erroneously promote individuals that select *all* pixels. Conversely, a fitness function that performs only penalization would encourage evolution of individuals that do not select *any* pixels at all. Thus, we need to take into account both these factors simultaneously.

Formally the fitness $f(S)$ ($f \in [0,1]$) of an individual S is been defined as:

$$f(S) = \frac{1}{2|I|} \sum_{i=1}^{|I|} \left(\frac{t_i}{p_i} + \frac{n_i - f_i}{n_i} \right) \tag{1}$$

In Formula (1), I denotes the set of training images and i is the image index. Next, t_i is the number of all pixels that are localized within the target object (tennis ball) and lit (true positives), f_i refers to the number of all pixels that are *not* localized within the target object and lit (false positives), p_i refers to the number of all pixels contained in target object (max positives), and n_i refers to all remining pixels in frame $\#i$ (max negatives). Note that this form of the fitness function makes it equally sensitive to false positive and false negative errors.

For the purpose of the experiment, we designed a software framework based on two libraries: Evolutionary Computation in Java (ECJ, [6]) and Java Advanced Imaging (JAI, [13]). The most relevant parameters of evolutionary algorithm are set as follows: algorithm type: generational, number of generations: 75, population size: 100. To populate the consecutive generations, we designed a 'breeding pipeline' according to ECJ's standards. In this pipeline, parent individuals undergo recombination (crossover) and their offspring is transferred to the next generation with probability 0.9. Alternatively, the parent solutions are transferred into the next generation without undergoing any changes; this happens with probability 0.1. In the reproduction stage, we apply tournament selection with pool size 7. In the recombination phase, we perform crossover and mutation using operators described in Section 3.

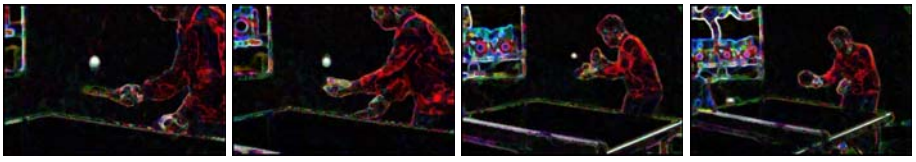


Fig. 3. The output images produceds by one of the best individuals (run #1) for the training set

To provide for statistical significance of the obtained results, we repeat the evolutionary run 10 times, starting with different initial populations. Each such run lasted for approx. 3 hours. Figure 2 shows the fitness graph reflecting the average dynamics of all 10 runs. This graph has been obtained by averaging the fitness of best individual of each generation over 10 runs. The whiskers show the 0.95-confidence intervals for the mean value, and the two accompanying series depict the maximum and minimum fitness of best individuals in all 10 runs. The leveling-off observed in these series clearly indicates that the number of generations (75) is sufficient. Further significant improvement of obtained solutions is highly unlikely.

The resulting aggregated performances of*best individuals* found in all 10 evolutionary runs are shown in Table 2. On the average, the learning process produces solutions that perform almost perfectly on the training data. On the testing set, we observe a significant reduction of performance (overfitting). This effect could be reduced by using more frames in the training set; this, however, would make the experiments last much longer. Nevertheless, taking into account how much different are our testing frames from the training ones (see Fig. 1), this performance should be considered good.

For comparison, the last row of Table 2 displays the fitness of the raw input image (or, in other words, the fitness of an 'identity' individual that produces a copy of input at its output). Note also that our fitness should not be interpreted as accuracy of tracking or as percentage of 'hits' (see Formula (1)).

Figure 3 shows the processing performed by one of the best individuals found in evolutionary runs for training images. It may be clearly seen that the individual was able to effectively get rid of the bright pixels representing objects other than the tennis ball. In particular, this applies also to the large white poster in the back of the scene.

Figure 4 shows the progress of two selected evolutionary runs in terms of image processing. The two left-hand columns show the processing performed in run #1,

Table 2. Aggregate performances computed based on best solution found in each run

	Training set	Testing set
Average fitness over all runs	0.9770±0.0133	0.8096±0.0806
Maximum fitness over all runs	0.9997	0.9487
Fitness of raw input image	0.7770	0.7500

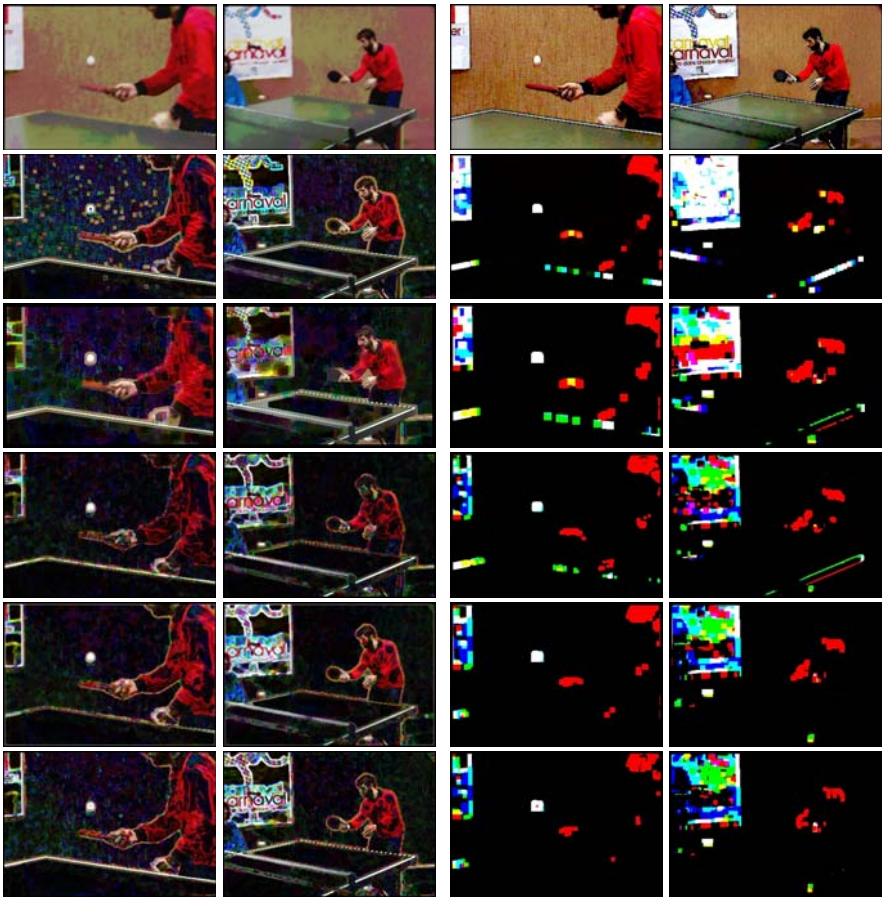


Fig. 4. The progress of best solutions of evolutionary run #1 (two left-hand columns) andrun #8 (two right-hand columns)

whereas the two right-hand columns show the processing performed in run #8. Within each run, each column corresponds to a selected input image (there are two of them) and shows, in consecutive rows, the output produced for that image for the best individual of generation 1, 15, 30, 45, 60, and 75. In initial generation, the best solutions are apparently straightforward as they do not change much the image contents. Then, the performance of the best individual is improved by gradually getting rid of undesired pixels. Note also that for the two presented runs, the best performing solutions follow significantly different strategy of image processing to select the tennis ball.

In Figure 5, we present the code of the best GP individual together with the processing it performs on one of the input images. Note that the result produced by the code fragment composed of operations #1, 2, and 3 is used twice in the following operations. This fact clearly demonstrates that the option of code

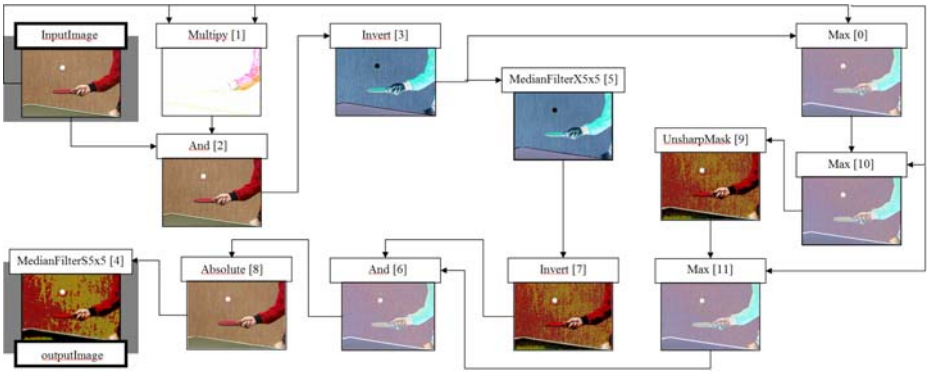


Fig. 5. The best individual found during one of evolutionary runs

re-usage offered by GPP is advantageous and reduces the complexity of evolved solutions.

5 Conclusions

In this paper, we presented a novel framework for GP-based image processing and interpretation. The major feature of the proposed approach is the use of advanced graph-based genetic programming to successfully recognize objects in real-world views of 3D scenes. The obtained results are encouraging, taking into account that the temporal nature of video sequence has not been taken into account – the ball is being located in each frame independently. Providing the method with extra information on estimated object locations in preceding frames would probably increase the performance of the approach. Note also that the proposed approach may be easily adapted to learn image processing procedures for other kinds of visual tasks, like, e.g., image enhancement or object recognition.

Acknowledgement

This work has been supported by KBN research grant 3 T11C 050 26.

References

1. Bhanu, B., Lin, Y., Krawiec, K. (2005) *Evolutionary Synthesis of Pattern Recognition Systems*. SpringerVerlag, New York.
2. Draper, B., Hanson, A., Riseman, E. (1993) "Learning blackboard-based scheduling algorithms for computer vision," International Journal of Pattern Recognition and Artificial Intelligence, vol. 7, 309-328, March.
3. Johnson, M.P., Maes, P., Darrell, T. (1994) "Evolving visual routines," in: R.A. Brooks, P. Maes (red.) Artificial Life IV: proceedings of the fourth international workshop on the synthesis and simulation of living systems, Cambridge, MA: MIT Press, 373-390.

4. Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G. (2003) Genetic Programming IV: Routine Human-Competitive Machine Intelligence. Boston, MA: Kluwer Academic Publishers.
5. Krawiec, K., Bhanu, B. Visual Learning by Coevolutionary Feature Synthesis, *IEEE Trans. on Systems, Man, and Cybernetics*, Part B, Vol. 35, Issue 3, June 2005, pp. 409-425.
6. Lijewski, P., *Automatic Decomposition of the Problem Representation in Coevolutionary Algorithms*, Master's Thesis, Poznań University of Technology, Poznań, Poland, 2005.
7. Luke, S. (2002) "ECJ Evolutionary Computation System," http://www.cs.umd.edu/projects/plus/ec/ecj/.
8. Maloof, M.A., Langley, P., Binford, T.O., Nevatia, R., Sage, S. (2003) "Improved rooftop detection in aerial images with machine learning," Machine Learning, vol. 53, 157-191.
9. Marek, A., Smart, W.D., and Martin, M.C. (2003) "Learning Visual Feature Detectors for Obstacle Avoidance using Genetic Programming," In Proceedings of the IEEE Workshop on Learning in Computer Vision and Pattern Recognition, Madison, WI.
10. Ohio State University repository of motion imagery repository, 2004, http://sampl.ece.ohio-state.edu/data/motion/tennis/index.htm.
11. Rizki, M., Zmuda, M., Tamburino, L. (2002) "Evolving pattern recognition systems," IEEE Transactions on Evolutionary Computation., vol. 6, 594-609.
12. Segen, J. (1994) "GEST: A learning computer vision system that recognizes hand gestures," In: R.S. Michalski and G. Tecuci (ed.) Machine learning. A Multistrategy Approach. Volume IV, San Francisco, CA: Morgan Kaufmann, 621-634.
13. Sun Microsystems, Inc., (2001) "Java Advanced Imaging API Specification", Version 1.2.
14. Teller, A., Veloso, M.M. (1997) "PADO: A new learning architecture for object recognition," In: K. Ikeuchi and M. Veloso (ed.) Symbolic Visual Learning, Oxford Press, 77-112.

Selective Motion Analysis Based on Dynamic Visual Saliency Map Model

Inwon Lee[1], Sang-Woo Ban[2], Kunihiko Fukushima[3], and Minho Lee[1]

[1] School of Electrical Engineering and Computer Science,
Kyungpook National University, 1370 Sankyuk-Dong, Puk-Gu, Taegu 702-701, Korea
[2] Dept. of Information and Communication Engineering,
Dongguk University, 707 Seokjang-Dong, Gyeongju, Gyeongbuk 780-714, Korea
[3] Graduate School of Informatics,
Kansai University, 2-1-1 Ryozenji, Takatsuki, Osaka 569–1095, Japan
mholee@knu.ac.kr
http://abr.knu.ac.kr

Abstract. We propose a biologically motivated motion analysis model using a dynamic bottom-up saliency map model and a neural network for motion analysis of which the input is an optical flow. The dynamic bottom-up saliency map model can generate a human-like visual scan path by considering dynamics of continuous input scenes as well as saliency of the primitive features of a static input scene. Neural network for motion analysis responds selectively to rotation, expansion, contraction and planar motion of the optical flow in a selected area. The experimental results show that the proposed model can generate effective motion analysis results for analyzing only an interesting area instead of considering the whole input scenes, which makes faster analysis mechanism for dynamic input scenes.

1 Introduction

The optical flow has been popularly used as one of ways to analysis of dynamic information of an input scene, which can represent rotation, expansion and contraction and is useful for tracking and navigation [1].

In our brain, there are many middle superior temporal (MST)-cells that responds selectively to global features of optical flow, such as rotation, expansion, contraction and planar motion, which covers a large area of the visual field [2,3]. Various kinds of MST models and hypotheses can be categorized as two major streams of the hypotheses of MST-cell: the direction mosaic hypothesis and the vector-field hypothesis [2,3]. There are some computational models where neural networks are self-organized by learning [3]. In most of the models, however, the cells in hidden layers show a behavior similar to that of the direction mosaic hypothesis. In recent, Fukushima et. al. proposed a neural network model for analyzing optical flow based on the vector-field hypothesis [4]. The outputs of the neural network describe the motion information, but it analyzes the whole

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 814–822, 2006.
© Springer-Verlag Berlin Heidelberg 2006

area of input scene and resultantly the computational complexity is so high that the model can not be applied for real application. Moreover, the model is based on an assumption for retina-topic optical flow signal.

On the other hand, there is strong evidence that a specialized processing of object motion takes place in the dorsal area including the V5 and the middle temporal (MT) area. The lateral intra-parietal cortex (LIP), which is essential organ for attention integrator, receives motion information from the V5 together with static information from the V4 in the ventral area [5]. Thus, the motion analysis is highly correlated with selective attention function. If we consider the selective attention mechanism in motion analysis, we can develop more efficient method for analyzing a dynamic input scene. The human visual system can effortlessly detect an interesting area or object within natural or cluttered scenes through the selective attention mechanism. The selective attention mechanism allows the human vision system to process input visual scenes more effectively with a higher level of complexity. The human visual system sequentially interprets not only a static input image but also a dynamic input scene with the selective attention mechanism.

In this paper, we propose a selective motion analysis model using a biologically motivated dynamic bottom-up saliency map model in conjunction with a neural network for motion analysis. Our proposed model is based on understanding of human visual pathway in the brain. Also, we use a maximum entropy approach to obtain a dynamic saliency map with scale information from successive static saliency maps.

2 The Proposed Model

Fig. 1 shows the procedure to get the proposed selective motion analysis using the dynamic bottom-up saliency map (SM) model. The dynamic bottom-up SM model imitates the roles of human visual brain from retina to MT and MST through the lateral geniculate nucleus (LGN), V1 and V2 including the LIP [6].

After the selective attention region is decided by the SM model, the selected area is processed as the low-resolution intensity with Guassian filter which mimics the foveated retina image. Then, the optical flow is obtained by successive image frames and used for the neural network model for motion analysis proposed by Fukushima [4]. The proposed model can generate effective motion analysis considering only an interesting area. The following subsections describe the dynamic bottom-up saliency map (SM) model and the selective motion analysis.

2.1 The Dynamic Bottom-Up Saliency Map Model

As shown in Fig. 1, in order to model the human-like visual attention mechanism for a static input scene, we use the three bases of edge, intensity, and color information, for which the roles of the retina cells and the LGN. The feature maps ($\bar{I}, \bar{O}$ and $\bar{C}$) are constructed by center surround difference and normalization (CSD& N) of three bases, which mimics the on-center and off-surround mechanism in our brain. We modified and enhanced our previous bottom-up saliency

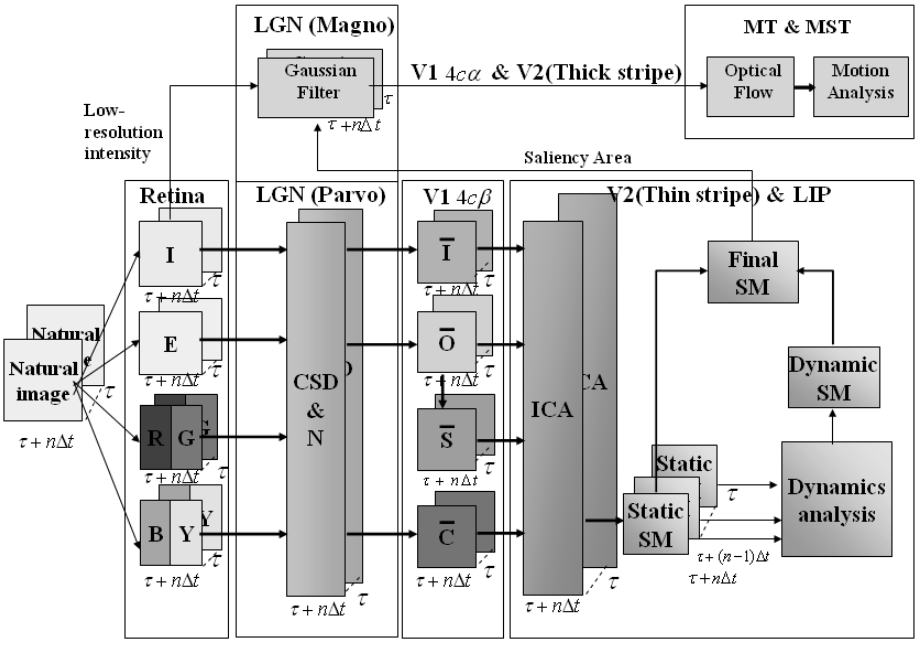


Fig. 1. The proposed selective motion analysis model (I: intensity image, E:edge image, S: symmetry image, RG: red-green opponent coding image, BY: blue-yellow opponent coding image, CSD & N: center-surround difference and normalization, $\bar{I}$: intensity feature map, $\bar{O}$: orientation feature map, $\bar{S}$: symmetry feature map, $\bar{C}$: color feature map, ICA: independent component analysis, LIP: lateral intra-parietal cortex, Static SM: static saliency map, Dynamic SM: dynamic saliency map)

map model by considering the orientation feature based on the edge feature and the symmetry feature based on the orientation feature, which seems more plausible in mimicking the biological mechanism more correctly. Among the feature maps, the orientation feature map is generated using orientation feature that is extracted using Gabor filters from edge feature, which mimics the orientation selective activation of the simple cells in the V1. Moreover, the symmetry feature map is constructed from symmetry features obtained by a noise tolerant generalized symmetry transformation (NTGST) algorithm from orientation feature, which mimics the higher-order analysis of complex cells and hyper-complex cells in the V1. The constructed four feature maps ($\bar{I}, \bar{O}, \bar{S}$ and $\bar{C}$) are then integrated by an independent component analysis (ICA) algorithm based on maximization of entropy [6].

Barlow's hypothesis is that human visual cortical feature detectors might be the end result of a redundancy reduction process [7], and Sejnowski's and Lee's results are that the ICA is the best way to reduce redundancy [8]. After the convolution between the channel of the feature maps and the filters is obtained by ICA learning, the saliency map is computed by integrating all feature maps for every location [6]. The LIP plays a role to provide a retinotopic spatio-feature

map that is used to control the spatial focus of attention and fixation, which is able to integrate feature information in its spatial map [9]. As an integrator of spatial and feature information, the LIP provides the inhibition of return (IOR) mechanism required here to prevent the scan path returning to previously inspected sites [9,8].

Fig. 1 shows the procedure to get a dynamic SM from natural input images. The human visual system sequentially interprets not only a still input image but also dynamic input scenes. A conventional bottom-up selective attention model, however, considers only static scenes. All selective attention models, including our previous model [6], considered only a static scene. Human beings, however, decide what constitutes an interesting area within a dynamic scene, as well as static images. The proposed dynamic SM model is based on successive static SMs. The entropy maximum is considered to analyze the dynamics of the successive static SMs, which is an extension of Kadir's approach [10] since the proposed model considers time varying property as well as spatial features. Fig. 1 shows the procedure to get a final SM by integrating both the static SM and the dynamic SM from natural input images. For the first frame at time τ, the most appropriate scale $\mathbf{x_s}$ for each area centered at location $\mathbf{x}$ is obtained by Eq. (1) which aims to consider spatial dynamics at each location:

$$\mathbf{x_s} = argmax_s\{H_D(s, \mathbf{x}, \tau) \times W_D(s, \mathbf{x}, \tau)\} \tag{1}$$

where D is the set of all descriptor values which consist of the intensity values corresponding the histogram distribution in local region s around an attended location $\mathbf{x}$ in a static SM at time τ, and $H_D(s, \mathbf{x}, \tau)$ is the entropy defined by Eq. (2) and $W_D(s, \mathbf{x}, \tau)$ is the inter-scale measure defined by Eq. (3):

$$H_D(s, \mathbf{x}, \tau) \equiv -\sum_{d \in D} p_{d,s,\mathbf{x},\tau} log_2 p_{d,s,\mathbf{x},\tau} \tag{2}$$

$$W_D(s, \mathbf{x}, \tau) \equiv \frac{s^2}{2s-1} \sum_{d \in D} \mid p_{d,s,\mathbf{x},\tau} - p_{d,s-1,\mathbf{x},\tau} \mid \tag{3}$$

where $p_{d,s,\mathbf{x},\tau}$ is the probability mass function obtained from the histogram of pixel values at time τ for scale s, position $\mathbf{x}$, and the descriptor value d which takes on values in D.

The probability mass function for a dynamic scene $p_{H_D(x_s,x,\tau+k\triangle t)}$ is obtained from the histogram of the entropy values for a sequence of the static SMs in $(k+1)$ frames from τ to $\tau + k\triangle t$ where k is the number of continuous frames and $\triangle t$ denotes a sampling time. The entropy value at location $\mathbf{x}$ is calculated using the histogram of pixel values of the local area centered at $\mathbf{x}$ with size $\mathbf{x_s}$ in a static SM. Using the probability mass function for a dynamic scene $p_{H_D(x_s,x,\tau+k\triangle t)}$, the time varying entropy $T_D(\cdot)$ is calculated by Eq. (4):

$$T_D(\mathbf{x_s}, \mathbf{x}, \tau + n\triangle t) \equiv -\sum_{d \in D} p_{H_D(x_s,x,\tau+n\triangle t)} log_2 p_{H_D(x_s,x,\tau+n\triangle t)} \tag{4}$$

The entropy value $T_D(\cdot)$, at each pixel $\mathbf{x}$, represents a fluctuation of visual information according to time, through which the proposed model can generate a dynamic SM. Finally, the proposed attention model decides the attention areas based on a final SM which is generated by integration of the static SM and the dynamic SM. Therefore, the proposed dynamic bottom-up attention model can selectively decide an attention area by considering not only static saliency but also dynamic feature information obtained from consecutive input scenes.

2.2 The Selective Motion Analysis Model

Fig. 2 shows the block diagram of the selective motion analysis. The dynamic SM model decides on the attention area, and the successive input images with low resolution intensity are used as input data to calculate the optical flow that is obtained by Horn & Schunck model [11]. The calculated optical flow is used as input for the feed-forward neural network in the Fukushima's model in order to analyze motion information [4]. The Fukushima's previous model is applied for analyzing motion information. However, the V1-cells' response is calculated as the optical flow in proportion to the speed of the stimuli with their receptive fields. As shown in Fig. 2, the MT_{abs}-cells extract absolute-velocity of stimuli. The MT_{abs}-cells consist of two sub-layers, namely exc-cell and inh-cell. Only the receptive-field sizes are different between them, that is, the receptive-field of an inh-cell is larger than that of an exc-cell. Receiving antagonistic signals from exc- and inh-cell of MT_{abs}-cells, MT_{rel}-cells extract relative velocity of the stimuli. Finally, MST-cells integrates responses of many MT_{rel}-cells simply by summation. Therefore by collecting signals from MT_{rel}-cells, MST-cells extract the global clockwise rotation, counter-clockwise rotation, expansion and contraction of optical flow [4].

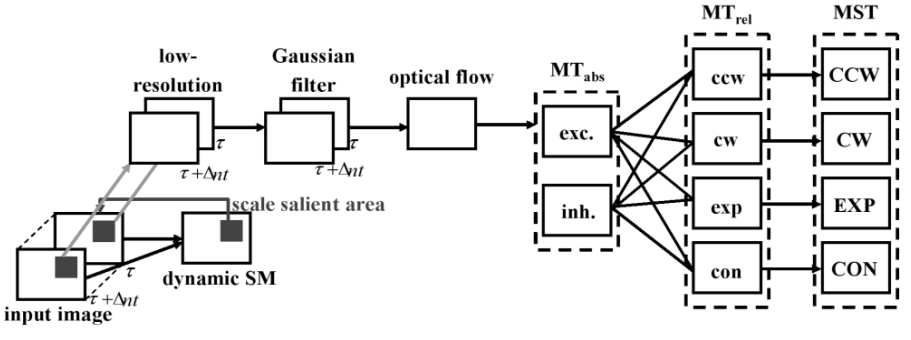


Fig. 2. A block diagram of the proposed selective motion analysis model (exc.: exc-cell, inh.: inh-cell, MT_{abs}: absolute-velocity of the stimuli, MT_{rel}: relative-velocity of the stimuli, ccw: counter-clockwise, cw: clockwise, exp: expansion, con: contraction, CCW: global counter-clockwise, CW: global clockwise, EXP: global expansion, CON: global contraction)

3 Experiments

Figs. 3 (a), (b), (c) and (g) show the scan path in static images with scale information at successive time with sampling time as 0.1sec, respectively. Scale information in each salient area is represented by a scaled box on the corresponding salient area in the figures. The numbers in each figure show the sequence of scan path generated by the static SMs. Figs. 3 (d), (e), (f) and (j) show the successive static SMs for the first, second, the third, and the fourth frame of a sequence of frames from time to time, respectively.

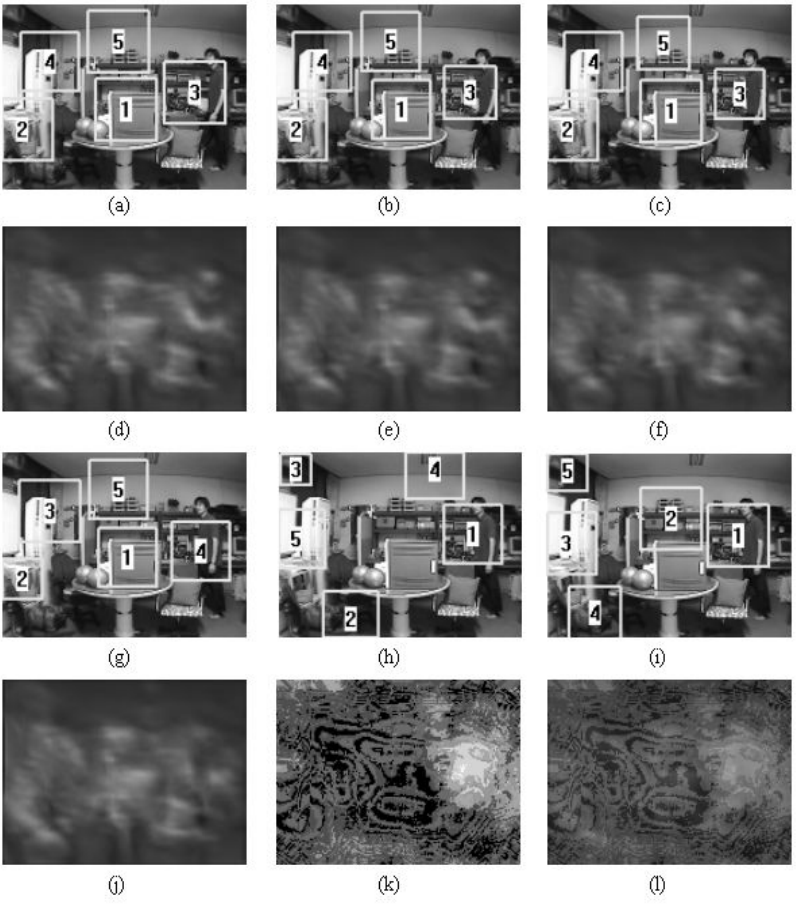


Fig. 3. Comparison the dynamic SM with static SM: (a) and (d) Scan path in static image and static SM for the first frame at time τ. (b) and (e) Scan path in static image and static SM at time $\tau + \triangle t$. (c) and (f) Scan path in static image and static SM at time $\tau + 2\triangle t$. (g) and (j) Scan path in static image and static SM at time $\tau + 3\triangle t$. (h) and (k) Scan path in dynamic image and dynamic SM from time τ to time $\tau + 3\triangle t$. (i) and (l) Scan path in dynamic image and integrated SM from time τ to time $\tau + 3\triangle t$.

Although the person moves slowly in the left and forward direction, the scan paths generated by the static SM are same with each other. Fig. 3 (h) shows the scan path in dynamic situation which is obtained from the successive static SMs shown in Figs. 3 (a), (b), (c) and (g). Fig. 3 (k) shows the dynamic SM corresponding to Fig. 3 (h). The scan path generated by the dynamic SM focuses on the moving person instead of paying attention to a dominant object in each static image. Finally, Fig. 3 (i) shows the final scan path generated by integrating the static SM and the dynamic SM. Fig. 3 (l) shows the scan path obtained from the final SM. As shown in Fig. 3, the scan path generated by the proposed dynamic model can pay attention to a dynamic area. Thus, for the same frame at time, the static bottom-up attention model can selectively pay attention to the salient areas shown in Fig. 3 (g). However, the proposed dynamic bottom-up attention model can dynamically decide more plausible attention areas as shown in Fig. 3 (l) than those in Fig. 3 (g). If the human vision system were a static system, the attention result shown in Fig. 3 (g) is reasonable. However, the human vision system contains dynamics as well as static characteristics. Therefore, the proposed model can generate a more reasonable attention result by focusing a dynamic area as well as static salient regions.

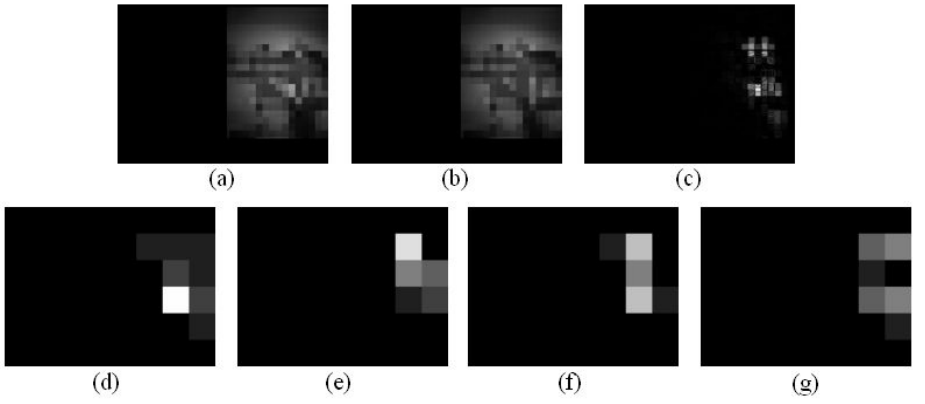


Fig. 4. The results of proposed selective motion analysis model: (a) Gaussian image after down-sampling for the first frame at time τ (b) Gaussian image after down-sampling for the fifth frame at time $\tau + 3\triangle t$. (c) Magnitude of optical flow between (a) and (b). (d) Counter-clockwise rotation. (e) Clockwise rotation. (f) Expansion. (g) Contraction.

After we get the scan path by the proposed dynamic SM, we obtain the gaussian filtered intensity image in which the gaussian function has the center value around in the first salient area. Figs. 4 (a) and (b) show the gaussian smoothed input images at τ and $\tau + 3\triangle t$, respectively. Fig. 4 (c) shows the optical flow obtained from Figs. 4 (a) and (b). Figs. 4 (d), (e), (f) and (g) show the motion analysis results for 4 directions such as counter-clockwise rotation, clockwise

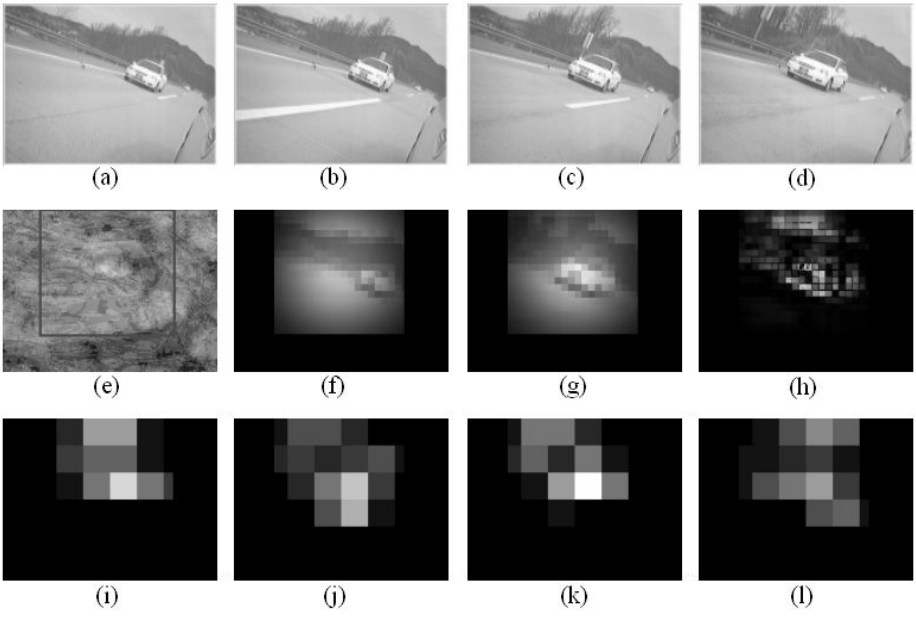


Fig. 5. Motion analysis result for blind spot monitoring in a car driving: (a), (b), (c) and (d) Input images at time τ, $\tau + \triangle t$, $\tau + 2\triangle t$ and $\tau + 3\triangle t$. (e) Dynamic SM result. (f) and (g) Gaussian image image after down-sampling at time τ and at time $\tau + 3\triangle$. (h) Magnitude of optical flow between (f) and (g). (i) Counter-clockwise rotation. (j) Clockwise rotation. (k) Expansion. (l) Contraction.

rotation, expansion and contraction. From the motion analysis result, we can understand that the people in the image move to left and forward direction.

We apply the proposed method to implement an automotive detector for blind spot monitoring in a car driving. Figs. 5 (a), (b), (c) and (d) show the successive images obtained from time τ to time $\tau + 3\triangle t$. Fig. 5 (e) show the dynamic SM for the successive input images, and Fig. 5 (f) and (g) show the gaussian images after down-sampling at time τ and at time $\tau + 3\triangle t$, respectively. Fig. 5 (h) shows the magnitude of optical flow. Figs. 5 (i), (j), (k) and (l) show the motion analysis results for 4 directions. From the motion analysis result, we can understand that the car in the image is approaching.

4 Conclusions

Computer simulation results showed that the proposed model can generate effective motion analysis results for analyzing only an interesting area instead of considering the whole input scenes, which makes faster analysis mechanism for dynamic input scenes. The proposed model can play an important role for motion detection or tracking process in a human-like robot system.

Acknowledgement

This research was funded by the Brain Neuroinformatics Research Program of the Ministry of Commerce, Industry and Energy and Components and Materials Technology Development Program, VISS(Vision-based Intelligent Steering System) Project by MANDO's sub-project called IPAS(Intelligent Parking Assistant System).

References

1. Giboson, J. J.: The perception of the visual world. MA, Boston, Houghton Mifflin (1950)
2. Duffy, C. J., Wurtz, R.H.: Sensitivity of MST neurons to optic flow stimuli. I. A continuum of response selectivity to large-field stimuli. Journal of Neurophysiology **65(6)** (1991) 1329–1345
3. Duffy, C. J., Wurtz, R.H.: Sensitivity of MST neurons to optic flow stimuli. II. Mechanisms of response selectivity revealed by small-field stimuli. Journal of Neurophysiology **65(6)** (1991) 1346–1359
4. Kazuya T., Fukushima K.: Neural network model for extracting optic flow. Neural Networks **18(5-6)** (2005) 1–8
5. Taylor, N.R., Hartley M., Taylor, J.G.: Coding of objects in low-level visual cortical area. Lecture Notes in Computer Science **3696** (2005) 57–63
6. Park, S.J., An, K.H., Lee, M.: Saliency map model with adaptive masking based on independent component analysis. Neurocomputing **49** (2002) 417–422
7. Barlow, H.B., Tolhust, D.J.: Why do you have edge detection ?. Optical Society of America Technical Digest **23** (1992) 172
8. Bell, J., Sejnowski, T.J.: The independent components of natural scenes are edge filters. Vision Research **37** (1997) 3327–3338
9. Lanyon, L.J., Denham, S.L.: A model of active visual search with object-based attention guiding scan paths. Neural Networks **17(5-6)** (2004) 873–897
10. Kadir, T., Brady, M.: Scale, saliency and image description. International Journal of Computer Vision **45(2)** (1996) 83–105
11. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence **17** (1981) 185–203

Efficient Ant Reinforcement Learning Using Replacing Eligibility Traces

SeungGwan Lee[1] and SeokMi Hong[2]

[1] School of Computer Science and Information Engineering, Catholic University
43-1, Yeokgok 2-Dong, Wonmi-Gu, Bucheon-Si, Gyeonggi-Do, 420-743, Korea
leesg@catholic.ac.kr
[2] School of Computer Information and Communication Engineering,
Sangji University
660 USan-Dong, WonJu-Si, KangWon-Do, 220-702, Korea
smhong@yahoo.co.kr

Abstract. The eligibility trace is one of the basic mechanisms in reinforcement learning to handle delayed reward. The traces are said to indicate the degree to which each state is eligible for undergoing learning changes should a reinforcing event occur. Formally, there are two kinds of eligibility traces(accumulating trace or replacing traces). In this paper, we propose an ant reinforcement learning algorithms using an eligibility traces which is called replace-trace methods(Ant-TD(λ)). This method is a hybrid of Ant-Q and eligibility traces. With replacing traces, the eligibility trace for the maximum($MaxAQ(s, z)$) state visited on the step is reset to 1 and the eligibility traces for another states decay by $\gamma\lambda$. Although replacing traces are only slightly different from accumulating traces, it can produce a significant improvement in optimization. We could know through an experiment that proposed reinforcement learning method converges faster to optimal solution than ACS and Ant-Q.

1 Introduction

Reinforcement Learning is learning by interaction as agent achieves learning doing interaction by trial-and-error. Agent attempts an action that can be taken in the given state, and changes its state and receive reward value for the action.

In the popular TD(λ) algorithm, the λ refers to the use of an eligibility traces. Almost any temporal-difference(TD) method, such as Q-learning can be combined with an eligibility traces to obtain a more general method that may learn more efficiently.

An eligibility trace is a temporary record of the occurrence of an event, such as the visiting of a state or the taking of an action. The trace marks the parameters associated with the event as eligible for undergoing learning changes. When a TD-error occurs, only the eligible states or actions are assigned credit or penalty for the error. Thus, eligibility traces help bridge the gap between events and training information, and a basic mechanism for temporal credit assignment.

In this paper we introduce Ant-Q algorithm [7], [8] for combinatorial optimization has been introduced by Colorni, Dorigo and Maniezzo, and we propose

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 823–832, 2006.
© Springer-Verlag Berlin Heidelberg 2006

new ant reinforcement learning algorithms using an eligibility traces are often called replace-trace methods(Ant-TD(λ)) to Ant-Q learning.

Proposed Ant-TD(λ) reinforcement learning method searches goal using TD-error [3], [4], [5] and an eligibility traces [13], there is characteristic that converge faster to optimal solution than original ACS [11], [12] and Ant-Q.

The rest of the paper is organized as follows. In section2, we introduce Ant-Q and Ant-TD reinforcement learning. Section3 describes new ant reinforcement learning algorithms using eligibility traces are called replace-trace methods (Ant-TD(λ)). Section 4 presents experimental results. Finally, Section 5 concludes the paper and describes directions for future work.

2 Related Work

2.1 Ant-Q

Ant-Q learning method [7], [8] that is proposed by Coloni, Dorigo and Mauiezzo is an extension of Ant System(AS) [1], [2], [6], [9], [10], it is reinforcement learning reinterpreting in view of Q-learning.

In Ant-Q, an agent(k) situated in node(r) moves to node(s) using the follow rule, called pseudo-random proportional action choice rule(or state transition rule):

$$s = \begin{cases} \arg\max\limits_{u \in J_k(r)} \left\{ [AQ(r,u)]^{\delta} \cdot [HE(r,u)]^{\beta} \right\} & if\ q \le q_0\ (exploitation) \\ S & otherwise\ (exploration) \end{cases} \quad (1)$$

$AQ(r,u)$ is Ant-Q value, be a positive real value associated to the edge(r,u), It is counterpart of Q-learning Q-values, and is intended to indicate how useful it is to move node(u) when in node(r).

$AQ(r,u)$ is changed at run time. $HE(r,u)$ is a heuristic value associated to edge(r,u) which allows an heuristic evaluation of which moves are better(in the TSP, the inverse of the distance).

Let k be an agent making a tour. $J_k(r)$ are nodes to be visited from the current node(r). Let initial parameter is set to $AQ(r,u)$=AQ_0= 1/(average length of edges·n).

Where δ and β is parameters which weigh the relative importance of the learned AQ-values and the heuristic values. q is a value chosen randomly with uniform probability in [0,1], $q_0 (0 \le q_0 \le 1)$ is a parameter, and S is a random variable selected according to the distribution given by Eq.(2) which gives the probability with which an agent in node(r) choose the node(s) to move to.

$$p_k(r,s) = \begin{cases} \frac{[AQ(r,s)]^{\delta} \cdot [HE(r,s)]^{\beta}}{\sum_{u \in J_k(r)} [AQ(r,u)]^{\delta} \cdot [HE(r,u)]^{\beta}} & if\ s \in J_k(r) \\ 0 & otherwise \end{cases} \quad (2)$$

The goal of Ant-Q is to learn AQ-values to find better solution as stochastic. AQ-values are updated by the following Eq.(3).

$$AQ(r,s) \leftarrow (1-\alpha) \cdot AQ(r,s) + \alpha \cdot (\Delta AQ(r,s) + \gamma \cdot \underset{z \in J_k(s)}{Max}\ AQ(s,z)) \quad (3)$$

$\alpha(0<\alpha<1)$ is the pheromone decay parameter and γ is discount rate. $MaxAQ(s,z)$ is maximum reinforcement value that receive from external environment, global reinforcement is zero.

Also, ΔAQ is reinforcement value, local reinforcement is always zero, while global reinforcement, which is given after all the agents have finished their tour, is computed by the following Eq.(4).

$$\Delta AQ(r,s) = \begin{cases} \frac{W}{L_{k_{ib}}} & if(r,s) \in tour\ done\ by\ the\ agent\ k_{ib} \\ 0 & otherwise \end{cases} \quad (4)$$

Where L_{kib} is the length of the tour done by the best agent which did the shortest tour in the current iteration, and W is a parameter, set to 10. Fig.1 shows the Ant-Q algorithm in pseudocode.

```
/* Initialization phase */
   Set an initial value for AQ-values, each pair AQ(r,s)=AQ_0.
/* Main algorithm */
Loop
   1. /* Initialization of agents data structures */
      Choose a starting node for agents
   2. /* In this step agents build tours and locally update AQ-values*/
      Each agent applies the state transition rule Eq.(1) to choose the
      node to go to, updates the set J_k and applies Eq.(3) to locally
      update AQ-values (in Eq.(3) ΔAQ(r,s)=0)
   3. /* In this step agents globally update AQ-values */
      The edges belonging to the tour done by the best agent are updated
      using Eq.(3) where ΔAQ(r,s) is given by Eq.(4)
Until (End condition = True)
```

Fig. 1. Pseudocode of Ant-Q Algorithm

2.2 Ant-TD

To solve temporal-credit assignment problems, reinforcement learning that uses TD can be a method to learn prediction.

Traditional learning methods such as supervised learning wait until final result happens to keep recording for prediction for all outputs that were calculated in intergrade. And then, as training error, we use difference between prediction value for output of each state and final result.

But, the reinforcement learning using TD without waits final result. At each learning step, training error uses difference with prediction for output of present state and prediction for output of next state and this is known as TD-error (Temporal Difference error).

The prediction for output of present state is updated to approximate with prediction for output of next state in TD-learning. Thus, it is known that reinforcement learning of simple form that uses difference of prediction values for output of each state is TD-learning.

TD-learning that use TD-error calculates Q-function value of present state with Eq.(5) using difference with prediction for output of present state and prediction for output of next state.

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot Q(s_t, a_t) + \alpha \cdot TD\,error \tag{5}$$

Here, α is learning rate, TD-error calculates with Eq.(6) as difference with prediction of present state and prediction of next state.

$$TDerror = r_{t+1} + \gamma[\underset{a \in A(s_t)}{Max}\, Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{6}$$

r_t is reinforcement value and γ is discount rate. The goal of Ant-Q is to learn AQ-values to find better solution as stochastic.

AQ-values are updated by the following Eq.(7). Appling Eq.(6) to Ant-Q, it is expressed with Eq.(8).

$$AQ(r,s) \leftarrow (1-\alpha) \cdot AQ(r,s) + \alpha \cdot (\Delta AQ(r,s) + \gamma \cdot \underset{z \in J_k(s)}{Max}\, AQ(s,z)) \tag{7}$$

$$TDerror = \Delta AQ(r,s) + \gamma[\underset{z \in J_k(s)}{Max}\, AQ(s,z) - AQ(r,s)] \tag{8}$$

Finally, an ant reinforcement learning model that apply TD-error in Ant-Q calculates Q-function value for node(r,s) of present state with Eq.(9)[14].

$$\begin{aligned} &AQ(r,s) \leftarrow (1-\alpha) \cdot AQ(r,s) \\ &\qquad +\alpha \cdot (\Delta AQ(r,s) \;+ \gamma \cdot [\underset{z \in J_k(s)}{Max}\, AQ(s,z) - AQ(r,s)]) \\ &where \quad \Delta AQ(r,s) = 0 \qquad\qquad if\ \ Local\ \ updating \\ &\qquad \underset{z \in J_k(s)}{Max}\, AQ(s,z) - AQ(r,s) = 0 \qquad if\ \ Global\ \ updating \end{aligned} \tag{9}$$

$\alpha(0<\alpha<1)$ is the pheromone decay parameter and γ is discount rate. $MaxAQ(s,z)$ is maximum reinforcement value that is received from external environment, global reinforcement is zero. Also, ΔAQ is reinforcement value, local reinforcement is always zero, while global reinforcement, which is given after all the agents have finished their tour, is computed by the following Eq.(10).

$$\Delta AQ(r,s) = \begin{cases} \frac{W}{L_{k_{ib}}} & if (r,s) \in tour\ done\ by\ the\ agent\ k_{ib} \\ 0 & otherwise \end{cases} \tag{10}$$

Where L_{kib} is the length of the tour done by the best agent which did the shortest tour in the current iteration, and W is a parameter, set to 10.

3 Ant Reinforcement Learning Using Replacing Eligibility Traces

An eligibility trace is one of the basic mechanisms of reinforcement learning. For example, in the popular TD(λ) algorithm, the reference to the use of an eligibility trace.

Almost any temporal-difference method, such as Q-learning, can be combined with eligibility traces to obtain a more general method that may learn more efficiently [13].

An eligibility trace is a temporary record of the occurrence of an event, such as the visiting of a state or the taking of an action. The trace marks the memory parameters associated with the event as eligible for undergoing learning changes.

When a TD error occurs, only the eligible states or actions are assigned credit for the error. Thus, eligibility traces help bridge the gap between events and training information. Like TD methods themselves, eligibility traces are a basic mechanism for temporal credit assignment.

In Ant-TD(λ), there is an additional memory variable associated with each state, its *eligibility trace*. The eligibility trace for edge(r,s) is denoted $e(r, s)$.

The traces are said to indicate the degree to which each state is eligible for undergoing learning changes should a reinforcing event occur.

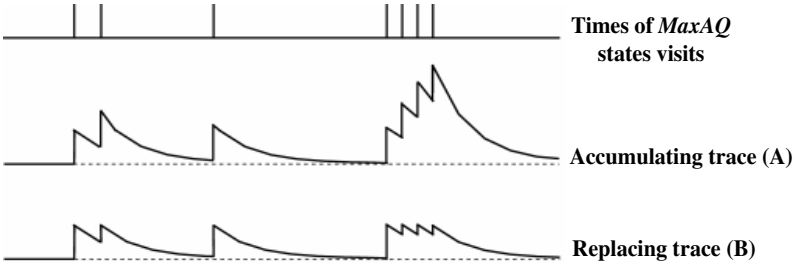


Fig. 2. Feature of Accumulating(A) and Replacing(B) Eligibility Traces

Formally, there are two kinds of eligibility traces: accumulating trace or replacing traces(Fig.1).

With accumulating trace(A), it accumulates each time the state is visited, then fades away gradually when the state is not visited. After all, on each step, the eligibility trace for the maximum($MaxAQ(s, z)$) state visited on the step is incremented by 1 and the eligibility traces for another states decay by $\gamma\lambda$.

Whereas with replacing traces(B), the trace is reset to 1, as illustrated below:

$$e(r,s) = \begin{cases} \gamma \cdot \lambda \cdot e(r_{t-1}, s_{t-1}) + 1 & if \;\; AQ(r,s) = \underset{z \in J_k(s)}{Max}\, AQ(s,z) \ldots (A) \\ 1 & if \;\; AQ(r,s) = \underset{z \in J_k(s)}{Max}\, AQ(s,z) \ldots (B) \\ \gamma \cdot \lambda \cdot e(r_{t-1}, s_{t-1}) & otherwise \end{cases} \tag{11}$$

In this paper, we use a replacing traces. Although replacing traces are only slightly different from accumulating traces, they can produce a significant improvement in optimization.

$$e(r,s) = \begin{cases} 1 & if \quad AQ(r,s) = \underset{z \in J_k(s)}{Max}\ AQ(s,z) \\ \gamma \cdot \lambda \cdot e(r_{t-1}, s_{t-1}) & otherwise \end{cases} \tag{12}$$

Where γ is the discount rate and λ is the trace-decay parameter. This kind of eligibility trace is called a *replacing trace* because it replaces each time the state is visited, and then fades away gradually when the state is not maximum.

Finally, Ant reinforcement learning algorithms using eligibility traces are called *replacing trace* methods(Ant-TD(λ)) updated by the following Eq.(13).

$$\begin{aligned} AQ(r,s) \leftarrow (1-\alpha) \cdot AQ(r,s) \\ +\alpha \cdot (\Delta AQ(r,s)\ +\ \gamma \cdot [\underset{z \in J_k(s)}{Max}\ AQ(s,z) - AQ(r,s)]) \cdot e(r,s) \end{aligned} \tag{13}$$

$$where \quad \Delta AQ(r,s) = 0 \qquad if\ Local\ updating$$

$$\underset{z \in J_k(s)}{Max}\ AQ(s,z) - AQ(r,s) = 0 \qquad if\ Global\ updating$$

$\alpha(0<\alpha<1)$ is the pheromone decay parameter and γ is discount rate. $MaxAQ(s,z)$ is maximum reinforcement value that is received from external environment, global reinforcement is zero. Also, ΔAQ is reinforcement value, local reinforcement is always zero, while global reinforcement, which is given after all the agents have finished their tour, is computed by the following Eq.(14).

$$\Delta AQ(r,s) = \begin{cases} \frac{W}{L_{k_{ib}}} & if (r,s) \in tour\ done\ by\ the\ agent\ k_{ib} \\ 0 & otherwise \end{cases} \tag{14}$$

Where L_{kib} is the length of the tour done by the best agent which did the shortest tour in the current iteration, and W is a parameter, set to 10. Fig.3 shows the proposed Ant-TD(λ) algorithm in pseudocode.

4 Performance Evaluation

Prediction of performance of ant reinforcement learning algorithms using eligibility trace is called replace-trace methods(Ant-TD(λ)). We measure performance through comparison with original ant model(ACS and Ant-Q).

We experimented the proposed model by using TSPLIB[15] which is a famous TSP example. Basis environment parameter for an experiment was decided as following, and optimum value decided by an experiment usually are β=2, α=0.1, q_0=0.9, γ=0.3, λ=0.7, m=n, W=10 and AQ_0= 1/(average length of edges·n).

The initial position of agents assigned one agent in an each node at randomly, and the termination condition is that a fixed number of cycles or the value known as the optimum value was found.

1. /* In this step, initialization phase */
   /* Set an initial value for AQ-values */
   For each pair(r,s) $AQ(r,s) := AQ_0$
   End-for
   /* Choose a starting node */
   For k := 1 to m do
     Let r_{k1} be the starting node for agent(k)
     $J_k(r_{k1})$:= {1, $\cdots$, n} - r_{k1}
   /* r_{k1} is the node set of yet to be visited nodes for agent(k) in node r_{k1} */
     $r_k := r_{k1}$
   End-for
2. /* In this step, agents build their tours and locally update AQ-values. Local reinforcement is always zero, only the next state evaluation is used to update AQ-values. */
   For i := 1 to n do
     If i $\neq$ n
       For k :=1 to m do
         /* Each agent choose the next node s_k according to the state transition rule Eq.(1), updates the set $J_k(s_k)$
         If i < n-1 Then $J_k(s_k) := J_k(r_k)$ - s_k
         If i = n-1 Then $J_k(s_k) := J_k(r_k)$ - s_k + r_k
         $Tour_k(i) := (r_k , s_k)$
       End-for
     Else
       For k := 1 to m do
         /* All the agents go back to the initial node r_k */
         $s_k := r_k$
         $Tour_k(i) := (r_k , s_k)$
       End-for
       For k := 1 to m do

       $AQ(r,s)$:= (1-α)·$AQ(r_k,s_k)$
                 +α·($\Delta AQ(r_k,s_k)$+γ[$MaxAQ(s_k,z)$-$AQ(r_k,s_k)$])·$e(r_k,s_k)$

         /* This above is Eq.(13), the locally reinforcement $\Delta AQ(r_k,s_k)$ is always zero */
         $r_k := s_k$ /* New node for agent(k) */
       End-for
   End-for
3. /* In this step, delayed reinforcement is computed and AQ-values are updated using Eq.(13), the next state evaluation term $\gamma[MaxAQ(s,z)\text{-}AQ(r,s)]$ is zero for all z */
   For k := 1 to m do
     Compute L_k
   End-for
   For each edge(r,s)
     Compute the delayed reinforcement $\Delta AQ(r,s)$ using Eq.(14)

Fig. 3. Pseudocode of Ant-TD(λ) Algorithm

$AQ(r,s)$:= (1-α)·$AQ(r_k,s_k)$
+α·($\Delta AQ(r_k,s_k)$+γ[$MaxAQ(s_k,z)$-$AQ(r_k,s_k)$])·$e(r_k,s_k)$

End-for
Update AQ-values applying a Eq.(13)
4. If (End condition = True)
then Print shortest of L_k
else goto Step 2

Fig. 3. (*continued*)

Table 1 shows optimal tour length and average tour length that are achieved by each algorithms(ACS, Ant-Q and Ant-TD(λ)) in case of repeated 1000 cycles in 10th trials about R×R grid problems, the performance of proposed method is excellent.

Table 2 shows best tour length and average tour length that are achieved by each algorithms(ACS, Ant-Q and nt-TD(λ)) in case of repeated 20000 cycles in 10th trials about TSPLIB problems, the performance of proposed method is excellent.

Table 1. Performance evaluation of Ant-TD(λ) (1)

Node	ACS		Ant-Q		Ant-TD(λ)	
	Average length	Best length	Average length	Best length	Average length	Best length
4×4	**160**	**160**	**160**	**160**	**160**	**160**
5×5	**254**	**254**	**254**	**254**	**254**	**254**
6×6	**360**	**360**	362	**360**	**360**	**360**
7×7	510	507	509	502	**494**	**494**
8×8	661	654	654	648	**640**	**640**
9×9	790	764	782	753	**775**	**749**
10×10	971	959	963	941	**952**	**936**

Table 2. Performance evaluation of Ant-TD(λ) (2)

Node	ACS		Ant-Q		Ant-TD(λ)	
	Average length	Best length	Average length	Best length	Average length	Best length
KroA150	28909	27824	28761	27231	26542	26524
Rat195	2572	2461	2514	2397	2441	2338
Gil262	2637	2526	2592	2493	2412	2380
A280	2893	2768	2841	2758	2637	2581
Pr299	53498	51395	52714	50278	49108	48320
Lin318	46244	44837	45319	43832	44083	42997

5 Conclusion and Future Work

In this paper, we proposed new ant reinforcement learning algorithms using eligibility traces are called replace-trace methods(Ant-TD(λ)).

Proposed Ant-TD(λ) learning is method that is proposed newly to improve Ant-Q, this converged faster to optimal solution by solving temporal-credit assignment problems to use TD-error while agents accomplish tour cycle.

Proposed Ant-TD(λ) ant model using eligibility traces used difference with prediction for output of present state and prediction for output of next state at each learning step, and updated to approximate with prediction for output of present state and prediction for output of next state in present state.

This method is a hybrid of Ant-Q and eligibility traces. With replacing traces, the eligibility trace for the maximum($MaxAQ(s, z)$) state visited on the step is reset to 1 and the eligibility traces for another states decay by $\gamma\lambda$. Although replacing traces are only slightly different from accumulating traces, it could produce a significant improvement in optimization.

Forward, we need study about several possible ways to generalize replacing eligibility traces in Ant-TD(λ).

References

1. Colorni, A., Dorigo, M., Maniezzo, V.: An Investigation of Some Properties of an Ant Algorithm. In: Manner, R., Manderick, B.(eds.): Proceedings of the Parallel Problem Solving from Nature Conference. Elsevier Publishing (1992) 509-520
2. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed Optimization by Ant Colonies. In: Varela, F., Bourgine, P.(eds.): Proceedings of the First European Conference of Artificial Life. Elsevier Publishing (1991) 134-144
3. Watkins, C.J.C.H.: Learning from Delayed Rewards. Ph.D. Thesis, King's College, Cambridge, U.K (1989)
4. Fiecher, C.N.: Efficient Reinforcement Learning. Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory (1994) 88-97
5. Barnald, E.: Temporal-Difference Methods and Markov Model. IEEE Trans. Systems, Man and Cybernetics 23 (1993) 357-365
6. Gambardella, L.M., Dorigo, M.: Solving Symmetric and Asymmetric TSPs by Ant Colonies. Proceedings of IEEE International Conference of Evolutionary Computation, IEEE Press (1996) 622-627
7. Gambardella, L.M., Dorigo, M.: Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. In: Prieditis, A., Russell, S.(eds.): Proceedings of ML-95, Twelfth International Conference on Machine Learning. Morgan Kaufmann (1995) 252-260
8. Dorigo, M., Gambardella, L.M.: A Study of Some Properties of Ant-Q. In: Voigt, H.M., Ebeling, W., Rechenberg, I., Schwefel, H.S.(eds.): Proceedings of PPSN IV, Fourth International Conference on Parallel Problem Solving From Nature. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg (1996) 656-665
9. Dorigo, M., Maniezzo, V., Colorni, A.: The Ant System: Optimization by a Colony of Cooperation Agents. IEEE Trans. Systems, Man and Cybernetics-Part B 26(1) (1996) 29-41

10. Stutzle, T., Hoos, H.: The Ant System and Local Search for the Traveling Salesman Problem. Proceedings of IEEE 4th International Conference of Evolutionary (1997)
11. Gambardella, L.M., Dorigo, M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Trans. Evolutionary Computation 1(1) (1997)
12. Stutzle, T., Dorigo, M.: ACO Algorithms for the Traveling Salesman Problem. In: Miettinen, K., Makela, M., Neittaanmaki, P., Periaux, J.(eds.): Evolutionary Algorithms in Engineering and Computer Science, Wiley (1999)
13. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (1998)
14. Lee, S.G.: Multiagent Reinforcement Learning Algorithm Using Temporal Difference Error. Lecture Notes in Computer Science, Vol. 3496. Springer-Verlag, Berlin Heidelberg (2005) 627-633
15. http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95

Face Recognition Using Correlation Between Illuminant Context

Mi Young Nam, Battulga Bayarsaikhan, and Phill Kyu Rhee

Dept. of Computer Science & Engineering, Inha University 253
Yong-Hyun Dong, Incheon, Korea
rera@im.inha.ac.kr, b_battulga_hm@hotmail.com, pkrhee@inha.ac.kr

Abstract. n this paper investigate how to aggregation method from face recognition varying environments. Face Images clustering is enhanced face recognition performance. Face image is clustered several cluster unsupervised or statistical method and we recognize using correlation between clusters. In this paper we adopted recognition algorithm by aggregation method. In this paper we present the recognition system using the table of fitness correlations between clusters for combining the results from the individual clusters. By training the different classifiers with different clusters of training data and adopting fusion method considering fitness correlation between clusters we found out better recognition performance than combining classifiers fed with same data.

1 Introduction

In an attempt to realize these computing environments, a new brand of computing technologies utilizing human sensory system has been developed. Face recognition technologies have been motivated from the application area of physical access, face image surveillance, people activity awareness, visual interaction for human computer interaction, and humanized vision. Dynamically changing illumination in a real world application poses one of the most challenging problems in face recognition systems.

Classifier decision methods for identification are illustrated their better reliance on recognition than single classifier and implemented in various ways[1, 2, 3]. Clustering the data set into different regions is added value to recognition systems by finding specific sophisticated system for particular region in ways as selection and fusion of classifiers [4, 5]. The outline of this paper is as follows. In section 2, we present the architecture of the proposed classifier fusion system. We give experimental results in section 3. Finally, we give concluding remarks in section 4.

2 Correlation Table Between Illuminant Context

As discussed in session 2, adaptive preprocessing and identification is required for robust face recognition under uneven environments.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 833–840, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2.1 Adaptive Gabor Feature Space

We employ the method of context-awareness in order to provide the capability of adaptation in preprocessing and feature representation stages. The proposed context-aware preprocessing together with adaptive Gabor feature space can perform well under uneven environments. That is generated by GA from best recognition performance. We determine Gabor weight for facial points using evolution algorithm. And we use three preprocessing, histogram equalization, contrast stretching and retinex algorithm [6, 7] for filter fusion.

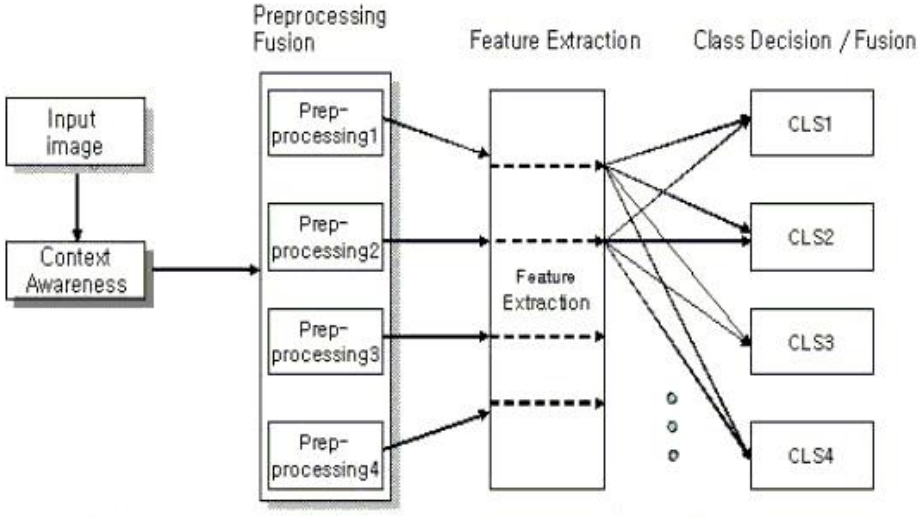


Fig. 1. Face recognition architecture based on the context-aware preprocessing

Clustering is researched many peoples [8, 9, 10]. In contrast to the huge amount of research in this active area [8], little work has been done on combining the specific classifier - the k nearest neighbor classifier (kNN) [11].

In this paper, we generate the method filter fusion as Bayesian based method. The system learns changing environments in the context-awareness stage, and adapts itself by restructuring its structure and/or parameters. The adaptation is guided by evolutionary computing module, genetic algorithm here. We adopt Fuzzy ART [12] for achieving an optimal illumination clustering architecture. In this paper, the clustering performance is improved by iterative learning method.

The system learns changing environments in the context-awareness stage, and adapts itself by restructuring its structure and/or parameters. The adaptation is guided by evolutionary computing module, genetic algorithm here. Filter fusion made following figure.

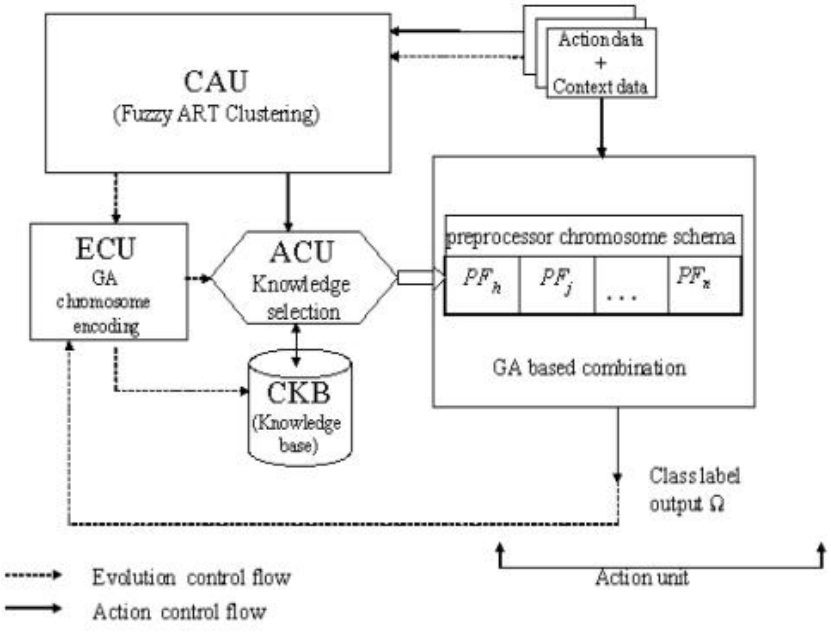


Fig. 2. GA based fusion for preprocessing filter

2.2 Classifier Fusion in Decision Step

Classifier outputs are usually made comparable by scaling to interval [0,1]. We assumed the outputs are also measurable as similarity of feature to classes. By the Fig1 the features in same cluster are measured more similar by classifiers.. In this paper, it is assumed that combination of Classifiers, each fed by data in one cluster is more steady in recognition rate. Classifier fusion assumes that all classifiers are trained over the whole feature space, and are thereby considered as competitive rather than complementary. But some methods as bagging, boosting and Ada-boosting made the classifiers individual from each other by selecting different training data sets [9, 10]. Thus, some solutions considered individualism between classifiers by correlation between them for making final decision.

The assumption that classifiers perform independent of each other might be invulnerable. But methods related to Boosting as Bagging [11], Boosting [9], AdaBoosting [9, 10] considered create each classifier in an ensemble independently of the other classifiers. We can look in way the classifier is simply compares the Test data with trained data. Same idea is introduced here to create the classifiers independent from each other and make the ensemble method fitness correlation more considerable and reasonable. Fig 4 shows the difference of independency of classifiers trained by different data set or whole.

Training Fitness Correlation Table

There is no guarantee that training data set contains features of each class in each cluster. Classifier fusion assumes that all classifiers are trained over

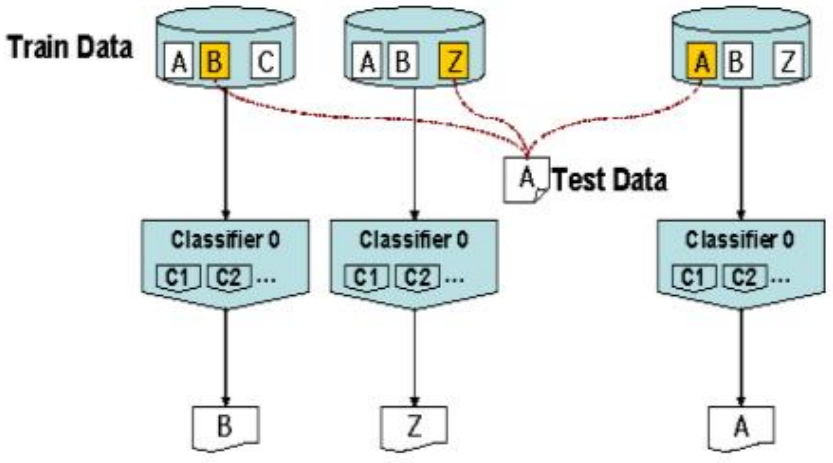


Fig. 3. Classifier Decision Architecture

the whole feature space, and are thereby considered as competitive rather than complementary [1]. In this paper, it is assumed that combination of Classifiers, each fed by data in one cluster is more steady in recognition rate. Let $\Omega = \{\omega_1, \omega_2, ..., \omega_c\}$ be a set of class labels.

Let $x = [x_1, x_2, ..., x_n]^T \in \Re^n$ be an n-dimensional column-vector describing an object. "Soft labels" classifiers are usually defined as the mapping from n-dimensional space to c-dimensional vector with supports to the classes.

That means $\tilde{D} : \Re^n \rightarrow [0,1]^c$ in other word $\tilde{D}(x) = [d_1(x), d_2(x), ..., d_c(x)]$. We can restrict the $d_i(x)$ within the interval [0,1] and say $d_i(x)$ is the degree of 'support' given by classifier $\tilde{D}$ to the hypothesis that x comes from class ω_i. But we actually will use classifiers more simple than "soft labels" for implementing Xcor. In a need of making the problem simple we will define the classifier D form $\widetilde{D}$ as follows:

$$\tilde{D} : \Re^n \rightarrow \Omega \times [0,1] \tag{1}$$

We will use the name "simple-soft-labels" for the classifiers as define above to facilitate further writing.

Xcor Table Traning

Training will consists of two main methods "classifier training" and the "Xcor Table training". Each will be accomplished by using Training Data Set T and Xcor Data Set U respectively.

$$T = \{t_i, t_i \in \Re^n\}, U = \{u_i, u_i \in Re^n\} \tag{2}$$

First we will cluster these data sets into K different sets by using clustering methods like k-means. Let $T_1, T_2, ...T_k$ are the clustered sets of T and $U_1, U_2, ...U_k$ are the clustered sets of U. Despite the T set contains elements of all classes, there is no guarantee that T_i should contain the elements of all classes. This assumption

actually the reason we are adopting Xcor method to keep the recognition rate of classifiers when the training set is not complete at all clusters.

Classifier Training. Train the "simple-soft-label" classifier D by sets $T_1, T_2, ...T_k$ and let $D_1, D_2, ...D_k$ are the trained classifiers respectively.

Acor Table Training. In this method we will create the Xcor Table by using classifiers', $D_1, D_2, ...D_k$, result of recognition on data sets $U_1, U_2, ...U_k$. This method would naturally be understood the training of second layer classifier.

Let Y_{ij}, $Y_{ij} = y, y = D_i(u)|u \in U_j$ is the set of results of D_i on U_j, simply can be represented as Y_{ij}^T. As we discussed in section 3.2 we can define three different sets of soft labels H_{ij},$overlineH_{ij}$, $widetildeH_{ij}$ from Y_{ij}, H_{ij} is set of $d(x)$ components of true results.

H_{ij} is set of $d(x)$components of wrong results despite T contains the true class of u,$u \in U_j$. $\widetilde{H}_{ij}$ is set of $d(x)$components of results when T_i does not contains the true class of $u, u \in U_j$.

Suppose that $H_{ij}, \overline{H}_{ij}, \widetilde{H}_{ij}$ perform normal distributions. The average and standard deviation of $H_{ij}, \overline{H}_{ij}, \widetilde{H}_{ij}$ are denoted by $\{\mu_{ij}, \sigma_{ij}\}$, $\{\overline{\mu}_{ij}, \overline{\sigma}_{ij}\}$, $\{\widetilde{\mu}_{ij}, \widetilde{\sigma}_{ij}\}$ respectively. The Xcor table, the final result of "Xcor Table training" method, would contain $\{\mu_{ij}, \sigma_{ij}\}$, $\{\overline{\mu}_{ij}, \overline{\sigma}_{ij}\}$, $\{\widetilde{\mu}_{ij}, \widetilde{\sigma}_{ij}\}$, m_{ij}, e_{ij} where $i, j \in \overline{1..k}$

Classifying Method

Let $x, x \in Re^n$ was the vector element that to be labeled into one of C classes. Let $\{\omega_1^{'}, d_1\}, \{\omega_2^{'}, d_2\}, ..., \{\omega_k^{'}, d_k\}$ are the results of classifiers $D_1, D_2, .., D_k$. Our main goal is to combine these results by using elements in Xcor table and determine the most valuable result of classifiers. In order to achieve overall performance, the cluster in which the element x belongs to has to be identified. Assume that x is a element of cluster numbered by $j, x \in R_j$. For each classifier, two kind of probability α_i and β_i will be generated using the elements in

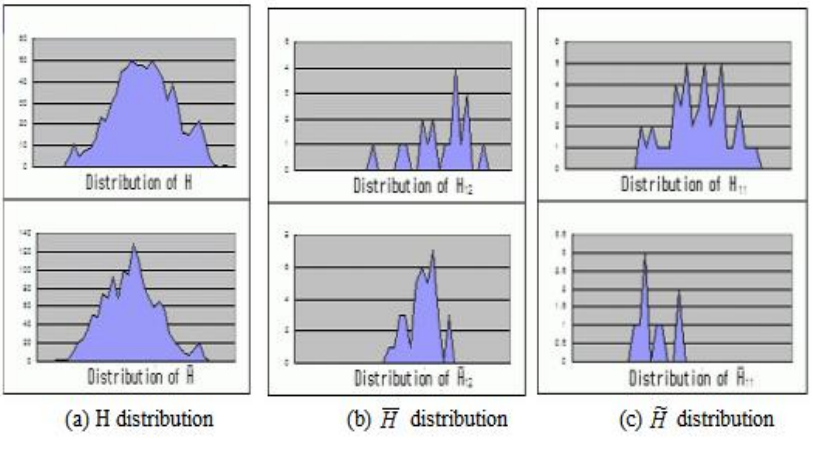


Fig. 4. Classifier Decision Architecture

Xcor table and the final result would be calculated by the α_i , β_i. In final fusion method we will find the probabilities $P_u = P(Y|whenxwaselementof\omega_u)$ and the final result would be $\omega_{u=\overline{1..c}} \in \Omega$.

Generating Probabilities α_i *and* β_i . In instance for classifier D_i two kind of trueness α_i and β_i of result $\{\omega_1^{'}, d_1\} = D_i(x), x \in R_j$ would be generated using the particular elements $\{\mu_{ij}, \sigma_{ij}\}$, $\{\overline{\mu}_{ij}, \overline{\sigma}_{ij}\}$, $\{\widetilde{\mu}_{ij}, \widetilde{\sigma}_{ij}\}$, m_{ij}, e_{ij} in Xcor table. α_i is probability of trueness of $\omega_i^{'}$.

In other word $\alpha_i = P(True|\omega_i^{'})$. We will calculate this probability by Bayesian decision theory in an assumption that T_i, the set of data used for training the classifier UD_i, contains the element of class $\omega^{'i}$.

3 Experiment

The feasibility of the proposed method has been tested using Inha, FERET[13], and Yale[14] database. Experiments have been carried out to compare the recognition performance of the filter fusion and identification based face recognition scheme and that of other methods. We used 1000 images of 100 persons from Inha DB, 60 images of 15 persons from Yale Face DB, and 2418 images of 1196 persons from FERET DB. Table 1 and 2 show that preprocessing fusion using feature weight is high performance for bed illumination images.

Table 3 shows the comparison between single classifier and proposed classifier fusion method. This is result from 5 clusters for illuminant information. It becomes apparent that proposed method shows the highest recognition performance, especially under varying illumination images.

Table 1. Frontal face detection result using single classifier

Method	Retnix + HE	HE + Retnix	
FERET fafc dataset	81.2%	78%	83.5%
Our lab dataset	97.2%	94.5%	98%
FERET fafb	82.2%	93%	96%
Yale DB	84.5%	82.5%	94%

Table 2. Performance evaluation of the proposed system comparing with other approaches

Algorithm Method	FERET(fafc)	FERET(fafb)
Ef_hist_dev_ml1	0.392	0.733
Mit_sep_96	0.32	0.948
Umd_mar_97	0.588	0.962
usc_mar_97	0.82	0.95
Proposed method	0.835	0.96

Table 3. In case of different between text image illuminant and enrolled illumination

Method and database	Yale DB	InhaDB
Gabor28	84.5%	95.1%
PCA	74.1%	87.9%
PCA[5]	95.1%	67.4%
FitCorr[5] with G28	94.4%	99.3%
FitCorr[5] with PCA	71.9%	69.2%

4 Concluding Remarks

We proposed the classifier fusion method from cluster's correlation. Therefore we enhanced face recognition ratio in external environment. The proposed preprocessing and decision fusion based on context-awareness performs well especially in changing illumination environments. Different clusters of training data and adopting fusion method considering fitness correlation between clusters we found out better recognition performance than combining classifiers fed with same data. From extensive experiment, we found that the performance of individual filtering methods for image enhancement is highly depending upon application environments. When the face images is different illuminant between enrolled face image and test face image, face recognition ratio is high other method.

References

1. Ludmilla I. Kuncheva, "Using measures of similarity and inclusion for multiple classifier fusion by decision templates," *Fuzzy Sets and Systems*, vol.122, no.3, (2001) pp.401–407
2. Ludmilla I. Kuncheva, "Clustering-and-selection model for classifier combination," In *4th International Conference on Knowledge-Based Intelligent Engineering Systems*, (2000)
3. Ludmila I. Kuncheva, James C. Bezdek, and Robert P. W. Duin, "Decision templates for multiple classifier fusion: An experimental comparison," Pattern Recognition, (2001) pp. 299–314
4. L. Kuncheva, "Switching Between Selection and Fusion in Combining Classifiers. AnExperiment," IEEE Transaction on Systems, Man and Cybernetics—PARTB, vol.32, no.2, APRIL (2002) pp.146-156
5. Deltas George. The small sample bias of the Gini coefficient: Results and implications for empirical research. *Review of Economics and Statistics*, vol.85, no.1, (2003) pp. 226–234
6. Daniel J. Jobson, Zia-ur Rahman, Glenn A. Woodell, " The Spatial Aspect of Color and Scientific Implications of Retinex Image Processing," Proc. SPIE, vol. 4388, (2001) pp. 117-128
7. Brian Funt, Kobus Barnard, "Luminance-Based Multi-Scale Retinex. normalize proceedings," AIC Colour 97 8th Congress of the International Colour Association, (1997).

8. L.Lim and C.Y.Suen. optimal combination of pattern classifiers. Pattern Recognition Letters, vol.16 (1995) pp.945-954
9. J.Kittler, M.Hatef, R.P.W. Duin, and J.Matas : On combining classifiers. IEEE Trans. Pattern Analysis and Machine Intelligence, vol.20, no.3, (1998) pp.226-239
10. Bay, SD.: Nearest neighbor classification from multiple feature subsets. Intelligent Data Analysis. vol.3, no.3 (1999) pp.191-209
11. Piotr Indyk. : Approximate nearest neighbor algorithm for frechet distance via product metrics. In Proc. Of Symposium on Computational Geometry, (2002) pp.27
12. Ramuhalli, P., Polikar, R., Udpa L., Udpa S., "Fuzzy ARTMAP network with evolutionary learning," Proc. of IEEE 25th Int. Conf. On Acoustics, Speech and Signal Processing (ICASSP 2000), vol. 6. Istanbul, Turkey, (2000) pp.3466-3469
13. P.Phillips : The FERET database and evoluation procedure for face recognition algorithms. Image and Vision Computing, vol.16, no.5, (1999) pp.295-306
14. http://cvc.yale.edu/projects/yalefaces/yalefaces.html

An Efficient Face and Eye Detector Modeling in External Environment

Mi Young Nam, Eun Jin Koh, and Phill Kyu Rhee

Dept. of Computer Science & Engineering, Inha University 253
Yong-Hyun Dong, Incheon, Korea
{rera, supaguri}@im.inha.ac.kr,
pkrhee@inha.ac.kr

Abstract. In this paper, we propose multi-class classifier and knowledge based face detection. Eye region and face location is used illuminant based Bayesian detector. We propose the efficient face and eye detection system using varying illuminant context modeling and multi–classifier. The face detection system architecture use cascade method by illuminant face model. Also, we detect eye region after face detection. Proposed eye detection frame is multiple illuminant Bayesian classifiers. Because face images have varying illuminant and this is vary difficult problem in face detection. Therefore, we made in context model using face illuminant. The multiple classifiers consist of face illuminant information. Multiple Bayesian classifiers are employed for selection of face and eye detection windows on illuminant face group. Finally, face and eye regions of the detected candidates are selected by context awareness.

1 Introduction

Detecting human face in image frames is an important task in many computer vision applications. Face detection, the first step of an automated face recognition system, determines the location and size of each human face from an input image. Closely related problems are face verification and identification. Face detection is one of the hardest problem in the research area of face recognition, and an important task in many computer vision applications such as human-computer interface, smart environment, ubiquitous computing, multimedia retrieval, etc. Object detection using a static image can be used in unconstrained environments with complex background Object detection systems from still image can be divided into three major categories. Eye detection is researched many people, Hero, Baskan [1, 2, 3]searched the profile method but is not researched very illuminant face images.

The major contribution of this paper is efficient classification using multiple Bayesian classifiers for efficient face and eye detection.

The outline of this paper is as follows. In section 2, we present the architecture of the proposed context awareness system. In section 3, we describe the method that search context based object detection. We give experimental results in section 4. Finally, we give concluding remarks in section 5.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 841–849, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Illuminant Context-Awareness by BP

The supervised learning method is classified the nine illuminant group. Fig.1 is shown the illuminant context group.

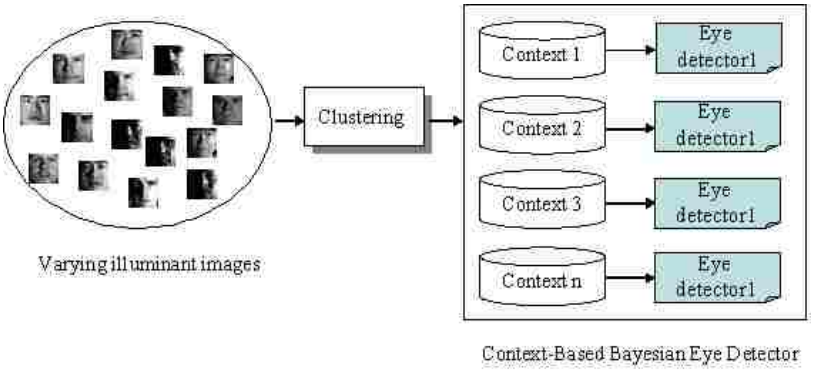


Fig. 1. Synthesized images using the face image modeling

We have tested three methods for illumination discrimination: the simple rule based discrimination(SR), the back propagation neural network based discrimination(BP). The training of evolutionary neural network is done 100 original face images accumulated in our lab, and 800 virtually generated mosaic face images using the image synthesis method described above. We assume that illumination variation in face images can be represented by the noise model. The modeling face image reflecting an brightness variation can be done by the additive, the multiplicative, and the hybrid functions[7]. Directional illumination variations are modeled by the sine and the cosine weight function.

3 Context Based Object Detection Scheme

In this paper, we proposed the method that is multiple Bayesian classifiers for selecting eye location and face detection. The final decision of face candidate window is done by post processing.

3.1 Face Detection

Face detection is statistical method using Mahalanobis distance and cascading face detection system, Search space is reduced by finding regions that has face color. The remaining regions are searched by multiple Bayesian classifiers under varying illuminant.

We can model a face in frequency domain using Haar wavelet transform(shown Fig.3). It is organize the image into sub-bands that are localized in orientation and frequency. In each sub-bands, each coefficient is spatially localized.

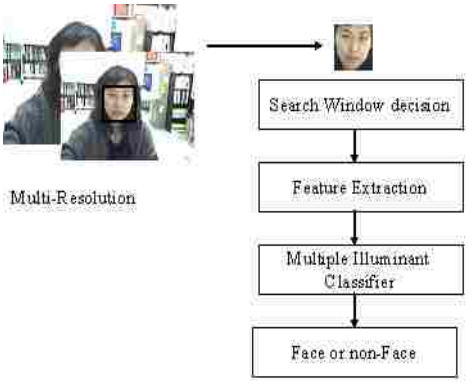


Fig. 2. Face detection system architecture

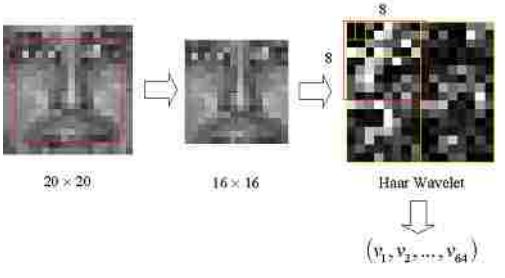


Fig. 3. Haar wavelet transform for face modeling

We use a wavelet transform based on 3 level decomposition producing 10 sub-bands. Generally, low frequency component has more discriminate power than higher. Also too high component has some noisy information. Then we use 3 level decomposition of Haar wavelet transform. Face image distribute using Back propagation.

These classifiers are constructed in cascade form of multi steps. The 64-dimensional feature from illuminant 1'st classifier is used as classifier in first step. The illuminant 2'st classifier is used in second step. These classifiers produce different false alarms when they work alone. Therefore cascade classifier reduces false alarms when each result is merged. Structure of cascade classifier is shown in middle block of Fig.4. Once a sub-window is determined to non-face by first classifier, it doesn't be tested by second classifier. First, search space can be reduced by skin color module. The remaining regions are searched by multiple Bayesian classifiers using integrated feature space. Final results are produced by post processing.

Merging heuristic that removes the overlapped face candidate windows is powerful process to remove most false detections and reduce multiple detections of a single face to one[4, 5]. This post processing can produce the results as Fig.5.

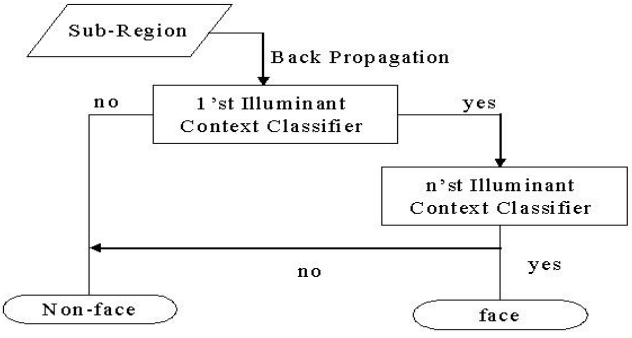


Fig. 4. Context-based face detection

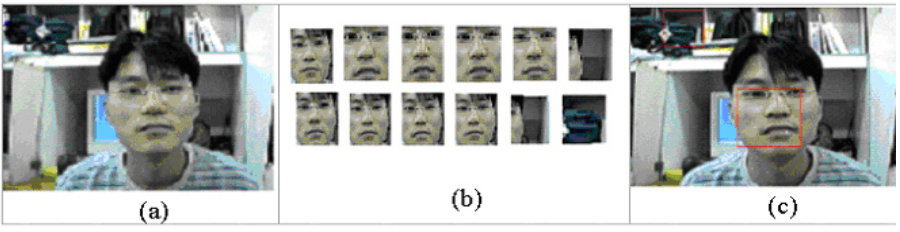


Fig. 5. Post processing. (a) Input image (b) Before post processing (c) After post processing.

3.2 Eye Detection

The outline of the proposed method is shown in Fig.6. Search space is reduced by finding eye regions. The remaining regions are searched by multiple Bayesian classifiers using integrated feature space to determine whether the 16×16 sub-window is eye region. These classifiers are constructed in cascade form of two steps. The 54-dimensional feature from intensity using sub-regions is used as feature in first step. Haar wavelet transform is used in second step. These two classifiers produce different false alarms when they work alone. Therefore cascade classifier reduces false alarms when each result is merged. Structure of cascade classifier is shown in middle block of Fig.6.Once a sub-window is determined to eye region by first classifier, it don't be tested by second classifier. In this paper, we use feature extraction method by the Haar wavelet transform.

We proposed the multi-classifier using Bayesian for illumination face and eye image.

Bayesian formula for each illuminant face and eye group is

$$P(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)\,P(\omega_i)}{p(\mathbf{x})} \tag{1}$$

where , i is illuminant eye group.

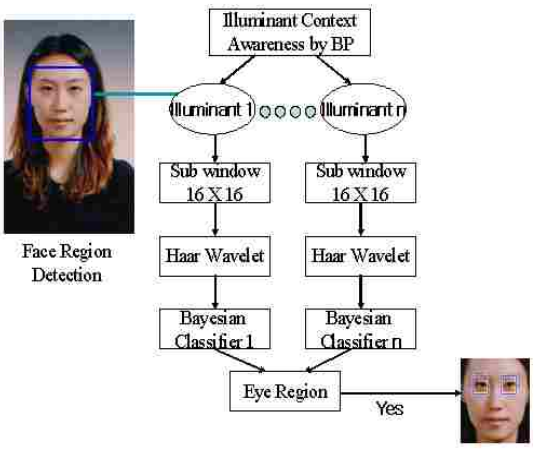


Fig. 6. The proposed scheme used for eye region detection

We applied different Bayesian classifier for cluster classification.

$$g_e(\mathbf{X}) = lnp(x|\omega_i) + lnP(\omega_\mathbf{i}) \tag{2}$$

$$g_{ne}(\mathbf{X}) = lnp(x|\omega_n) + lnP(\omega_\mathbf{n}) \tag{3}$$

But we can use one discriminant function, $g(\mathbf{x})$ as (3) since we try to classify a pattern into two categories. In this case, if $g(\mathbf{x})$ is greater than 0, we can decide input pattern as face. Otherwise, it is comparison other Bayesian classifier.

$$g(\mathbf{X}) = g_e(\mathbf{X}) - g_{ne}(\mathbf{X}) \tag{4}$$

The locations which output through previous methods are good at being precision detect. But because eyes are very important feature at face recognition, we want to make robust precision position of eye. We regard pupil is robust precision point of eye and make an effort to point pupil. We use Gaussian filter and projection function to point center of pupil. In this case, center of detected window is moved far away from earlier position. It is a critical error and a primary factor of decrease of detect rate. Therefore, we need to restrict center of window to move for away.

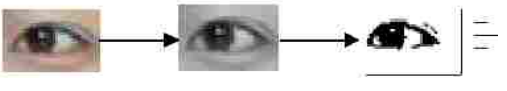


Fig. 7. An Eye detection sample under varying illuminant context images

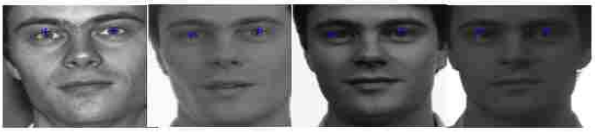


Fig. 8. Eye detection results under varying illuminant

4 Experiment

We used 3,816 FERET frontal face images as face data and 23 natural images as nonface data for training data. Because we include mirror images with the face data, total face data consist of 7,632 frontal face images.

In order to normalize size and position of eyes, each face image was rotated and scaled to 16×16 size as shown in Fig.9. Non-face data was gathered from 23 natural images. Because the face-like non-face samples were chosen by each Bayesian classifier with two different types of feature extraction methods. Thus each classifier uses different number of face-like non-face images.

We performed the experiments on two test sets constituted from images of MIT-CMU frontal face images and captured images. First set consists of 130 images include 507 faces. 117 images contain frontal faces and remains don't.

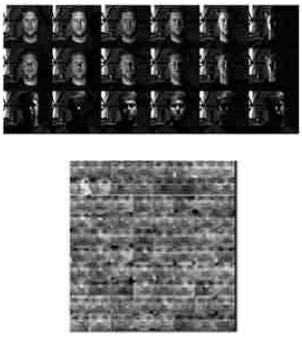


Fig. 9. Face training images

Fig. 10. Natural images that are not contained any face image

Table 1. Frontal face detection result using single classifier

Test Set	Face Detection	False detection	Face number
MIT -CMU	283	150	507
Our Lab.	2299	43	2539
FERET	1100	32	1196
Yale	92	20	100

Table 2. Frontal face detection result using multi-classifier using illuminant context-awareness

Test Set	Face Detection	False detection	Face number
MIT -CMU	350	30	507
Our Lab.	2445	10	2539
FERET	1153	6	1196
Yale	97	3	100

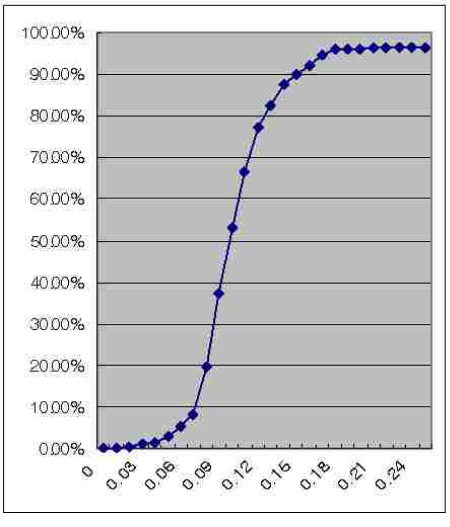


Fig. 11. ROC Curve of eye detection

Second set consists of 2,539 images that were captured in various environments. We show the experiment result in Table 1 and Table 2.

Table 3 and Fig.11 are shown that we achieved very encouraging experimental results in the error rate 0.18. The first set has the some faces that are not frontal. Thus the result in this set is not good. But in the second set, the images have only frontal faces and our skin color module worked. The result in this set is very good. We show the detected images in Fig.12.

Table 3. Without context – awareness eye location results

Err	AR	FA	AR
0	0	3816	0.00
0.04	54	3762	1.42
0.05	111	3705	2.91
0.06	201	3615	5.27
0.07	314	3502	8.23
0.08	744	3072	19.50
0.09	1424	2392	37.32
0.10	2021	1795	52.96
0.11	2544	1272	66.67
0.12	2942	874	77.10
0.13	3148	668	82.49
0.14	3342	474	87.58
0.15	3438	378	90.09
0.16	3521	295	92.27
0.17	3607	209	94.52
0.18	3670	146	96.17

Fig. 12. Detected images of MIT-CMU data set

5 Concluding Remarks

In this paper, we propose the efficient face and eye detection method. It consists of multiple Bayesian classifiers using clustering face and sys images. In varying illuminant images such that from cameras face detector module can reduce false detectio ratio. And face detection ratio is enhancement in dynamic environment. This proposed method can detect 96.17% of eye in Err 0.18. There are some works that can improve the proposed method. The second feature of our method, Haar wavelet transformation must be improved. Proposed multiple face and eye detector is high performance. The face and eye detection ratio must be improved in our propose method.

References

1. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)
2. A. Haro, M. F. : Detecting and tracking eyes by using their physiological properties, dynamics, and appearance. Proc. Of IEEE Conf. on CVPR (2000)
3. S.Baskan, M. M. B., V.Atalay : Projection based method for segmentation of human face and its evaluation.. *Pattern Recognition Letters* 23 (2002) pp.1623-1629
4. S.Lucey, S.Sridharan, V.Chandran : Improved facial feature detection for AVSP via unsupervised clustering and dicriminant analysis. *EURASIP Journal on Applied Signal Processing*, vol 3 (2003) pp.264-275
5. C. Liu. : A Bayesian Discriminating Features Method for Face Detection. IEEE Trans. Pattern Analysis and Machine Intelligence, vol.25 (2003) pp.725-740
6. T. V. Pham, et. Al.,: Face detection by aggregated Bayesian network classifiers. Pattern Recognition Letters 23, (2002) pp. 451-461
7. Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A.: Pfinder: Real-Time Tracking of the Human Body. Technical Report 353, Media Laboratory, Massachusetts Institute of Technology, (1995)
8. P.Phillips : The FERET database and evoluation procedure for face recognition algorithms. Image and Vision Computing, vol.16, no.5, (1999) pp.295-306

Keypoints Derivation for Object Class Detection with SIFT Algorithm

Krzysztof Slot[1] and Hyongsuk Kim[2]

[1] Institute of Electronics, Technical University of Lodz, Wolczanska 211/215, 90-924 Lodz and Academy of Humanities and Economics, Lodz, Poland
kslot@p.lodz.pl

[2] Division of Electronics and Information Engineering, Chonbuk National University, 561-756 Chonju, Republic of Korea
hskim@chonbuk.ac.kr

Abstract. The following paper proposes a procedure for SIFT keypoints derivation for the purpose of object class detection. The main idea of the method is to build appropriate object class keypoints by extracting information that corresponds to characteristic class features. The proposed procedure is composed of two main steps: clustering of similar SIFT keypoints and derivation of appropriate keypoint descriptors. Face detection in images has been selected as a sample application for the proposed approach performance evaluation.

1 Introduction

Scale-Invariant Feature Transform (SIFT) algorithm is a relatively new object recognition paradigm [1], which belongs to a class of feature-based object recognition strategies [2],[3],[4]. SIFT proved to be robust in realizing several hard recognition tasks, such as detection of scaled, rotated and partially occluded objects in highly cluttered environments. Objects in SIFT are characterized through keypoints and their descriptors. Keypoint descriptors are complex image gradient field characteristics, built in a way that makes them insensitive to image transformations and that provides very distinctive and unique representation of object features. High specificity of descriptors maximizes a chance of some particular object detection; however it becomes a drawback when a detection of object class is to be made, i.e. when some level of within-class object variability needs to be admitted.

The following paper proposes a procedure for building SIFT keypoints that could be used for detecting of object classes (a problem of object class recognition from local features has been studied e.g. in [5],[6],[7]). The main idea of the procedure is to identify, among all class' objects, pools of keypoints that share the same common properties, and to integrate this information in a form of target, object class keypoints and their descriptors. Face detection has been selected as a specific task for the proposed approach evaluation.

The paper has the following structure. After a brief presentation of the SIFT algorithm (Section 2), the proposed keypoint derivation procedure is explained (Section 3) along with its preliminary experimental evaluation (Section 4).

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 850–859, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 SIFT Algorithm

SIFT algorithm is a feature-based object recognition strategy that is composed of two major image processing phases [1]. First, a set of characteristic locations (keypoints) is extracted from an input image and then, keypoint and keypoint group matching is employed to test an image for a presence of objects of interest.

To ensure scale-invariant object recognition, keypoint derivation involves multi-resolution image analysis, so that characteristic image regions at various scales can become preliminary candidates for keypoints. These candidates are subject to further selection aimed at excluding of elements with insufficient magnitude (low feature stability) and edge responses (redundant representation). The remaining elements become keypoints and are assigned with descriptors that summarize information on local image gradient flows.

Object recognition step begins with matching of all keypoints, extracted from an image, to object prototype keypoints, recorded in a corresponding database. Keypoint-distinctiveness is used as a primary criterion for preliminary match selection, reducing a pool of elements considered in further analysis. Object recognition is done through keypoint group matching, which involves object candidate pre-selection (using Generalized Hough Transform), affine matching of considered regions and a probabilistic correspondence test, performed on aligned keypoints.

3 Object Class Keypoints Derivation Procedure

SIFT keypoint descriptors represent local gradient-field properties in a form of a set of orientation histograms, computed for sixteen regions within a keypoint neighborhood. This detailed representation provides high keypoint matching selectivity, crucial for reference object detection, however, it becomes a problem in detection of objects that somehow differ from a prototype, yet belong to the same class. Poor performance of the method in detecting objects of the same category has been illustrated in Fig. 1, where keypoints derived from sample image face (shown in upper right) allow for finding of only two faces in the image (including the prototype).

For object class detection one needs to look for keypoint descriptors that would reflect some typical appearance of features, which is common to all class' elements rather than to derive keypoints from some randomly drawn class member. The proposed object class keypoint derivation procedure, which is depicted in Fig. 2, explores similarities that exist among class objects.

The first phase of the procedure is aimed at determining groups of keypoints that are likely to correspond to various realizations of the same semantic features of training set objects. This is done through a sequence of keypoint clustering procedures, where spatial locations, orientations and scales are used as keypoint similarity criteria. The resulting keypoint clusters are then used for derivation of object class keypoints in the second phase of the algorithm. Keypoints extracted from training set images will be henceforth referred to as *object keypoints*, whereas target object class keypoints will be referred to as *class keypoints*.

Fig. 1. Face detection using object-derived keypoints: input image (left), detected keypoints (middle) and detected faces (right) with marked correct match locations

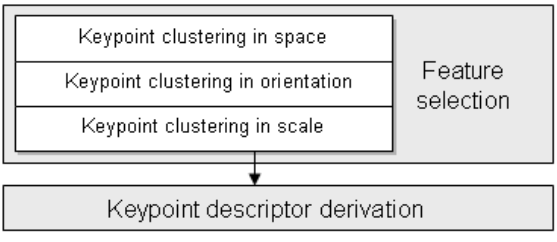


Fig. 2. Object class keypoint derivation procedure

3.1 Feature Selection

A derivation of object class representation begins with a search for groups of keypoints that appear consistently in location, scale and orientation among all training set objects, so that they can be regarded as markers of the same semantic feature. Spatial keypoint grouping can be performed using any of existing data clustering methods [8], however, for the considered face detection case, a set of characteristic regions that correspond to distinct facial features was selected manually (Fig. 3). Keypoints, which appear within any of pre-selected facial regions, are subject to clustering in scale and orientation. Sample keypoint distributions for the 'nose' facial region, expressed in a scale-orientation space, have been presented in Fig. 3, along with image intensity distributions that correspond to marked keypoint clusters. Each of the clusters of object keypoints with low-variance, both in scale and orientation, has been selected as a basis for further class keypoint derivation procedure. Since no distinctive clusters were found for some of pre-selected regions, these regions were dropped from further analysis. This means that some semantic features, such as e.g. forehead, will have no representation among class keypoints. On the other hand, some other features will have several representatives, as it happens in the case of the presented 'nose' region.

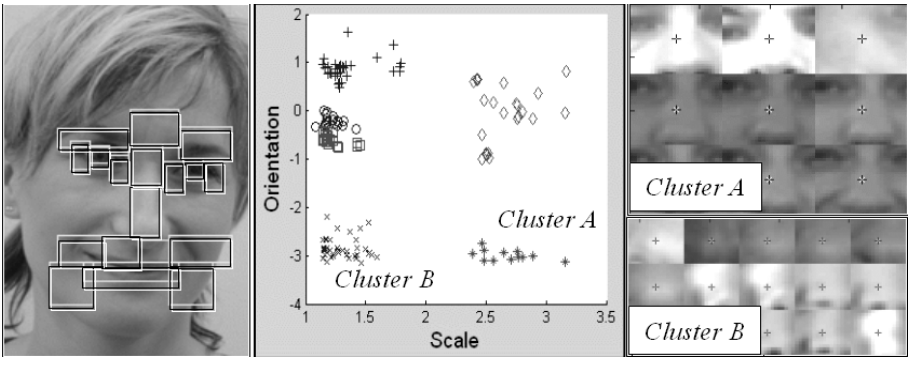


Fig. 3. Facial regions used for clustering of keypoints in space (left); keypoint clusters in the scale-orientation space (middle) and intensity distributions corresponding to the marked clusters (left)

3.2 Keypoint Descriptor Derivation

Object keypoint clusters identified in the first phase of the procedure correspond to semantic facial features and form domains for a derivation of class keypoints. A class keypoint is supposed to reflect common properties of cluster members. If we assume that orientation histograms of all cluster keypoints have similar distributions, an appropriate class keypoint descriptor can be generated by taking an average or median over these histograms. Since median is more robust with respect to outliers, it has been used for producing a class keypoint descriptor.

The major difference between object recognition and object class recognition by means of the SIFT algorithm emerges at final stages of image analysis, i.e. at keypoint elimination and keypoint group matching. In the original algorithm, keypoints extracted from a test image are discarded if they correspond to ambiguous matches, i.e. if a ratio of matching scores for the two winners drops below some pre-set threshold. This strategy is justified when corresponding keypoints from the same object are to be fit. However, a class keypoint is expected to reflect an average realization of some feature appearance and consequently, differs from any particular object keypoint. Therefore some level of keypoint mismatch is inevitable, which implies that an application of the original criterion for keypoint elimination is inappropriate. To enable discarding of false matches one can use another approach. Namely, one can evaluate (and record) an average mismatch that exists between a class keypoint and all object keypoints from the corresponding cluster. This value can be used as a reference for determining whether a match is to be accepted or rejected. Each generated class keypoint is therefore assigned with additional information - an average scatter, which is used for determining a keypoint match acceptance threshold.

Keypoint clustering based on orientation histograms. The presented class keypoint descriptor derivation strategy will produce a correct representation

of a feature only if gray level distributions, represented by descriptors of cluster object keypoints, are similar to each other. However, cluster keypoints typically correspond to various realizations of the underlying feature (Fig. 4a,b), which is usually represented by distinct, yet repetitive intensity distributions. Averaging object keypoint descriptors or taking their median is therefore likely to produce a class keypoint descriptor, which represents neither of characteristic modes of a typical feature appearance. Therefore, keypoint clusters identified in the feature selection phase of the procedure need to be further partitioned, to identify groups of keypoints that correspond to similar image intensity distributions. Since keypoint descriptor structure is region-oriented, a region-wise consistency among cluster member descriptors is examined. Region-wise histogram similarity is evaluated based on a parametric histogram representation, through modes and corresponding spreads of orientation distributions. As a result, object keypoint sub-clusters that represent similar image distributions are identified. A class keypoint descriptor that corresponds to such a sub-cluster can be computed by taking an element-wise median of sub-cluster member descriptors (Fig. 4c).

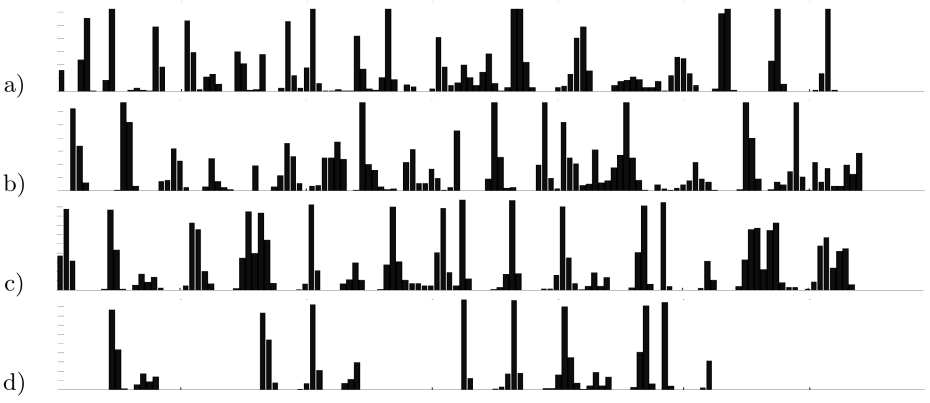


Fig. 4. Keypoint orientation histograms for two sample members of the nose-region cluster (a,b), a class keypoint descriptor candidate generated for one of its sub-clusters (c) and the resulting class keypoint descriptor with discarded regions shown blank (d)

The last refinement of the presented strategy is aimed at increasing class keypoint selectivity by focusing only on these regions around cluster keypoint locations, which have similar image intensity distributions. For example, some areas within a neighborhood of keypoints that are located in eye regions can be repetitive, whereas some others, such as eye-lids with eye-lashes, can vary significantly. Instead of admitting of inevitably large within-class scatters, a better strategy is to selectively discard parts of a descriptor from a matching process. This can be based on the assessment of a within-class scatter that exists for each neighborhood region r among a set of considered object keypoints:

$$s_r = \frac{1}{8} \sum_{b=1}^{8} \sum_{i=1}^{N} \left| c_{r,b} - o_{r,b}^{i} \right| \tag{1}$$

where $c_{r,b}$ is an element of the derived class keypoint descriptor that corresponds to $b-th$ bin of an $r-th$ region, $o_{r,b}^{i}$ is an $b-th$ element of an $r-th$ region of some $i-th$ object keypoint and N is a number of object keypoints in the considered sub-cluster. The following descriptor region acceptance threshold:

$$\theta = E(s_r) + k_\theta \sigma(s_r) \tag{2}$$

is then computed for a class keypoint, where $E(.)$ is an expectation, $\sigma(.)$ denotes a standard deviation and k_θ is a parameter. Only these descriptor regions where $s_r < \theta$ are included into a class keypoint descriptor. By varying the parameter k_θ, i.e. the threshold level (2), different sets of keypoints with varying numbers of excluded descriptor regions, can be generated. High descriptor region acceptance thresholds yield keypoints with larger admissible region-wise intensity differences, low thresholds yield more selective sets of keypoints. The class keypoint descriptor, derived according to the proposed strategy, will contain regions, which won't be considered in testing for keypoint similarity, which has been shown in Fig. 4d.

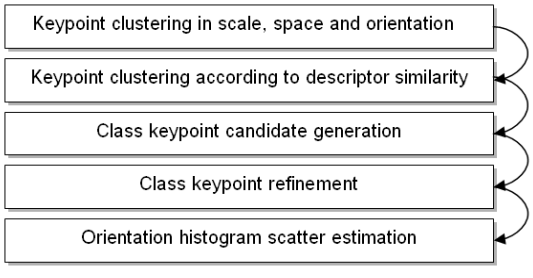


Fig. 5. Class keypoint descriptor derivation procedure

The adopted strategy for object class keypoint descriptor derivation has been summarized in Fig. 5. Clusters of features that are consistent in space, scale and location are subject to partitioning, based on region-wise orientation histogram consistency analysis. Element-wise median of object keypoint descriptors from resulting sub-clusters constitute preliminary candidates for corresponding class keypoint descriptors. These candidates are subject to further refinement, aimed at increasing their selectivity by excluding some descriptor regions. The last step of the class keypoint derivation procedure is an estimation of the keypoint match acceptance threshold, which is done by adding together within-class scatter measures s_r of all accepted regions. This value is assigned to every derived class keypoint.

4 Experimental Evaluation of the Procedure

A database of 1650 annotated frontal-view face images of 40 persons (taken mainly from the "BioId" database [9]) was used in our experiments. Training and test sets used for derivation of class keypoints as well as for image recognition, were disjoint, i.e. they were containing faces of different persons. SIFT algorithm parameters suggested in [1] (such as a number of scales used within each resolution octave, edge response elimination thresholds, details of orientation histogram generation etc.) were adopted throughout experiments. Fifteen hundred images of thirty persons were used to derive several sets of object class keypoints according to the presented procedure. Each of these sets, which can be considered as a 'SIFT face prototype', was constructed by setting different values of the procedure parameters (such as k_θ, clustering thresholds etc.). The constructed face prototypes were composed on average of 25 class keypoints.

The first objective of our experiments was to verify whether object class keypoints perform any better than object keypoints in detecting faces. To examine that, results of test image analysis by means of the derived class keypoint sets were compared to the results of processing with sets of object keypoints, extracted from multiple, randomly selected training images. For the case of class keypoint matching, an average number of correct matches per test face image varied between five and nine, depending on the selectivity of applied keypoints. In the latter case, results of object keypoint matching were significantly worse, regardless of a number of face images used for keypoint extraction (Fig. 6).

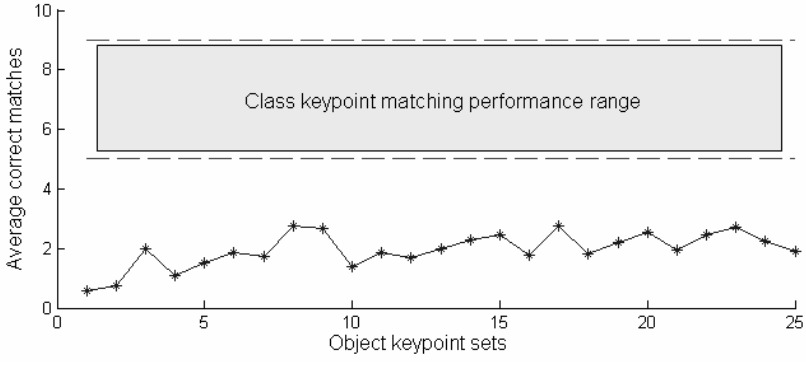


Fig. 6. Object keypoint matching versus class object matching performance

An objective of further experiments was to examine a performance of the derived different SIFT face prototypes in the context of object recognition efficiency. To achieve reasonable complexity of object class detection, a substantial number of false matches should be eliminated prior to subsequent SIFT algorithm steps. The adopted false match elimination mechanism, tests matching

scores and rejects test image keypoints with excessive mismatch. Keypoint filtering performance for two different sets of class keypoints, composed of less selective keypoints (obtained for setting k_θ from the equation (2) to 3) and more selective keypoints (with $k_\theta = 1$), has been shown in Fig. 7. An experiment objective was to estimate a percentage of correct matches and false matches as a function of the keypoint acceptance threshold. This threshold was varying from T_i to $3 * T_i$, where T_i is an average scatter, associated with an *i-th* class keypoint. As it can be seen, a transition between two extreme cases - all keypoints rejected and all keypoints accepted - is steep for SIFT face prototypes composed of less selective keypoints. Since wide transition regions provide larger tolerance for a threshold selection (it has to be preset prior to test image analysis, however its exact value is data-dependent), face prototypes composed of more selective keypoints should be used.

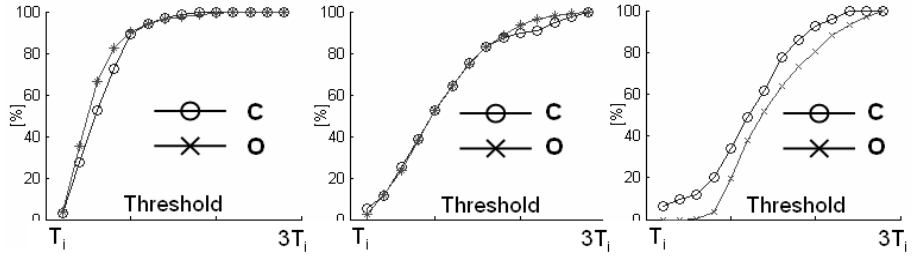


Fig. 7. Keypoint acceptance rates versus the acceptance threshold for three different SIFT face prototypes: low selective keypoint set (left), selective keypoint set (middle) and a set of well-performing selective keypoints (right). C denotes class keypoints, O - object keypoints.

Results of an application of two different keypoint acceptance thresholds in sample face detection task are shown in Fig. 8. As it can be seen, a number of incorrect matches significantly exceeds a number of the correct ones, whereas a proportion between the two remains similar (as shown in Fig. 7).

To increase an efficiency of object class detection it would be desirable to change a proportion between correct and false match acceptance rates in favor of the former category. To do that we decided to determine keypoints that account for large numbers of correct matches as well as these ones that produce large number of false matches. Typical correct and false match histograms for a sample class keypoint set has been presented in Fig. 9.

As it can be seen, some keypoints unevenly contribute to correct and to false match frequencies. Therefore, we decided to discard from the created SIFT face prototypes all keypoints that produced an excessive amount of false matches. A performance of sample resulting set is summarized in the right part of Fig. 7, where 10 to 20 percent increase in a correct to false match acceptance ratio can be observed.

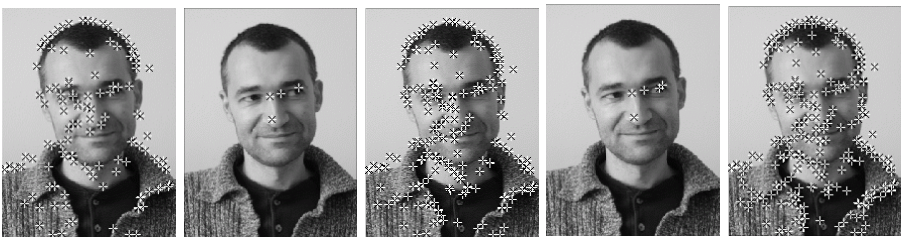


Fig. 8. Keypoint-matching results for two different keypoint acceptance thresholds T_1 and $T_2 > T_1$: from left to right - keypoints accepted for T_1, correct matches for T_1, keypoints accepted for T_2, correct matches for T_2, all image keypoints

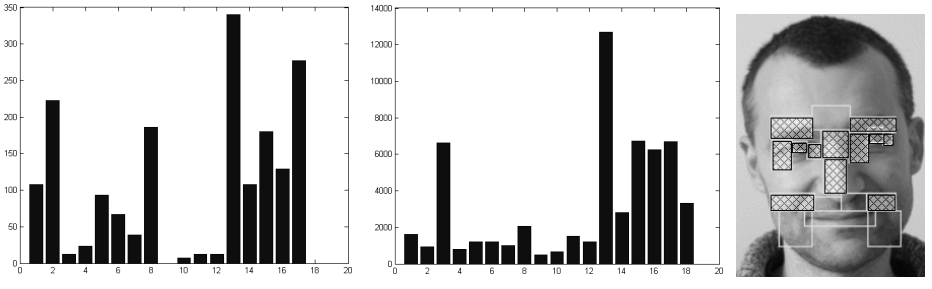


Fig. 9. Sample histograms of correct matches (left) and false matches (middle) for various feature regions; filled regions with well-performing keypoints (right)

5 Conclusion

A strategy for object class keypoint set derivation has been presented in the paper. The proposed procedure includes two phases - a selection of keypoints that correspond to characteristic class object features and a derivation of appropriate class keypoint descriptors. It has been shown that using the proposed approach it is possible to generate sets of keypoints that are capable of detecting object classes. Also, a strategy for false match elimination, which is crucial for the computational feasibility of the SIFT algorithm, has been proposed. Since this strategy relies on data-driven keypoint acceptance threshold estimation, further experiments involving other data sets still need to be performed, before any decisive conclusions regarding SIFT robustness in object class detection can be made.

Acknowledgment

This research was supported by the MIC (Ministry of Information and Communication), Korea, Under the ITFSIP (IT Foreign Specialist Inviting Program) supervised by the IITA (Institute of Information Technology Assessment).

References

1. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. In: International Journal of Computer Vision, Vol. 60 (2). Kluwer Academic Publishers, (2004), 91-110
2. Harris, C. and Stephens, M.: A combined corner and edge detector. Fourth Alvey Vision Conference, Manchester, UK, (1988), 147-151.
3. Mikolajczyk, K., Zisserman, A., Schmid, C.: Shape recognition with edge-based features. British Machine Vision Conference, vol.2 Norwich, U.K. (2003)779-788
4. K. Mikolajczyk and C. Schmid.: A performance evaluation of local descriptors. Conference on Computer Vision and Pattern Recognition (2), Madison, Wisconsin, (2003), 257-263.
5. Fergus R., Perona P., Zisserman A.: Object class recognition by unsupervised scale-invariant learning. Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin (2003), pp. 264–271
6. Helmer S., Lowe D.G.: Object recognition with many local features. Workshop on Generative Model Based Vision, Washington, D.C. (2004)
7. Zhang W., Bing Yu, Zelinsky G.J., Samaras D.: Object Class Recognition Using Multiple Layer Boosting with Heterogeneous Features. International Conference on Computer Vision and Pattern Recognition, Vol. 2, (2005), 323-330.
8. L. Kaufman and P. J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*, John Wiley & Sons, 1990
9. The BioID Face Database - http://www.bioid.com/downloads/facedb/

Gray Image Contrast Enhancement by Optimal Fuzzy Transformation

Roman Vorobel and Olena Berehulyak

Institute of Physics and Mechanics,
Ukrainian Academy of Sciences,
5 Naukova Str., Lviv, 79601, Ukraine
{vorobel, oberehulyak}@ipm.lviv.ua

Abstract. The brief analysis of methods for contrast enhancement of gray images is performed. The application of fuzzy logic for image binarization and contrast enhancement is emphasized. The drawbacks of known methods are shown. To transfer from spatial domain to fuzzy one by the way of additional optimization of the of S-type membership function shape over its steepness by the change of order, which can be both whole number and fractional one, is proposed. The new method of image reconstruction from the smoothed one after the local contrast enhancement in the fuzzy domain is applied. The effectiveness of proposed method is illustrated on the examples.

1 Introduction

Fuzzy logic, introduced by L. Zadeh in 1965, has found wide application in image processing, particularly in different analysis systems, in 70^{th}-80^{th} years of last century. The distinguishing feature of this period is the fact that problem of binary image formation from its halftone initial one was solved. Several papers will be enumerated here, among which the most typical is [6], where the classical for nowadays way of halftone image binarization is described. In the Ref. [5] method of construction of fuzzy membership function is proposed. It also was implemented in the halftone image processing with the purpose of the formation of its binary representation. Simultaneously, the automation of the process of text and sign recognition also caused further development of image processing methods based on fuzzy logic [3, 7]. And only in posterior papers [8, 9] the methods of gray image enhancement in the sense of their detalization and quality increase for visual perception were proposed.

These approaches, as well as described in [1, 2], made the application of fuzzy logic an effective tool for image quality enhancement, in spite of some processing time increasing in comparison with classical method in the spatial domain [4]. Their advantage is a possibility to represent image details on the multiplex nonuniform background in more qualitative way. But the disadvantage of existing approaches is an unsolved problem of improved image contrast management and receiving of its significant increasing without the distortion of multiplex elements of the background. It is known, the image contrast enhancement with

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 860–869, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the lost of weak contrast details, when significant contrast of image is reached by actual transformation of halftone image into close to binary one – is not a problem. The purpose of this paper is the development of approach to widening of range of halftone image contrast intensification in general, and preserving the structure of fine details.

2 Membership Functions

The approach to image contrast enhancement, proposed by us, uses some known steps, namely connected with information lost minimization at the transition to fuzzy domain and noise influence suppression. Lets consider them further.

2.1 One Parameter Optimization of Membership Function

In the Ref. [1], which was taken for the basis of our modification of image contrast enhancement approach, was proposed the way of selection for the S-type function such one, which is specified by parameters a, b, and c and is defined as

$$\mu_X(x_{mn}) = S(x_{mn}, a, b, c) = \begin{cases} 0, & if\ x \le a, \\ \dfrac{(x-a)^2}{(b-a)(c-a)}, & if\ \ a < x \le b, \\ 1 - \dfrac{(c-x)^2}{(c-b)(c-a)}, & if\ \ b < x \le c, \\ 1, & if\ \ x > c, \end{cases} \tag{1}$$

where x_{mn} – gray level value for the pixel of input image, which is considered as a set of fuzzy singletons, with dimension $M \times N$, $m = 1, 2, \ldots M$, $n = 1, 2, \ldots N$.

Notice that expression (1) is a generalization of the following membership functions

$$\mu(x) = S(x, a, b, c) = \begin{cases} 1 - 2(1-x)^2 & , \quad if\ x \le 0.5, \\ 2x^2 & , \quad if\ \ x > 0.5, \end{cases} \tag{2}$$

and [13, 14]

$$\mu(x) = S(x, a, b, c) = \begin{cases} 1 - 2^{n-1}(1-x)^n & , \quad if\ x \le 0.5, \\ 2^{n-1}x^n & , \quad if\ \ x > 0.5, \end{cases} \tag{3}$$

where $n \in N$.

However, the class of membership functions can be significantly expanded by the nonlinear transformation

$$\mu^*(x) = F\left(\mu(x)\right), \tag{4}$$

satisfying requirements:

$$F\left(\mu(x)\right) \in [0, 1], \quad F\left(\mu(x_2)\right) \ge F\left(\mu(x_1)\right) \ \text{ when } \ x_2 \ge x_1.$$

That is those transformations create new monotone membership function $\mu^*(x)$ with range [0,1].

As an example for the nonlinear transformation function $F(\cdot)$ power function with positive real power γ can be chosen:

$$F(\mu(x)) = (\mu(x))^{\gamma}. \tag{5}$$

Figure 1 demonstrates shapes of function (3) with transformation by power function (5) depending on values of exponent n and γ.

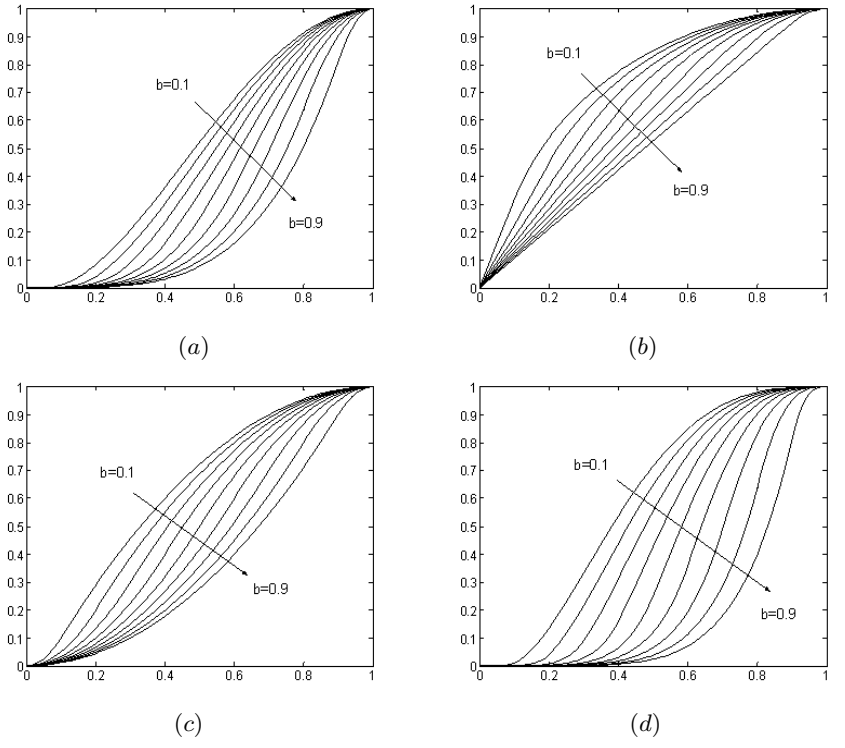


Fig. 1. Shape of membership function (5) depending on values of powers n and γ: (*a*) n=2, γ=2; (*b*) n=2, γ=0,5; (*c*) n=2, γ=1; (*d*) n=3, γ=2

It follows that S-function is integer-order, mainly second-order [5]. At the same time we'll notice, that parameters a and c are chosen from the reflection of noise influence minimization [1], and parameter b is determined as the solution of optimization problem, reasoning from the initial conditions $b \in [a+1, c-1]$ and find an optimum value b_{opt} such that fuzzy entropy H, by which the transformed by means of membership function image is estimated, is maximized:

$$H_{\max}(X; a, b_{opt}, c) = \max\{H(X; a, b, c \mid L_{\min} \le a < b < c \le L_{\max})\}, \tag{6}$$

where $L_{\min}$, $L_{\max}$ minimum and maximum grayscale values of input image elements. Entropy of the fuzzy set is defined as

$$H(x) = \frac{1}{MN} \sum_{n=1}^{N} \sum_{m=1}^{M} S_f(\mu_X(x_{mn})), \tag{7}$$

where $S_f(\cdot)$ is a Shannon function

$$S_f(\mu_X(x_{mn})) = -\mu_X(x_{mn}) \log_2 \mu_X(x_{mn}) - (1 - \mu_X(x_{mn})) \log_2 (1 - \mu_X(x_{mn})). \tag{8}$$

The drawback of the known method is the fact that optimization of membership function for the a and c parameters determination is held only by the way of choosing such a parameter b, at which fuzzy entropy (7) of transformed image in fuzzy domain is maximized. At that, fact that S-shaped function can be not necessary second order, but another one, including fractional, isn't taken into consideration. That's why the optimization should be performed both at the selection of $b \in [a+1, c-1]$ and at another parameter – selection of order α of S^{α}-function, such that $\alpha \in [\alpha_{min}, \alpha_{max}]$. Furthermore, a new membership function (Fig.1) gained by nonlinear transformation (4) can be used.

2.2 Membership Functions of Real Order

In [11] we enlarged the possibility of construction of S-type membership functions by adding to the existing S-functions of integer order – second, third, fourth [5], new ones of real order S^{α} for which the power α is positive real number $\alpha \geq 1$. At that the function $S^{\alpha}(x)$ is constructed in such a way, that at position of joint of two units of power functions derivative $dS^{\alpha}(x)/dx$ exists and is a continuous function.

3 Three Stage Image Processing

Three-stage image processing consists in optimal transformation of input image from spatial domain into fuzzy one through calculation of optimal membership function. At the second stage the amplification of local contrasts at the fuzzy domain is performed. At the third stage the improved image is reconstructed by the way of its transformation from fuzzy domain to spatial one.

3.1 Two-Parameter Optimization of Membership Function

The minimum α_{min} and maximum α_{max} values of power α of membership function order were specified by us for the two-stage optimization of membership function, described as

$$\mu_X^{(\alpha)}(x; a, b, c) = \begin{cases} 0, & if\ x \leq a, \\ \dfrac{(x-a)^{\alpha}}{(b-a)^{\alpha-1}(c-a)}, & if\ \ a < x \leq b, \\ 1 - \dfrac{(c-x)^{\alpha}}{(c-b)^{\alpha-1}(c-a)}, & if\ \ b < x \leq c, \\ 1, & if \quad x > c. \end{cases} \tag{9}$$

After that the optimal value of parameter b, which maximizes fuzzy entropy, was determined for the chosen membership function $\mu_X^{(\alpha)}$

$$H^{\alpha}_{\max}(X; a, b_{opt}, c, \alpha) = \max\left\{H(X; a, b, c, \alpha \ | L_{\min} \le a < b < c \le L_{\max})\right\}, \tag{10}$$

and for the optimal maximal among all $H^{\alpha}_{\max}$ was chosen:

$$H_{\max} = \max_{\alpha_{\min} \le \alpha \le \alpha_{\max}} \{H^{\alpha}_{\max}(X; a, b, c, \alpha).\} \tag{11}$$

Thus the optimal fuzzy membership function $\mu_X^{(\alpha)}$ is selected.

3.2 Local Contrast Intensification in Fuzzy Domain

Since the goal of this paper is widening of range of halftone image contrast intensification in general, and preserving the structure of fine details, the different from accepted in [1] way of local contrast intensification in fuzzy domain was chosen by us. Approach described in [10] was taken for the basis.

Local contrast of image element in fuzzy domain is defined as in [1]

$$C_{\mu}(x_{mn}) = \frac{|\mu^{\alpha}(x_{mn}) - E_{\mu}(x_{mn})|}{[\mu^{\alpha}(x_{mn}) + E_{\mu}(x_{mn})]}, \tag{12}$$

where $E_{\mu}(x_{mn})$ – mean gray value in moving window W of size $r \times s$ in fuzzy domain for pixel with coordinates (m, n).

Nonlinear enhancement of local contrast was held using power function

$$C^{*}_{\mu}(x_{mn}) = \left(C_{\mu}(x_{mn})\right)^{\sigma}, \tag{13}$$

but the averaged value was reconstructed in accordance with [11]

$$E'_{\mu}(x_{mn}) = \begin{cases} \mu^{\alpha}(x_{mn}) \dfrac{\left(1 + C^{*}_{\mu}(x_{mn})\right)}{\left(1 - C^{*}_{\mu}(x_{mn})\right)} : \mu^{\alpha}(x_{mn}) \le E_{\mu}(x_{mn}), \\ \mu^{\alpha}(x_{mn}) \dfrac{\left(1 - C^{*}_{\mu}(x_{mn})\right)}{\left(1 + C^{*}_{\mu}(x_{mn})\right)} : \mu^{\alpha}(x_{mn}) > E_{\mu}(x_{mn}), \end{cases} \tag{14}$$

After that the reconstruction of enhanced fuzzy value of image element magnitude was held by the expression

$$\mu^{*}(x_{mn}) = \mu^{\alpha}(x_{mn}) + \beta \cdot r \cdot s \cdot (E_{\mu}(x_{mn}) - E'_{\mu}(x_{mn})), \tag{15}$$

where β is normalizing factor.

3.3 Reconstruction of Enhanced Image

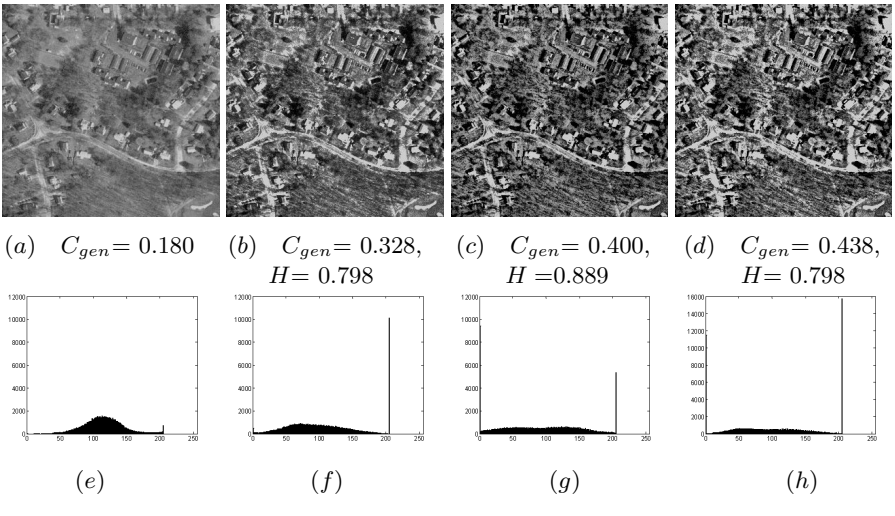
For the reconstruction of enhanced image the described in [11] formula of transition from fuzzy domain to spatial one by the function inverse to membership function (9) is used:

$$x_{ij}^{*} = \begin{cases} L_{\min}, & \mu^{*}(x_{ij}) = 0, \\ L_{\min} + \dfrac{L_{\max} - L_{\min}}{(c-a)} \left[(b-a)^{\alpha-1}(c-a)\mu^{*}(x_{ij})\right]^{\frac{1}{\alpha}}, & \\ & 0 < \mu^{*}(x_{ij}) \le \dfrac{b-a}{c-a}, \\ L_{\min} + \dfrac{L_{\max} - L_{\min}}{(c-a)} \left\{c - a - \left[(c-b)^{\alpha-1}(c-a)(1-\mu^{*}(x_{ij}))\right]\right\}^{\frac{1}{\alpha}}, & \\ & \dfrac{b-a}{c-a} \le \mu^{*}(x_{ij}) < 1, \\ L_{\max}, & \mu^{*}(x_{ij}) = 1, \end{cases} \tag{16}$$

that allows to form image with enhanced local contrasts and amplified image contrast as a whole.

4 Experimental Studies

The described approach (9)-(16) is illustrated on the example of contrast enhancement of *Orthophoto* image. The results of those investigations are presented in the Figure 2. Sliding window W of size $r = s$=15 and local contrast enhancement (13) with power σ=0.75 and reconstruction with $\beta = 0.01$ were used. The generalized contrast C_{gen} [12] and fuzzy entropy H were calculated for every image, that additionally demonstrates effectiveness of proposed approach. The value α, which is the order of membership function $\mu_x^{(\alpha)}$(11), is also specified. At the Fig. 2 are presented: (*a*) – *Orthophoto* image; (*e*) – histogram of image (*a*); (*b*) – result of image (*a*) processing by method [1]; (*f*) – histogram of image (*b*); (*c*) – result of image (*a*) processing by the proposed method (α=1.1); (*g*) – histogram of image (*c*); (*d*) – result of image (*a*) processing by the proposed method without the optimization by the parameter α; (*h*) – histogram of image (*g*).



(*a*) C_{gen}= 0.180 (*b*) C_{gen}= 0.328, H= 0.798 (*c*) C_{gen}= 0.400, H =0.889 (*d*) C_{gen}= 0.438, H= 0.798

(*e*) (*f*) (*g*) (*h*)

Fig. 2. Contrast enhancement for image *Orthophoto* and their histograms

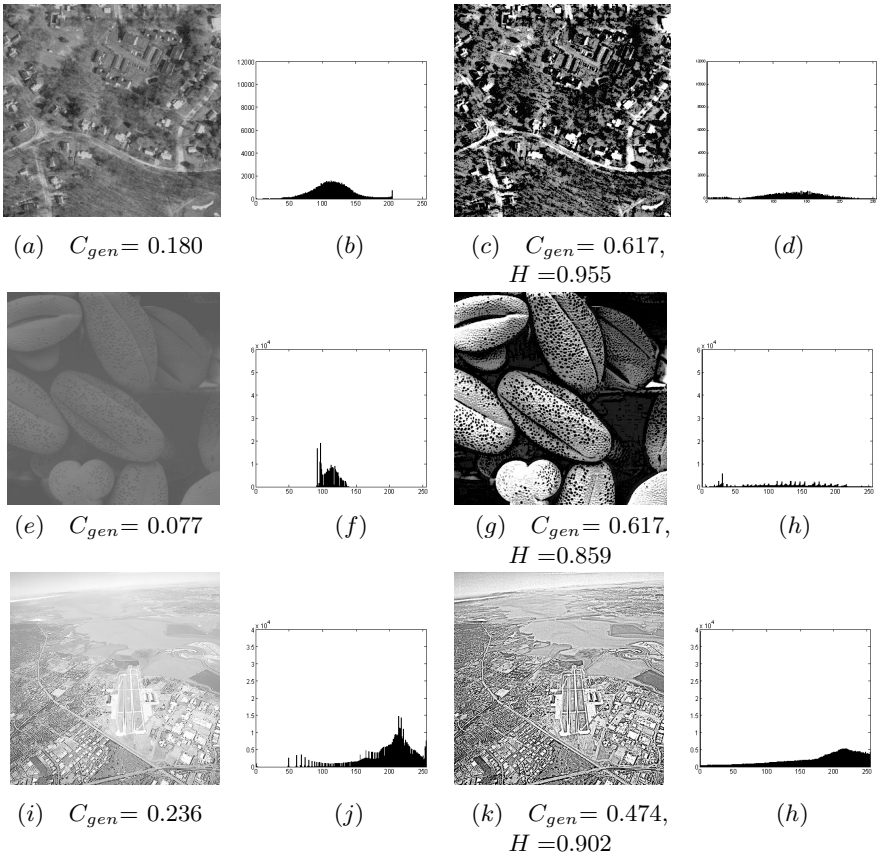


Fig. 3. Contrast enhancement, using approach with power transformation (5) of membership function (9) – input images, enhanced images and their histograms

The analysis of images (a), (b), (c) and (d) confirms effectiveness of proposed approach by the receiving of greater values of generalized contrast and better amplification of pure contrast images. The effectiveness of two parameters optimization of fuzzy membership function is well illustrated by Fig. 2 (c) and (d). The last one is received without optimization by the parameter α that allowed getting greater generalized contrast C_{gen} then in the Fig. 2 (c), but at the same time lower contrast of fine details.

The results of two-parameter optimization with the usage of membership function nonlinear transformation (5), in particular power function, are shown in Figure 3 on the examples of *Orthophoto*, *Beans* and *Town* images. Sliding window $W$of size $r = s$=15 and local contrast enhancement (9) with power σ=0.75 and $\beta = 0.01$ were used. At the Fig. 3 are presented: (a), (e), (i) – original images, (b), (f), (j) – histograms of original images, : (c), (g), (k) – results

of images (a), (e), (i) processing, (d), (h), (l) – histograms of images (c), (g), (k) accordingly. Values of generalized contrast $C_{gen}$and fuzzy entropy H are shown.

Some artifacts can appear while processing images with homogeneous background. We propose method for such artifact suppression which consist in modification of image reconstruction in fuzzy domaine (11):

$$\mu'^{\alpha}(x_{mn}) = \begin{cases} \mu^{\alpha}(x_{mn}) + q \cdot \beta \cdot r \cdot s \cdot \left| \frac{1}{q} \left(E_{\mu}(x_{mn}) - E'_{\mu}(x_{mn}) \right) \right|^{\lambda} \\ \qquad\qquad \left| E_{\mu}(x_{mn}) - E'_{\mu}(x_{mn}) \right| \leq q, \\ \mu^{\alpha}(x_{mn}) + \beta \cdot r \cdot s \cdot (E_{\mu}(x_{mn}) - E'_{\mu}(x_{mn})), \\ \qquad\qquad \left| E_{\mu}(x_{mn}) - E'_{\mu}(x_{mn}) \right| > q, \end{cases} \tag{17}$$

where $\lambda \in (1, 2]$, $q \in [0.1, 0.5]$ –coefficients of artifacts formation decrease on homogeneous regions of large size. It is necessary to select specified values of parameters λ and q for each image individually.

Essence of this modification can be illustrated by the graph shown on Fig.4.

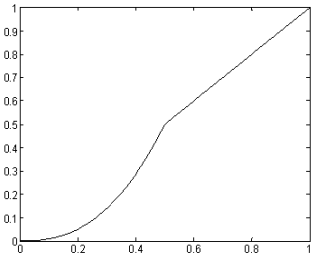


Fig. 4. Graph of reconstruction modification (17) for artifacts suppression (λ=1.2, q=0.5)

Variation of additional parameter λ allows to reduce artifacts in enhanced image.

This algorithm is illustrated on the example of *Capitol* image on Fig.5. Here (a) is an original image; (d) – its histogram; (b) – image (a) enhanced using approach with power transformation of membership function (5) (α=0.5, γ=2, β = 0.01); (e) – its histogram; (c) – image (a) enhanced using approach with power transformation of membership function (5) and reconstruction modification in fuzzy domain (17) (α=0.5, γ=2, β = 0.01, λ=1.4); (g) – its histogram. From comparison of images (b) and (c) one can see that proposed method significantly reduces artifacts on homogeneous background, though slightly decreases a value of generalized contrast.

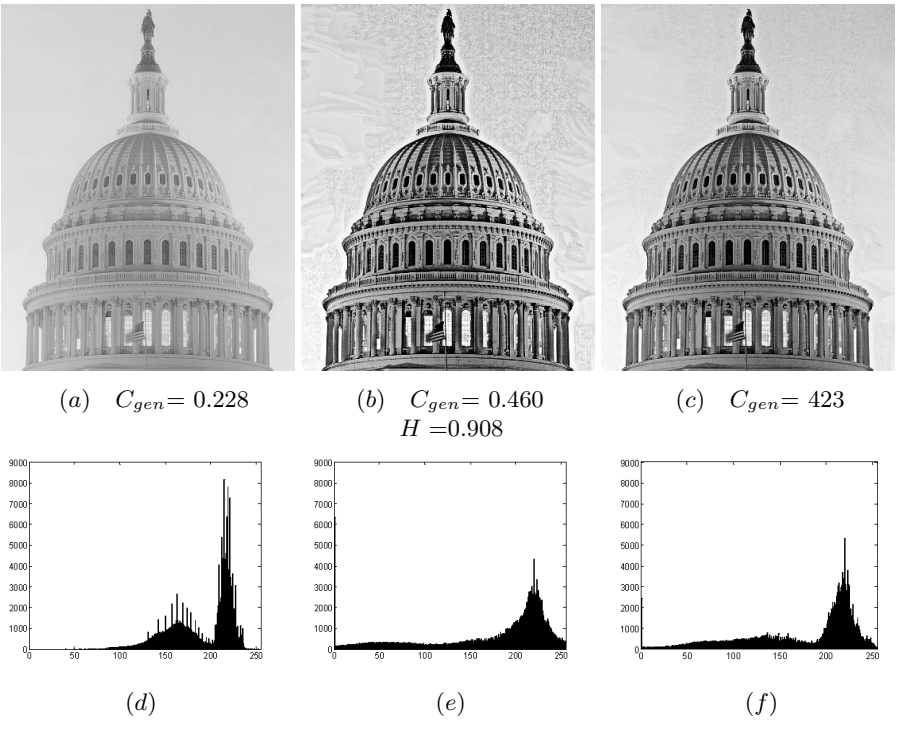


Fig. 5. Illustration of algorithm for artifact removal

5 Conclusion

A new approach of two-parameter optimization of membership function by the maximum of fuzzy entropy of halftone image criterion is proposed. It is realized by the construction of membership function of real order. It allows to reach greater values of generalized contrast in transformed images at simultaneous preserving of fine details that made these images more informative and more acceptable for the analysis. Proposed approach to the construction of new membership functions through nonlinear transformation of classical S-type membership functions allows to enlarge the region of search of optimal transition to fuzzy domain by the principal of fuzzy entropy maximum. It becomes apparent in better visualization of fine details on the images with poor contrast, regardless of the dark or light background of these details. If image has homogenous object of large size, which can be interpreted as background than developed method decreases possibility of artifact formation through abundant intensification of small differences of pixel local contrast.

References

1. Cheng, H.D., Huijuan Xu: A novel fuzzy logic approach to contrast enhancement. Pattern Recognition. – Vol. 36, No. 5, pp. 809–819, 2000.
2. Cheng, H.D., Mei Xue, Shi, X.J.: Contrast enhancement based on a novel homogeneity measurement. Pattern Recognition. – Vol. 36, No. 4, pp. 2687–2697, 2003.
3. Choi, Y.S., Krishnapuram, R.: A robust approach to image enhancement based on fuzzy logic. IEEE Trans. on Image Processing. – Vol. 6, No. 10, pp. 811-825, 1997.
4. Gonzales R.C., Woods R.E.: Digital Image Processing. Prantis Hall, Upper Saddle River, 2002.
5. Li, H., Yang, H.S.: Fast and reliable image enhancement using fuzzy relaxation technique. IEEE Transactions on Systems, Man and Cybernetics, Vol. 19, No. 5, pp. 1276-1281, 1989.
6. Pal S.K., King R.A.: Image Enhancement using fuzzy set. Electronics Letters. Vol. 16, No. 10, pp. 376-378, 1980.
7. Sattar, F., Tay, D.B.H.: Enhancement of document images using multiresolution and fuzzy logic techniques. IEEE Signal Processing Letters. – Vol. 6, No. 6, pp.811-825, 1999.
8. Tizhoosh, H.R.: Fuzzy image enhancement: an overview. Kerre, E, Nachtegal, M. (Eds.): Fuzzy Techniques in Image Processing, Springer, Studies in Fuzziness and Soft Computing, pp. 137-171, 2000.
9. Tizhoosh, H.R., Michaelis, B.: Image enhancement based on fuzzy aggregation techniques. Proc. 16th Int. conference IEEE IMTC 99, Venice, Italy, Vol. 3, pp. 1813-1817, 1999.
10. Vorobel, R.A.: A method for image reconstruction with contrast improvement. Information Extraction and Processing. Issue 17(93), pp. 122-126, 2002.
11. Vorobel, R., Datsko, O.: Image contrast improvement using change of membership function parameters. Bulletin of National University "Lvivska Politechnika". Computer engineering and Information Technologies, No. 433, pp. 233-238, 2001.
12. Vorobel, R.A.: Perception of the subject images and quantitative evaluation of their contrast based on the linear description of elements of contrast. Reports of the Ukrainian Academy of Sciences. No. 9, pp. 103-108, 1998.
13. Kacprzyk J. Fuzzy sets in system analysis. Warsaw, PWN, 1986. (In Polish).
14. C. Alsina, E. Trillas, L. Valverde, Do we need Max, Min and 1-j in Fuzzy Sets Theory? Fuzzy Sets and Possibility Theory, Ed. R. Yager, Pergamon, New York, pp. 275-297 1982.

Non-negative Matrix Factorization with Quasi-Newton Optimization

Rafal Zdunek* and Andrzej Cichocki**

Laboratory for Advanced Brain Signal Processing
BSI, RIKEN, Wako-shi, Japan

Abstract. Non-negative matrix factorization (NMF) is an emerging method with wide spectrum of potential applications in data analysis, feature extraction and blind source separation. Currently, most applications use relative simple multiplicative NMF learning algorithms which were proposed by Lee and Seung, and are based on minimization of the Kullback-Leibler divergence and Frobenius norm. Unfortunately, these algorithms are relatively slow and often need a few thousands of iterations to achieve a local minimum. In order to increase a convergence rate and to improve performance of NMF, we proposed to use a more general cost function: so-called Amari alpha divergence. Taking into account a special structure of the Hessian of this cost function, we derived a relatively simple second-order quasi-Newton method for NMF. The validity and performance of the proposed algorithm has been extensively tested for blind source separation problems, both for signals and images. The performance of the developed NMF algorithm is illustrated for separation of statistically dependent signals and images from their linear mixtures.

1 Introduction and Problem Formulation

Non-negative matrix factorization (NMF) [1, 2, 3, 4, 5] decomposes the data matrix $\boldsymbol{Y} = [\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(K)] \in \mathbb{R}^{M \times K}$ as a product of two matrices $\boldsymbol{A} \in \mathbb{R}^{M \times R}$ and $\boldsymbol{X} = [\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(K)] \in \mathbb{R}^{R \times K}$ having only non-negative elements. Although some decompositions or matrix factorizations provide an exact reconstruction data (i.e., $\boldsymbol{Y} = \boldsymbol{AX}$), we shall consider here decompositions which are approximative in nature, i.e.,

$$\boldsymbol{Y} = \boldsymbol{AX} + \boldsymbol{V}, \qquad \boldsymbol{A} \geq 0, \;\; \boldsymbol{X} \geq 0 \tag{1}$$

or equivalently $\boldsymbol{y}(k) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{v}(k), \;\; k = 1, 2, \ldots, K$ or in a scalar form as $y_m(k) = \sum_{r=1}^{R} a_{mr} x_r(k) + \nu_m(k), \;\; m = 1, \ldots, M,$ where $\boldsymbol{V} \in \mathbb{R}^{M \times K}$ represents noise or error matrix, $\boldsymbol{y}(k) = [y_1(k), \ldots, y_M(k)]^T$ is a vector of the observed signals (typically nonnegative) at the discrete time instants[1] k while

* On leave from Institute of Telecommunications, Teleinformatics and Acoustics, Wroclaw University of Technology, Poland.

** On leave from Warsaw University of Technology, Poland.

[1] The data are often represented not in the time domain but in a transform domain such as the time frequency domain, so index k may have different meaning.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 870–879, 2006.
© Springer-Verlag Berlin Heidelberg 2006

$\boldsymbol{x}(k) = [x_1(k), \ldots, x_R(k)]^T$ is a vector of components or source signals at the same time instant [6]. Our objective is to estimate the mixing (basis) matrix $\boldsymbol{A}$ and sources $\boldsymbol{X}$ subject to nonnegativity constraints all entries. Usually, in Blind Source Separation (BSS), to which NMF is applied in this paper, $K >> M \geq R$ and R is known or can be relatively easily estimated using SVD or PCA. Through this paper, we use the following notations: $x_r(k) = x_{rk}$, $y_m(k) = y_{mk}$ and $z_{mk} = [\boldsymbol{AX}]_{mk}$ means mk-element of the matrix $(\boldsymbol{AX})$, the mr-th element of the matrix $\boldsymbol{A}$ is denoted by a_{mr}.

The basic approach to NMF is alternating minimization or alternating projection: the specified loss function is alternately minimized with respective to two sets of the parameters $\{x_{rk}\}$ and $\{a_{mr}\}$, each time optimizing one set of arguments while keeping the other one fixed [7, 2, 6].

One of the NMF algorithms, which was proposed by Lee and Seung [2], alternatively minimizes the Kulback-Leibler (KL) divergence

$$D_{KL}(\boldsymbol{AX}||\boldsymbol{Y}) = \sum_{mk} \left(y_{mk} \log \frac{y_{mk}}{[\boldsymbol{AX}]_{mk}} + [\boldsymbol{AX}]_{mk} - y_{mk} \right) \tag{2}$$

$$\text{s. t.} \quad x_{rk} \geq 0, \quad a_{mr} \geq 0, \quad \|\boldsymbol{a}_r\|_1 = \sum_{m=1}^{M} a_{mr} = 1.$$

with multiplicative update rules based on a gradient descent approach [6]. This leads to the following algorithm

$$x_{rk} \leftarrow x_{rk} \frac{\sum_{m=1}^{M} a_{mr} \, (y_{mk}/[\boldsymbol{A}\,\boldsymbol{X}]_{mk})}{\sum_{q=1}^{M} a_{qr}}, \tag{3}$$

$$a_{mr} \leftarrow a_{mr} \frac{\sum_{k=1}^{K} x_{rk} \, (y_{mk}/[\boldsymbol{A}\,\boldsymbol{X}]_{mk})}{\sum_{p=1}^{K} x_{rp}}. \tag{4}$$

This algorithm extends (by alternating minimization) the well-known EMML or Richardson-Lucy algorithm (RLA) [8]. Another Lee-Seung algorithm minimizes the square Euclidean distance (Frobenius norm) with the same alternating approach.

The multiplicative descent algorithms are known to be very slowly-convergent and easily stuck in local minima. One of the way to speed up the convergence is to modify the learning rate in an iterative scheme. In this paper, we address this issue with second-order approximations of the loss function, i.e. with the quasi-Newton method.

2 Quasi-Newton Optimization

The KL divergence (2) is a particular case of the Amari alpha-divergence [9, 10, 11] defined as

$$D_A(\boldsymbol{AX}||\boldsymbol{Y}) = \sum_{mk} y_{mk} \frac{(y_{mk}/z_{mk})^{\alpha-1} - 1}{\alpha(\alpha-1)} + \frac{z_{mk} - y_{mk}}{\alpha}, \quad z_{mk} = [\boldsymbol{AX}]_{mk} \tag{5}$$

This case takes place if $\alpha \to 1$, and for $\alpha \to 0$ the dual KL can be derived. For $\alpha = 2, 0.5, -1$, we obtain the Pearson's, Hellinger and Neyman's chi-square distances, respectively.

Applying the quasi-Newton method to (5), we have

$$\boldsymbol{X} \leftarrow \left[\boldsymbol{X} - [\boldsymbol{H}_{D_A}^{(\mathbf{X})}]^{-1}\nabla_{\mathbf{X}} D_A\right]_{\epsilon}, \quad \boldsymbol{A} \leftarrow \left[\boldsymbol{A} - [\boldsymbol{H}_{D_A}^{(\mathbf{A})}]^{-1}\nabla_{\mathbf{A}} D_A\right]_{\epsilon}, \tag{6}$$

where $\boldsymbol{H}_{D_A}^{(\mathbf{X})}$ and $\boldsymbol{H}_{D_A}^{(\mathbf{A})}$ are Hessians, $\nabla_{\mathbf{X}} D_A$ and $\nabla_{\mathbf{A}} D_A$ are gradients matrices for (5) with respect to $\boldsymbol{X}$ and $\boldsymbol{A}$, respectively. The nonlinear operator $[\cdot]_{\epsilon} = \max\{\cdot, \epsilon\}$ enforces nonnegativity.

The gradients with respect to $\boldsymbol{X}$ can be expressed as

$$\boldsymbol{G}_{D_A}^{(\mathbf{X})} = \nabla_{\mathbf{X}} D_A = \frac{1}{\alpha}\boldsymbol{A}^T\left(\underline{1} - (\boldsymbol{Y}./(\boldsymbol{A}\boldsymbol{X}))^{\alpha}\right) \in \mathbb{R}^{R\times K}, \tag{7}$$

where ./ is a Hadamard division. The Hessian has the form: $\forall i \in \{1,\ldots,R\}, j \in \{1,\ldots,K\}$:

$$[\boldsymbol{H}_{D_A}^{(\mathbf{X})}]_{ij} = \frac{\partial^2 D_A}{\partial x_{rk}\partial x_{ij}} = \begin{cases} \sum_{m=1}^{M} \frac{a_{mr}\, y_{mk}^{\alpha}\, a_{mi}}{(\sum_{s=1}^{R} a_{ms}\, x_{sk})^{\alpha+1}}, & \text{for } j=k,\, i=s \\ 0, & \text{otherwise} \end{cases}, \tag{8}$$

or in a block matrix

$$\boldsymbol{H}_{D_A}^{(\mathbf{X})} = \frac{1}{\alpha}\,\mathrm{diag}\{[\boldsymbol{h}_k^{(\mathbf{X})}]_{k=1,\ldots,K}\} \in \mathbb{R}^{RK\times RK} \tag{9}$$

where

$$\boldsymbol{h}_k^{(\mathbf{X})} = \boldsymbol{A}^T\,\mathrm{diag}\{[\boldsymbol{Y}^{\alpha}./(\boldsymbol{A}\boldsymbol{X})^{\alpha+1}]_{*,k}\}\boldsymbol{A} \in \mathbb{R}^{R\times R}$$

Similarly for $\boldsymbol{A}$, we get

$$\boldsymbol{G}_{D_A}^{(\mathbf{A})} = \nabla_{\mathbf{A}} D_A = \frac{1}{\alpha}\left(\underline{1} - (\boldsymbol{Y}./(\boldsymbol{A}\boldsymbol{X}))^{\alpha}\right)\boldsymbol{X}^T \in \mathbb{R}^{M\times R}. \tag{10}$$

The Hessian has the form: $\forall i \in \{1,\ldots,M\}, j \in \{1,\ldots,R\}$:

$$[\boldsymbol{H}_{D_A}^{(\mathbf{A})}]_{ij} = \frac{\partial^2 D_A}{\partial a_{mr}\partial a_{ij}} = \begin{cases} \sum_{k=1}^{K} \frac{x_{rk}\, y_{mk}^{\alpha}\, x_{jk}}{(\sum_{s=1}^{R} a_{ms}\, x_{sk})^{\alpha+1}}, & \text{for } j=s,\, i=m \\ 0, & \text{otherwise} \end{cases}, \tag{11}$$

or in a block matrix

$$\boldsymbol{H}_{D_A}^{(\mathbf{A})} = \frac{1}{\alpha}\,\mathrm{diag}\{[\boldsymbol{h}_m^{(\mathbf{A})}]_{m=1,\ldots,M}\} \in \mathbb{R}^{MR\times MR} \tag{12}$$

where

$$\boldsymbol{h}_m^{(\mathbf{A})} = \boldsymbol{X}\,\mathrm{diag}\{[\boldsymbol{Y}^{\alpha}./(\boldsymbol{A}\boldsymbol{X})^{\alpha+1}]_{m,*}\}\boldsymbol{X}^T \in \mathbb{R}^{R\times R}$$

Since the Hessian is usually ill-conditioned, especially if we have sparse representations of the image to be estimated, some regularization of the Hessian is essential, which leads to a quasi-Newton iterations. We applied the Levenberg-Marquardt approach with a small fixed regularization parameter $\lambda = 10^{-12}$. Additionally we control the convergence by a slight relaxation of the iterative updates. To reduce substantially a computational cost, the inversion of the Hessian is replaced with the Q-less QR factorization computed with LAPACK. Thus the final form of the algorithm with the quasi-Newton algorithm is

$$\begin{aligned} \boldsymbol{X} &\leftarrow [\boldsymbol{X} - \gamma \boldsymbol{R}_{\mathbf{X}} \backslash \boldsymbol{W}_{\mathbf{X}}]_{\epsilon}, \qquad \boldsymbol{A} \leftarrow [\boldsymbol{A} - \gamma \boldsymbol{R}_{\mathbf{A}} \backslash \boldsymbol{W}_{\mathbf{A}}]_{\epsilon}, \\ \boldsymbol{W}_{\mathbf{X}} &= \boldsymbol{Q}_{\mathbf{X}}^T \nabla_{\mathbf{X}} D_A, \qquad \boldsymbol{Q}_{\mathbf{X}} \boldsymbol{R}_{\mathbf{X}} = \boldsymbol{H}_{D_A}^{(\mathbf{X})} + \lambda \boldsymbol{I}_{\mathbf{X}}, \\ \boldsymbol{W}_{\mathbf{A}} &= \boldsymbol{Q}_{\mathbf{A}}^T \nabla_{\mathbf{A}} D_A, \qquad \boldsymbol{Q}_{\mathbf{A}} \boldsymbol{R}_{\mathbf{A}} = \boldsymbol{H}_{D_A}^{(\mathbf{A})} + \lambda \boldsymbol{I}_{\mathbf{A}}, \end{aligned} \tag{13}$$

where $\boldsymbol{I}_{\mathbf{X}} \in \mathbb{R}^{RK \times RK}$, $\boldsymbol{I}_{\mathbf{A}} \in \mathbb{R}^{MR \times MR}$ are identity matrices, $\boldsymbol{R}_{\mathbf{X}}$ and $\boldsymbol{R}_{\mathbf{A}}$ are upper triangular matrices, and γ controls the relaxation. We set $\gamma = 0.9$. The $\backslash$ in (13) means the Gaussian elimination.

For $\alpha \to 0$, the Amari alpha-divergence converges to the dual I-divergence (generalized K-L divergence), i.e.

$$\begin{aligned} D_{KL2}(\boldsymbol{Y} || \boldsymbol{A}\boldsymbol{X}) &= \lim_{\alpha \to 0} D_A(\boldsymbol{A}\boldsymbol{X} || \boldsymbol{Y}) \\ &= \sum_{mk} \left(z_{mk} \log \frac{z_{mk}}{y_{mk}} + y_{mk} - z_{mk} \right), \quad z_{mk} = [\boldsymbol{A}\boldsymbol{X}]_{mk}, \end{aligned} \tag{14}$$

and consequently the gradient and Hessian matrices simplify as follows:

– For $\boldsymbol{X}$:

$$\boldsymbol{G}_{D_{KL2}}^{(\mathbf{X})} = \nabla_{\mathbf{X}} D_{KL2} = \boldsymbol{A}^T \log\left((\boldsymbol{A}\boldsymbol{X})./\boldsymbol{Y}\right) \in \mathbb{R}^{R \times K}, \tag{15}$$

and

$$\boldsymbol{H}_{D_{KL2}}^{(\mathbf{X})} = \operatorname{diag}\{[\boldsymbol{h}_k^{(\mathbf{X})}]_{k=1,\ldots,K}\} \in \mathbb{R}^{RK \times RK}, \tag{16}$$

where

$$\boldsymbol{h}_k^{(\mathbf{X})} = \boldsymbol{A}^T \operatorname{diag}\{[1./(\boldsymbol{A}\boldsymbol{X})]_{*,k}\} \boldsymbol{A} \in \mathbb{R}^{R \times R}.$$

– For $\boldsymbol{A}$:

$$\boldsymbol{G}_{D_{KL2}}^{(\mathbf{A})} = \nabla_{\mathbf{A}} D_{KL2} = \log\left((\boldsymbol{A}\boldsymbol{X})./\boldsymbol{Y}\right) \boldsymbol{X}^T \in \mathbb{R}^{M \times R}, \tag{17}$$

and

$$\boldsymbol{H}_{D_{KL2}}^{(\mathbf{A})} = \operatorname{diag}\{[\boldsymbol{h}_m^{(\mathbf{A})}]_{m=1,\ldots,M}\} \in \mathbb{R}^{MR \times MR}, \tag{18}$$

where

$$\boldsymbol{h}_m^{(\mathbf{A})} = \boldsymbol{X} \operatorname{diag}\{[1./(\boldsymbol{A}\boldsymbol{X})]_{m,*}\} \boldsymbol{X}^T \in \mathbb{R}^{R \times R}.$$

In each alternating step, the l_1-norm of the columns of $\boldsymbol{A}$ are normalized to a unity, i.e. we have: $a_{mr} \leftarrow \dfrac{a_{mr}}{\sum_{m=1}^{M} a_{mr}}$.

3 Fixed-Point Algorithm

In our application, $\boldsymbol{X}$ has much larger dimensions than $\boldsymbol{A}$, and hence, the computation of $\boldsymbol{X}$ with the Newton method may be highly time-consuming or even intractable, even though the Hessian is very sparse. Let us assume some typical case: $M = 20$, $R = 10$, and $K = 1000$. Thus the Hessian $\boldsymbol{H}^{(\mathbf{A})}$ has size 200 by 200 with $MR^2 = 2 \times 10^3$ non-zero entries, but the size of $\boldsymbol{H}^{(\mathbf{X})}$ is 10^4 by 10^4 with $KR^2 = 10^5$ non-zero entries. For this reason, we do not apply the Newton method for updating $\boldsymbol{X}$. This can be also justified by the fact that the computation of $\boldsymbol{A}$ needs to solve the system which is much more over-determined than for $\boldsymbol{X}$, and hence, this may be better done with the second order method since the information about the curvature of the cost function is exploited. In our approach, we apply the Newton method to the generalized cost function (Amari alpha-divergence).

In this paper, the sources $\boldsymbol{X}$ are basically estimated with the EMML and Fixed-Point (FP) algorithms.

In general, the FP algorithm solves a least-squares problem

$$\boldsymbol{X}^* = \arg\min_{\mathbf{X}} \left\{ \frac{1}{2} ||\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}||_F^2 \right\} \tag{19}$$

with the Moore-Penrose pseudo-inverse of a system matrix, i.e. in our case, the matrix $\boldsymbol{A}$. Since in NMF $M \geq R$, we formulate normal equations as $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{X} = \boldsymbol{A}^T\boldsymbol{Y}$, and the least-squares solution of minimal l_2-norm to the normal equations is $\boldsymbol{X}_{LS} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{Y} = \boldsymbol{A}^+\boldsymbol{Y}$, where $\boldsymbol{A}^+$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{A}$. The cost function given by the square Euclidean distance as in (19) works the best with Gaussian noise (matrix $\boldsymbol{V}$ in (1)), however, the computation of $\boldsymbol{A}$ uses the Amari alpha-divergence which is optimal for a wide spectrum of signal distributions.

The computation of $\boldsymbol{X}$ is usually improved with the prior knowledge about the source representations, such as sparsity and/or smoothing. The information about a structure of the estimated sources is usually incorporated to the cost function in the form of the additional term that regularizes the solution. Thus, the cost function in (19) is extended to the regularized squares Euclidean distance, and the problem to be solved becomes:

$$\boldsymbol{X}^* = \arg\min_{\mathbf{X}} \left\{ \frac{1}{2} ||\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}||_F^2 + \alpha_{\mathbf{X}} \Omega(\boldsymbol{X}) \right\}, \tag{20}$$

where $\Omega(\boldsymbol{X})$ is a regularization function, and $\alpha_{\mathbf{X}} \geq)$ is a regularization parameter. The minimal-norm least-square solution to (20) is given by:

$$\boldsymbol{X}_{LS} = (\boldsymbol{A}^T\boldsymbol{A} + \alpha_{\mathbf{X}}\boldsymbol{C})^{-1}\boldsymbol{A}^T\boldsymbol{Y}, \tag{21}$$

where $\boldsymbol{C} \in \mathbb{R}^{R \times R}$ is some discrete representation of the regularization term $\Omega(\boldsymbol{X})$.

There are many possibilities for defining $\Omega(\boldsymbol{X})$. For example, we have the basic Tikhonov regularization for $\Omega(\boldsymbol{X}) = ||\boldsymbol{X}||_F^2$, which leads to $\boldsymbol{C} = \boldsymbol{I}_R$, where

$\boldsymbol{I}_R \in \mathbb{R}^{R\times R}$ is an identity matrix. This operator enforces a smooth solution. In many applications, $\Omega(\boldsymbol{X})$ is a first or second derivative of the solution, or it is given by the Total Variation (TV) term. For more regularization operators, e.g. see [12, 13, 14].

Due to sparse solutions in NMF, we introduce some new approach, i.e. we assume that $\boldsymbol{C} = \boldsymbol{E} \in \mathbb{R}^{R\times R}$, where $\boldsymbol{E}$ is a matrix composed from all ones entries. The regularization parameter is set according to the exponential rule, i.e.

$$\alpha_{\mathbf{X}} = \alpha_{\mathbf{X}}^{(k)} = \alpha_0 \exp\{-\tau k\}, \tag{22}$$

where k is a number of alternating steps. This rule is motivated by a temperature schedule in the simulated annealing that steers the solution towards a global one. Thus, larger parameter α_0 and smaller τ should give better results but at the cost of high increase in a number of alternating steps. In our simulations, we set $\alpha_0 = 20$ and $\tau = 0.02$ for 1000 alternating steps.

Another simple approach that can be used for controlling sparsity of estimated variables is to apply nonlinear projections with suitable nonlinear monotonic functions. In this paper, the EMML updates are modified by a very simple nonlinear transformation $x_{rk} \leftarrow (x_{rk})^{1+\alpha_{sX}}$, $\forall k$, where α_{sX} is a small coefficient, typically from 0.001 to 0.005, and it is positive or negative to increase or decrease the sparseness, respectively.

Since the Newton method does not ensure a nonnegative solution, we enforce a nonnegative solution through a very simple nonlinear projection as in (6), which is applied to both sets of the arguments ($\boldsymbol{X}$ and $\boldsymbol{A}$). The same nonlinear projection is also applied to (21). Moreover, since $\boldsymbol{E}$ is singular, and $\boldsymbol{A}^T\boldsymbol{A}$ may be very ill-conditioned, especially for sparse solutions, the inversion in (21) is done with the Moore-Penrose pseudo-inverse instead of the standard one. Thus the updating of $\boldsymbol{X}$ in the $(k+1)$-th alternating step is performed with the novel algorithm:

$$\boldsymbol{X}^{(k+1)} \leftarrow \max\left\{\varepsilon, ([\boldsymbol{A}^T\boldsymbol{A}]^{(k)} + \alpha_{\mathbf{X}}^{(k)}\boldsymbol{E})^{+}[\boldsymbol{A}^T]^{(k)}\boldsymbol{Y}\right\}, \tag{23}$$

where $\boldsymbol{A}^{(k)}$ is the update of $\boldsymbol{A}$ from the k-th alternating step.

4 Results

The proposed algorithms have been extensively tested for many benchmarks of signals and images. The original 4 images [Fig. 1(a)] and the original 9 signals [Fig. 2 (a)] have been mixed with uniform distributions, where $\boldsymbol{A}^{(i)} \in \mathbb{R}^{9\times 4}$ and $\boldsymbol{A}^{(s)} \in \mathbb{R}^{18\times 9}$ are dense mixing matrices for the images and signals, respectively. The mixtures are shown in Figs. 1 (b) and 2 (b). The results obtained with the traditional Lee-Seung algorithm (3) and (4), which has been applied to estimation of both $\boldsymbol{A}$ and $\boldsymbol{X}$, are presented in Figs. 1 (c) and 2 (c). The separations are quantified with Signal-to-Interference Ratios (SIRs) that have the following

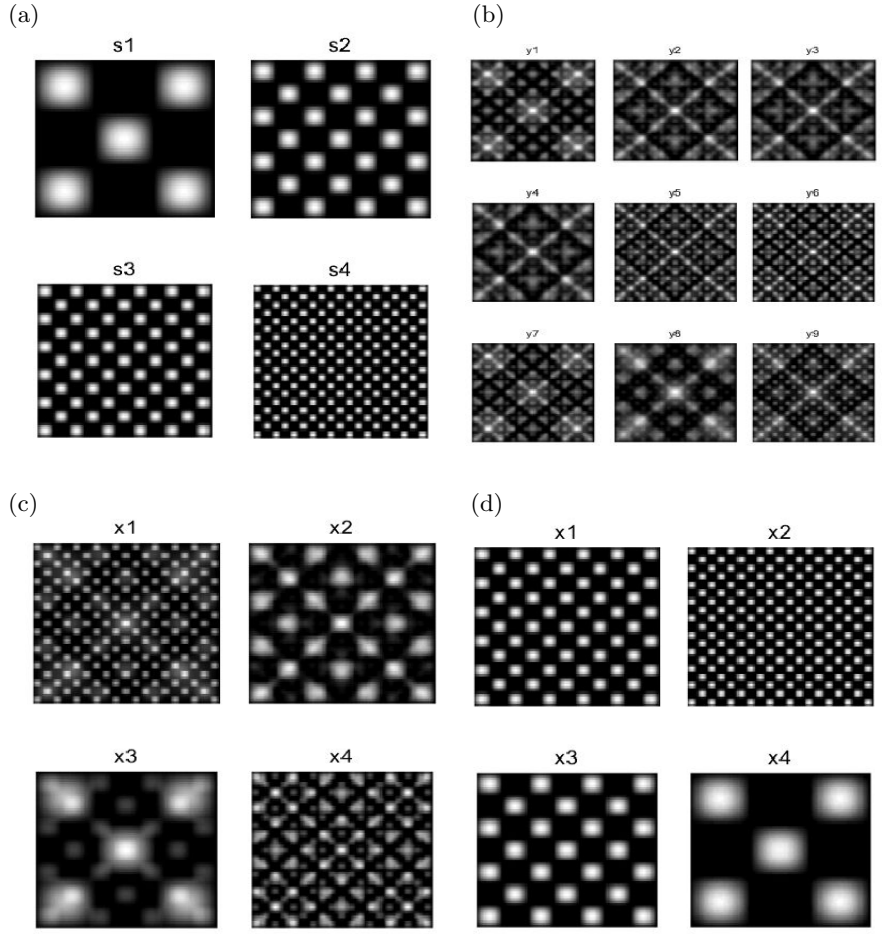


Fig. 1. Example 1: (a) Original 4 source images; (b) observed 9 mixed images; (c) Estimated source images using the standard Lee-Seung algorithm for KL function (2) (SIR = 5.5dB, 12.5dB, 9dB, 6dB, respectively); (d) Estimated source images using the new EMML-Newton algorithm for $\alpha = 2$ with nonlinear projection $\alpha_{sX} = 0.002$ with SIR=47dB, 45dB, 50dB, 44dB, respectively

values: 5.5, 12.5, 9, 6 [dB] for the images, and 1, 7.6, 5.9, 4.9, 4.4, 9.8, 5.9, 10.7, 3.7 [dB] for the signals. Applying the quasi-Newton method only to estimation of $\boldsymbol{A}$, and the nonlinear transformation with $\alpha_{sX} = 0.002$ to the same Lee-Seung algorithm (EMML) we obtained much better results which are shown in Fig. 1 (d). The SIRs are as follows: 47, 45, 50, 44 [dB]. The similar performance can be obtained for signals. Furthermore, the results can be even much better if the sources $\boldsymbol{X}$ are estimated with the Fixed-Point algorithm – see Fig. 2 (d). For this case, we have all the SIRs above 110 [dB], which is nearly a perfect result.

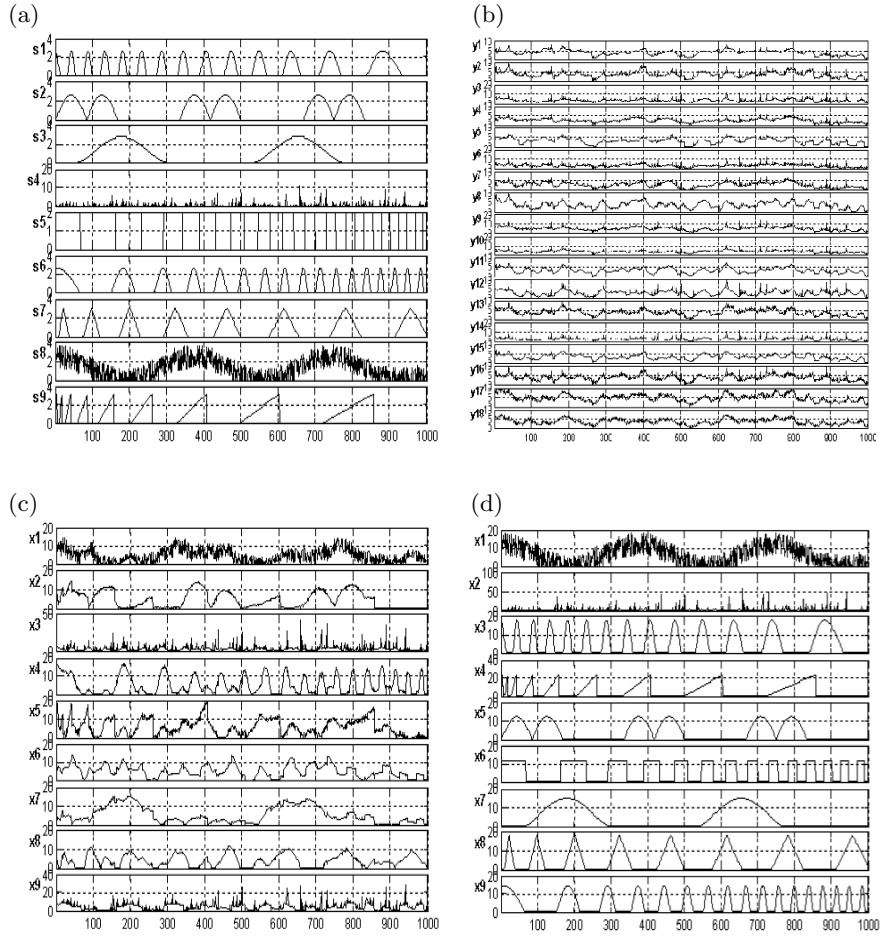


Fig. 2. Example 2: (a) Original 9 source signals; (b) observed 18 mixed signals; (c) Estimated source signals with the standard Lee-Seung algorithm for KL function (2) (SIR = 1, 7.6, 5.9, 4.9, 4.4, 9.8, 5.9, 10.7, 3.7 [dB], respectively); (d) Estimated source signals with new Fixed-Point – Quasi-Newton algorithm with SIR = 130.8, 126.6, 135.9, 129.6 135.9, 129.5, 133.7, 119.5, 137.4 [dB], respectively

For a sparse mixing matrix, the results can be even better. For estimating $\boldsymbol{A}$ we set $\alpha = 2$, but satisfactory results can be obtained for $\alpha \in [-1, 2]$.

The cost functions (2) and (19) are convex with respect to only one set of the arguments ($\boldsymbol{X}$ or $\boldsymbol{A}$). In the whole set of the arguments, both functions are non-convex, and hence, the alternating minimization may get stuck easily in local minima. To check how plausible are the single estimations given by the tested algorithms, the Monte Carlo analysis is carried out from 100 SIR samples in each case. We tested four different algorithms. Fig. 3 (a) presents the histogram of the SIR samples obtained with the traditional Lee-Seung algorithm. Applying

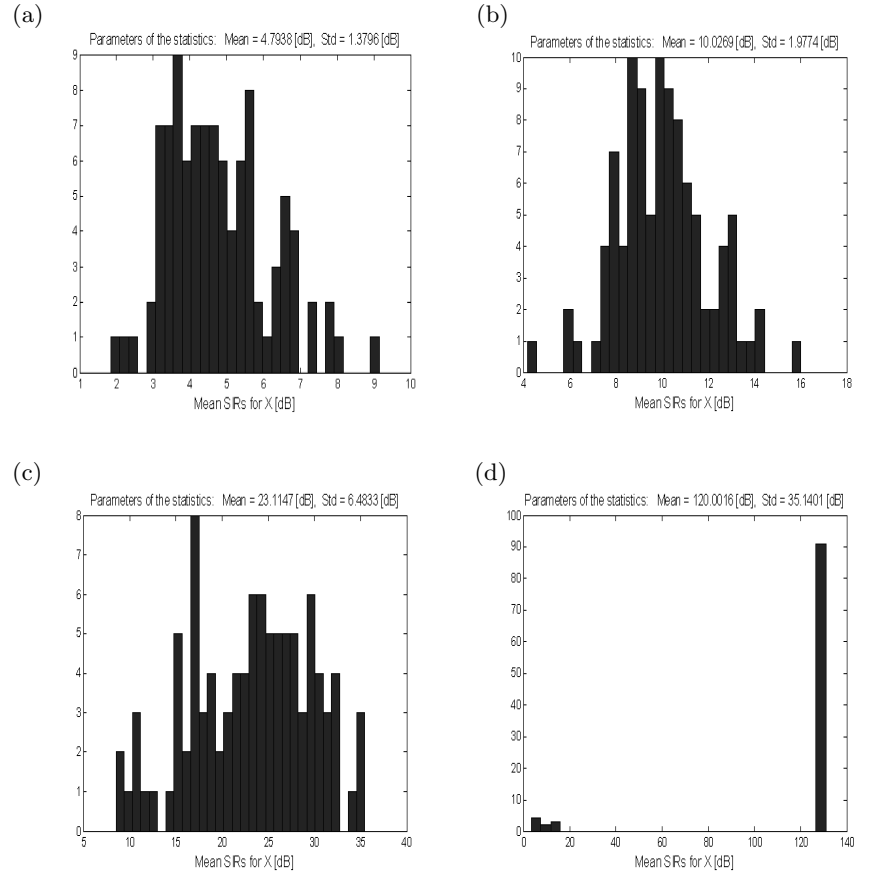


Fig. 3. Histograms from 100 SIR samples generated with the following algorithms initialized from uniformly distributed random initial matrices $\boldsymbol{A}$ and $\boldsymbol{X}$: (a) $\boldsymbol{A}$ – EMML, $\boldsymbol{X}$ – EMML; (b) $\boldsymbol{A}$ – Quasi-Newton, $\boldsymbol{X}$ – EMML; (c) $\boldsymbol{A}$ – Quasi-Newton, $\boldsymbol{X}$ – EMML with 3 inner iterations and $\alpha_{sX} = 0.001$; (d) $\boldsymbol{A}$ – Quasi-Newton, $\boldsymbol{X}$ – Fixed-Point (with exponential rule for damping parameter)

the quasi-Newton only to estimation of $\boldsymbol{A}$, the mean-SIR performance increased more than twice – see Fig. 3 (b). Then, improving the estimation of $\boldsymbol{X}$ with the nonlinear projection with $\alpha_{sX} = 0.001$, and using a few inner iterations for updating $\boldsymbol{X}$ in each alternating step, the performance substantially goes up [Fig. 3 (c)]. However, the extremely good performance is obtained with the hybrid connection of the FP and quasi-Newton algorithm, which is illustrated in Fig. 3 (d). The global solution with the exponential rule is reached 90% times for 100 trials, and such a good performance is not possible to get with the other tested algorithms. However, the rest 10% trials are still quite far from the desired solution, and this problem will be analyzed in our further research.

5 Conclusions

In this paper, we proposed a new hybrid algorithm for NMF, which demonstrates a very good performance. We have confirmed by the extensive simulations that the proposed algorithm can successfully separate signals and images, especially if a suitable regularization/projection is applied. Changing parameter α in the Amari alpha-divergence, we can tune the algorithm to minimize the influence of noisy disturbances in data. The free parameters in the exponential rule (22) steer the updates towards the global solution. All the parameters can be estimated from data, but this issue will be a subject of our future research. The detailed description of our other algorithms for NMF can be found in [15].

References

[1] Paatero, P., Tapper, U.: Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values. Environmetrics **5** (1994) 111–126
[2] Lee, D.D., Seung, H.S.: Learning of the parts of objects by non-negative matrix factorization. Nature **401** (1999) 788–791
[3] Dhillon, I., Sra, S.: Generalized nonnegative matrix approximations with Bregman divergences. In: Neural Information Proc. Systems, Vancouver, Canada (2005)
[4] Chu, M., Plemmons, R.J.: Nonnegative matrix factorization and applications. Bulletin of the International Linear Algebra Society **34** (2005) 2–7
[5] Pauca, V.P., Shahnaz, F., Berry, M.W., Plemmons, R.J.: Text mining using non-negative matrix factorizations. In: Proc. SIAM Inter. Conf. on Data Mining, Orlando, FL (2004)
[6] Cichocki, A., Amari, S.: Adaptive Blind Signal And Image Processing (New revised and improved edition). John Wiley, New York (2003)
[7] Amari, S.: Information geometry of the EM and em algorithms for neural networks. Neural Networks **8**(9) (1995) 1379–1408
[8] Byrne, C.: Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative (RBI) methods. IEEE Transactions on Image Processing **7** (1998) 100 – 109
[9] Amari, S.: Differential-Geometrical Methods in Statistics. Springer Verlag (1985)
[10] Cressie, N.A., Read, T.: Goodness-of-Fit Statistics for Discrete Multivariate Data. Springer, New York (1988)
[11] Cichocki, A., Zdunek, R., Amari, S.: Csiszar's divergences for non-negative matrix factorization: Family of new algorithms. LNCS **3889** (2006) 32–39
[12] Cullum, J.: The effective choice of the smoothing norm in regularization. Math. Comp. **3** (1979) 149–170
[13] Björck, Å.: Numerical Methods for Least Squares Problems. SIAM, Philadelphia (1996)
[14] Hansen, P.C.: Rank-Deficient and Discrete Ill-Posed Problems. SIAM, Philadelphia (1998)
[15] Cichocki, A., Zdunek, R.: NMFLAB for Signal Processing. Technical report, Laboratory for Advanced Brain Signal Processing, BSI RIKEN, Saitama, Japan (2006) http://www.bsp.brain.riken.jp.

Active Mining Discriminative Gene Sets (Invited)

Feng Chu and Lipo Wang

College of Information Engineering, Xiangtan University,
Xiangtan, Hunan, China
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
elpwang@ntu.edu.sg

Abstract. Searching for good discriminative gene sets (DGSs) in microarray data is important for many problems, such as precise cancer diagnosis, correct treatment selection, and drug discovery. Small and good DGSs can help researchers eliminate "irrelavent" genes and focus on "critical" genes that may be used as biomarkers or that are related to the development of cancers. In addition, small DGSs will not impose demanding requirements to classifiers, e.g., high-speed CPUs, large memorys, etc. Furthermore, if the DGSs are used as diagnostic measures in the future, small DGSs will simplify the test and therefore reduce the cost. Here, we propose an algorithm of searching for DGSs, which we call active mining discriminative gene sets (AM-DGS). The searching scheme of the AM-DGS is as follows: the gene with a large t-statistic is assigned as a seed, i.e., the first feature of the DGS. We classify the samples in a data set using a support vector machine (SVM). Next, we add the gene with the greatest power to correct the misclassified samples into the DGS, that is the gene with the largest t-statistic evaluated with only the mis-classified samples is added. We keep on adding genes into the DGS according to the SVM's mis-classified data until no error appears or overfitting occurs. We tested the proposed method with the well-known leukemia data set. In this data set, our method obtained two 2-gene DGSs that achieved 94.1% testing accuracy and a 4-gene DGS that achieved 97.1% testing accuracy. This result showed that our method obtained better accuracy with much smaller DGSs compared to 3 widely used methods, i.e., T-statistics, F-statistics, and SVM-based recursive feature elimination (SVM-RFE).

1 Introduction

Accurate classification of homogenous cancers is a key problem for disease diagnosis, treatment selection, pathology research, and drug discovery. In recent years, gene expression profiles have been extensively applied to classifying cancers at the molecular level [5,13,14]. A typical gene expression data set can be described as a high dimensional $n \times m$ matrix B. In B, each column stands for a cancer sample (i.e., an observation) and each row stands for a gene. Here

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 880–889, 2006.
© Springer-Verlag Berlin Heidelberg 2006

m usually ranges from several tens to over one hundred and n usually ranges from several thousands to tens of thousands. Since n is much larger than m, it is of great importance to select a group genes (rather than use all of them) for classification because of the following two points. First, among all the genes, only a part of them have discriminating power. Furthermore, some genes even act as "noise" and undermine the classification accuracy. Second, some genes are highly correlated and their expression profiles behave very similarly in classification. Excluding some of such correlated genes will reduce redundancy in the discriminative gene sets (DGS).

Since mid-1990s, a number of gene selection approaches [8,9,11,12,10,15] have been proposed. Most of these methods can be regarded as filter schemes [20], which first rank genes according to their discriminative ability and then select a certain number (e.g., 20, 50, or 100) of top-ranked genes for classification. Although these top-ranked genes can lead to highly accurate classification results, they may still contain great redundancy. Some other methods use wrapper scheme [20]. In [1], a support vector machine based recursive feature elimination method (SVM-RFE) is proposed, which eliminates unimportant genes (i.e., the genes with little or no discriminating power) or redundant genes one by one from the initial gene set that includes all the genes. Since the SVM-RFE usually has to eliminate several hundreds or thousands genes to obtain a final DGS, it requires a large amount of computing time. In [17], a method called *Markov blanket* was used to reduce redundancy in DGSs. Since the Markov blanket mainly focus on reducing redundancy, it does not guarantee that the resulting DGS has very good discriminating power. In [7], Wang *et al.* proposed a method that uses unsupervised clustering to identify the redundancy in DGSs and then reduced the redundancy by "collapsing dense clusters". They firstly rank all the genes and then select some top-ranked genes. After that, they cluster these "pre-selected" genes and pick out a representative gene for each cluster. The DGSs were formed using these representative genes. Although this method is able to reduce the redundancy of DGSs, the obtained DGSs are often not optimal because of the following reasons. (a) The cooperation among clusters and their representatives are not optimal; (b) A gene sometimes cannot represent the whole cluster, especially when the cluster contains more genes than other clusters. In [24], Liu *et al.* used entropy to reduce the redundancy of DGSs. However, the computaion of entropy needs to know or estimate the very complicated probability density function of training samples, which prevents the entropy-based method becoming popular for this application.

Here we propose a simple yet very effective and efficient method of searching for DGSs that lead to high classification accuracy. Our method is a top-down forward wrapper search scheme, which is much more computationally efficient than the SVM-RFE scheme [1] and is able to greatly reduce the redundancy of DGSs by considering the cooperation among genes.

The rest of this paper is organized as follows. In Section 2, we introduce our SVM-based method of searching for DGSs, i.e., active mining discriminative gene sets (AM-DGS), and its related techniques. In Section 3, we apply our

SVM-based AM-DGS algorithm to the well-known benchmark gene expression data sets, i.e, the leukemia data set [5]. In Section 4, we discuss our results and conclude the paper.

2 Active Mining Discriminative Gene Sets

Recently, *active learning* has attracted great attention in the machine learning field because of its self-learning ability [2,3,22,23]. An active learner, AL, has three components $\{X, F, Q\}$. Here X is the input matrix. F is the mapping function from input space to output space that describes the objective (or function) of the AL. Q is a query function that is used to determine the sequence of unlabelled samples to be learned by the AL according to the current state of the AL, i.e., the AL has the ability to choose the "new things" that will "benefit" its learning. Compared to passive learners, which only contain X and F but no Q, ALs are able to select data for themselves based on the learners' present performance and therefore has the potential to obtain better learning results.

For almost all the active learning approaches proposed to date, the function Q is used to search for the unlabelled *samples*, i.e., *observations*, to be learned by the AL. In the following parts of this section, we will propose a learning scheme with a query function $\tilde{Q}$ that is used to search for *features* (i.e., *genes* in this application) according to the current state of the learner (i.e., the SVM classifier in this application) and its objective. Hence we call our algorithm *active mining* as opposed to *active learning*. In addition, our proposed method is a forward searching scheme that is more straight-forward and efficient than backward searching schemes are.

2.1 T-Statistic

In the first step of our scheme, we rank all the features (genes) according to their t-statistics (TSs). The TS of gene i is defined as follows [16].

$$TS_i = |\frac{\overline{x}_{c1} - \overline{x}_{c2}}{s_{pi}\sqrt{1/n_1 + 1/n_2}}| \tag{1}$$

where

$$\overline{x}_{c1} = \sum_{j \in C_1} \overline{x}_{ij}/n_1 \tag{2}$$

$$\overline{x}_{c2} = \sum_{k \in C_2} \overline{x}_{ik}/n_2 \tag{3}$$

$$s_{pi}^2 = \frac{\sum_{j \in C_1}(x_{ij} - \overline{x}_{c1})^2 + \sum_{k \in C_2}(x_{ik} - \overline{x}_{c2})^2}{n_1 + n_2 - 2} \tag{4}$$

There are 2 classes, i.e., C_1 and C_2, which include n_1 and n_2 samples, respectively. x_{ij} and x_{ik} are the expression values of gene i in C_1 and C_2, respectively. $\overline{x}_{c1}$ and $\overline{x}_{c2}$ are the mean expression values of C_1 and C_2. s_{pi} is the pooled standard deviation of gene i.

2.2 Seeds

After ranking all the genes with TSs, the gene with the largest TS is selected as the first feature in the discriminative gene set (DGS). We call this first feature the *seed*. The best seed that leads to the highest accuracy may not necessarily be the No.1 gene in the TS ranking result (the gene with the greatest TS). It can be the No.2 gene, the No.3 gene and so on. In our application, we use a number of top genes as seeds to search for the best DGS with the highest classification accuracy.

2.3 Support Vector Machines

We use support vector machines (SVMs) [18] [19] as our classifier, i.e., we input our DGS into an SVM to carry out training and classification.

A standard SVM classifier aims to solve the following problem. Given l training vectors $\{\mathbf{x}_i \in R^n, i = 1, ..., l\}$ that belong to two classes, with desired output $y_i \in \{-1, 1\}$, find a decision boundary:

$$\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) + b = 0, \tag{5}$$

where $\mathbf{w}$ is the weight vector and b is the bias. $\boldsymbol{\phi}(\mathbf{x}_i)$ is the function that maps $\mathbf{x}_i$ to a potentially much higher dimensional feature space. This decision boundary is determined by minimizing the cost function:

$$\psi = \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{l} \xi_i, \tag{6}$$

subject to:

$$y_i(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) + b) \geq 1 - \xi_i, \tag{7}$$

$$\xi_i \geq 0. \tag{8}$$

where $\{\xi_i, i = 1, 2, ..., l\}$ are slack variables and C is a constant that determines the tradeoff between the training error and the generalization capability of the SVM. This optimization problem has a quadratic programming (QP) dual problem:

$$\text{maximize: } Q(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \boldsymbol{\phi}(\mathbf{x}_i)^T \boldsymbol{\phi}(\mathbf{x}_j), \tag{9}$$

subject to:

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \tag{10}$$

$$C \geq \alpha_i \geq 0, \tag{11}$$

where $\{\alpha_i, i = 1, 2, ..., l\}$ are Lagrange multipliers. For this problem, we use the sequential minimum optimization [4] as the QP-solver.

2.4 Correction Score

We define a ranking scheme, which we call correction score (CS), to measure a feature's ability to separate the samples that are misclassified by the DGS obtained in the previous round of training. (Here we define the process of picking out a feature and adding it into a DGS as a *round* of training.) The CS of gene i is defined as:

$$CS_i = S_{bi}/S_{wi} \tag{12}$$

where

$$S_{bi} = \sum_{j \in C_1} (e_{ij} - \overline{x}_{c1})^2 + \sum_{k \in C_2} (e_{ik} - \overline{x}_{c2})^2 \tag{13}$$

$$S_{wi} = \sum_{j \in C_1} (e_{ij} - \overline{x}_{c2})^2 + \sum_{k \in C_2} (e_{ik} - \overline{x}_{c1})^2 \tag{14}$$

where e_{ij} and e_{ik} are the expression values of *misclassified* samples in $C1$ and $C2$, respectively. $\overline{x}_{c1}$ and $\overline{x}_{c2}$ are defined in Eq.2 and Eq.3. S_{bi} is the sum of squares of the inter-class distances [21] (the distances between samples of different classes) among the misclassified samples. S_{wi} is the sum of squares of the intra-class distances (the distances of samples within the same class) among the misclassified samples.

2.5 Adding Features According to Misclassification

We input the feature with the largest CS into the SVM in the next round of learning. Our method of searching for the discriminating gene sets is analogous to an AL in the sense that our method has the ability to choose the feature (i.e., the gene) to be included in the next round of learning based on the present state of the learner (i.e., the SVM).

2.6 SVM-Based AM-DGS

The whole process to obtain a DGS is summarized as follows.

Algorithm: SVM-based AM-DGS

Inputs:

Training samples: $\mathbf{X}_{tr} = [\mathbf{x}_{tr1}, \mathbf{x}_{tr2}, ..., \mathbf{x}_{trl}]^T$, validation samples: $\mathbf{X}_v$, testing samples $\mathbf{X}_{test}$

Class labels for training, validation, and testing samples: $\mathbf{Y}_{tr} = [y_{tr1}, y_{tr2}, ..., y_{trl}]^T$, $\mathbf{Y}_v$, $\mathbf{Y}_{test}$

The number of top-ranked genes to search for DGSs: M.

Initialize:

Initialize DGS to an empty matrix: DGS=[ ].
Initialize the training error to 1: $E_{tr} = 1$.
Initialize the validation error to 0: $E_v = 0$.
Initialize the repeat counter to 0: $Rpt = 0$.
Choose a seed:
Calculate the TS for each feature in $\mathbf{X}_{tr}$.
for($m = 1$;until $m < M$; $m++$)
{
  Select a feature with the m-th largest TS as the seed ($\mathbf{S}$).
  $\mathbf{S} \rightarrow$DGS.
  Repeat until: $E_{tr} = 0$ or $E_v < E_{vpre}$ or $Rpt > 2$:
  {
    $E_{trpre} = E_{tr}$;
    $E_{vpre} = E_v$;
    Train an SVM with DGS then obtain E_{tr}.
    Pick out the misclassified samples $\mathbf{X}_e = [\mathbf{x}_{e1}, \mathbf{x}_{e2}, ..., \mathbf{x}_{et}]^T$.
    Validate the SVM using $\mathbf{X}_v$ and obtain E_v.
    If $E_{vpre} = E_v$, $Rpt = Rpt + 1$.
    Calculate CS for each feature in $\mathbf{X}_e$.
    Pick out the feature with the largest CS and put it into the DGS.
  }
}
Output
DGS

3 Experimental Results

We tested our method in the well-known leukemia data set [5]. The leukemia data set [5] (http://www-genome.wi.mit.edu/cancer/) contains two types of leukemia samples, i.e., acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). Golub *et al.* divided the data into 38 samples for training and the other 34 independent samples for testing. Among the 38 training samples, there are 27 ALL samples and 11 AML samples. Among the 34 testing samples, there are 20 ALL samples and 14 AML samples. The entire leukemia data set contains the expression values of 7129 genes. We normalized this data set by subtracting the mean and dividing the standard deviation across each sample.

We processed the leukemia data set with our SVM-based AM-DGS algorithm and showed the results in Table 1. Here we list the 8 DGSs whose seeds are the top 8 genes according to their TSs. For each DGS, the first gene (i.e., the first line in the DGS) is its seed. The second, third (and so on) genes are the genes included in the DGS in the corresponding round, respectively.

From these results, we found that our SVM-based AM-DGS is very effective and efficient in finding good DGSs. Let us use DGS 1 to illustrate this. Since

Table 1. Training and testing accuracies for various DGSs obtained by our SVM-based AM-DGS algorithm to the leukemia data set

Set No.	Gene Sets	Training Accuracy (%)	Testing Accuracy (%)
1	U50136_rna1_at	92.11	79.41
	X17042_at	100	82.35
2	X95735_at	97.37	94.12
	M23197_at	100	94.12
3	M55150_at	97.37	82.35
	M84526_at	92.11	82.35
	M23197_at	97.37	91.18
4	M16038_at	92.11	79.41
	U22376_cds2_s_at	94.74	82.35
5	Y12670_at	94.74	64.71
	U22376_cds2_s_at	100	82.35
6	M23197_at	92.11	85.29
	U22376_cds2_s_at	97.37	88.24
	M63138_at	97.37	94.12
7	D49950_at	97.37	94.12
	U22376_cds2_s_at	86.84	67.65
	X04085_rna1_at	94.74	76.47
	U50136_rna1_at	100	82.35
8	X17042_at	89.47	79.41
	U22376_cds2_s_at	89.47	85.29
	M86406_at	94.74	64.71
	X95735_at	100	97.06

DGS 1 contained only 2 genes, we plotted the gene expression values of the two genes in Fig.1. In the first round of training, only the seed, i.e., gene $U50136_rna1_at$, was input to the SVM. Because gene $U50136_rna1_at$ has a high TS, the SVM misclassified only three samples that were indicated with arrows. In the second round training, the algorithm selected the gene that had the best capability to separate the three misclassified samples, i.e., gene $X17042_at$. We found in Fig.1(a) that gene $X17042_at$ "dragged" the misclassified samples away from the classes which these 3 samples were mistakenly assigned to in the previous round of training. Therefore, with the help of the second gene $X17042_at$, DGS 1 increased its training accuracy from 92.11% to 100%: the 38 training samples were perfectly separated by DGS 1.

The best testing accuracy was obtained by DGS 8, which included 4 genes. The SVM obtained 100% training accuracy and 97.1% testing accuracy (i.e., 1 errors in the 34 testing samples) using DGS 8. In this data set, we used the 8 genes with the largest TSs as the seeds (M=8 in our algorithm summarized in the previous section). If more seeds were used, more DGSs could be found.

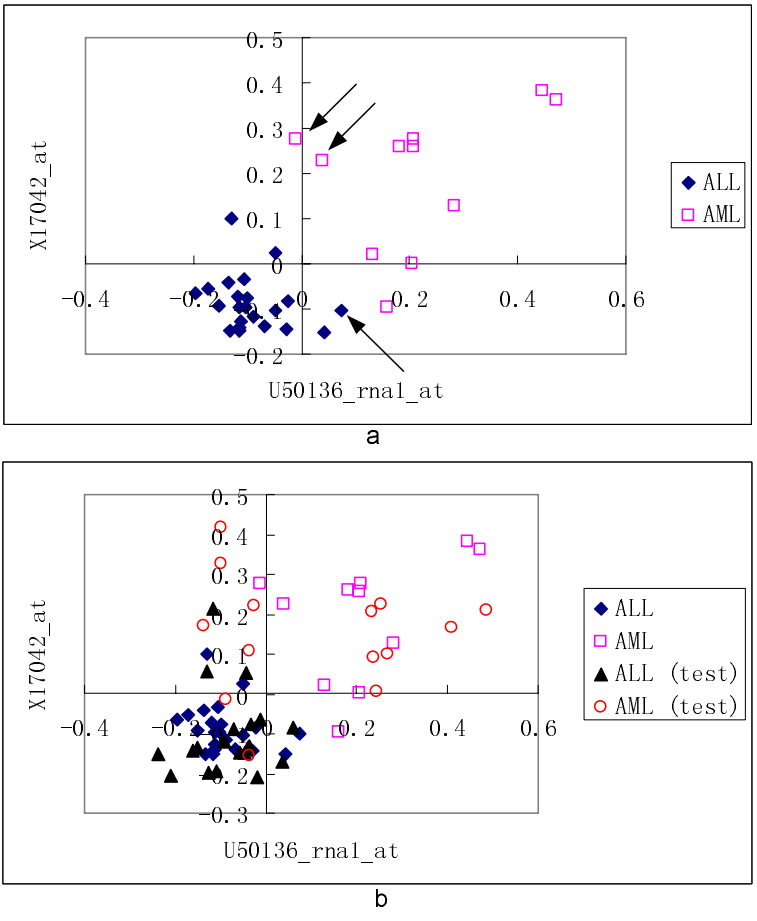


Fig. 1. Gene expression values for the two genes in DGS 1 in the leukemia data set. (a) a plot includes only the training samples; (b) a plot includes all the training and testing samples.

4 Discussion

The results of leukemia data set visually indicate the effectiveness of our SVM-based AM-DGS algorithm. Except the seeds, all the genes in a DGS are selected according to their capability to correct misclassified samples. Therefore, the SVM-based AM-DGS can optimize the cooperation among genes and hence leads to good accuracy and smaller DGSs. Compared with the filter approaches, e.g., TS and FS, the SVM-based AM-DGS can greatly reduce the redundancy in a DGS.

In conclusion, the SVM-AMDGS proposed here is effective and computationally efficient in searching for good DGSs, the simulation using the leukemia data set shows that our algorithm leads to highly accurate classifications with the smallest gene sets found in the literature.

References

1. Guyon, I., Wecton, J., Barnhill, S., Vapnik, V.: Gene Selection for Cancer Classification using Support Vector Machines. Machine Learning. **46** (2002) 389-422
2. Mitra, P., Murthy, C. A., Pal, S. K.: A Probabilistic Active Support Vector Learning Algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence. **26** (2004) 413-418
3. Tong, S., Koller, D.: Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research. **2** (2002) 45-66
4. Platt, J. C.: Sequential Minimum Optimization: A Fast Algorithm for Training Support Vector Machines. Microsoft Research, Cambridge, U.K., Technical Report, (1998)
5. Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science. **286** (1999) 531-537
6. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays. Proc. Natl. Acad. Sci. USA. **96** (1999) 6745-6750
7. Wang, Y., Makedon, F., Ford, J., Pearlman, J.: Hykgene: a Hybrid Approach for Selecting Marker Genes for Phenotype Classification Using Microarray Gene Expression Data. Bioinformatics. **21** (2005) 1530-1537
8. Li, L., Weinberg, C. R., Darden, T. A., Pedersen, L. G.: Gene Selection for Sample Classification Based on Gene Expression Data: Study of Sensitivity to Choice of Parameters of the GA/KNN Method. Bioinformaitcs. **17** (2001) 1131-1142
9. Cho, J. H., Lee, D., Park, J. H., Lee, I. B.: Gene Selection and Classification from Microarray Data Using Kernel Machine. FEBS Letters. **571** (2004) 93-98
10. Li, J., Wong, L.: Identifying Good Diagnostic Gene Groups from Gene Expressin Profiles Using the Concept of Emerging Patterns. Bioinformatics. **18** (2002) 725-734
11. Lai, Y., Wu, B., Chen, L., Zhao, H.: Statistical Method for Identifying Differential Gene-Gene Coexpression Patterns. Bioinformatics. **21** (2005) 1565-1571
12. Broet, P., Lewin, A., Richardson, S., Dalmasso, C., Magdelenat, H.: A Mixture Model-Based Strategy for Selecting Sets of Genes in Multiclass Response Microarray Experiments. Bioinformatics. **20** (2004) 2562-2571
13. Alizadeh, A. A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., et al.: Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling. Nature. **403** (2000) 503-511
14. Khan, J. M., Wei, J.S., Ringner, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C., et al.: Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks. Nature Medicine. **7** (2001) 673-679

15. Deutsch, J. M.: Evolutionary Algorithms for Finding Optimal Gene Sets in Microarray Prediction. Bioinformatics. **19** (2003) 45-52
16. Devore, J., Peck, R.: Statistics: the Exploration and Analysis of Data. 3rd edn. Duxbury Press, Pacific Grove, CA (1997)
17. Xing, E. P., Jordan, M. I., Karp, R. M.: Feature Selection for High-Dimensional Genomic Microarray Data. Proc. of the 18th International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., (2001) 601-608
18. Vapnik, V.: Statistical Learning Theory, Wiley, New York, (1998)
19. Wang L. P. (ed.): Support Vector Machines: Theory and Applications, Springer, Berlin, (2005)
20. Devijver, P., Kittler, J.: Pattern Recognition: a Statistical Approach, Prentice Hall, London, (1982)
21. Fu, X., Wang, L. P.: Data Dimensionality Reduction with Application to Simplifying RBF Network Structure and Improving Classification Performance. IEEE Trans. on Systems, Man, and Cybernetics-Part b: Cybernetics. **33**, (2003) 399-409
22. Ji, S., Krishnapuram, B., Carin, L.: Hidden Markov Models and Its Application to Active Learning. IEEE Trans. on Pattern Analysis and Machine Intelligence. **28** (2006) 522-532
23. Riccardi, G., Hakkani-Tur, D.: Active Learning: Theory and Application to Automatic Speech Recognition. IEEE Trans. on Speech and Audio Processing. **13** (2005) 504-511
24. Liu, X., Krishnan, A., Mondry, A.: An Entropy-Based Gene Selection Method for Cancer Classification Using Microarray Data. BMC Bioinformatics. **6** (2005) 76

A Novel Approach to Image Reconstruction from Discrete Projections Using Hopfield-Type Neural Network

Robert Cierniak

Technical University of Czestochowa, Departament of Computer Engineering, Armii Krajowej 36, 42-200 Czestochowa, Poland

Abstract. Presented paper shows a novel approach to the problem of image reconstruction from projections using recursive Hopfield-type neural network. The reconstruction process is performed during the minimizing of the energy function in this network. Our method is of a great practical use in reconstruction from discrete parallel beam projections. Experimental results show that the appropriately designed neural network is able to reconstruct an image with better quality than obtained from conventional algorithms.

1 Introduction

Computerized tomography is regarded as one of the most important inventions of the twentieth century. The remarkable feature of this medical imaging method is a possibility of using it to examine the inside of an object, for example the human body. A three-dimensional image of the object is given by applying an appropriate method of projection and an image reconstruction algorithm. There are several reconstruction methods to solve this problem, for example the most popular reconstruction algorithm using convolution and back-projection [7] and the algebraic reconstruction technique (ART) [2]. This work presents an approach to image reconstruction from projections problem using the Hopfield-type neural network. The idea of an application of a neural network to computerized tomography is shown for example in [1]. In this paper we present a new approach to reconstruction problem based on the commonly applied transformation methodology. An algorithm based on the algebraic approach was proposed in [8]. A great advantage of our algorithm is a possibilty of a hardware implementation and a direct practical application to medical divices.

2 Image Reconstruction Algorithm

Presented in this paper reconstruction neural network algorithm resembles the ρ-filtered layergram method [4]. The main difference between these two methods is a realization of the filtering. In our case the neural network is implemented instead of the two-dimensional filtering of the blurred image obtained after the back-projection operation. The scheme of the proposed reconstruction method using the Hopfield-type neural network is shown in Fig.1.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 890–898, 2006.
© Springer-Verlag Berlin Heidelberg 2006

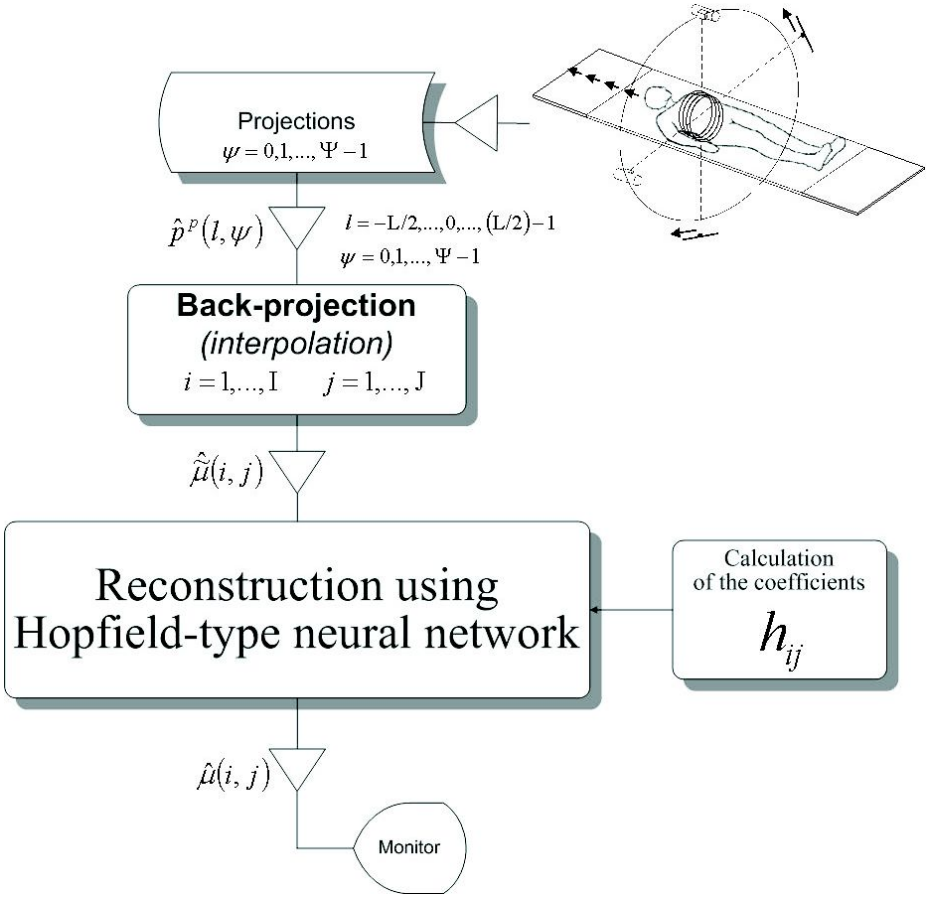


Fig. 1. A neural network image reconstruction algorithm with parallel beam geometry of the scanner

2.1 Projections

In the first step of the presented reconstruction algorithm a set of parallel beam projections $p^p(s,\alpha^p)$; $s = (l+1/2)\cdot\Delta_s$, $l = -L/2,\ldots,0,\ldots,L/2-1$, L—even number of detectors; $\alpha^p = \psi\cdot\Delta_\psi$, $\Delta_\psi = \pi/\Psi$, $\psi = 0,\ldots,\Psi-1$, Ψ—number of projections, is obtained. A projection can be interpreted as the depth of the shadow cast by the object onto a screen positioned opposite the radiation source. In the case of parallel geometry of the scanner (see Fig.2) this is called the Radon's transformation [6] and in continuous domain it can be expressed as follows

$$p^p(s,\alpha^p) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \mu(x,y)\cdot\delta(xcos\alpha^p + ysin\alpha^p - s)dxdy, \qquad (1)$$

where: α^p—is the angle of parallel projection; x, y —the co-ordinates of the examined object; $\Delta s = (xcos\alpha^p + ysin\alpha^p - s)$—a distance from the centre of rotation to the axis of the ray falling on the projection screen; $\mu(x, y)$—a distribution of the attenuation of x-rays in analysed cross-section.

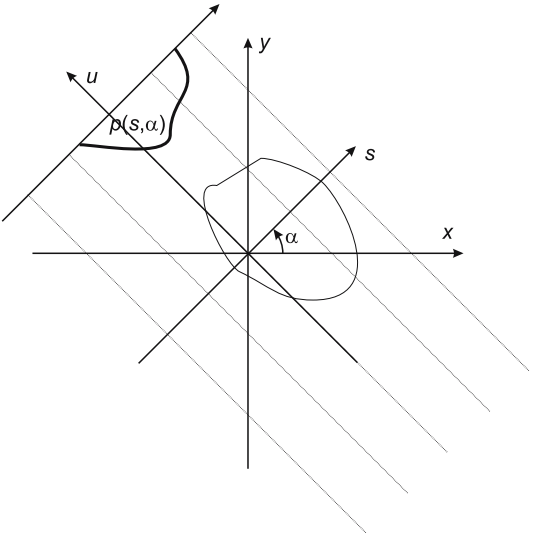


Fig. 2. A single parallel projection

2.2 Interpolation and Back-Projection

The next step in the proceeding sequence is the back-projection operation. This can be expressed as

$$\widetilde{\mu}(x, y) = \int_0^{\pi} p^p(s, \alpha^p) d\alpha^p. \tag{2}$$

Function $\widetilde{\mu}(x, y)$ denotes a blurred image obtained after operations of projection and back-projection. According to equations (1) and (2) after some algebra we can write

$$\widetilde{\mu}(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mu(\ddot{x}, \ddot{y}) \int_0^{\pi} \delta(\ddot{x}cos\alpha^p + \ddot{y}sin\alpha^p - xcos\alpha^p - ysin\alpha^p) d\alpha^p d\ddot{x} d\ddot{y}. \tag{3}$$

In practical realization of the proposed reconstruction algorithm it is highly possible that for any given projection no ray passes through a certain point (x, y) of the image. To take this into account we can apply interpolation expressed by the equation

$$\dot{p}^p(\dot{s}, \alpha^p) = \int_{-\infty}^{+\infty} p^p(s, \alpha^p) \cdot I(\dot{s} - s) ds, \tag{4}$$

where $I(\Delta s)$ is an interpolation function. Owing to relation (5) it is possible to define the obtained, after back-projection operation, image in the following way

$$\widetilde{\mu}(x,y) = \int_0^{\pi} \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mu(\ddot{x},\ddot{y}) \cdot \delta(\ddot{x}cos\alpha^p + \ddot{y}sin\alpha^p - \dot{s}) d\ddot{x}d\ddot{y} \right) \cdot I(\dot{s}-s) ds d\alpha^p. \tag{5}$$

According to the properties of the convoluation we can transform formula (5) to the form

$$\widetilde{\mu}(x,y) = \int_{-\infty}^{+\infty} \mu(\ddot{x},\ddot{y}) \left(\int_0^{\pi} I(\ddot{x}cos\alpha^p + \ddot{y}sin\alpha^p - xcos\alpha^p - ysin\alpha^p) d\alpha^p \right) d\ddot{x}d\ddot{y}. \tag{6}$$

2.3 Discrete Reconstruction Problem

In presented method we take into consideration the discrete form of images $\mu(x,y)$ and $\widetilde{\mu}(x,y)$. That means we will substitute continuous functions of images in equation (6) for their discrete equivalents $\hat{\mu}(i,j)$ and $\hat{\widetilde{\mu}}(i,j)$; $i = 0,1,\ldots,I$; $j = 0,1,\ldots,J$, where I and J—are numbers of pixels in horizontal and vertical directions, respectively. Additionally, we approximate the 2-D convolution function by two finite sums. In this way we express relation (6) in the following form

$$\hat{\widetilde{\mu}}(i,j) \approxeq \sum_{\ddot{i}} \sum_{\ddot{j}} \hat{\mu}(i-\ddot{i}, j-\ddot{j}) \cdot h_{\ddot{i}\ddot{j}}, \tag{7}$$

where

$$h_{\ddot{i}\ddot{j}} = \Delta_\alpha^p (\Delta_s)^2 \cdot \sum_{\psi=0}^{\Psi-1} I\left(\ddot{i}\Delta_s cos\psi\Delta_\alpha^p + \ddot{j}\Delta_s sin\psi\Delta_\alpha^p\right). \tag{8}$$

As one can see from equation (7), the original image in a given cross-section of the object, obtained in the way described above, is equal to the amalgamation of this image and the geometrical distortion element given by (8). The number of coefficients $h_{\ddot{i}\ddot{j}}$ is equal to $I \cdot J$ and owing to expression (8) values of these coefficients can be easily calculated.

The discrete reconstruction from projections problem can be formulated as following optimisation problem [5]

$$\min_{\Omega} \left(p \cdot \sum_{\ddot{i}=1}^{I} \sum_{\ddot{j}=1}^{J} f\left(e_{\ddot{i}\ddot{j}}(\Omega)\right) \right), \tag{9}$$

where: $\Omega = [\hat{\mu}(i,j)]$—a matrix of pixels from original image; p—suitable large positive coefficient; $f(\bullet)$—penalty function and

$$e_{\ddot{i}\ddot{j}}(\Omega) = \sum_{i} \sum_{j} \hat{\mu}(i,j) \cdot h_{\ddot{i}-i,\ddot{j}-j} - \hat{\widetilde{\mu}}(\ddot{i},\ddot{j}). \tag{10}$$

If a value of coefficient p tends to infinity or in other words is suitably large, then the solution of (9) tends to the optimal result. Our research has shown that the following penalty function yields the best result

$$f\left(\ddot{e}_{ij}(\Omega)\right) = \lambda \cdot \ln\cosh\left(\frac{\ddot{e}_{ij}(\Omega)}{\lambda}\right), \tag{11}$$

and derivation of (11) has the convenient form

$$f'\left(\ddot{e}_{ij}(\Omega)\right) = \frac{df\left(\ddot{e}_{ij}(\Omega)\right)}{d\ddot{e}_{ij}(\Omega)} = \frac{1-\exp\left(\ddot{e}_{ij}(\Omega)/\lambda\right)}{1+\exp\left(\ddot{e}_{ij}(\Omega)/\lambda\right)}, \tag{12}$$

where: λ—slope coefficient.

2.4 Hopfield-Type Neural Network Realizing Reconstruction Process

Now we can start to formulate the energy expression

$$E^t = p \cdot \sum_{\ddot{i}=1}^{I}\sum_{\ddot{j}=1}^{J} f\left(\ddot{e}_{ij}\left(\Omega^t\right)\right). \tag{13}$$

which will be minimized by the constructed neural network to realize the deconvolution task expressed by equation (9). In order to find a minimum of function (13) we calculate the derivative

$$\frac{dE^t}{dt} = p \cdot \sum_{\ddot{i}=1}^{I}\sum_{\ddot{j}=1}^{I}\sum_{i=1}^{I}\sum_{j=1}^{J} \frac{\partial f\left(\ddot{e}_{ij}\left(\Omega^t\right)\right)}{\partial\left(\ddot{e}_{ij}\left(\Omega^t\right)\right)} \frac{\partial\left(\ddot{e}_{ij}\left(\Omega^t\right)\right)}{\partial\hat{\mu}^t(i,j)} \frac{d\hat{\mu}^t(i,j)}{dt}. \tag{14}$$

If we let

$$\frac{d\hat{\mu}^t(i,j)}{dt} = -p\sum_{\ddot{i}=1}^{I}\sum_{\ddot{j}=1}^{I} \frac{\partial f\left(\ddot{e}_{ij}\left(\Omega^t\right)\right)}{\partial\left(\ddot{e}_{ij}\left(\Omega^t\right)\right)} \frac{\partial\left(\ddot{e}_{ij}\left(\Omega^t\right)\right)}{\partial\hat{\mu}^t(i,j)} = -p\sum_{\ddot{i}=1}^{I}\sum_{\ddot{j}=1}^{I} f'\left(\ddot{e}_{ij}(\Omega)\right) h_{ij}, \tag{15}$$

equation (14) takes the form

$$\frac{dE^t}{dt} = -\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\frac{d\hat{\mu}^t(i,j)}{dt}\right)^2. \tag{16}$$

The structure of the recursive neural network performing the reconstruction process consists of two layers and is depicted in Fig. 3.

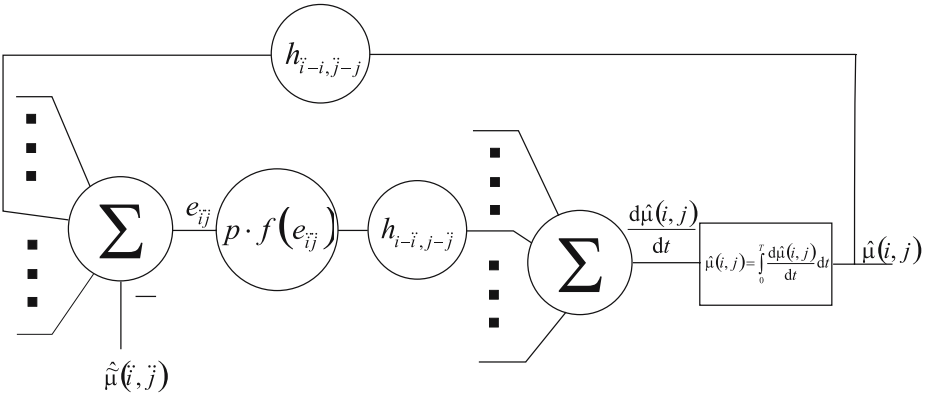


Fig. 3. The structure of the designed Hopfield-type neural network

3 Experimental Results

A mathematical model of the projected object, a so-called phantom, is used to obtain projections during simulations. The most common matematical phantom of head was proposed by Kak (see eg. [3]). In our experiment the size of the image was fixed at $I \times J = 129 \times 129$ pixels. A view obtained from this

Fig. 4. A view obtained from the mathematical model of the cross-section of the skull

mathematical model is depicted in Fig.4. The discret approximation of the interpolation operation expressed by equation (4) takes the form

$$\hat{p}^p(s,\psi) = \sum_{l=-L/2}^{L/2-1} \hat{p}^p(l,\psi) \cdot I(s - l\Delta_\psi^p). \tag{17}$$

The interpolation function $I(\Delta s)$ can be defined for example as linear interpolation function

$$I_L(\Delta s) = \begin{cases} \frac{1}{\Delta_s}\left(1 - \frac{|\Delta s|}{\Delta_s}\right) & if\,|\Delta s| \geq \Delta_s \\ 0, & if\,|\Delta s| > \Delta_s \end{cases} \tag{18}$$

or the Dirichlet kernel of order $2L$

$$I_D(\Delta s) = \frac{sin\,(\pi \Delta s)}{2Lsin\,(\pi \Delta s/2L)}. \tag{19}$$

where $s = (icos\psi \Delta_\alpha^p + jsin\psi \Delta_\alpha^p)$.

In Fig.5 we show views of images obtained after the back-projection operation using above presented interpolation functions. The image was next subjected to

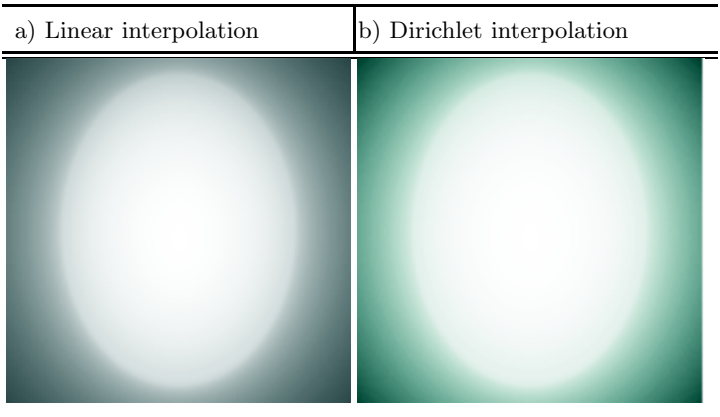


Fig. 5. Images obtained after the back-projection operation: a) the linear interpolation function; b) the Dirichlet kernel

a process of reconstruction using the Hopfield-type neural network presented in section 2. The Euler's method (see eg. [8]) was used to approximate (15) in following way

$$\mu(i,j)^{t+1} := \mu(i,j)^t + \Delta t \left(-p \sum_{\ddot{i}=1}^{I} \sum_{\ddot{j}=1}^{I} f'\left(e_{ij}^{\cdots}(\Omega)\right) h_{ij}^{\cdots} \right), \tag{20}$$

where $e_{ij}^{\cdots}$ is expressed by (10) and Δt is a sufficient small time step.

The progress in reconstruction process is shown in Fig.6.

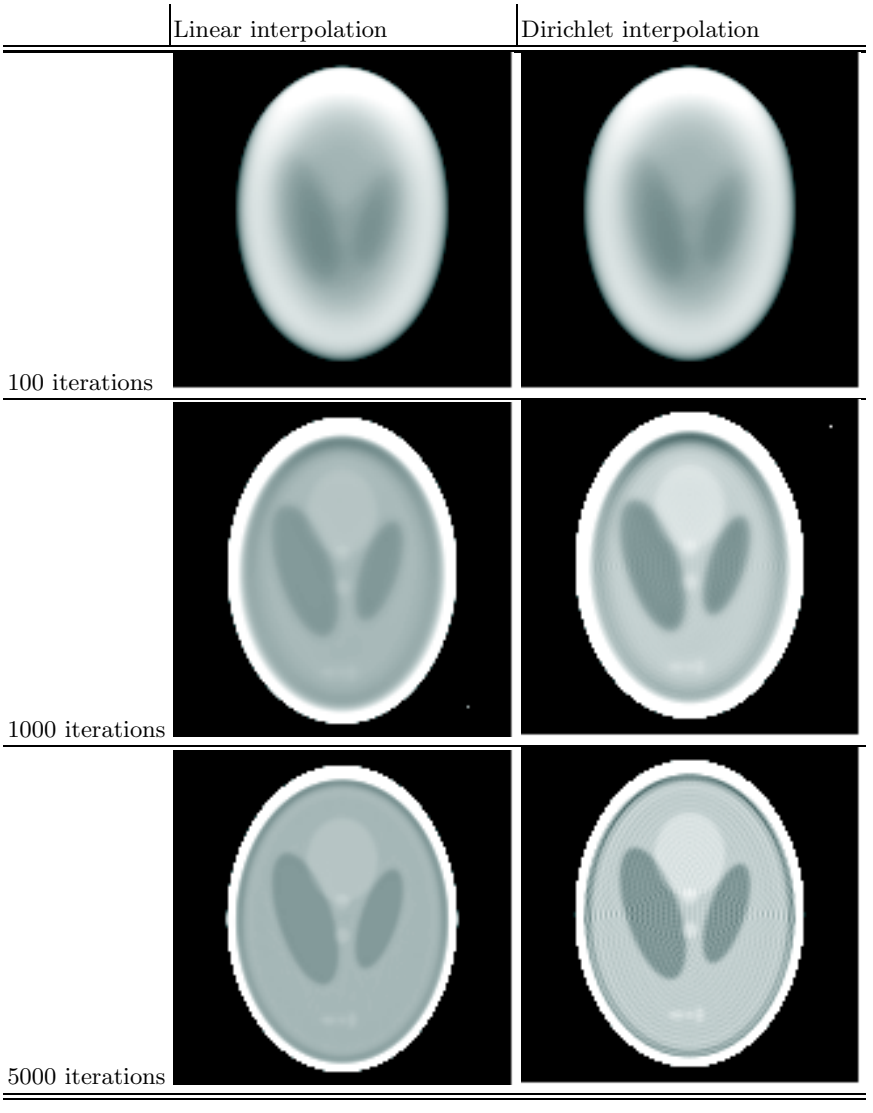


Fig. 6. Progress in reconstruction process using a mathematical model of the cross-section of the skull

4 Conclusions

The performed simulations demonstrated a convergence of the image reconstruction from projections algorithm based on the Hopfield-type neural network described in this work. In both cases of interpolation functions the reconstructed images in Fig.6 have satisfactory level of quality after about five thousand iterations. Therefore, one can say that at this point the image is reconstructed and

the process can be stopped. The Dirichlet interpolation is more time consuming and gives no advanteges in quality. Then linear interpolation seems to be more adequate in this application. Although our procedure is time consuming, the hardware implementation of the described neural network structure could give incomparable better results than other reconstruction methods.

References

1. Frieden, B. R., Zoltani, C. R.: Maximum bounded entropy: application to tomographic reconstruction. Appl. Optics **24** (1985) 201–207
2. Jaene, B.: Digital Image Processing - Concepts, Algoritms and Scientific Applications. Springer-Verlag Berlin, Heidelberg (1991)
3. Jain, A. K.: Fundamentals of Digital Image Processing. Prentice Hall, New Jersey (1989)
4. Lewitt, R. M.: Reconstruction algorithms: transform methods. Proceeding of the IEEE **71** (1983) 390–408
5. Luo, Fa-Long, Unbehauen R.: Applied Neural Networks for Signal Processing. Cambridge University Press (1998)
6. Radon, J.: Ueber die Bestimmung von Functionen durch ihre Integralwerte Tangs gewisser Mannigfaltigkeiten. Berichte Saechsiche Akad. Wissenschaften, Math. Phys. Klass **69** Leipzig (1917) 262–277
7. Ramachandran, G. N., Lakshminarayanan, A. V.: Three-dimensional reconstruction from radiographs and electron micrographs: II. Application of convolutions instead of Fourier transforms. Proc. Nat. Acad. Sci. **68** (1971) 2236–2240
8. Wang, Y., Wahl F. M.: Vector-entropy optimization-based neural-network approach to image reconstruction from projections. IEEE Transaction on Neural Networks **8** (1997) 1008–1014

Leukemia Prediction from Gene Expression Data—A Rough Set Approach

Jianwen Fang[1,⋆] and Jerzy W. Grzymala-Busse[2]

[1] Bioinformatics Core Facility
and
Information and Telecommunication Technology Center
University of Kansas, Lawrence, KS 66045, USA
jwfang@ku.edu
[2] Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA
and
Institute of Computer Science Polish Academy of Sciences, 01-237 Warsaw, Poland
jerzy@ku.edu
http://lightning.eecs.ku.edu/index.html

Abstract. We present our results on the prediction of leukemia from microarray data. Our methodology was based on data mining (rule induction) using rough set theory. We used a novel methodology based on rule generations and cumulative rule sets. The final rule set contained only eight rules, using some combinations of eight genes. All cases from the training data set and all but one cases from the testing data set were correctly classified. Moreover, six out of eight genes found by us are well known in the literature as relevant to leukemia.

1 Introduction

In our research we used data mining methodology, based on rough set theory, to predict leukemia on the basis of gene expression monitoring by DNA microarrays. Microarray technology has provided biologists with the ability to simultaneously study thousands of genes comprising a large part of the genome. The development of this promising technology has motivated interest of its use in clinical diagnosis and drug discovery. The key step of these types of applications is to identify subsets of genes, often referred to as biomarkers, which distinguish cases with different labels, e.g., different tumor types, cancer versus non-cancer, response to therapy. In addition, these biomarkers are potential drug targets for treatment since they are relevant to the disease under study.

In our data set there were 72 leukemia cases including 47 acute myeloid leukemia (AML) and 25 patients with lymphoblastic leukemia (ALL). The data set contains the expression levels of 6817 human genes measured by Affymetrix

⋆ This research has been partially supported by the K-INBRE Bioinformatics Core, NIH grant P20 RR016475.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 899–908, 2006.
© Springer-Verlag Berlin Heidelberg 2006

high-density genechips. Following Golub *et al.* [5], the data set was split into 38 training cases (27 ALL and 11 AML) and 34 testing cases (20 ALL and 14 AML).

Distinguishing ALL from AML is critical for successful treatment of leukemia patients since the two types require chemotherapy to concentrate on different regimens. It has been found the cure rates arise and unwarranted toxicities are diminished if the subtypes have been correctly classified.

The data set has been widely used in classification and feature selection papers in the literature. Using 50 informative genes, Golub *et al.* [5] were able to predict 29 of the 34 testing cases using a weighted voting scheme. Their model was not able to predict the remaining five instances. Lee [14] used a hierarchical Bayesian model to select five genes and then perform predictions on the testing data. Only one case was misclassified. Yeung *et al.* [24] proposed a Bayesian model averaging method for gene selection and classification of microarray data. They also misclassified one case. Nguyen and Rocke [16] reported 1–3 classification errors on the testing data set using 50–1500 selected genes. They used partial least squares to reduce the dimension and logistic discrimination and quadratic discrimination analysis to classify the instances. Since one case (#66) has been consistently misclassified in the literature, it has been suggested that the case might be incorrectly labeled [24].

2 Data Mining Tools

The main tool for our experiments was the MLEM2 (Modified Learning from Examples Module, version 2) rule induction module of the LERS (Learning from Examples based on Rough Sets) data mining system, [6, 7, 8, 9].

In the first step of processing the input data file, the data mining system LERS checks if the input data file is *consistent* (i.e., if the file does not contain conflicting examples). If the input data file is inconsistent, LERS computes lower and upper approximations [17, 18] of all concepts. Rules induced from the lower approximation of the concept *certainly* describe the concept, so they are called *certain* [6]. On the other hand, rules induced from the upper approximation of the concept describe the concept only *possibly* (or *plausibly*), so they are called *possible* [6].

LEM2 learns the smallest set of minimal rules, describing the concept. The module LEM2 of LERS is most frequently used since—in most cases—it gives best results. LEM2 explores the search space of attribute-value pairs. Its input data file is a lower or upper approximation of a concept, so its input data file is always consistent. In general, LEM2 computes a local covering and then converts it into a rule set. We will quote a few definitions to describe the LEM2 algorithm.

The LEM2 algorithm is based on an idea of an attribute-value pair block. For an attribute-value pair $(a,\ v) = t$, a *block* of t, denoted by $[t]$, is a set of all cases from U such that for attribute a have value v. For a set T of attribute-value pairs, the intersection of blocks for all t from T will be denoted by $[T]$. Let B be a nonempty lower or upper approximation of a concept represented by

a decision-value pair (d, w). Set B *depends* on a set T of attribute-value pairs $t = (a, v)$ if and only if

$$\emptyset \neq [T] = \bigcap_{t \in T} [t] \subseteq B.$$

Set T is a *minimal complex* of B if and only if B depends on T and no proper subset T' of T exists such that B depends on T'. Let $\mathcal{T}$ be a nonempty collection of nonempty sets of attribute-value pairs. Then $\mathcal{T}$ is a *local covering* of B if and only if the following conditions are satisfied:

(1) each member T of $\mathcal{T}$ is a minimal complex of B,

(2) $\bigcup_{t \in \mathcal{T}} [T] = B$, and

$\mathcal{T}$ is minimal, i.e., $\mathcal{T}$ has the smallest possible number of members.

The procedure LEM2 is presented below.

Procedure LEM2
(**input**: a set B,
output: a single local covering $\mathcal{T}$ of set B);
begin
$G := B$;
$\mathcal{T} := \emptyset$;
while $G \neq \emptyset$
begin
$T := \emptyset$;
$T(G) := \{t | [t] \cap G \neq \emptyset\}$;
while $T = \emptyset$ **or** $[T] \not\subseteq B$
begin
select a pair $t \in T(G)$ such that $|[t] \cap G|$
is maximum; if a tie occurs, select a pair $t \in T(G)$
with the smallest cardinality of $[t]$;
if another tie occurs, select first pair;
$T := T \cup \{t\}$;
$G := [t] \cap G$;
$T(G) := \{t | [t] \cap G \neq \emptyset\}$;
$T(G) := T(G) - T$;
end {while}
for each $t \in T$ **do**
if $[T - \{t\}] \subseteq$ B **then** $T := T - \{t\}$;
$\mathcal{T} := \mathcal{T} \cup \{T\}$;
$G := B - \cup_{T \in \mathcal{T}} [T]$;
end {while};
for each $T \in \mathcal{T}$ **do**
if $\bigcup_{S \in \mathcal{T} - \{T\}} [S] = B$ **then** $\mathcal{T} := \mathcal{T} - \{T\}$;
end {procedure}.

MLEM2 is a modified version of the algorithm LEM2. The original algorithm LEM2 needs discretization, a preprocessing, to deal with numerical attributes. MLEM2 recognizes integer and real numbers as values of attributes, and labels such attributes as numerical. For numerical attributes MLEM2 computes blocks in a different way than for symbolic attributes. First, it sorts all values of a numerical attribute. Then it computes cutpoints as averages for any two consecutive values of the sorted list. For each cutpoint c MLEM2 creates two blocks, the first block contains all cases for which values of the numerical attribute are smaller than c, the second block contains remaining cases, i.e., all cases for which values of the numerical attribute are larger than c. The search space of MLEM2 is the set of all blocks computed this way, together with blocks defined by symbolic attributes. Then MLEM2 combines attribute-value pairs relevant to a concept and creates rules describing the concept. In addition, MLEM2 handles missing attribute values during rule induction [9]. The previous version of MLEM2, LEM2, induced certain rules from incomplete decision tables with missing attribute values interpreted as lost. Recently, MLEM2 was further extended to induce both certain and possible rules from a decision table with some missing attribute values being lost and some missing attribute values being "do not care" conditions, while attributes may be numerical.

3 Classification System

The classification system of LERS is a modification of the *bucket brigade algorithm* [1, 11]. The decision to which concept a case belongs is made on the basis of three factors: strength, specificity, and support. They are defined as follows: *strength* is the total number of cases correctly classified by the rule during training. *Specificity* is the total number of attribute-value pairs on the left-hand side of the rule. The matching rules with a larger number of attribute-value pairs are considered more specific. The third factor, *support*, is defined as the sum of scores of all matching rules from the concept, where the score of the rule is the product of its strength and specificity. The concept C for which the support, i.e., the following expression

$$\sum_{matching\ rules\ R\ describing\ C} Strength_factor(R) * Specificity_factor(R)$$

is the largest is the winner and the case is classified as being a member of C.

In the classification system of LERS, if complete matching is impossible, all partially matching rules are identified. These are rules with at least one attribute-value pair matching the corresponding attribute-value pair of a case. For any partially matching rule R, the additional factor, called *Matching _ factor* (R), is computed. Matching_factor (R) is defined as the ratio of the number of matched attribute-value pairs of R with a case to the total number of attribute-value pairs of R. In partial matching, the concept C for which the following expression is the largest

$$\sum_{\substack{partially\ matching\\ rules\ R\ describing\ C}} Matching_factor(R) * Strength_factor(R) * Specificity_factor(R)$$

is the winner and the case is classified as being a member of C.

Every rule induced by LERS is preceded by three numbers: specificity, strength, and the rule domain size (the total number of training cases matching the left-hand side of the rule).

4 Rule Generations

Ordinarily, in data mining just one rule set is induced. In our research we used a novel approach to data mining based on the idea of inducing many rule generations.

The original rule set is the first generation rule set. In our rule induction method, dominant attributes involved in the first rule generation were excluded from the data set and then the second rule generation was induced from such prepared, new data set. The procedure was repeated iteratively.

The idea of inducing many rule generations is not always feasible. However, for microarray type of data, where the number of attributes (gene expressions) is so large compared with the number of cases, it is quite natural.

Obviously, the first rule generation is, in general, more valuable than the second rule generation, the second rule generation is more valuable than the third rule generation, etc.

Let us illustrate the rule generation method on the leukemia data set. The first rule generation, called First.r, was

1, 27, 27
(X95735, -674..994) –> (label, ALL)
1, 11, 11
(X95735, 994..6218) –> (label, AML)

After removal of the attribute X95735 from the original data set, the second rule generation, called Second.r, was

1, 25, 25
(M27891, -376..946.5) –> (label, ALL)
2, 2, 2
(M27891, 946.5..1419.5) & (M31166, -22..83.5) –> (label, ALL)
2, 1, 1
(M27891, 946.5..1419.5) & (M55150, 1346..2693) –> (label, AML)
1, 10, 10
(M27891, 1419.5..17863) –> (label, AML)

As is clear from a close inspection of Second.r, among the four rules two are outliers (the second and the third rules). Such rules were deleted. However, to

compensate for such a rule deletion, remaining rules in the same rule generation were compensated by increasing their strength. The dominant attribute of Second.r was M27891. This attribute was additionally deleted from the original data set. Thus, two attributes were missing: X95735 and M27891. From that new data set the third rule generation was induced (called Third.r)

2, 27, 27
(M31166, -22..83.5) & (D88422, -9..658) –> (label, ALL)
2, 11, 11
(M55150, 1346..2693) & (M77142, -20..80.5) –> (label, AML)

The process continued until the cumulative rule sets, described in the next section, were not better in terms of the error rate.

5 Cumulative Rule Sets

Rule generations were gradually collected together into new rule sets. For example, from the first and second rule generations, a new cumulative rule set, called Two.r was created:

1, 2, 27
(X95735, -674..994) –> (label, ALL)
1, 3, 25
(M27891, -376..946.5) –> (label, ALL)
1, 4, 11
(X95735, 994..6218) –> (label, AML)
1, 6, 10
(M27891, 1419.5..17863) –> (label, AML)

Note that rule strengths were changed. We are using here a few mechanisms to change rule strengths. Normally, all rules from First.r would have rule strengths twice as large as rule strengths from Second.r. But the original Second.r had outliers, so we increase each rule strength for rules from a new Second.r by two. Additionally, following the method presented in [10], we increase the rule strength for every rule describing AML (the weaker concept) multiplying it by two, since the leukemia data set is imbalanced. Thus, the strength of the second rule is $1 + 2$, the strength of the third rule is 2×2, and the strength of the fourth rule is $(1 + 2) \times 2$.

Similarly, the cumulative rule set Three.r, created from First.r, Second.r and Third.r was

1, 3, 27
(X95735, -674..994) –> (label, ALL)
1, 4, 25
(M27891, -376..946.5) –> (label, ALL)
2, 1, 27
(M31166, -22..83.5) & (D88422, -9..658) –> (label, ALL)
1, 6, 11

(X95735, 994..6218) –> (label, AML)
1, 8, 10
(M27891, 1419.5..17863) –> (label, AML)
2, 2, 11
(M55150, 1346..2693) & (M77142, -20..80.5) –> (label, AML)

Here, initially all rules from First.r had strengths equal to three, rules from Second.r had strengths equal to two, and rules from Third.r had strengths equal to one. Additionally, we compensate the strength for every rule from Second.2 by adding two for lack of outliers. Finally, we multiply the resulting strength for every rule describing AML by two to take into account that the training data set is imbalanced.

6 Experiments

Results of our experiments are presented in Tables 1–2. Table 1 shows the total number of errors for both data sets: training and testing and for consecutive rule generations. Note that three errors for the rule set Second.r on the training data set are caused by deletion of the two weak rules (outliers).

Table 1. Number of errors for rule generations

	Number of errors on	
	training data set	testing data set
First.r	0	4
Second.r	3	5
Third.r	0	8
Fourth.r	0	6
Fifth.r	0	6
Sixth.r	0	10

Table 2 also presents the total number of errors for the training and testing data sets, but for the cumulative rule sets. The simplest cumulative rule set with the smallest number of errors for the testing data set was Four.r:

1, 4, 27
(X95735, -674..994) –> (label, ALL)
1, 5, 25
(M27891, -376..946.5) –> (label, ALL)
2, 2, 27
(M31166, -22..83.5) & (D88422, -9..658) –> (label, ALL)
2, 1, 27

Table 2. Number of errors for cumulative rule sets

	Number of errors on	
	training data set	testing data set
First.r	0	4
Two.r	0	3
Three.r	0	2
Four.r	0	1
Five.r	0	1
Six.r	0	2

(D88422, -9..658) & (M21551, -47..398.5) –> (label, ALL)
1, 8, 11
(X95735, 994..6218) –> (label, AML)
1, 10, 10
(M27891, 1419.5..17863) –> (label, AML)
2, 4, 11
(M55150, 1346..2693) & (M77142, -20..80.5) –> (label, AML)
2, 2, 11
(M77142, -20..80.5) & (U46499, 156.5..3107) –> (label, AML)

7 Conclusions

First of all, our final rule set Four.r is very simple and it classifies well all cases from the training data set and all but one cases from the testing data set. Genes from the rule set Four.r are characterized in Table 3. Cross-references are citations of papers in which the respective gene was identified as a gene relevant for leukemia. Number of references from Table 3 is the number of the references that were retrieved by PubMed and PubMed Central query consisting of the gene name and the word "leukemia".

Most of these genes that have been found are relevant to leukemia. Zyxin (encoded in X95735) possesses LIM domain which is known to interact with leukemogenic bHLH proteins (TAL1, TAL2 and LYL1) [23]. Cystatin C (CST3) expression, measured by reverse transcription polymerase chain reaction (RT-PCR), confirmed that the gene was significantly increased in AML patients [19]. TSG-14 plays a critical role in controlling acute inflammatory response in part via the modulation of tumor necrosis factor (TNF)-alpha expression [20]. Cystatin A (acid cysteine proteinase inhibitor, i.e., ACPI) is a natural inhibitor of cysteine proteinases. It has been suggested that an inverse correlation exists between cystatin A and malignant progression [13]. M21551 encodes neuromedin B, which is known to be readily hydrolyzed by neutral endopeptidase, a common acute lymphoblastic leukemia antigen [4]. TIA-1, encoded in M77142, has been found to be a good diagnostic marker for certain type of cell leukemia [15].

Table 3. Relevant genes found by LERS

Gene ID	Gene name	Description	Cross-references	Number of references
X95735	Zyxin	Zyxin	[2, 3, 5, 12, 21]	23
M27891	CST3	cystatin C (CST3) gene, exon 3	[2, 3, 24]	5
M31166	TSG-14	tumor necrosis factor-inducible (TSG-14) mRNA. Tumor necrosis factor inducible gene		3
D88422	cystatin A	Gene for cystatin A	[2, 12]	16
M21551		nuromedin B mRNA	[21]	8
M77142	TIA-1	polydentylate binding protein (TIA-1) mRNA	[2]	42
M55150		fumarylacetoacetate hydrolase mRNA	[2, 3, 5, 12]	4
U46499	MGST1	microsomal glutathione transferase	[2, 3, 22]	4

References

1. Booker, L. B., Goldberg, D. E., and Holland, J. F.: Classifier systems and genetic algorithms. In *Machine Learning. Paradigms and Methods*. Carbonell, J. G. (ed.), The MIT Press, Menlo Park, CA, 235–282, (1990).
2. Broberg P.: Statistical methods for ranking differentially expressed genes. *Genome Biology* **4** (2003) (http://genomebiology.com).
3. Chu W., Ghahramani Z., Falciani F., Wild D. L.: Biomarker discovery in microarray gene expression data with Gaussian processes. *Bioinformatics* **21** (2005) 3385–3393.
4. Cohen A. J., Franklin W. A., Magill C., Sorenson J., Miller Y. E.: Low neutral endopeptidase levels in bronchoalveolar lavage fluid of lung cancer patients. *American Journal of Respiratory and Critical Care Medicine* **159** (1999) 907–910.
5. Golub T. R., Slonim D. K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J. P., Coller H., Loh M. L., Downing J. R., Caligiuri M. A., *et al.*: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* **286** (1999) 531–537.
6. Grzymala-Busse, J. W.: Knowledge acquisition under uncertainty—A rough set approach. *Journal of Intelligent & Robotic Systems* **1** (1988), 3–16.
7. Grzymala-Busse, J. W.: LERS—A system for learning from examples based on rough sets. In Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory, ed. by R. Slowinski, Kluwer Academic Publishers, Dordrecht, Boston, London (1992) 3–18.
8. Grzymala-Busse, J. W.: A new version of the rule induction system LERS. *Fundamenta Informaticae* **31** (1997) 27–39.

9. Grzymala-Busse, J. W.: MLEM2: A new algorithm for rule induction from imperfect data. Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002, Annecy, France, July 1–5, 2002, 243–250.
10. Grzymala-Busse, J. W., Goodwin, L. K., Grzymala-Busse W. J., and Zheng X.: An approach to imbalanced data sets based on changing rule strength. Learning from Imblanced Data Sets, AAAI Workshop at the 17th Conference on AI, AAAI-2000, Austin, TX, July 30–31, 2000, 69–74.
11. Holland, J. H., Holyoak, K. J., and Nisbett, R. E.: Induction. Processes of Inference, Learning, and Discovery. The MIT Press, Menlo Park, CA, (1986).
12. Jirapech-Umpai T., Aitken S.: Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC Bioinformatics* 2005 **6** (http://www.biomedcentral.com).
13. Kuopio T., Kankaanranta A., Jalava P., Kronqvist P., Kotkansalo T., Weber E., Collan Y.: Cysteine proteinase inhibitor cystatin A in breast cancer. *Cancer Research* **58** (1998) 432–36.
14. Lee K. E., Sha N. J., Dougherty E. R. , Vannucci M., Mallick B. K.: Gene selection: a Bayesian variable selection approach. *Bioinformatics* **19** (2003) 90–97.
15. Mori N., Murakami Y. I., Shimada S., Iwamizu-Watanabe S., Yamashita Y., Hasegawa Y., Kojima H., Nagasawa T.: TIA-1 expression in hairy cell leukemia. *Modern Pathology* **17** (2004) 840–846.
16. Nguyen D. V., Rocke D. M.: Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* **18** (2002) 39–50.
17. Pawlak, Z.: Rough Sets. *International Journal of Computer and Information Sciences* **11** (1982) 341–356.
18. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht, Boston, London (1991).
19. Sakhinia E., Faranghpour M., Yin J. A. L., Brady G., Hoyland J. A., Byers R. J.: Routine expression profiling of microarray gene signatures in acute leukaemia by real-time PCR of human bone marrow. *British Journal of Haematology* **130** (2005) 233–248.
20. Souza D. G., Soares A. C., Pinho V., Torloni H., Reis L. F. L., Martins M. T., Dias A. A. M.: Increased mortality and inflammation in tumor necrosis factor-stimulated gene-14 transgenic mice after ischemia and reperfusion injury. *American Journal of Pathology* **160** (2002) 1755–1765.
21. Thomas J. G., Olson J. M., Tapscott S. J., Zhao L. P.: An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles. *Genome Research* **11** (2001) 1227–1236.
22. Vinterbo S. A., Kim E. Y., Ohno-Machado L.: Small, fuzzy and interpretable gene expression based classifiers. *Bioinformatics* **21** (2005) 1964–1970.
23. Wadman I., Li J. X., Bash R. O., Forster A., Osada H., Rabbitts T. H., Baer R.: Specific in-vivo association between the Bhlh and Lim proteins implicated in human T-cell leukemia. *EMBO Journal* **13** (1994) 4831–4839.
24. Yeung K. Y., Bumgarner R. E., Raftery A. E.: Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data. *Bioinformatics* **21** (2005) 2394–2402.

Random Forest of Dipolar Trees for Survival Prediction

Małgorzata Krętowska

Faculty of Computer Science
Białystok Technical University
Wiejska 45a, 15-351 Białystok, Poland
mmac@ii.pb.bialystok.pl

Abstract. In the paper the method of using the ensemble of dipolar trees for survival prediction is presented. In the approach the random forest is applied to calculate the aggregated Kaplan-Meier survival function for a new patient. The induction of individual dipolar regression tree is based on minimization of a piece-wise linear criterion function. The algorithm allows using the information from censored observations for which the exact survival time is unknown. The Brier score is used to evaluate the prediction ability of the received model.

1 Introduction

The development of prognostic tools is one of the major tasks in survival analysis. The physicians are concerned not only with the prediction of the exact survival time for a given patient but also with discovering the factors that influence the survival. The most common statistical method used in analysis of survival data is Cox's proportional hazard model [4]. Its application is limited by additional assumptions required for the analyzed phenomenon. This limitation concerns also other statistical methods. If the requirements are difficult to obey some other techniques are adopted. Among them artificial neural networks and regression trees are considered as ones of the most promising tools.

The analysis of survival data may be treated either as the regression or classification task. In both cases the problem how to treat censored data arises. Censored observations include incomplete knowledge about the exact time of event occurrence. We only know that the true survival time is not less than their follow-up time.

The proposed algorithms of regression trees induction include modifications which allow coping with censored data. Its application in survival analysis aimed at identifying subgroups that are homogeneous in their survival experience. Marubini *et al.* [13] proposed an approach based on Cox's proportional hazards model and partial likelihood approach. Similar method was developed by Ciampi *et al.* [3]. Davis and Anderson [5] assumed an exponential model for the survival distribution and as a goodness-of-split criterion exploited exponential log-likelihood. Krętowska [12] proposed induction of the multivariate tree based on the minimization of a dipolar criterion function.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 909–918, 2006.
© Springer-Verlag Berlin Heidelberg 2006

The problem that arises while analyzing the results calculated on the base of the single tree is instability, especially in discovering the risk factors. To stabilize the predictions, the ensembles of several models are used. Hothorn *et al.* [8] proposed boosting survival trees to create aggregated survival function. The approach proposed by Ridgeway [14] allows minimizing the partial likelihood function (boosting Cox's proportional hazard model). The Hothorn *et al.* [9] developed two approaches for censored data: random forest and gradient boosting. Breiman [2] provided the software that allows induction of the random forest for censored data.

In the paper the random forest consisting of dipolar survival trees is analyzed. The method of building the ensemble of trees is based on the approach proposed by Hothorn *at al.* [8]. The method enables calculation the aggregated Kaplan-Meier survival function for a new patient. The Brier score [7] is used to evaluate the prediction ability of the received model.

The paper is organized as follows. Section 2 describes the survival data and introduces the idea of Kaplan-Meier survival function. In Section 3 induction of dipolar survival tree is presented. Section 4 contains the algorithm how to build the aggregated survival function based on random forest. Experimental results are presented in Section 5. The experiments were carried out on the base of two real datasets. The first one contains the feature vectors describing the patients with primary biliary cirrhosis of the liver [6], the other includes the information from the Veteran's Administration lung cancer study [10]. Section 6 summarizes the results.

2 Survival Data

Let T^0 denotes the true survival time and C denotes the true censoring time with distribution functions F and G respectively. We observe random variable $O = (T, \Delta, \mathbf{X})$, where $T = \min(T^0, C)$ is the time to event, $\Delta = I(T \leq C)$ is a censoring indicator and $\mathbf{X} = (X_1, ..., X_N)$ denotes the set of N covariates from a sample space χ. We observe the learning sample $L = (\mathbf{x}_i, t_i, \delta_i)$, $i = 1, 2, ..., n$, where $\mathbf{x}_i$ is N-dimensional covariates vector, t_i - survival time and δ_i - failure indicator, which is equal to 0 for censored cases and 1 for uncensored.

The distribution of random variable T may be described by the marginal probability of being event free up to a time $t > 0$ ($S(t) = P(T > t)$). The estimation of the survival function $S(t)$ may be done by using the Kaplan-Meier product limit estimator [11]. The Kaplan-Meier function is calculated on the base of learning sample and is denoted by $\hat{S}(t)$:

$$\hat{S}(t) = \prod_{j|t_{(j)} \leq t} \left(\frac{m_j - d_j}{m_j}\right) \tag{1}$$

where $t_{(1)} < t_{(2)} < \ldots < t_{(D)}$ are distinct, ordered survival times from the learning sample L, in which the event of interest occurred, d_j is the number of events at time $t_{(j)}$ and m_j is the number of patients at risk at $t_{(j)}$ (i.e.,

the number of patients who are alive at $t_{(j)}$ or experience the event of interest at $t_{(j)}$).

The 'patients specific' survival probability function is given by $S(t|\mathbf{x}) = P(T > t|\mathbf{X} = \mathbf{x})$. The conditional survival probability function for the new patient with covariates vector $\mathbf{x}_n$ is denoted by $\hat{S}(t|\mathbf{x}_n)$.

3 Dipolar Survival Tree Induction

Hierarchical and sequential structure of a tree recursively partitions the feature space. The tree consists of terminal nodes (leaves) and internal (non-terminal) nodes. An internal node contains a split, which tests the value of an expression of the covariates. In the proposed approach the split is equivalent to the hyper-plane $H(\mathbf{w}, \theta) = \{(\mathbf{w}, \mathbf{x}) :< \mathbf{w}, \mathbf{x} >= \theta\}$. For given covariate vector $\mathbf{x}$, the result of the test is equal to 0, if the inner product $< \mathbf{w}, \mathbf{x} >$ is less then θ and 1, otherwise. Each distinct outcome of the test generates one child node, which means that all non-terminal nodes have two child nodes. A terminal node generates no descendant.

The tree induction aims at establishing the structure of the tree (the number of internal nodes) and the values of hyper-planes parameters. The proposed algorithm [12] is based on the concept of dipoles [1]. The dipole is a pair of different covariate vectors $(\mathbf{x}_i, \mathbf{x}_j)$ from the learning set. Mixed and pure dipoles are distinguished. Mixed dipoles are formed between objects that should be separated, while pure ones between objects that are similar from the point of view of the analyzed criterion. The aim is to find such a hyper-plane $H(\mathbf{w}, \theta)$ that divides possibly high number of mixed dipoles and possibly low number of pure ones. It is done by minimization of the dipolar criterion function.

Two types of piece-wise linear and convex penalty functions $\varphi_j^+(\mathbf{v})$ and $\varphi_j^-(\mathbf{v})$ are considered:

$$\varphi_j^+(\mathbf{v}) = \begin{cases} \delta_j - < \mathbf{v}, \mathbf{y}_j > & \text{if} < \mathbf{v}, \mathbf{y}_j > \leq \delta_j \\ 0 & \text{if} < \mathbf{v}, \mathbf{y}_j > > \delta_j \end{cases} \tag{2}$$

$$\varphi_j^-(\mathbf{v}) = \begin{cases} \delta_j + < \mathbf{v}, \mathbf{y}_j > & \text{if} < \mathbf{v}, \mathbf{y}_j > \geq -\delta_j \\ 0 & \text{if} < \mathbf{v}, \mathbf{y}_j > < -\delta_j \end{cases} \tag{3}$$

where δ_j is a margin ($\delta_j = 1$), $\mathbf{y}_j = [1, x_1, \ldots, x_N]^T$ is an augmented covariate vector and $\mathbf{v} = [-\theta, w_1, \ldots, w_N]^T$ is an augmented weight vector. Each mixed dipole $(\mathbf{y}_i, \mathbf{y}_j)$, which should be divided, is associated with function $\varphi_{ij}^m(\mathbf{v})$ being a sum of two functions with opposite signs ($\varphi_{ij}^m(\mathbf{v}) = \varphi_j^+(\mathbf{v}) + \varphi_i^-(\mathbf{v})$ or $\varphi_{ij}^m(\mathbf{v}) = \varphi_j^-(\mathbf{v}) + \varphi_i^+(\mathbf{v})$). For pure dipoles that should remain undivided we associate function: $\varphi_{ij}^p(\mathbf{v})$ ($\varphi_{ij}^p(\mathbf{v}) = \varphi_j^+(\mathbf{v}) + \varphi_i^+(\mathbf{v})$ or $\varphi_{ij}^c(\mathbf{v}) = \varphi_j^-(\mathbf{v}) + \varphi_i^-(\mathbf{v})$). A dipolar criterion function is a sum of the penalty functions associated with each dipole:

$$\Psi_d(\mathbf{v}) = \sum_{(j,i) \in I_p} \alpha_{ij} \varphi_{ij}^p(\mathbf{v}) + \sum_{(j,i) \in I_m} \alpha_{ij} \varphi_{ij}^m(\mathbf{v}) \tag{4}$$

where α_{ij} determines relative importance (price) of the dipole $(\mathbf{y}_i, \mathbf{y}_j)$, I_p and I_m are the sets of pure and mixed dipoles, respectively.

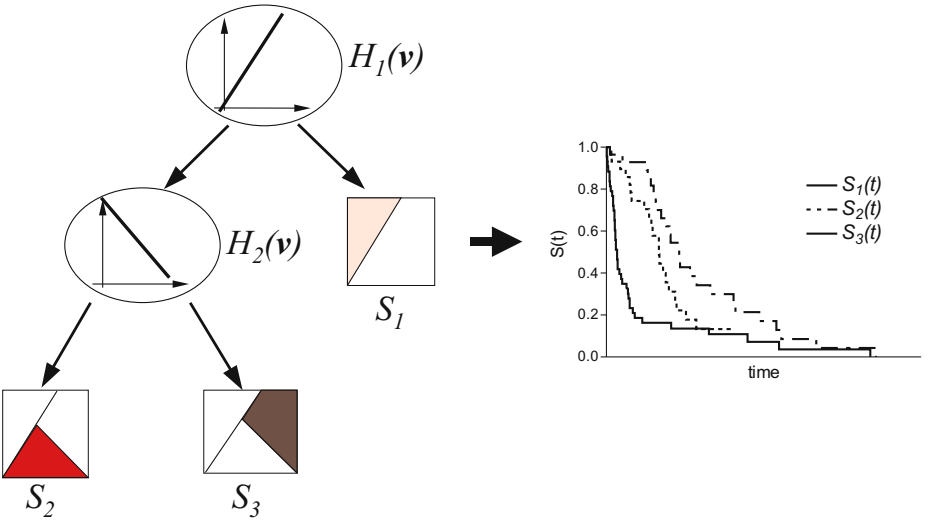


Fig. 1. An example of survival tree

The rules of dipoles formations depend on the purpose of our research. Assuming that the analysis aims at dividing the feature space into such areas, which would include the patients with similar survival times (see Fig. 1), pure dipoles are created between pairs of feature vectors, for which the difference of failure times is small, mixed dipoles - between pairs with distant failure times. Taking into account censored cases the following rules of dipole construction can be formulated:

1. a pair of feature vectors $(\mathbf{x}_i, \mathbf{x}_j)$ forms the pure dipole, if
   - $\sigma_i = \sigma_j = 1$ and $|t_i - t_j| < \eta$
2. a pair of feature vectors $(\mathbf{x}_i, \mathbf{x}_j)$ forms the mixed dipole, if
   - $\sigma_i = \sigma_j = 1$ and $|t_i - t_j| > \zeta$
   - $(\sigma_i = 0, \sigma_j = 1$ and $t_i - t_j > \zeta)$ or $(\sigma_i = 1, \sigma_j = 0$ and $t_j - t_i > \zeta)$

Parameters η and ζ are equal to quartiles of absolute values of differences between uncensored survival times. The parameter η is fixed as 0.2 quartile and ζ - 0.6. The hyper-planes in the internal nodes of the tree are computed by minimization of dipolar criterion function, starting from the root. The function in a given node is designed on the base on those feature vectors that have reached the node. The induction of survival tree is stopped if one of the following conditions is fulfilled: 1) all the mixed dipoles are divided; 2) the set that reach the node consists of less than 5 uncensored cases.

4 Random Forest Algorithm

The random forest method [8] allows estimation the conditional survival function $\hat{S}(t|\mathbf{x}_n)$ on the base of k learning samples $(L_1, L_2, \ldots, L_k)$ drawn with replacement from the given sample L. For each learning sample L_i $(i = 1, 2, \ldots, k)$ the set of observations $L_i(\mathbf{x}_n)$ which are close to covariates vector $\mathbf{x}_n$ is distinguished.

The dipolar survival tree is calculated for each learning set L_i, $i = 1, 2, \ldots, k$. The covariates vector $\mathbf{x}_i$ is included to the set $L_i(\mathbf{x}_n)$ when it belongs to the same leaf of the survival tree as $\mathbf{x}_n$ itself. Having k sets $L_i(\mathbf{x}_n)$, aggregated sample $L_A(\mathbf{x}_n)$ is built:

$$L_A(\mathbf{x}_n) = [L_1(\mathbf{x}_n); L_2(\mathbf{x}_n); \ldots; L_k(\mathbf{x}_n)]$$

The aggregated conditional Kaplan-Meier survival function, calculated on the base of the set $L_A(\mathbf{x}_n)$ can be referred to as $\hat{S}_A(t|\mathbf{x}_n)$.

To summarize the above considerations, the random forest algorithm leading to receive the aggregated survival function is as follows:

1. Draw k bootstrap samples $(L_1, L_2, \ldots, L_k)$ of size n with replacement from L
2. Induction of dipolar survival tree $T(L_i)$ based on each bootstrap sample L_i
3. Build aggregated sample $L_A(\mathbf{x}_n) = [L_1(\mathbf{x}_n);\ L_2(\mathbf{x}_n), \ldots, L_k(\mathbf{x}_n)]$
4. Compute the Kaplan-Meier aggregated survival function for a new observation $\mathbf{x}_n$: $\hat{S}_A(t|\mathbf{x}_n)$.

For the evaluation of prediction ability of the method the Brier score introduced by Graf *at al.* [7] was used. The Brier score as a function of time is defined by

$$BS(t) = \frac{1}{n}\sum_{i=1}^{N}(\hat{S}(t|\mathbf{x}_i)^2 I(t_i \le t \wedge \sigma_i = 1)\hat{G}(t_i)^{-1} + (1 - \hat{S}(t|\mathbf{x}_i))^2 I(t_i > t)\hat{G}(t)^{-1}) \quad (5)$$

where $\hat{G}(t)$ denotes the Kaplan-Meier estimator of the censoring distribution. It is calculated on the base of observations $(t_i, 1 - \delta_i)$. $I(condition)$ is equal to 1 if the condition is fulfilled, 0 otherwise.

5 Experimental Results

The analysis was conducted on the base on two datasets. The first data is from the Mayo Clinic trial in primary biliary cirrhosis (*PBC*) of the liver conducted between 1974 and 1984 [6]. A total of 424 PBC patients, referred to Mayo Clinic during that ten-year interval, met eligibility criteria for the randomized placebo controlled trial of the drug D-penicillamine. The first 312 cases in the data set participated in the randomized trial and contain largely complete data. The additional 106 cases did not participate in the clinical trial, but consented to have basic measurements recorded and to be followed for survival. The analysis

was done on the base of 418 patients described by the following features: age, sex, presence of edema, serum bilirubin in mg/dl, albumin in gm/dl, platelets per cubic ml/1000, prothrombin time in seconds, histologic stage of disease. The number of days between registration and the earlier of death, transplantation, or study analysis time in July 1986 was available.

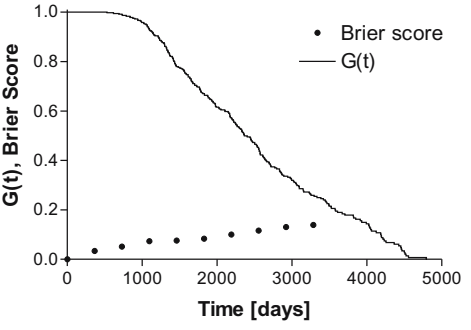


Fig. 2. Kaplan-Meier estimator $G(t)$ of the censoring distribution $G(t) = P(C > t)$; Brier score for selected values of t

In Fig. 2 Kaplan-Meier estimator of the censoring distribution together with the Brier score received for selected values of time (1, 2, 3..., 10 years) are presented. We can see that the Brier score values are quite small for the lower values of time and increase over time. The maximum value of Brier score for t=10 years is equal to 0.14.

In Fig. 3 we can observe the differences between Kaplan-Meier survival functions for 50 years old women and men with two different levels of serum bilirubin and four histologic stages of disease. Other features, which were not considered in the analysis, were fixed to their median values: absence of edema; albumin = 3; platelets = 251; prothrombin time = 11.

The impact of the level of serum bilirubin for survival can be observed in the figures. The survival for patients with lower value of serum bilirubin is much better than for patients with the value equal to 5. Taking into account two upper figures (see Fig. 3) one can see that the differences between survival functions for the first three histologic stages are not significant. The worse survival prediction is for men with histologic stage 4 (median survival time equal to 4079 days). The survival prediction for serum bilirubin equal to 5 is worse in all analyzed cases. For men we can see significant differences between the function for the first histologic stage and other stages. Median survival time is equal to 3244, 1657, 1478 and 1462 [days] for the consecutive histologic stages. The survival functions for women with different stages of disease are more diverse. The best prediction is for the first stage and is getting worse as the stage number increases. Median survival time is equal to 3358 (1st stage), 2081, 1297 and 1152 for the 4th histologic stage. We can say that women response better for the given treatment

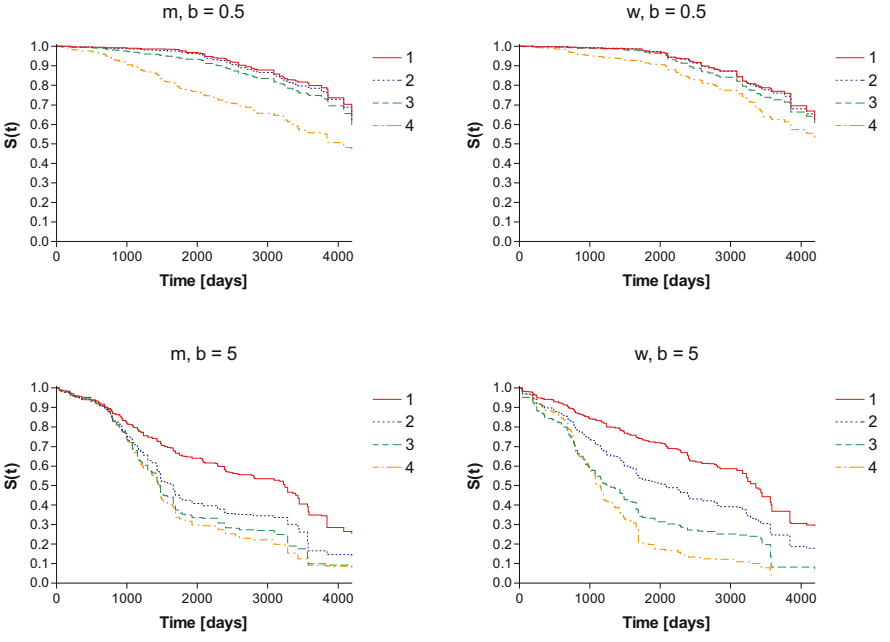


Fig. 3. Kaplan-Meier survival functions for women (w) and men (m) with two different values of serum bilirubin (b=0.5 and b=5) for each histologic stage of disease

for the first and second stage of disease. For the third and fourth stage of disease better survival prediction is for men.

The other analyzed dataset contains the information from the Veteran's Administration (*VA*) lung cancer study [10]. In this trial, male patients with advanced inoperable tumors were randomized to either standard (69 subjects) or test chemotherapy (68 subjects). Only 9 subjects from 137 were censored. Information on cell type (0 - squamous, 1 - small, 2 - adeno, 3 - large), prior therapy, performance status at baseline (Karnofsky rating - KPS), disease duration in months and age in years at randomization, was available.

In Fig. 4 the Brier score for selected times (0, 50, ..., 950 days) and estimated curve of censoring distribution for *VA* lung cancer study are presented. In contrast to the *PBC* dataset, the Brier score values decrease over time. It is due to the small number of censoring cases in the dataset. The shape of function $G(t)$ suggests that there are no censored cases with survival times greater than 250 days.

The analysis aims at discovering the factors that influence the survival. Therapy, KPS and cell type were taken into account. The estimated survival functions for 50 years old patients without prior therapy and five months of disease duration are shown in Fig. 5.

The results suggest that the cell type does not influence the survival, especially when the standard therapy is applied. Median survival times (Table 1) obtained

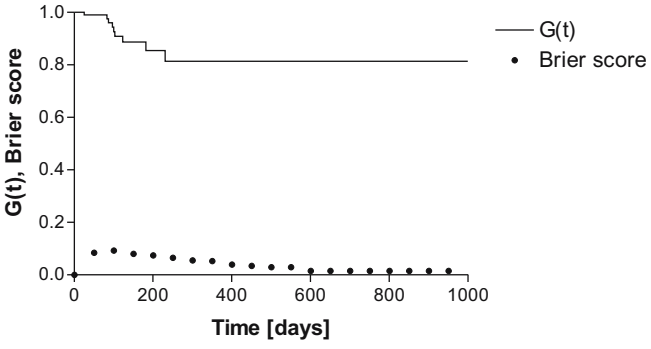


Fig. 4. Kaplan-Meier estimator $G(t)$ of the censoring distribution $G(t) = P(C > t)$ for *VA* lung cancer study; Brier score for selected values of t

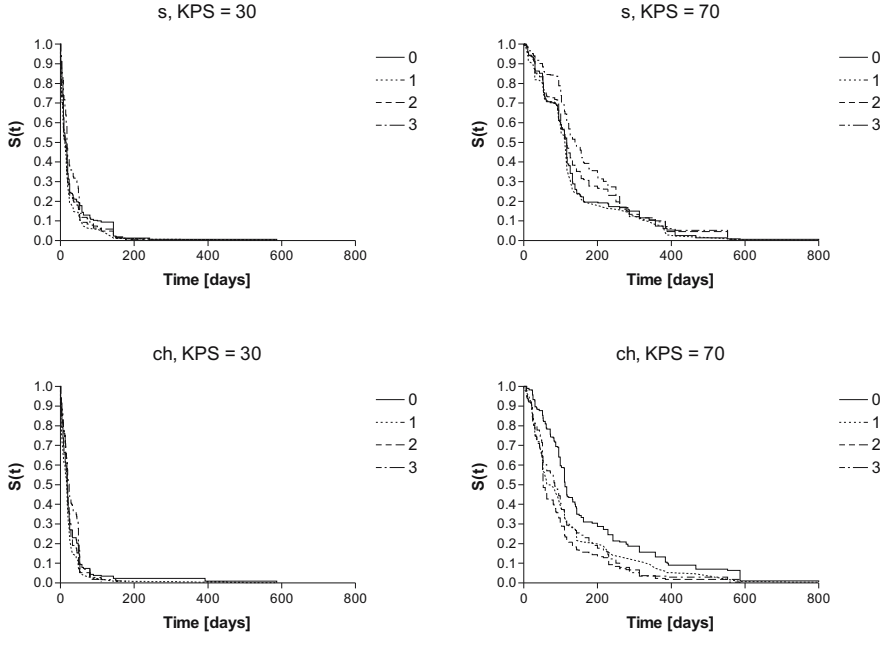


Fig. 5. Kaplan-Meier survival functions for standard (s) and chemotherapy (ch) with two different values of KPS (KPS=30 and KPS=70) for each cell type

for different cell types (with the same values of the remaining attributes) do not differ significantly. The only difference may be observed for patients with chemotherapy with KPS equal to 70. The survival function received for patients with squamous cell type indicates better prognosis (median survival time equal to 112 days) than for other patients (median survival times equal to 72, 53 and 83, respectively).

Table 1. Median survival times for *VA* lung cancer study

	Cell type			
	0	1	2	3
Standard th.				
KPS=30	16	12	13	19
KPS=70	117	112	117	139
Chemotherapy				
KPS=30	19	18	19	24
KPS=70	112	72	53	84

Significant differences between survival curves are visible for different values of KPS. As we can see the survival functions obtained for either standard and test chemotherapy for KPS equal to 30 are similar. The median survival time varies from 12 to 24 days for patients with large cell type and chemotherapy. Much better prognosis is for patients with performance status at baseline equal to 70. The smallest median survival time is for patients with adeno cell type and chemotherapy (53 days) and the best one for individuals with large cell type and standard therapy (139 days).

6 Conclusions

In the paper the random forest consisting of dipolar survival trees was analyzed. The applied method enables calculation of the aggregated Kaplan-Meier survival function for a new patient. The survival tree induction as well as the estimation of aggregated survival function enables using the information from censored cases.

Experiments were carried out on the base of two real datasets. The first one contains the feature vectors describing the patients with primary biliary cirrhosis of the liver, the other includes the information from the Veteran's Administration lung cancer study. The method was used to analyze the influence of histologic stage and serum bilirubin for the survival of patients with primary biliary cirrhosis of the liver and the influence of therapy, KPS and cell type for survival of patients form *VA* lung cancer study. The goodness of prediction was calculated using Brier score. As it was noticed, the values of the Brier score for the first dataset are increasing over time. It may mean that the prediction is better for short time prognosis, but on the other hand the percentage of censored cases increases over time, what also affects the value of Brier score. The *VA* lung cancer dataset contains only 6% of censored cases with respectively short follow-up time. The Brier score values are increasing over the first, short period of time and then starting to decrease. In this case, the method performs well even for long time prognosis.

Acknowledgements. This work was supported by the grant W/WI/4/05 from Białystok Technical University.

References

1. Bobrowski L., Krętowska M., Krętowski M., Design of neural classifying networks by using dipolar criterions. Proc. of the Third Conference on Neural Networks and Their Applications, Kule, Poland (1997) 689–694
2. Breiman L., How to use survival forest. [URL http://stat-www.berkeley.edu/users/breiman]
3. Ciampi A., Negassa A., Lou, Z., Tree-structured prediction for censored survival data and the Cox model. Journal of Clinical Epidemiology **48**(5) (1995) 675-689
4. Cox D.R., Regression models and life tables (with discussion). Journal of the Royal Statistical Society B **34** (1972) 187–220
5. Davis R. B., Anderson J. R., Exponential survival trees. Statistics in Medicine **8**(1989) 947–961
6. Fleming T.R., Harrington D.P., Counting Processes and Survival Analysis. John Wiley & Sons, Inc. (1991)
7. Graf E., Schmoor C., Sauerbrei W., Schumacher M., Assessment and comparison of prognostic classification schemes for survival data. Statistics in Medicine **18** (1999) 2529–2545
8. Hothorn T., Lausen B., Benner A., Radespiel-Troger M., Bagging survival trees. Statistics in medicine **23** (2004) 77–91
9. Hothorn T., Buhlmann P., Dudoit S., Molinaro A. M., van der Laan M. J., Survival ensembles. [URL http://www.bepress.com/ucbbiostat/paper174] U.C. Berkeley Division of Biostatistics Working Paper Series **174** (2005)
10. Kalbfleisch J.D., Prentice R.L., The statistical analysis of failure time data. John Wiley & Sons, New York (1980)
11. Kaplan E.L., Meier P., Nonparametric estimation from incomplete observations, Journal of the American Statistical Association **5** (1958), 457–481
12. Krętowska M., Dipolar regression trees in survival analysis. Biocybernetics and biomedical engineering **24** (3) (2004) 25–33
13. Marubini E., Morabito A., Valsecchi G., Prognostic factors and risk groups: Some results given by using an algorithm suitable for censored survival data. Statistics in Medicine **2** (1983) 295–303
14. Ridgeway G., The state of boosting. Computing Science and Statistics **31** (1999) 1722–1731

Interpretation of Medical Images Based on Ontological Models

Juliusz L. Kulikowski

Institute of Biocybernetics and Biomedical Engineering PAS, 4 Ks. Trojdena Str.
02-109 Warsaw, Poland
jlkulik@ibib.waw.pl

Abstract. It is presented a concept of using ontological models as a form of medical knowledge representation for computer-aided interpretation of medical images. The models are based on hyper-relations linking the concepts of visual objects, visualized medical objects and related non-visualized real objects. It is shown that due to the algebra of hyper-relations such ontological models provide a possibility to describe large variety of situations being of particular interest in image interpretation tasks. It is defined an association area of a given concept. In the case of visual objects this makes possible characterization of a class of questions that within the given ontological model can be posed and answered by interpretation of the visual objects under examination.

1 Introduction

Interpretation of images is aimed at logically based answering questions concerning the visualized objects: their nature, behavior, relations to other objects in a given environment, etc. [1] Advanced computer-aided interpretation of images related to any application area should be based on resources of knowledge about the real world (or its part) under examination. A computer-readable, structured knowledge of this type constitutes an *ontology* of the world. According to [2], a formal description of the ontology can be given by a quadruple:

$$O = [C, R, A, Top] \tag{1}$$

where C is a non-empty set of *concepts* (including the relation concepts), R is a set of all *assertions* in which two or more concepts are related to each other and A is a set of *axioms*; among the relations a *taxonomy* of all concepts with the *highest-level* concept *Top* is defined. A *concept* means here an abstract representation of any real or abstract object, person, event, process, etc. which can be distinguished and about which some statements (logically validated propositions) can be formulated.

Despite a generality of the above-given definition of ontology which suits well to the solution of many information retrieval, text processing, knowledge management, e-business, automatic control and other tasks it seems that for image interpretation purposes it should be extended and modified. First, the classical concept of *relation* should be replaced by a more general concept of *hyper-relation*

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 919–924, 2006.
© Springer-Verlag Berlin Heidelberg 2006

(in the sense proposed in [3]), including *relation* as its particular case. Second, hyper-relations should be considered as constituting a sort of composite formal structures consisting of hyper-relations of various levels. Third, in practice, some *domain ontologies* related to some *areas of interest*, i.e. limited to some parts and/or to *selected aspects* of the world, should be considered. That is why in this paper a specific, oriented to image interpretation, approach to ontologies is presented. In this approach a basic role is played by *ontological models*. Basic concepts and perspectives of using ontological models to image interpretation were indicated in [4]; in this paper they are presented in a more extended form.

2 Ontological Models

We start with presentation of several examples of typical questions that may be posed in connection with interpretation of some medical images:

- Is the *light structure* visible in the left-upper part of the mammogram of a *spiky form*?
- Is the *visible structure* typical for advanced cancerous changes?
- Is the detected *cancerous structure* morphologically adjacent to any *blood artery* in the breast?
- Does the *detected structure* need a *biopsy* in order to prove its *cancerous character*?
- Is the *cancerous structure* recommended to be medically treated by *radiological* or by *surgical* means?

etc. It can be observed that five types of questions have been represented here: 1^{st} the ones concerning the details of images under observation, 2^{nd} concerning the relations between the details in the image and the visualized objects, 3^{rd} concerning relations among several visualized objects, 4^{th} concerning the visualized objects and some related to them external objects, in the image directly non-visible, and 5^{th} concerning relations among external objects related to some other objects visualized in the image.

Answering any type of questions within a given objective area needs using due ontological models from an adequate *domain-oriented ontology*. The last, denoted by Θ, is here considered as a *composition* of a finite set of *ontological models* $\{M_\kappa\}$, $\kappa = 1, 2, \ldots, k$, describing a given area of interest from some strongly defined points of view:

$$\Theta = [\{M_\kappa\}, H], \tag{2}$$

H denoting a set of constraints imposed on the models. Each ontological model is given by a pair:

$$M_\kappa = [C_\kappa, \{r_{\kappa,i}\}] \tag{3}$$

where C_κ is a subset of concepts, and $\{r_{\kappa,i}\}$ is a finite subset of *relations* described on finite Cartesian powers of C_κ . Relations are here considered in a large

sense that will be explained in the next section. Then two types of constraints may be contained by H: 1^{st} having the form of additional relations imposed on selected subsets of $\{C_\kappa\}$, and 2^{nd} the ones in the form of super-relations imposed on some combinations of relations of various models M_κ.

For image interpretation purposes the following classes of objects (and related concepts) should be taken into account:

- *Visual objects* i.e. any geometrical, morphological, statistical etc. details or structures occurring in the image under examination;
- *Visualized real objects* i.e. those existing in real world, represented by visual objects in the examined image;
- *Related objects* i.e. objects, processes, relations etc. not visible in the examined image but in certain sense related (depending, caused, influencing etc.) to the visualized objects.

Consequently, the following categorization of ontological models for image interpretation can be established:

1) *visual objects* oriented models,
2) *visual objects* - visualized real objects oriented models,
3) *visualized real objects* oriented models,
4) *visualized real objects* - related objects oriented models,
5) *related objects* oriented models.

They may be distinguished among them *homogenous models*, based on relations between concepts of the same class, and *heterogenous models*, based on relations linking concepts of different classes.

Visual objects oriented models are a domain of investigation of structural image analysis methods [5, 6, 7]. Such models, related to visual structures only, are not able *per se* to help answering questions concerning physical or biological meaning or the nature of observed real objects. However, if certain structures in a strongly defined context can be considered as visualization of some real objects then a second-type relationship is established. This type makes it possible to interpret the observed image details in the higher-level terms associated with real world concepts, as it will be shown below.

Some relations of $\{r_{\kappa,i}\}$ may establish taxonomies of the concepts of C_κ. A taxonomy is then represented by a tree whose root corresponds to C_κ as a whole and the leafs represent the subsets of the lowest hierarchical level concepts: their single (individual) exemplifications, admitted values of concepts defined as parameters, etc. The middle-level nodes correspond to taxonomic subclasses of concepts. Other relations of $\{r_{\kappa,i}\}$ can be then described on taxonomic subclasses of concepts.

3 Relational Structures for Image Interpretation

As mentioned above, a classical definition of a n-th order relation as a subset:

$$r_\kappa \subseteq C_1 \times C_2 \times \ldots \times C_n, \tag{4}$$

where $C_1 \times C_2 \times \ldots \times C_n$ is a Cartesian product of a linearly ordered family of sets, is not quite suitable to the description of widely understood relationships occurring in the reality. For example, a *configuration of a finite number of points located on a circular contour of finite diameter on a plane* cannot be easily described as a fixed-order relation between points on the plane.

This shortcoming can be overcome by replacing the concept of relation by hyper-relation, as formally defined in [3]. Roughly speaking, a *first-type hyper-relation (h-relation)* described on a finite family of sets $\{C_\nu\}$, $\nu = 1, 2, \ldots, n$, is a sum of any relations described on any subfamilies of $\{C_\nu\}$ used in any linear orders.

We call *syndromes* of h-relation the satisfying it strings of elements. As syndromes of h-relation are thus admitted strings of elements drawn from any and in any way linearly ordered subfamilies of $\{C_\nu\}$. In particular, if only one linear order in $\{C_\nu\}$ is fixed and only one Cartesian product described on it is taken into account then the h-relation becomes a relation in the classical sense. On the other hand, a fully symmetrical h-relation (i.e. such that together with any its syndrome it is satisfied by all its permutations) is equivalent to a *hyper-graph* in the sense of Berge [8] described on the sum of sets $C = \bigcup_\nu C_\nu$ considered as the set of its nodes.

For any given set C it can be defined a set U_C of all possible strings of elements drawn from all possible and in any possible ways linearly ordered subfamilies of C. We call U_C an *universal extension* of C. For example, if $C = \{a, b, c\}$ then

$$U_C = \{[a,b,c],[a,c,b],[b,a,c],[b,c,a],[c,a,b],[c,b,a],\\ [a,b],[b,a],[a,c],[c,a],[b,c],[c,b],[a],[b],[c]\}$$

Any h-relation described on C is thus a subset of the universal extension U_C . Denoting by 2^{UC} a family of all possible subsets of U_C including U_C itself and an empty subset $\oslash$ one can define an algebra of h-relations as the algebra of subsets of U_C . This gives us the possibility to describe any h-relation on the given basis C in several ways: 1^{st} by a direct listing of its syndromes, 2^{nd} as a subset of syndromes of another h-relation described on C, 3^{rd} as an algebraic combination of some other h-relations described on C and 4^{th} by selection of its syndromes from U_C using corresponding logical tests.

The concept of h-relation together with the algebra of h-relations provide us with effective and flexible tools for construction of ontological models. The tools are also universal in the sense that basic types of statements like: factographical, inferring, deontical or evaluating, usually stored in factographical databases can be represented as syndromes of h-relations [18].

Let us consider an ontological model M_κ given by (3). It will be taken into account a concept $u_i \in C_\kappa$ belonging to M_κ. It may occur in the syndromes of some h-relations of $\{r_{\kappa,i}\}$ together with some other, associated with it concepts $u_j \in C_\kappa$. To each u_i in M_κ it thus may be assigned a subset $S_i \subseteq C_\kappa$ of all concepts associated with u_i; it will be called an *association area* of u_i. The last idea has a simple interpretation in pattern recognition, as it shows the following example.

Example

It is considered a system of epithelial neoplasms early detection by a computer-aided analysis of microscopic cytological images. For this purpose an ontological model based on two types of concepts will be taken into account: C_1 - a selected part of a medical taxonomy of diseases [10], C_2 - a set of parameters describing the form of cells and of their clusters in cytological specimens [11]. Typical examples of the concepts are:

C_1: *neoplasma epitheliale benignum, neoplasma epitheliale, carcinoma*, etc.
C_2: *object specific area, Feret diameter, elliptic shape factor*, etc.

For the sake of simplicity the parameters can be discretized, for example:

$C_{2,1}$ (*object specific area*) =$\{$*very small, small, moderate,. . . , extremely large*$\}$.

Therefore, we put $C_2 = [C_{2,1}, C_{2,2}, \ldots, C_{2,k}]$. A heterogenous ontological model is based on a h-relation linking the levels of visualized and visual objects. For this purpose it is considered a linearly ordered family of sets $F = [C_1, C_{2,1}, C_{2,2}, \ldots, C_{2,k}]$. A general structure of syndromes of the h-relation will be as follows:

$$\mathbf{s} = [\sigma, \gamma_\alpha, \gamma_\beta, \ldots, \gamma_\tau]$$

where $\sigma \in C_1$, $\gamma_\alpha \in C_{2,\alpha}$, $\gamma_\beta \in C_{2,\beta}$,. .., $\gamma_\tau \in C_{2,\tau}$ and $C_{2,\alpha}$, $C_{2,\beta}, \ldots, {}_{2,\tau}$ are selected components of C_2 taken without changing the order established in F.

The syndromes to each type of epithelial neoplasm assign a set of values of cells' parameters sufficient to conclude on a given certainty level that this type of neoplasm occurs in the given cytological specimen. Let us assume that the h-relation is given, and a certain i $\sigma \in C_1$ (say, denoting *carcinoma*) is considered. Then $S_i \subseteq C_{2,1} \times C_{2,2} \times \ldots \times C_{2,k}$ represents a similarity class of observations assigned to the class σ_i •

Let us take into account a concept u_i and its association area S_i. Let $u_j \in S_i$ be another concept, associated to the former one. Then we can take into consideration its association area S_j. Of course, u_i and u_j both belong to the intersection $S_i \cap S_j$. However, they may exist some concept, say $u_m \in S_j \setminus S_i$, not directly but through u_j associated to u_i. They extend the association area S_i The difference $S'_{i,j} = S_j \setminus S_i$ thus can be called an extension of S_i induced by the associated concept u_j. The sum of all such extensions induced by all concepts $u_j \in S_i$ will be called a *second-order association area of* u_i. In similar way the third-, fourth- and higher-order extended association areas of u_i can be defined. A practical sense of this is such that if a certain concept u_s belongs to an extended m-th order associated area of u_i than it exists a chain of at most m concepts connected with u_i by syndromes of relations or hyper-relations. If this condition is satisfied then an answer is possible by periphrases of the corresponding relations between the associated concepts. Of course, not all so-obtained answers are of practical value for the users. Therefore, they need an additional pragmatic validation.

4 Conclusions

Formal ontologies are a suitable form of knowledge representation for computer interpretation of images. In any given medical application area they can be represented by an adequate collection of ontological models. The models should be based on sets of concepts related to visual objects extracted from images, visualized medical objects, and related to them non-visualized objects. Ontological models should describe relationships between the concepts. For this purpose it is desirable to use hyper-relations as an extension of classical relations. On the basis of hyper-relations it is possible to outline an associative area of any identified visual object within which reasonable questions concerning this object on the basis of the ontological model can be answered. This indicates a real way to the design of computer-aided medical image interpretation systems.

References

1. Kulikowski J.L.: From Pattern Recognition to Image Interpretation. Biocybernetics and Biomedical Engineering, 22(2-3): 177-197, 2002.
2. Shamsfard M., Abdollahzadeh Barforoush A.: The State of the Art in Ontology: a Framework for Comparison. The Knowledge Eng. Rev. 18(4): 293-316, 2003. http://delicias.dia.fi.upm.es/WORKSHOP/ECAI00/accepted-papers.html
3. Kulikowski J.L. Recognition of Hyperrelations in Computer Knowledge Bases (in Polish). In: Inzynieria Wiedzy i Systemy Ekspertowe, OW Politechniki Wroclawskiej, Wroclaw, 281-291, 2000.
4. Kulikowski J.L.: The Role of Ontological Models in Pattern Recognition. In: Computer Recognition Systems. Int. Conf. CORES'05. Springer, 2005: 43-52.
5. Fu K.S.: Syntactic Methods in Pattern Recognition. Academic Press, New York, 1974.
6. Miclet M: Structural Methods in Pattern Recognition. North Oxford Academic Publ., 1986.
7. Kulikowski J.: Algebraic Methods in Pattern Recognition. ICMS Courses and Lectures No 85. Springer Verlag, Wien, 1971.
8. Berge C.: Graphs and Hypergraphs. North Holland, Amsterdam, 1973.
9. Kulikowski J.L.: Principles of Structural Description of Resources of Distributed Databases (in Polish). INFOBAZY'2005, Politechnika Gdanska: 29-38, 2005.
10. SNOMED - a Systemised Medical Nomenclature. Morphology (in Polish)). IBIB PAN, Warsaw, 1997.
11. Zielinski K.W., Strzelecki M:. Computer Analysis of Medical Images. An Introduction to Numerical Morphometry and Pathology (in Polish). PWN, Warsaw, 2002.

Fuzzy Logic in Stuttering Therapy

Halina Kwasnicka and Blazej Zak

Institute of Applied Informatics, Wroclaw University of Technology

Abstract. Artificial intelligence (AI) was always widely explored by medical systems. We follow the trend, and try to use AI techniques to develop *Orator* – a system that would be helpful in a therapy of stuttering patients. The paper describes very shortly the background and the solution of intelligently controlling therapy parameters. Logopedist and patients opinions are shown as the evaluation of developed system.

1 Stuttering – The Short Introduction

Speech fluency is a natural speech flow. The four basic factors make speech considered to be fluent:

1. Continuity – it basically tells if the pauses are in syntactically and semantically right places of the speech stream.
2. Rate, expressed in syllables per second, tells how fast do we speak.
3. "Rhythm of speech is the sense of the flow of speech one gets from the stress, duration and timing of syllables. In English, stressed syllables, which are longer, louder, and higher-pitched than unstressed syllables, occur at more or less equal intervals interspersed with one or more unstressed syllables."[1].
4. Effort is the physical and mental power that one uses to produce a speech.

Stuttering is a complex multi-dimensional speech fluency disorder [2]. It involves disorders in all four factors mentioned above. What is interesting, persons who stutter usually have tendency of trying to speak "too fast", but unattended blocks just make them unable to do so. The most common observed type of stutters are [1, 8]:

- Repetitions of sound, syllable, word or phrases: *I t-t-talk like this; I ta-ta-talk like this; I talk talk like this; I talk I talk like this.*
- Sound prolongation: *I aaaaate my lunch.*
- Blocks: *I . . . . . don't know* – the inability to produce audible speech at the beginning of an utterance or word (tense pause) or in the middle of a word.

There are many theories trying to explain the stuttering phenomenon[2, 3]: Physiological, Psychological, Linguistic, and Neurological. It is observed that most of those who stutters seem to try to speak very fast "to say everything before a block occurs". Fluency training aims to change this strategy and teach them to speak slow, but do it fluently[1]. Very interesting technique making stutterers speak just slightly slower, is chorus speaking. The variation of this technique

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 925–930, 2006.
© Springer-Verlag Berlin Heidelberg 2006

is called Delayed Auditory Feedback (DAF), it involves producing echo for a stuttering person, so he hears himself with a little delay[4, 5, 7]. The similar technique to DAF is Frequency Altered Feedback (FAF), it involves producing pitch-shifted echo for the stutterer. Person using FAF simply hears himself speaking in higher or lower tone. The most effective is technique merging the two – FAF and DAF, therefore we propose the system *Orator* with frequency altered and delayed auditory feedback.

2 The *Orator* System

The *Orator* system is a PC software that can also act in the same way as a traditional DAF device. It implements sound filters necessary to carry on echo-therapy, those are: echo effect, reverb effect and chorus effect. However, unlike traditional DAF device, *Orator* can also adjust all the therapy parameters automatically, and "tune" them adaptively to the patient. Those parameters are controlled adaptively in real time basing on information about training progress. There are three basic components of the system: Stuttering Detector – it analyses sound and detects whether stuttering occurred or not; Therapy Controller – a component controlling parameters of therapy according to therapy progress; Provider – a component providing Delayed Auditory Feedback – delay, reverb and chorus effects.

The *Orator* can work in two modes of interaction with patients: manual and automatic.

In the *manual mode* user uses sliders to adjust parameters of therapy. The system acts virtually the same as a traditional DAF device. There are four basic parameters: Master volume – the global feedback volume; Echo delay; Reverb volume; Chorus volume.

In the *automatic mode* there are no sliders at all, since the therapeutic session is fully automatically controlled. Anyway before starting a session patient has to go through three simple steps to setup and calibrate the system, those are:

1. Choosing initial volume, so the system will not hurt his ears.
2. Recording noise sample for noise filtering component – noise filtering is unfortunately necessary if we do not use high-end microphone and sound card. To perform FFT noise filtering a noise sample has to be recorded beforehand,
3. Reading three sentences of text provided to calibrate the stuttering detection component.

After these three steps the system is ready to use, and does not really interact with the patient through other than auditory channel. The patient task is just to talk or read some text laudly, the system will adjust all the parameters itself. The flowchart in Fig. 1 presents an overview of the whole process of detecting stutters in voice. In the paper we focus on the methods of stuttering detection and automatic control of therapy applied in the *Orator*. Our stuttering detection system is based on two measures scattered in time: wave energy, and spectral distance. Using them we calculate the *repetition* signal [6] which represents quite

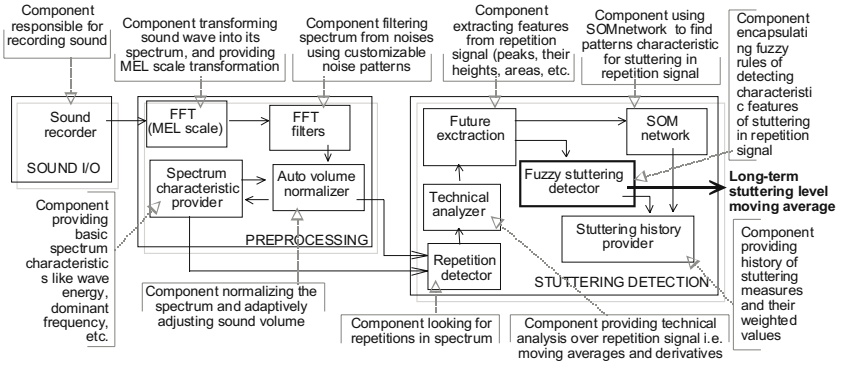


Fig. 1. The general scheme of Orator system

well repetitions in recorded sound as far as they can be observed in spectrum. The next task is to find patterns characteristic for stuttering in the repetition signal. We tested SOM neural networks for this task, but the results were not good. This was probably because stutters are usually surrounded by blocks that are simply silence (see Fig. 2), but unfortunately SOM network did not distinguish them from normal pauses, and it was usually yielding stutter at ends of words where pause was starting.

Fuzzy logic approach

Many observations show that during a simple syllable or sound repetition stutter, the repetition signal looks like in the Fig. 2. Those observations have also a theoretical base. The graph represents double stutter – a double sound repetition (that is represented by the peaks) that is interleaved by speech blocks. Speech block is simply a silence and hence is represented as a very low plateau

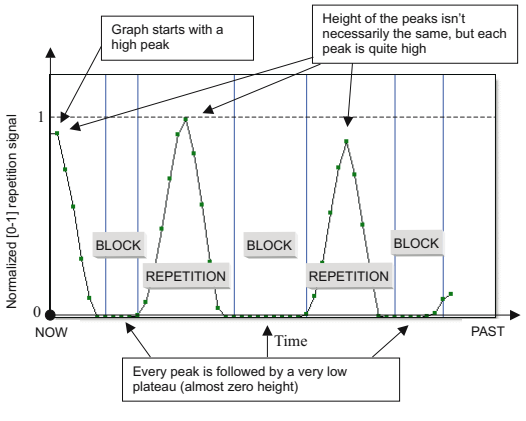


Fig. 2. Syllable repetition stutter characteristic features

on repetition signal graph. Such repetition stutters can be used as a base for the stuttering classifier. Fuzzy logic seems to be a right tool for solving not sharply defined problems, such as finding mentioned above pattern. For finding the pattern just two rules were necessary. The rules are as simple as follows (Fig. 3 shows used fuzzy sets):

```
if (Peak1 IS High) and (Peak2 IS Low) and (Peak3 is High) then SingleStutterDetected;
if (SingleStutterDetected) and (Peak4 is Low) and (Peak5 is High) then DoubleStutterDetected;
```

Observations show that patterns similar to single stutter occur also in fluent speech so relying on them will not give the best discrimination results. However, patterns described by double stutter rule can be rarely observed in fluent speech whereas they are very common in stuttering. Hence when evaluating numerical value of stuttering level we give double stutter ten times bigger stuttering discrimination value than when single stutter is detected. We assume simple ramp and triangular membership functions defined on a value of relative peak height. The used numerical values were found after numerous experiments. Such a fuzzy classifier works well enough to provide quality data for the control component. We feed control component with long term moving average of stuttering level, so single false alarms or not detected stutters does not play important role in the quality of long term moving average. Moreover the controller is self-tuning so the stuttering level does not have to denote any objective measure, but just should be able to tell if a person is stuttering less or more than some time ago.

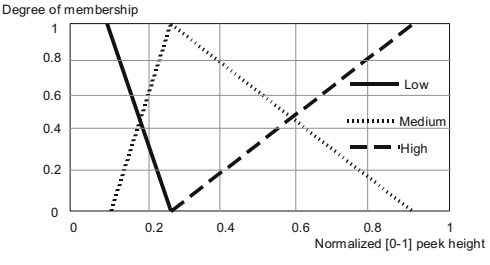


Fig. 3. Peak height fuzzy set

The *Therapy Controller* was implemented as a fuzzy logic controller. As an input value for the controller we provide long term moving average of stuttering measure provided by the Stuttering Detector, the output values are just delay and volume of the echo. A part of Perkins' stuttering therapy program described in [2] was used as a base for automatic therapy control component. The idea how to control single therapeutic session was consulted with logopedist. The general idea is that we should start the session with 250ms delayed echo, and decrease it every time we observe significant progress, by the end of the therapeutic session we should also fade out volume, to give the patients impression that at the end they are speaking without help of the system, and so they can carry on with fluent speech when they walk away from the computer. *Session*

progress is a vital measure determining the how the component should act. The progress is calculated as the difference between current stuttering level and the last checkpoint stuttering level expressed in percent. *Checkpoints* are points in therapeutic session when we consider therapy stage to change. Five basic therapy stages were highlighted, those are: *starting stage*, *early stage*, *middle*, *advanced* and *final stage*. Another measure playing important part in fuzzy rules of the controller is *significance of progress*. This is the relation between change in stuttering level and maximum stuttering level observed in the current session. This measure makes the controller relative changes sensitive, whereas the absolute values of stuttering level are insignificant. Such design was necessary since the absolute stuttering measure provided by stuttering detector might vary among different people even if they stutter more or less the same. Quite simple fuzzy rules describe the way the controller works:

```
if Significance is small or medium then TherapyDelta is zero;
if Significance is large and Progress is negative then TherapyDelta is negative;
if Significance is large and Progress is zero then TherapyDelta is zero;
if Significance is large and Progress is positive then TherapyDelta is positive;
TherapyStage = TherapyStage + TherapyDelta;

if TherapyStage is starting then Delay is very_large;
if TherapyStage is early then Delay is large;
if TherapyStage is middle then Delay is medium;
if TherapyStage is advanced then Delay is small;
if TherapyStage is final then Delay is very_small;
if TherapyStage is starting then Volume is very_large;
if TherapyStage is early then Volume is very_large;
if TherapyStage is middle then Volume is very_large;
if TherapyStage is advanced then Volume is large;
if TherapyStage is final then Volume is small;
```

The changes of controlled variables (*delay* and *volume*) are smoothed by short term moving average to prevent rapid changes that would be hearable to the user. This makes the delay and volume adjustments slow enough so they do not disturb a patient nor grab his attention. Defined the fuzzy controller does not need very high quality input data (the stuttering level) to perform well. This makes the system still effective even if stuttering detector does not detect all the different types of stutters. The *Orator* system gives patients possibilities of everyday echo-based therapeutic sessions what is very imortant for achieving good and stable results.

3 Summary

The developed *Orator* system was proved to be useful in therapy of real patients in Psychological and Pedagogical Clinic No. 1 in Wroclaw. It is difficult to provide objective measures of how the software really influences the therapy process, but a subjective logopedists opinions provide a feeling how the system performs. Bellow an important opinion is quoted: *"Orator application is highly useful in therapy of stuttering children, particularly the fact that a child can use it at home. Everyday work is very important for the therapy results and traditional echo-correction devices are often unavailable for our patients because they are simply too expensive.*

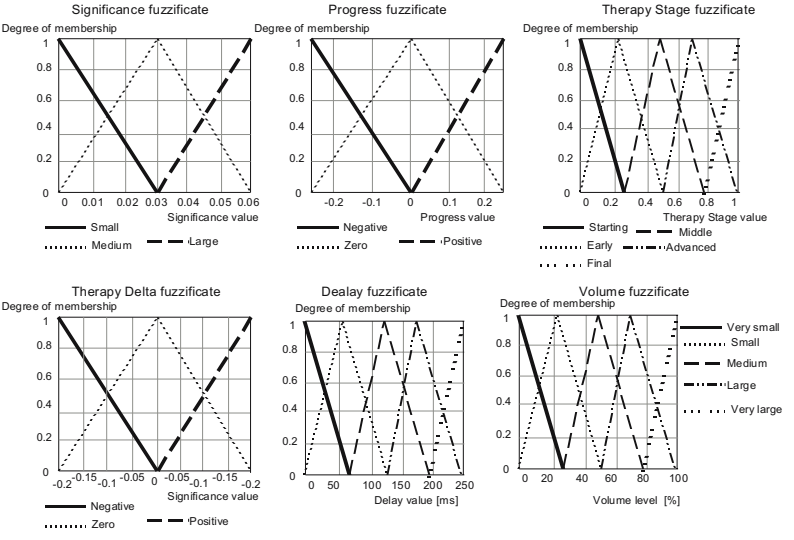


Fig. 4. Fuzzy sets for the therapy controller

The option of automatic echo delay control seems to be interesting and promising. Therapeutic session carried out with it seems to give much better results than one carried just with a traditional device. I recommend the Orator system to every logopedist using delayed auditory feedback in therapy of his patients." (K. Marszewska, Logopedist from Psychological and Pedagogical Clinic).

The stuttering detection component can be improved either by fine tuning the fuzzy sets or introducing some more sophisticated rules into fuzzy stuttering detector, or by using completely different approach.

References

1. VENKATAGIRI H.: Fluency Tutorials. Iowa State University. 2003 http://www.public.iastate.edu/čmdis476/tutorials/fluency_tutorials/definition_of_stuttering.html
2. TARKOWSKI Z.: Suttering (in polish). Wydawnictwo Naukowe PWN. Warszawa 1999
3. GOLDMAN-EISLER F.: Psycholinguistics. London, 1968
4. ROLAND-MIESZKOWSKI M.: DSA (Digital Speech Aid) – a New Device to Decrease or Eliminate Stuttering. 1st World Congress on Fluency Disorders 1994. Accessed in 2003 http://www.digital-recordings.com/publ/pubdsa1.html
5. Delayed Audio Feedback System for Treatment of Stuttering. Luminaud, Inc. http://www.luminaud.com/DAF.htm
6. PICONE J.: Signal Modeling Techniques In Speech Recognition. Texas Instr. Syst. & Inform. Sc. Lab., Japan. Accessed in 2003 http://www.isip.msstate.edu/ publications/journals/ieee_proceedings/1993/signal_modeling/paper_v2.pdf
7. Stuttering Therapy Devices. Casa Futura Techn. http://www.casafuturatech.com/
8. Sheehan Stuttering Center. http://www.stutterssc.org

Using Most Similarity Tree Based Clustering to Select the Top Most Discriminating Genes for Cancer Detection

Xinguo Lu[1], Yaping Lin[2,1], Xiaolin Yang[1], Lijun Cai[1], Haijun Wang[1], and Gustaph Sanga[1]

[1] College of Computer and Communication, Hunan University, Changsha, 410082, China
hnluxinguo@hotmail.com
[2] College of Software, Hunan University, Changsha, 410082,China
yplin@hnu.cn

Abstract. The development of DNA array technology makes it feasible to cancer detection with DNA array expression data. However, the research is usually plagued with the problem of "curse of dimensionality", and the capability of discrimination is weakened seriously by the noise and the redundancy that are abundant in these datasets. This paper proposes a hybrid gene selection method for cancer detection based on clustering of most similarity tree (CMST). By this method, a number of non-redundant clusters and the most discriminating gene from each cluster can be acquired. These discriminating genes are then used for training of a perceptron that produces a very efficient classification. In CMST, the Gap statistic is used to determine the optimal similarity measure λ and the number of clusters. And a gene selection method with optimal self-adaptive CMST(OS-CMST) for cancer detection is presented. The experiments show that the gene pattern pre-processing based on CMST not only reduces the dimensionality of the attributes significantly but also improves the classification rate effectively in cancer detection. And the selection scheme based on OS-CMST can acquire the top most discriminating genes.

1 Introduction

The development of DNA array makes it feasible to cancer detection with DNA array expression data. In cancer detection, the research is usually plagued with large number of gene variables versus the small number of records of experiment, which is known as the "curse of dimensionality". The gene patterns selection and the determination for the number of most discriminating patterns are very important issues to cancer detection.

Feature selection is the process of choosing the most appropriate features when creating the model of the process [1]. In cancer detection, many feature selection methods were used to obtain the appropriate gene patterns. In Veer's approach [2], the genes were correlated against the disease outcome and then ranked; and a set of 70 genes was then identified. In Shipp's approach [3], signal

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 931–940, 2006.
© Springer-Verlag Berlin Heidelberg 2006

noise ratio was used to identify 30 genes. Cho et al. evaluated the performances of feature selection methods systematically[4]. These methods reduced the noise but co-expressing redundancy among genes as they were applied across the entire data set [5, 6]. The gene patterns selection methods based on clustering [7, 8, 9] were proposed to deal with this issue. Mateos et al. proposed a way of reducing the dimensionality of the dataset by pre-clustering gene expression patterns [8]. Then, the average expression values of the clusters of co-expression genes were used with a supervised neural network for classifying the experiment conditions. In Conde's approach [9], a hierarchical based self Organizing Maps (SOM), the Self Organizing Tree Algorithm (SOTA), was used to group gene expression profiles into a number of clusters, and the average values of these clusters were used for training of a perceptron to classify the gene expression profiles of the conditions. However, these schemes are sensitive to the noise and outlier.

In this paper, the approach of gene selection based on clustering of most similarity tree(CMST)[10] is proposed to cancer detection. Firstly, CMST is used and many clusters, in which the redundancy is minimized from each other, are produced. Then the most informative and discriminating gene is selected from each cluster with feature selection method as used in the entire data set, which removes the noise and outlier in the cluster. This proposed scheme can efficiently eliminate the disturbance of redundancy and noise. Hence it improves the quality of representative genes in the classification of experiment conditions with gene expression data. These discriminating genes are used to train a perceptron that produces a very efficient classification.

How to determine the number of discriminating gene patterns is also a topic in the field of cancer detection. In the proposed schemes, the number of gene patterns for cancer detection is pre-fixed arbitrarily. Different from these schemes, in this paper the Gap statistic [11] with similarity measure is used to determine the most optimal similarity measure in CMST, and the gene selection based on the optimal self-adaptive CMST (OS-CMST) is proposed. The experiment shows that this scheme can acquire the top most discriminating genes without inputting parameters.

The paper is organized as follows. Section II describes the related methodologies on gene pattern pre-processing in gene classification and cancer detection. Section III presents the gene clustering of most similarity tree(CMST), the gene clustering of optimal self-adaptive CMST(OS-CMST) and the hybrid method of gene selection based on CMST/OS-CMST. In Section IV, the cancer detection scheme is described , and thorough experimental results with the Alizadeh's dataset [12] are presented.

2 Related Methodologies

Information gain (IG) is a measurement to the correlation between the viable and the classes. According to the IG, the genes are ranked and selected. It is described using following formula:

$$IG(g_i, c_j) = P(g_i, c_j) log \frac{P(g_i, c_j)}{P(g_i)P(c_j)} + P(\overline{g}_i, c_j) log \frac{P(\overline{g}_i, c_j)}{P(\overline{g}_i)P(c_j)} \quad (1)$$

where $P(c_j)$ is the number of features in class c_j, $P(g_i)$ and $P(\overline{g}_i)$ are the numbers of features when are induced and regressed respectively.$P(g_i, c_j)$ is the number of cases when the g_i is induced in class c_j , and $P(\overline{g}_i, c_j)$is the number of cases when the g_i is regressed in class c_j.

Signal-to-noise ratio (SNR)is a calculated ranking number for each variable to define how well this variable discriminates different classes. In gene selection the following formula is used:

$$SNR(g_i, c_j) = \frac{\mu(g_i, c_j) - \mu(g_i, \overline{c}_j)}{\sigma(g_i, c_j) - \sigma(g_i, \overline{c}_j)} \quad (2)$$

where $\mu(g_i, c_j)$ and $\mu(g_i, \overline{c}_j)$ are the mean values of g_i for the samples in c_j and not in c_j respectively and $\sigma(g_i, c_j)$ and $\sigma(g_i, \overline{c}_j)$ are the corresponding standard deviations.

The Self Organizing Tree Algorithm (SOTA) is a hierarchical version of Self Organizing Maps(SOM). It is a divisive method and constructs a hierarchical relationships among the entities(here the gene expression profiles in the dataset). The genes clustering resolution is obtained according to the level of the tree constructed by SOTA. In Ref.[9],to reduce the dimensionality the average values of these clusters are used for the training of a perceptron to classify the conditions in DNA microarray gene expression data.

3 Gene Selection Based on CMST/OS-CMST

In this section a hybrid method for gene selection is introduced to cancer detection. Firstly, the clustering method of CMST/OS-CMST is used to partition the genes into some clusters. Secondly, in each cluster the genes are ranked using Information gain(IG)/Signal-to-noise ratio (SNR), and then the representative gene is selected. It can eliminate the affection of redundancy and noise efficiently and improve the accuracy of cancer detection.

3.1 Gene Clustering of CMST/OS-CMST

The algorithm of CMST is based on the idea of equivalence relation and equivalence classes in set theory and graph theory. There is an equivalence relation R to objects g_i,g_j in cluster C,

$$R = < g_i, g_j > | g_i, g_j \in C. \quad (3)$$

If R is an equivalence relation to cluster C, below characters will be satisfied.

$$< g_i, g_i > \in R, if g_i \in C. \quad (4)$$

$$< g_j, g_i > \in R, if < g_i, g_j > \in R \wedge g_i, g_j \in C. \quad (5)$$

$$< g_i, g_k > \in R, if < g_i, g_j > \in R$$
$$\wedge < g_j, g_k > \in R \wedge g_i, g_j, g_k \in C. \tag{6}$$

According to the equivalence relation R, the equivalence classes $[g_i]R$, where g_i is the representative object in its classe, can be obtained from the objects domain G. All objects in G are partitioned effectively by these classes. The most similarity tree is an extensive method of equivalence relation theory combined with graph theory.

Let $G = \{g_1, g_2, ..., g_n\}$ be a set of genes, and a member of the set $g_i = \{g_{i1}, g_{i2}, ..., g_{im}\}$,where g_{ij} is the jth character of the ith gene in the gene expression data. The similarity measure between gene g_i and g_j is defined as $SM(g_i, g_j) = s_{ij}$,

$$s_{ij} = \frac{\sum_{k=1}^{m} |g_{ik} - \overline{g}_i||g_{jk} - \overline{g}_j|}{\sqrt{\sum_{k=1}^{m} (g_{ik} - \overline{g}_i)^2 \cdot \sum_{k=1}^{m} (g_{jk} - \overline{g}_j)^2}} \tag{7}$$

where $\overline{g}_i = \frac{1}{m} \sum_{k=1}^{m} g_{ik}$,$\overline{g}_j = \frac{1}{m} \sum_{k=1}^{m} g_{jk}$. And the similarity measurement matrix, *SMM*, is built and $SM(g_i, g_j)$ is its ijth element.

A most similarity tree T<v,e>is a weighted graph built with *SMM*, the weight of an edge in this tree is the similarity of the two connecting neighbor vertexes. There exist a path between two vertexes in the tree and the path weight,$PW(g_i, g_j)$, is the least weight of edge via this path.

In gene domain G, the relation $R(\lambda, G)$ of similarity measure is defined as:

$$R(\lambda, G) = < g_i, g_j > |PW(g_i, g_j) > \lambda \wedge g_i, g_j \in G. \tag{8}$$

If the relation $R(\lambda, G)$ satisfies the three characters above, it is an equivalence relation. A partition of G with relation $R(\lambda, G)$ groups genes into clusters. In each cluster, the similarity measure of pair of genes is greater than λ.

A simple method for clustering is to remove the edge in which its weight is less than the customized threshold λ. In each cluster, the relation of R is the equivalence relation with similarity measure of λ.

Suppose the gene data set G, according to similarity measure λ, has been clustered into k clusters $C_1, C_2, ..., C_k$, with C_r denoting the indices of observations in rth cluster, and $n_r = |C_r|$.

Let WS_r be within similarity of cluster $|C_r|$.

$$WS_r = \sum_{g_i, g_j \in C_r \wedge g_i \neq g_j} SM^2(g_i, g_j), \tag{9}$$

Where WS_r is the sum of pairwise squared similarity measures for all genes in rth cluster. And $AS(\lambda)$ is set as follows.

$$AS(\lambda) = \sum_{i=1}^{k} \frac{1}{n_i} WS_i. \tag{10}$$

The idea of our approach is to standardize the graph of $log(AS(\lambda))$ by comparing it to its expectation under an appropriate null reference distribution. Hence we defined that

$$Gap(\lambda) = E^*(log(AS(\lambda))) - log(AS(\lambda)). \tag{11}$$

where $E^*(log(AS(\lambda)))$ is estimated by an average of B copies $log(AS(\lambda)^*)$, each of which is computed from a Monte Carlo sample $g_1^*, g_2^*, ..., g_n^*$ drawn from the reference distribution. For convenience, the reference distribution used here is generating each reference feature uniformly in the range of the observed values for that feature.

$$E^*(log(AS(\lambda))) = \frac{1}{B}\sum_b log(AS(\lambda)_b^*) \tag{12}$$

$$sd(\lambda) = \sqrt{\frac{1}{B}\sum_b \Big(log\big(AS(\lambda)_b^*\big) - \frac{1}{B}\sum_b log\big(AS(\lambda)_b^*\big)\Big)^2} \tag{13}$$

$$s(\lambda) = sd(\lambda)\sqrt{1+\frac{1}{B}} \tag{14}$$

where $sd(\lambda)$ denotes the standard deviation of the B Monte Carlo replicates $log(AS(\lambda)^*)$, account for the simulation error in $E^*(log(AS(\lambda)))$. Using this,the similarity measure is chosen to be the greatest one that makes Equ.(15) satisfied.

$$Gap(\lambda_i) \geq Gap(\lambda_{i+1}) - s(\lambda_{i+1}). \tag{15}$$

3.2 Gene Selection Based on CMST/OS-CMST

Gene selection based on CMST/OS-CMST is a hybrid method to select the top most discriminating genes for cancer detection. Firstly, the clustering method of CMST/OS-CMST partitions the genes into some clusters, in which there is little redundancy between the genes from different clusters. Secondly, in each of the most similarity clusters the genes are ranked using IG/SNR, and the representative genes are selected. These genes have the character of discriminating the class of samples in the corresponding cluster furthest, which eliminate the affection of noise and outlier efficiently. Then the genes selected as the two steps above are used to train the classification model. The hybrid method of gene selection combines the strongpoints of different data processing methods at muti-steps, so it has merits with least redundancy and non-sensitive to outlier and noise.

The algorithm of gene selection based on CMST/OS-CMST is implemented as follows:

1) Construct the similarity measurement matrix SMM according to the similarity measure of genes, $SM(g_i, g_j) = s_{ij}$.
2) Depict all the genes g_i,$g_i \in G$.
3) Connect the edge of the genes according to the rank of the similarity measure, the two genes with greatest similarity measure are selected and connected with a weighted edge, $weight(g_i, g_j) = SM(g_i, g_j)$, if it is guaranteed that there is not a loop, a most similarity tree is built, and the number of edges of the tree is $|T|$.

4) If CMST is used, remove the edge whose weight is less than the inputting threshold λ and go to step 13.
5) Rank the $|T|$ similarity measures λ, the weight of edge, in the tree by sort descending.
6) For each similarity measure $\lambda_i, 1 \leq i \leq |T|$.
7) Cluster genes in the tree T giving within similarity measure $AS(\lambda_i)$.
8) Generate B reference data sets and build B most similarity trees.
9) Cluster genes in each of the B most similarity trees according to similarity measure λ_i giving within similarity measures $AS(\lambda_i)_b, 1 \leq b \leq B$.
10) Compute the Gap statistic $Gap(\lambda) = E^*(log(AS(\lambda))) - log(AS(\lambda))$ and $s(\lambda_i)$.
11) Until $Gap(\lambda_i) \geq Gap(\lambda_{i+1}) - s(\lambda_{i+1})$, else go to step 4.
12) λ_i is selected as the most optimal similarity measure threshold, and the genes is clustered using CMST with similarity measure λ_i .
13) IG/SNR is used to these clusters and the top most discriminating genes are selected for cancer detection.

4 Experiments

4.1 Cancer Detection with Gene Selection Based on CMST

We applied this hybrid gene selection method of CMST and IG/SNR for classifying cell lines on the Alizadeh's dataset [12], which includes nine different types of cell lines. In the experiment we compared the classification results to that of using pre-processing of SOTA[9], IG and SNR via entire genes.

This experiment composes of two parts. In the first part, SOTA,IG and the hybrid method of CMST and IG are used to select the gene patterns respectively. The numbers of gene patterns with SOTA and IG are customized to 15,40 and 75 respectively.In the second part,SNR is used instead of IG. The numbers of gene patterns with SOTA and SNR are customized to 12,35 and 72 respectively. In both parts, the number of clusters by CMST is chosen to be the closest one to the corresponding number of gene patterns produced by other methods. Then these patterns are used for the training of a perceptron with a single input layer with k (the number of gene patterns) input nodes and nine nodes in the output layer, corresponding to the nine types of cell lines of Alizadeh's data. The weight-update rule is used as follows:

$$\Delta w_{ij} = \eta * (D - Y) * x_i \tag{16}$$

where Δw_{ij} is the weight update from node i to unit j,η is the leaning rate and is set to 0.5, D is the desired output, and Y is the actual output,x_i is the input from node i. The classification results are tested through the leave one out method for validation and shown in table1 and table2.

From table1 and table2, we can see that the classification accuracies with gene selection of our hybrid method are better than with the pre-processing by SOTA, IG and SNR at different numbers of gene patterns. The reasons are as

Table 1. Classification results with different gene patterns selection methods(SOTA, IG,CMST)

Cell line	Total	SOTA (15)	IG (15)	CMST (14)	SOTA (40)	IG (40)	CMST (38)	SOTA (75)	IG (75)	CMST (76)
No.1	46	41	39	43	42	40	45	42	40	43
No.2	2	1	1	2	2	1	2	2	1	2
No.3	2	1	1	1	1	1	2	1	1	1
No.4	10	8	6	8	9	8	10	8	6	9
No.5	6	5	5	6	4	5	6	5	4	6
No.6	6	5	5	5	5	4	5	4	5	5
No.7	9	8	7	8	9	8	8	8	6	8
No.8	4	3	3	3	3	3	3	3	2	4
No.9	11	9	8	10	10	8	9	9	7	9
Total	96	81	75	86	85	78	90	82	72	87

Table 2. Classification results with different gene patterns selection methods(SOTA, SNR,CMST)

Cell line	Total	SOTA (12)	SNR (12)	CMST (11)	SOTA (35)	SNR (35)	CMST (36)	SOTA (72)	SNR (72)	CMST (70)
No.1	46	40	37	42	41	39	45	41	38	44
No.2	2	1	1	2	2	1	2	1	2	2
No.3	2	2	1	1	1	2	2	1	1	1
No.4	10	8	6	9	9	7	9	7	8	8
No.5	6	4	6	5	5	5	6	5	4	5
No.6	6	4	4	5	5	4	5	5	4	5
No.7	9	8	7	8	9	7	8	8	4	7
No.8	4	3	3	4	3	4	4	2	2	4
No.9	11	9	7	9	8	8	10	10	8	10
Total	96	79	72	85	83	77	91	80	71	86

follows:firstly, there is a lot of redundant information in the gene dataset which composes of 4026 genes. The redundancy is included in the selected genes by IG and SNR via whole dataset and some discriminating genes are missed. Secondly, the noise and outlier are abundant in the gene expression dataset. The gene patterns pre-processed by SOTA for classification are disturbed severely and the classification accuracy is weakened. However, Our approach combines the strongpoints of different data processing methods at different steps. It eliminates redundancy and removes noise simultaneously, so the accuracy of cancer detection is improved.

4.2 Cancer Detection with Gene Selection Based on OS-CMST

In cancer detection, the accuracy doesn't increase as the number of gene patterns increases, which are shown in table1 and table2. When the number of clusters is a magnitude between 38 and 55, it achieves the peak value. In all the proposed gene selection methods, the number of gene patterns for cancer detection is pre-fixed by the researches arbitrarily. In this experiment, OS-CMST is used to produce the global optimal clusters, with minimized similarity and least redundant information from each other. From Fig.1, the most appropriate number of clusters is 46, which is estimated by the Gap statistic. In Ref.[9], one appropriate number of clusters is 44 via method of exhaustion with the pre-processing by SOTA .

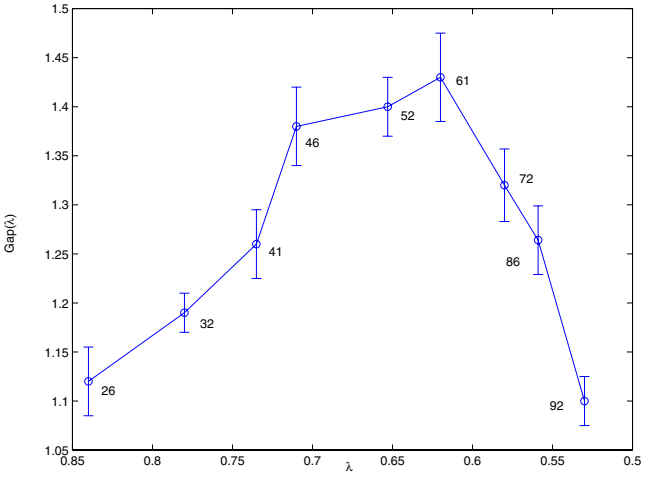


Fig. 1. Gap statistic with similarity measure threshold λ on clusters using CMST, the numeric near the label circle is corresponding number of clusters to this threshold

Then IG/SNR is used to select the most representative gene from each cluster for the training of the perceptron with a single input layer with 46 input nodes, which is 44 by SOTA, and nine nodes in the output layer. The results of classification with the most discriminating gene patterns, which pre-processed by SOTA, IG/SNR via the entire genes and our hybrid method respectively, are in table 3. The expression in the second row is the same as the last row in table 1. Table 3 shows that the classification based on the hybrid gene selection of OS-CMST and IG gains the exciting results and the efficiency is 94 out of 96, which is better than the accuracy by pre-processing with SOTA, while using IG via the entire genes is 86 out of 96. And using SNR the accuracy is only 82 out of 96, while combined with OS-CMST the accuracy advances to 92 out of 96.

Table 3. Classification of cell lines with gene patterns pre-processed by SNR,IG, SOTA and the hybrid method respectively

Total	SNR	IG	SOTA	OS-CMST(SNR)	OS-CMST(IG)
96	82	86	91	92	94

5 Conclusion

In this paper we present a approach of gene selection based on CMST/OS-CMST for cancer detection. This hybrid approach combines the strongpoints of different data processing methods.The experiment results indicate that this scheme can eliminate the redundancy and noise efficiently,acquire the top most discriminating genes and generate the better cancer detection accuracy. Also the OS-CMST based gene selection approach can obtain the most optimal gene patterns for cancer detection without parameters.

References

1. N. Kasabov,"Evolving Connectionist Systems, Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines",Verlag Springer,2002.
2. L.J.V.T. Veer, H. Dai,, M.J.V.D. Vijver, Y.D. He, and et al., "Gene expression profiling predicts clinical outcome of breast cancer", Nature,415,530-536,2002.
3. M.A. Shipp, K.N. Ross, P. Tamayo, A.P. Weng, J.L. Kutok, R.C.T. Aguiar , M. Gaasenbeek, and et al.,"Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning", Nature Medicine,8,68-74,2002.
4. S.B. Cho, H.H. Won,"Machine Learning in DNA Microarray Analysis for Cancer Classification". In Proc. of the First Asia-Pacific Bioinformatics Conference(APBC 2003), 189-198, 2003.
5. L. Goh, Q. Song,and N. Kasabov,"A Novel Feature Selection Method to Improve Classification of Gene Expression Data" . In Proc. of Bioinformatics 2004 Second Asia-Pacific Bioinformatics Conference (APBC 2004), 161-166, 2004.
6. X. Hu and I. Yoo,"Cluster Ensemble and its application in gene expression analysis". the 2nd Asia-Pacific Bioinformatics Conference(APBC2004),Dunedin,New Zealand.Conferences in Reseach and Practice in Information Technology, Vol.29, 2004.
7. A.V. Lukashin, and R. Fuchs, "Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters", Bioinformatics, 17, 405-414, 2001.
8. A .Mateos, J. Herrero, J. Tamames, J. Dopazo, "Supervised neural networks for clustering conditions in DNA array data after reducing noise by clustering gene expression profiles",In Microarray data analysis II.Kjuwer Academic,2002.
9. L. Conde, A. Mateos, J. Herrero, and J. Dopazo, "Unsupervised Reduction of the Dimensionality Followed by Supervised Learning with a Perceptron Improves the Classification of Conditions in DNA Microarray Gene Expression Data", Neural Networks for Signal Processing XII. IEEE Press (New York). Eds. Boulard, Adali, Bengio, Larsen, Douglas. pp.77-86,2002.

10. X.G. Lu, Y.P. Lin, X.L. Li, Y.Q. Yi, L.J. Cai, H.J. Wang, "Gene Cluster Algorithm Based on Most Similarity Tree",In Proc. of the 8th International Conference on High Performance Computing in Asia Pacific Region(HPC Asia2005),652-656,2005.
11. T. Hastie, T. Tibshirani and G. Walther,"Estimating the number of clusters in a dataset via the gap statistic". Tech. report. March 2000. Published in JRSSB 2000.
12. A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma,and et al.,"Distinct types of disuse lare B-cell lymphoma identified by gene expression profiling".Nature, vol 403, pp.503-511, 2000.

Nonambiguous Concept Mapping in Medical Domain

Paweł Matykiewicz[1,2], Włodzisław Duch[1,3], and John Pestian[2]

[1] Department of Informatics, Nicolaus Copernicus University, Toruń, Poland
[2] Dept. of Biomedical Informatics, Cincinnati Children's Hospital Medical Center, OH, USA
[3] School of Computer Engineering, Nanyang Technological University, Singapore
pawelm@phys.uni.torun.pl, Google: Duch, john.pestian@cchmc.org

Abstract. Automatic annotation of medical texts for various natural language processing tasks is a very important goal that is still far from being accomplished. Semantic annotation of a free text is one of the necessary steps in this process. Disambiguation is frequently attempted using either rule-based or statistical approaches to semantical analysis. A neurocognitive approach for a nonambiguous concept mapping is proposed here. Concepts are taken from the Unified Medical Language System (UMLS) collection of ontologies. An active part of the whole semantic memory based on these concepts forms a graph of consistent concepts (GCC). The text is analyzed by spreading activation in the network that consist of GCC and related concepts in the semantic network. A scoring function is used for choosing the meaning of the concepts that fit in the best way to the current interpretation of the text. ULMS knowledge sources are not sufficient to fully characterize concepts and their relations. Annotated texts are used to learn new relations useful for disambiguation of word meanings.

1 Introduction

The Unified Medical Language System (UMLS) is a collection of 88 medical knowledge sources. The most recent edition of UMLS (2005AB ed.) contains 1 196 265 unique concepts, each labeled by a Concept Unique Identifier (*CUI*), and 2 873 310 unique phrases (*SUI*) [1]. Annotation of texts requires mapping of noun phrases, words, abbreviations and acronyms discovered in the unstructured text to the unique UMLS concepts. The *MetaMap* software (MMTx) [2] is frequently used to discover UMLS concepts in texts. The software has been developed by experts in the *U.S. National Library of Medicine*, a part of the *National Institute of Health*. The *MetaMap* algorithm is rather slow and quite complicated. It is aimed at discovering all possible terms in the text without carrying much about ambiguity of the output. As a result some words are given many annotations, listing all possible meanings and various phrases they appear in, making the semantic search even more difficult than with the raw text.

The goal of our research is to overcome these drawbacks and create fast, precise and unambiguous concept mappings. There are many statistical, pattern

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 941–950, 2006.
© Springer-Verlag Berlin Heidelberg 2006

recognition and syntactical approaches to the general word sense disambiguation (WSD) problem [3], but most experiments have been conducted on a small scale, while the number of medical concepts that need to be taken into account exceeds one million. Moreover, although some word meanings are easily distinguishable other are quite difficult to capture and even human annotators agree only in no more than 80% [4]. Despite that fact experts have no problem with understanding medical or technical texts. The only system capable of language understanding at the human competence level is the human brain, and it should be the source of inspirations for development of semantic annotation systems.

General philosophy of our neurocognitive approach to natural language processing (NLP) is presented in the next section. To approximate formation of primed semantic subnetwork providing interpretation of the text graphs of consistent concepts (GCCs) are constructed. The concept mapping algorithm, presented in the third section, is based on this approach, although many other variants and applications are possible [5,6,7]. The algorithm for phrase sense disambiguation that adds new relations and determines relations strength between concepts with the use of prior knowledge and the acquisition of new knowledge are also discussed in this section. The last section contains conclusions and future plans.

2 Neurocognitive Approach to NLP

Analysis of texts, independent of the purpose, requires three main steps:

- recognition of tokens, or mapping from strings of letters to unique terms;
- resolving ambiguities, grouping terms into phrases and mapping them to concepts;
- semantic representation of the whole text capturing relations among entities that are involved, facilitating inferences, and thus understanding and answering questions about its content.

These three steps roughly correspond to the function of three kinds of human memory [8]: recognition memory, semantic memory and episodic memory. NLP research usually ignores this fact, focusing on formal approaches (grammar, logics, statistical correlations). Neurocognitive approach to NLP follows inspirations from brain science focusing on approximated models of memory and other neural processes. The long-term goal is to reach human-level competence in natural language processing.

Recognition memory helps to ignore most spelling errors. As long as the first and the last letter of the word is not changed even severely distorted texts containing *wrods wtih many paris of letres trasnpoesd* is read without much troubles, a phenomenon that is of interest to spammers and cognitive scientists. It is rather obvious that context and anticipation plays a major role in correct recognition. Although we do not consider problems at the recognition level here unstructured medical texts need a lot of data cleaning. Lexical Systems Group of the US National Library of Medicine has developed a spelling suggestion tool Gspell and

the SPECIALIST lexicon containing many biomedical terms and general English words, that Gspell is using to provide suggestions. Without the use of context and understanding the topic of the text Gspell makes many spelling suggestions, although humans recognize a single term, frequently paying no attention to the misspellings. It is clear that recognition memory cannot be separated from other memory systems, doing much more than just searching for similar terms in the lexicon. Reading text leads to priming effects: expectation and anticipation of a few selected words, and inhibition of many others that do not come to our mind.

The *semantic priming* (SP) phenomenon has been known in cognitive psychology since more than 30 years (see the review in [9]). Each word excites brain subnetworks that encode different meanings of that word [10]. In such coding identical phonological representations of words may be shared among several concepts without leading to any problems. Words that have been processed earlier (context) have already activated many brain subnetworks, increasing the probability of a particular meaning of the new concept, and inhibiting all other meanings. This competition, leading to inhibition of subnetworks coding alternative meanings of the word, makes it hard to think about alternatives when one of the meanings (interpretations) fits really well to the current context. Statistical language processing models applied to a large text corpus used for training allow for prediction of the next word in a sequence with high reliability [11], partially capturing this anticipation, although statistical algorithms do not approximate well real brain processes behind this phenomenon. Anticipation may help to disambiguate word senses, facilitating the mapping of terms into concepts.

Semantic memory encodes in the activity of brain's subnetworks information about objects and concepts, together with their properties and relations. Formal models of semantic networks, computational structures inspired by psychological ideas about semantic memory, are known since more than 30 years [8,12,13,14]. Semantic networks are used in artificial intelligence as knowledge representation tools [15,16] and may provide a model to approximate functions of biological semantic memory (SM). Each node in semantic network is in fact a subnetwork, with similarities and associations between concepts resulting in sharing some parts of the subnetworks. Activations of semantic subnetworks are responsible for semantic priming, building an episode that may be memorized and retrieved later, reinstating a particular configuration of brain activities, or an episode.

Episodic memory is based on semantic relations of concepts found in the analyzed text, understanding or rough categorization of the text topic, and binding different entities in a specific way for this particular text. Our assumption here is that a simple model of episodic memory may be provided by priming the semantic network during text reading, forming an active subnetwork of main concepts found in the text, their mutual relations and their relations to concepts forming background knowledge that has not been explicitly mentioned. One way to achieve it is to scan the text to find main, unambiguous concepts that form the skeleton of the active subnetwork, and add other concepts selecting the meaning that increase overall consistency. In this way graphs of consistent concepts

(GCCs) may be formed, capturing the meaning of the text and facilitating its unambiguous annotation. Some relations are not defined directly at the level of relations between individual concepts, but at the higher ontological level (this is probably done by the non-dominant hemisphere, where representations of abstract concepts may be stored [17]. UMLS contains *Semantic Network*, but it has only 132 highest level broad subject categories (Semantic Types).

An alternative to the network model is provided by vector space models, for example the High Dimensional Semantic Space memory model [18]. An ambiguous word, for example cold, corresponding to several different concepts (UMLS Methathesaurus has 6 senses of cold) is represented by 6 different vectors. Each vector component measures statistical co-occurrences of each particular word sense to all other lexicon terms, defining how likely it is that this term will appear in the context window of a given word. To select the correct sense the context window is used to find how similar is a given vector representation to the current context. This approach has been used with some success in general word sense disambiguation tasks. Vector models may be understood as an approximation to the activation of network nodes, and the search for consistent concepts that is used in the GCC algorithm may be formulated as the search for vectors that create smallest volume.

Memory-based process are rarely acknowledged in natural language processing research. Semantic Knowledge Representation (SKR) project at the National Library of Medicine (NLM) has a very ambitious goal [19], although it is only loosely inspired by psychological ideas about semantic memory, rather then being a model of semantic memory as implemented by the brain. Yet it is obvious that without recognition, semantic and episodic memory understanding texts would not be possible. Each concept has numerous properties and relations that are encoded in the structure of subnetworks that encode it in the brain [10]. Our goal should be to approximate some of these processes. This leads to the extension of the idea of semantic networks [12,13,15,16], providing a model to approximate functions of biological semantic memory (SM). Concepts should be represented by distributed subnetworks that contain phonological representations and by semantic extensions of these representations, linking to all the properties of a given concept and to all concepts that may be associated in some way with them. Two main processes in such networks are spreading activation and competition. Competitive processes should not be considered as the "winner-takes-all" only, as there are many winners and the activity of the whole subnetwork providing consistent interpretation of the text being analyzed is growing.

The challenge is to collect sufficient knowledge about concepts and their relations that allows humans to understand language and to interpret texts. The largest collection of ontologies combined by specialists is available in medical domain. UMLS ontologies have hierarchical structures and thus do not provide strong concept descriptions. An expert knows much more about basic medical concepts than can be found in the UMLS. Thus UMLS may serve only as a poor approximation to the real semantic memory. Relations contained in the UMLS

may be represented as excitatory connections between concepts. UMLS (2005AB edition) contains 4 435 387 unique relations but it is not a comprehensive medical encyclopedia, so a lot of relations are missing. It is not clear how to score most relations because only co-occurrence relations in UMLS are numerical and all others are logical. Lack of knowledge about concept properties and relations is the major obstacle to annotate in an unambiguous way clinical texts.

To approximate formation of primed semantic subnetwork providing interpretation of the text, graphs of consistent concepts (GCCs) are constructed. An algorithm for adding new relations and determining relations strength between concepts with the use of prior knowledge is described below

3 Concept Mapping Algorithm

UMLS includes three main modules used in our approach: GSPELL, a tool for spelling correction; and two UMLS Knowledge Sources: *Specialist Lexicon*, a general lexicon that includes both common English words and biomedical vocabulary; *Methathesaurus*, describing biomedical and health-related concepts, a very large, multi-lingual and multi-purpose vocabulary. Overall the whole UMLS installation needs 26 GB of storage space. The GCC algorithm uses only part of UMLS. For normalizing and varying terms following files from the SPECIALIST LEXICON are used: `DM.DB, SM.DB, LRABR, LRAGR, LRNOM, LRSPL`. For binding concepts with phrases and sources the following file are used: `MRCONSO.RRF, MRXNS_ENG.RRF`.

In order to map a noun and verb phrases to concepts *TreeTagger* software [20] is used to annotate text with parts of speech (POS). Every word (EUI) in a text is mapped to its normalized form (WUI) - a singular noun. Unique string identifiers (SUI) are composed in turn from normalized words (WUI). Every phrase (SUI) has on average 2.4 different CUIs associated with it. The following schema for mapping is used:

$$EUI \mapsto WUI \mapsto SUI \mapsto CUI$$

Figure 1 presents a simplified schema for mapping a text to concepts. Words in that example are already normalized.

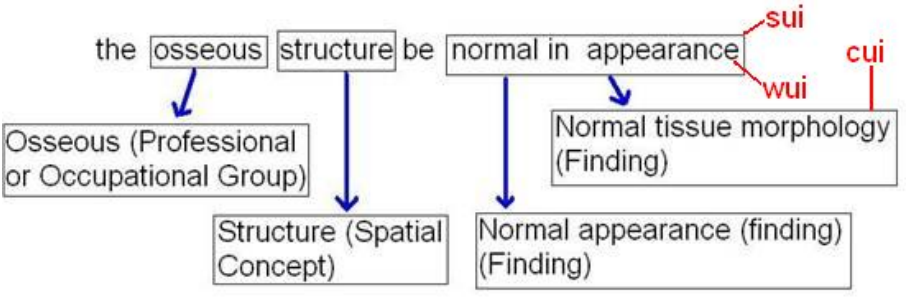


Fig. 1. A simplified schema for mapping normalized words to concepts

In order to map phrases (SUI) to concepts (CUI) following algorithm was used:

```
1. Assign part of speech tags to every token.
2. Map all the words to their normalized forms.
3. Scan normalized words from the end of the text.
  3a. If a POS tag matches one of the symbols:
    CD,RB,JJ,N,VV,LS,SYM
    start scanning the text from the current position
    towards beginning of the text,
    add words to a phrase that match mentioned POS tags
    until there is a phrase that is not in the UMLS.
  3b. Resume after position where last UMLS
  phrase was found.
4. Finish when at the beginning of the text.
```

This is a very fast and simple mapping algorithm. The following text from ultrasonography dictation has been mapped to a concept space:

```
Fever, left flank pain, pyelonephritis. The right kidney is
normal in sonographic appearance with no evidence of scarring,
hydronephrosis or calculi. It measures XXXX cm which is normal
for patient's age. The left kidney is enlarged. It measures xx
cm in length. No focal areas of abnormal echogenicity or scarring
are seen. No hydronephrosis or calculi are identified. Images of
the bladder demonstrate no abnormality. Enlargement of the left
kidney which may suggest acute pyelonephritis. This could also
represent a normal variant. Normal appearing right kidney.
```

The algorithm found 30 concepts in this text, 11 of which are ambiguous. Next step is to find which $SUI \mapsto CUI$, or phrases map to concepts. In order to disambiguate SUI phrases relational table from UMLS (`MRREL.RRF` file) is used. This file contains 5 499 792 unique relations, with the same relations found in many different ontologies. Some relations are rather peculiar and appear only in one specialized ontology, while important relations are found in many ontologies. A weight matrix for all relations is constructed. Following definitions are used: $N(CUI_i)$ – number of occurrence of a CUI_i concept in the relational table, $C(CUI_i, CUI_j)$ – number of co-occurrences of CUI_i and CUI_j concepts in the relational table row, $W = \{w_{ij}\}$ – matrix storing weights between ith and jth concept. The weights are defined as conditional probabilities:

$$w_{ij} = P(j|i) = \frac{C(CUI_i, CUI_j)}{N(CUI_i)} \tag{1}$$

Once a text is mapped to a set of ambiguous concepts a graph of consistent concepts is created, with nodes corresponding to concepts and edges to relations. Each node corresponding to the concept found in the text has an initial activity $a_i(t = 0)$ that spreads to other nodes according to W matrix:

$$a_i(t+1) = \alpha a_i(t) + \sum_j w_{ij} H(a_j(t)) \tag{2}$$

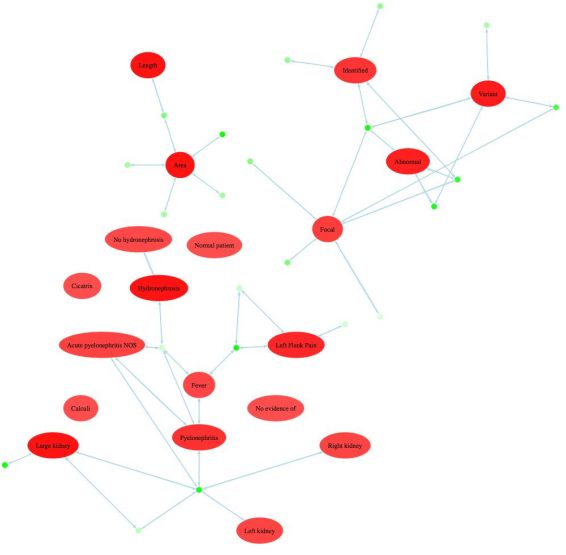


Fig. 2. Example of a graph of consistent concepts after 4 iteration of spreading activations

where H is the Heaviside step function and α is a spontaneous decay parameter. Similar function was considered in [21]. The main problem for spreading activation in networks without inhibition is to prevent the infinite growth of all node activities. α decays should be sufficiently large to achieve this; all experiments in this paper are with $\alpha = 0.73$.

Propagation of activations in the semantic network should lead to the decay of concepts that are not supported by other active concepts. After a few iterations only most consistent concepts forming the GCC graph should have activations above certain threshold. These concepts should give the right sense of a phrase (SUI). Figure 2 shows an example of the GCC graph after 4 iterations.

The initial weights created from UMLS relations help to disambiguate phrases only in a limited way. The UMLS is a big but quite general knowledge base, and it frequently lacks more specific knowledge. Enriching UMLS relations means simply adding $N(CUI_i)$ and $C(CUI_i, CUI_j)$ for all pairs of concepts from an annotated text. For pilot purposes two small radiology corpuses were created. For knowledge acquisition Cincinnati Children's Hospital Medical Center clinical texts from radiology were used. These texts are dictated by physicians and changed into a text form by medical voice recognition software. 60 of those documents where chosen for manual annotation and divided in two parts. Every set of documents has 30 chest x-ray dictations for 6 different diseases. Special web application was created with an easy to use interface that allows a specialist to annotate a text. Figure 3 shows the main interface done with Asynchronous JavaScript And XML (AJAX) technology [22].

In order to check the usefulness of this approach pilot project accuracy measure that focuses only on the ambiguous mappings was used. If the maximally

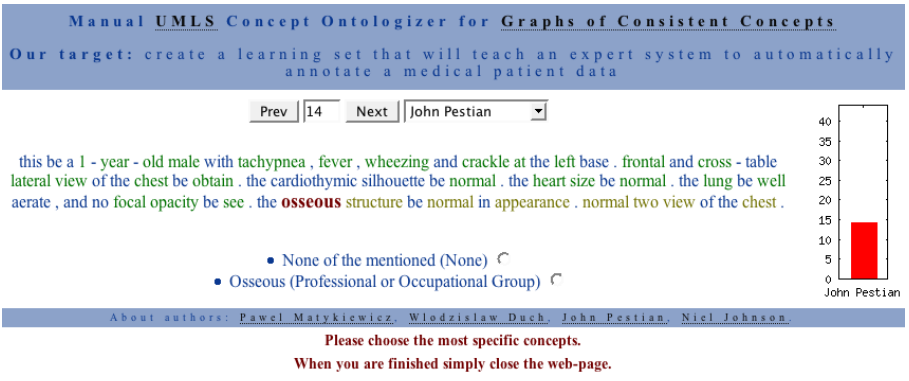


Fig. 3. Influence of learning on CUI selection

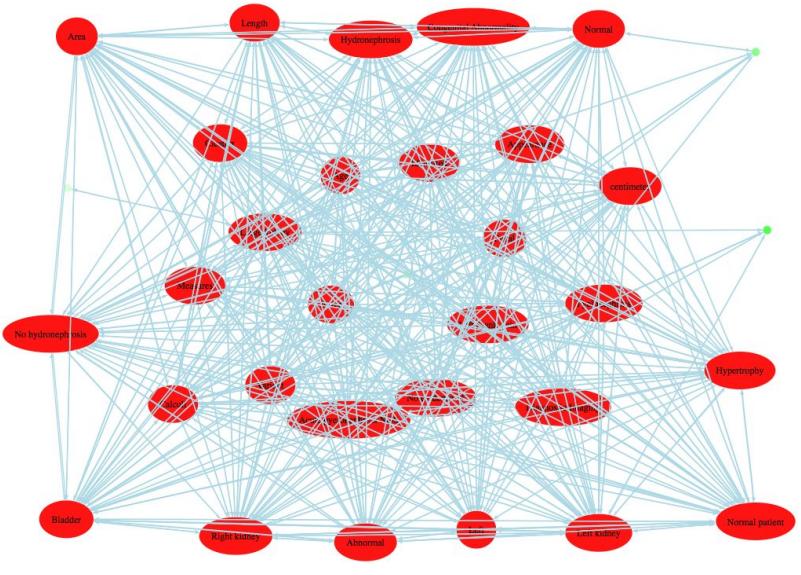


Fig. 4. Example of a graph of consistent concepts with enriched UMLS relations

activated CUI corresponds to the manually chosen CUI then a correct recognition is counted. Overall *Corpus I* has 140 ambiguous phrases and *Corpus II* 301 ambiguous phrases. Table 1 shows comparison of accuracies with no training, training using Corpus I and training using Corpus II. The second corpus seems to be much more difficult to learn but overall results are promising.

The initial weights were able to give maximum activation to only 64% of correct concepts but manually adding the relations found in the radiology corpus to this toy example gave perfect disambiguation in all cases. Figure 4 shows the GCC graph after adding new co-occurrence relations. This figure presents more compact and consistent graph. Only the right senses of the phrases (SUIs) have maximum activation potential.

Table 1. Comparison of GCC disambiguation accuracies with and without additional training

	no training	training	
		Corpus I	Corpus II
Corpus I	79%	96%	86%
Corpus II	57%	64%	79%

4 Conclusions

General neurocognitive approach to the natural language processing has been described and an algorithm for nonambiguous medical concept mapping, based on this approach, has been presented. Unfortunately even with the use of ULMS parsed clinical texts showed that many relations needed for correct annotations are still missing. A software tool for enriching UMLS relation by creating manually annotated texts and learning from them has been presented. Experiments performed on two small corpuses showed significant influence of additional knowledge on the disambiguation performance of GCC graphs.

Annotation of unstructured medical texts is quite difficult. Sometimes the UMLS mappings do not make sense for medical experts. In such cases they have additional CUI to choose: *None of the mentioned* (Fig. 3). This means that none of the CUI that are mapped to a SUI should be included in the graph (or they should at least have a very small activation). This and many other issues remain still to be investigated.

GCCs are a promising tool for Natural Language Processing tasks. They provide a better approximation to brain processes then vector models, yet computationally they are relatively simple, using only a single vector with node activations and a weight matrix estimating strength of relations. There may be many variants of GCC-based algorithms, with different strategies for initial node activations and subsequent activation spreading, weighting of relations, and overall consistency scoring evaluations. To add new knowledge large manually annotated corpus should be created, and better concept descriptions created using medical textbooks and dictionaries (the problem of automatic creation of semantic memory has been considered in [6]). Combining ideas from cognitive neuroscience with ideas from medical information retrieval literature algorithms that reach human level performance should finally be achieved.

Acknowledgement. WD is grateful for the support by the Polish Committee for Scientific Research, research grant 2005-2007.

References

1. UMLS Knowledge Server web site: http://umlsks.nlm.nih.gov
2. MetaMap web site: http://mmtx.nlm.nih.gov

3. M. Stevenson, "Word Sense Disambiguation: The Case for Combinations of Knowledge Sources", The University of Chicago Press, Chicago IL, 2003
4. P. Edmonds, and A. Kilgarriff, "Introduction to the special issue on evaluating word sense disambiguation systems", Journal of Natural Language Engineering, 8(4), 279-291, 2002.
5. J.P. Pestian, L. Itert, C. Andersen, W. Duch, "Preparing Clinical Text for Use in Biomedical Research." Journal of Database Management 17(2), 1-11, 2006.
6. J. Szymanski, T. Sarnatowicz, W. Duch, "Towards Avatars with Artificial Minds: Role of Semantic Memory". Journal of Ubiquitous Computing and Intelligence (in print).
7. W. Duch, J. Szymanski, T. Sarnatowicz, "Concept description vectors and the 20 question game". In: Intelligent Information Processing and Web Mining, Eds. M.A. Klopotek, S.T. Wierzchon, K. Trojanowski, Advances in Soft Computing, Springer Verlag, pp. 41-50, 2005.
8. J.R. Anderson, "Learning and Memory". J. Wiley and Sons, NY 1995.
9. T.P Mcnamara, "Semantic Priming; Perspectives from Memory and Word Recognition (Essays in Cognitive Psychology)", Psychology Press, UK, 2005
10. F. Pulvermller, "The Neuroscience of Language. On Brain Circuits of Words and Serial Order". Cambridge Uni. Press, 2003.
11. R. Hecht-Nielsen, "Cogent confabulation". Neural Networks 18(2): 111-115, 2005.
12. A.M. Collins and M.R. Quillian, "Retrieval time from semantic memory". Journal of Verbal Learning and Verbal Behavior 8, 2407, 1969.
13. A.M. Collins, E.F. Loftus, "A spreading-activation theory of semantic processing". Psychological Reviews 82, 40728, 1975.
14. J.L. McClelland and T.T. Rogers, "The Parallel Distributed Processing Approach to Semantic Cognition". Nature Reviews Neuroscience 4, 310-322, 2003.
15. J.F. Sowa, ed. "Principles of Semantic Networks: Explorations in the Representation of Knowledge". Morgan Kaufmann Publishers, San Mateo, CA, 1991.
16. F. Lehmann, ed. "Semantic Networks in Artificial Intelligence". Pergamon Press, Oxford, 1992.
17. W. Duch, "Computational Creativity". World Congres on Computational Intelligence, Vancouver, 16-21 July 2006
18. C. Burgess, "Representing and resolving semantic ambiguity: A contribution from high-dimensional memory modeling". In: Gorfein, D.S. (Ed.), On the Consequences of Meaning Selection: Perspectives on Resolving Lexical Ambiguity. APA Press, 2001.
19. T.C. Rindflesch and A.R. Aronson. "Semantic processing for enhanced access to biomedical knowledge". In: V. Kashyap, L. Shklar (eds), Real World Semantic Web Applications, IOS Press, pp. 157-172, 2002.
20. Tree-Tagger web site: http://www.ims.uni-stuttgart.de//projekte/corplex/TreeTagger/DecisionTreeTagger.html
21. C. Rocha, D. Schwabe, M. P. Aragao, A hybrid approach for searching in the semantic web Source, International World Wide Web Conference archive Proceedings of the 13th international conference on World Wide Web, New York, NY, USA
22. AJAX Sun web site http://developers.sun.com/channel/01_06/index.jsp?cid=59754

Feature Selection and Ranking of Key Genes for Tumor Classification: Using Microarray Gene Expression Data

Srinivas Mukkamala*, Qingzhong Liu, Rajeev Veeraghattam, and Andrew H. Sung*

Department of Computer Science, New Mexico Tech, Socorro, NM 87801
* Institute for Complex Additive Systems and Analysis, New Mexico Tech
{srinivas, liu, rajeev, sung}@cs.nmt.edu

Abstract. In this paper we perform a t-test for significant gene expression analysis in different dimensions based on molecular profiles from microarray data, and compare several computational intelligent techniques for classification accuracy on Leukemia, Lymphoma and Prostate cancer datasets of broad institute and Colon cancer dataset from Princeton gene expression project. Classification accuracy is evaluated with Linear genetic Programs, Multivariate Regression Splines (MARS), Classification and Regression Tress (CART) and Random Forests. Linear Genetic Programs and Random forests perform the best for detecting malignancy of different tumors. Our results demonstrate the potential of using learning machines in diagnosis of the malignancy of a tumor.

We also address the related issue of ranking the importance of input features, which is itself a problem of great interest. Elimination of the insignificant inputs (genes) leads to a simplified problem and possibly faster and more accurate classification of microarray gene expression data. Experiments on select cancer datasets have been carried out to assess the effectiveness of this criterion. Results show that using significant features gives the most remarkable performance and performs consistently well over microarray gene expression datasets we used. The classifiers used perform the best using the most significant features expect for Prostate cancer dataset.

1 Introduction

Though most cells in our bodies contain the same genes, not all of the genes are used in each cell. Some genes are turned on, or "expressed" when needed. Many genes are used to specify features unique to each type of cell. Microarray technology looks at many genes at once and determines which are expressed in a particular cell type. Using DNA microarray analysis thousands of individual genes can be spotted on a single square inch slide. DNA targets are arrayed onto glass slides (or membranes) and explored with fluorescent or radioactively labeled probes [1]. Obtaining genome-wide expression data from cancerous tissues gives insight into the gene expression variation of various tumor types, thus providing clues for cancer classification of individual samples. One of the key challenges of microarray studies is to derive biological insights from the unprecedented quantities of data on gene expression patterns. Partitioning genes into closely related groups has become an element of practically all

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 951 – 961, 2006.
© Springer-Verlag Berlin Heidelberg 2006

analyses of microarray data [2]. But identification of genes faces with many challenges. The main challenge is the overwhelming number of genes compared to the smaller number of available training samples. In machine learning terminology, these data sets have high dimension and small sample size. And many of these genes are irrelevant to the distinction of samples. These irrelevant genes have negative effect on the accuracies of the classifier. Another challenge is that DNA array data contain technical and biological noise. Thus, it is critical to identify a subset of informative genes from a large data that will give higher classification accuracy.

Many methods have been proposed in the past to reduce the dimensionality of gene expression data [3]. Several machine learning techniques have been successfully applied to cancer classification using microarray data [4]. One of the early methods is a hierarchical algorithm developed by Eisen et al. [5]. Other popular algorithms, such as neural networks, K-Nearest Neighbor (KNN), support vector machines, kernel based classifiers, genetic algorithms and Self-Organizing Maps (SOM) are widely applied for tumor classification [3, 6].

In this paper, we extract different dimensional gene data based on t-test and apply Regression Splines (MARS), Classification and Regression Tress (CART) Random Forests and Linear Genetic Programs (LGP) to extracted datasets, and compare the classification accuracy on microarray data.

This paper is organized as follows: section 2 presents gene expression data and t-test analysis to extract key features; section 3 introduces to the problem of feature ranking. Datasets used for experiments and gene data selection specific to the datasets used is described in section 4. Section 5 introduces *Multivariate Regression Splines (MARS)* and section 6 *Classification and Regression Tress (CART)*. Random forests are described in section 7. A brief introduction to Linear Genetic Programs (LGP) is given in section 8. Classifier performance is presented in section 9. Summary and conclusions are given in section 10.

2 Gene Expression Data Selection

For a given classifier and a training set, the optimality of a gene identification algorithm can be ensured by an exhaustive search over all possible gene subsets. For a data set with n genes, there are 2^n gene subsets. Due to the high dimension of microarrays data, it is impractical to search whole space exhaustively. In our experiments, we choose the significant data based on Student's *t*-test.

2.1 Student's *t*-Test

Student's *t*-test deals with the problems associated with inference based on "small" samples. The unpaired t method tests the null hypothesis that the population means related to two independent, random samples from an approximately normal distribution are equal [7].

Under the assumption of equal underlying population means, if t < 0, "P(T <= t) one-tail" gives the probability that a value of the t-Statistic would be observed that is more negative than t. If t >=0, "P(T <= t) one-tail" gives the probability that a value of the t-Statistic would be observed that is more positive than t. "t Critical one-tail"

gives the cutoff value so that the probability of observing a value of the t-Statistic greater than or equal to "t Critical one-tail" is Alpha.

"P(T <= t) two-tail" gives the probability that a value of the t-Statistic would be observed that is larger in absolute value than t. "P Critical two-tail" gives the cutoff value so that the probability of an observed t-Statistic larger in absolute value than "P Critical two-tail" is Alpha.

Assuming unequal variances, eqation 1 is used to determine the statistic value *t* and equation 2 is used to calculate the degrees of freedom, df:

$$d = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (1)$$

$$df = \frac{\left[\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right]^2}{\frac{\left(s_1^2/n_1\right)^2}{n_1 - 1} + \frac{\left(s_2^2/n_2\right)^2}{n_2 - 1}} \quad (2)$$

3 Feature Ranking

The feature ranking for microarray gene expression data is similar in nature to various engineering problems that are characterized by:

- Having a large number of input variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ of varying degrees of importance to the output **y**; i.e., some elements of **x** are essential, some are less important, some of them may not be mutually independent, and some may be useless or irrelevant (in determining the value of **y**)
- Lacking an analytical model that provides the basis for a mathematical formula that precisely describes the input-output relationship, $\mathbf{y} = \boldsymbol{F}(\mathbf{x})$
- Having available a finite set of experimental data, based on which a model (e.g. neural networks) can be built for simulation and prediction purposes
- Excess features can reduce classifier accuracy
- Excess features can be costly to collect
- If real time classification is important, excess features can reduce classifier operating speed independent of data collection
- If storage is important, excess features can be costly to store

4 Experiments

Leukemia, Lymphoma and Prostate cancer data sets are obtained from broad institute [19]. Colon cancer data set is obtained from Princeton gene expression project [20]. Significant gene data obtained from t-test is used for measuring the performance of the classifiers. Fifty percent of the data is used for training and the rest is used for testing. Leukemia data set has (37 training samples and 38 testing samples). Lymphoma data set has (40 training samples and 39 testing samples). Prostate data set has (52 training samples and 52 testing samples). Colon data set has (32 training samples and 32 testing samples).

Data sets used in our experiments.

- Leukemia data set comes from a study of gene expression in two types of acute Leukemia: 48 acute lymphoblastic Leukemia (ALL) samples and 25 acute myeloblastic Leukemia (AML) samples. It was studied in [15].
- Lymphoma data set consists of 58 diffuse large B-cell lymphoma (DLBCL) samples and 19 follicular lymphoma (FL) samples. It was studied in [16]. The data file, lymphoma_8_lbc_fscc2_rn.res, and the class label file, lymphoma_8_lbc_fscc2.cls are used in our experiments for identifying DLBCL and FL.
- Prostate data set in [17] contains 52 prostate tumor samples and 50 non-tumor prostate samples.
- The Colon data set in [18] consists of 40 tumor and 22 normal colon tissues.

4.1 Gene Data Selection Based on t-Test

Different thresholds are set in our experiments and different dimension of the most significant gene data are extracted as feature space. Figure 1 (a, b, c, d) shows the dimensions of the filtered significant data according to different p-value thresholds for Leukemia, Lymphoma**,** Colon**,** and Prostate data sets.

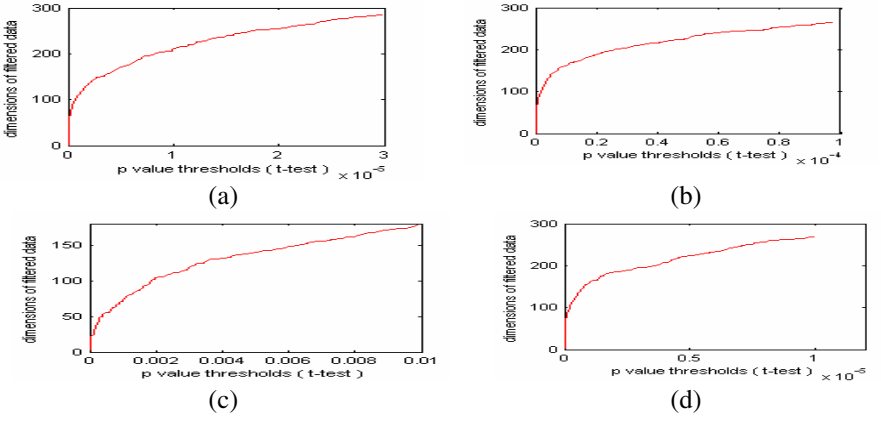


Fig. 1. The dimensions of filtered significant data for Prostate (a), Leukemia **(b),** Colon **(c),** and Lymphoma **(d)** data sets, respectively. The p-values of filtered data are smaller than the corresponding thresholds in x-label. Figure 1 (a, b, c, and d) indicates that the significance levels of Prostate**,** Lymphoma and Leukemia data sets are higher than Colon data set.

5 Multivariate Adaptive Regression Splines (MARS)

Multivariate Adaptive Regression Splines (MARS) is a nonparametric regression procedure that makes no assumption about the underlying functional relationship between the dependent and independent variables. Instead, MARS constructs this relation from a set of coefficients and basis functions that are entirely "driven" from the

data [8]. The method is based on the "divide and conquer" strategy, which partitions the input space into regions, each with its own regression equation. This makes MARS particularly suitable for problems with higher input dimensions, where the curse of dimensionality would likely create problems for other techniques [8,9].

The MARS Model: the basis functions together with the model parameters (estimated via least squares estimation) are combined to produce the predictions given the inputs. The general MARS

$$y = f(x) = \beta_o + \sum_{m-1}^{M} \beta_m h_m(X) \quad (6)$$

Where the summation is over the M nonconstant terms in the model, y is predicted as a function of the predictor variables X (and their interactions); this function consists of an intercept parameter (β_o) and the weighted by (βm) sum of one or more basis functions $h_m(X)$ [9].

Model Selection: after implementing the forward stepwise selection of basis functions, a backward procedure is applied in which the model is pruned by removing those basis functions that are associated with the smallest increase in the (least squares) goodness-of-fit. A least squares error function (inverse of goodness-of-fit) is computed. The so-called Generalized Cross Validation error is a measure of the goodness of fit that takes into account not only the residual error but also the model complexity as well. It is given by

$$GCV = \sum_{i=1}^{N} (y_i - f(x_i))^2 / (1 - c/n)^2 \quad (7)$$

with $C = 1 + cd$.

Where N is the number of cases in the data set, d is the effective degrees of freedom, which is equal to the number of independent basis functions. The quantity c is the penalty for adding a basis function. Experiments have shown that the best value for C can be found somewhere in the range $2 < d < 3$ [9].

6 Classification and Regression Trees (CART)

CART builds classification and regression trees for predicting continuous dependent variables (regression) and categorical predictor variables (classification) [10].

The decision tree begins with a root node t derived from whichever variable in the feature space minimizes a measure of the impurity of the two sibling nodes. The measure of the impurity at node t, denoted by i(t), is as shown in the following equation:

$$i(t) = -\sum_{f=1}^{k} p(w_j / t) \log p(w_j / t) \quad (8)$$

Where p(wj | t) is the proportion of patterns xi allocated to class wj at node t. Each non-terminal node is then divided into two further nodes, tL and tR, such that pL , pR

are the proportions of entities passed to the new nodes tL, tR respectively. The best division is that which maximizes the difference given in:

$$\Delta i(s,t) = i(t) - pi_L(t_L) - pi_R(t_R) \quad (9)$$

The decision tree grows by means of the successive sub-divisions until a stage is reached in which there is no significant decrease in the measure of impurity when a further additional division s is implemented. When this stage is reached, the node t is not sub-divided further, and automatically becomes a terminal node. The class wj associated with the terminal node t is that which maximizes the conditional probability p(wj | t). Each of the terminal node describes a data value; each record is classifies into one of the terminal node through the decisions made at the non-terminal node that lead from the root to that leaf [8,10].

7 Random Forests

A random forest is a classifier consisting of a collection of tree structured classifiers $\{\mathbf{h}(\mathbf{x},\Theta_k), k=1, \ldots\}$ where $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class of input $\mathbf{X}$. The common element in random trees is that for the $\mathbf{K}^{th}$ tree, a random vector Θ_k is generated, independent of the past random vectors $\Theta_1, \ldots \Theta_{k-1}$ but with the same distribution; and a tree is grown using the training set and Θ_k, resulting in a classifier $\mathbf{h}(\mathbf{x},\Theta_k)$ where $\mathbf{x}$ is an input vector. For instance, in bagging the random vector Θ is generated as the counts in $\mathbf{N}$ boxes resulting from $\mathbf{N}$ darts thrown at random at the boxes, where N is number of examples in the training set. In random split selection Θ consists of a number of independent random integers between 1 and K. The nature and dimensionality of Θ depends on its use in tree construction. After a large number of trees are generated, they vote for the most popular class [11].

The random forest error rate depends on two things:

- ✓ The ***correlation*** between any two trees in the forest. Increasing the correlation increases the forest error rate.
- ✓ The ***strength*** of each individual tree in the forest. A tree with a low error rate is a strong classifier. Increasing the strength of the individual trees decreases the forest error rate.

8 Linear Genetic Programming (LGP)

Linear Genetic Programming (LGP) is a variant of the genetic programming technique that acts on linear genomes. The linear genetic programming technique used for our current experiment is based on machine code level manipulation and evaluation of programs. Its main characteristic, in comparison to tree-based GP, is that the evolvable units are not the expressions of a functional programming language (like LISP); instead, programs of an imperative language (like C) are evolved [12,13,14].

In the automatic induction of machine code by GP, individuals are manipulated directly as binary code in memory and executed directly without passing through an

interpreter during fitness calculation. The LGP tournament selection procedure puts the lowest selection pressure on the individuals by allowing only two individuals to participate in a tournament. A copy of the winner replaces the loser of each tournament. The crossover points only occur between instructions. Inside instructions the mutation operation randomly replaces the instruction identifier.

In GP an intron is defined as part of a program that has no influence on the fitness calculation of outputs for all possible inputs. Fitness F of an individual program p is calculated as

$$F(p) = \frac{1}{nm}\sum_{j=1}^{n}\left(o_{ij}^{pred} - o_{ij}^{des}\right)^2 + \frac{w}{n}CE = MSE + wMCE \tag{10}$$

i.e., the mean square error (MSE) between the predicted output (o_{ij}^{pred}) and the desired output (o_{ij}^{des}) for all n training samples and m outputs. The classification error (CE) is defined as the number of misclassifications. Mean classification error (MCE) is added to the fitness function while its contribution is determined by the absolute value of weight (w) [12].

9 Classifier Performance

We applied MARS, CART, Random forests and LGPs to Leukemia (6,27,53), Lymphoma (7,28,55), Colon (7,15,27,54) and Prostate (6,26,52) cancer data sets, for detecting malignancy of a tumor with different data dimensionalities given in the parenthesis. Classification accuracies are summarized in tables 1 to 4. Table 1 summarizes Leukemia classification accuracies of MARS, CART, LGP and Random forests on 6, 27 and 53. Table 2 summarizes Prostate cancer classification accuracies. Table 3 summarizes Colon cancer classification accuracies. Table 4 summarizes Lymphoma cancer classification accuracies.

Detection rates and false alarms are evaluated for the cancer data sets, and the obtained results are used to form the ROC curves. The point (0,1) is the perfect classifier, since it classifies all positive cases and negative cases correctly. Thus an ideal system will initiate by identifying all the positive examples and so the curve will rise

Table 1. Leukemia Classification Accuracies

	No of Features					
	6		**27**		**53**	
	Class 1	**Class2**	**Class 1**	**Class2**	**Class 1**	**Class2**
MARS	75	100	83.33	76.92	100	84.62
CART	95.83	92.3	91.66	92.3	91.66	92.3
LGP	**100**	**100**	**100**	**100**	**100**	**100**
Random Forests	91.66	100	95.83	100	95.83	100

Table 2. Prostate Cancer Classification Accuracies

	No of Features					
	6		26		52	
	Class 1	Class2	Class 1	Class2	Class 1	Class2
MARS	44	**92.31**	60	**96.15**	88	92.31
CART	64	92.3	60	**96.15**	60	**96.15**
LGP	**92**	**92.31**	**96**	**96.15**	**100**	**96.15**
Random Forests	68	92.3	80	88.46	80	88.46

Table 3. Colon Cancer Classification Accuracies

	No of Features					
	7		27		54	
	Class 1	Class2	Class 1	Class2	Class 1	Class2
MARS	63.64	80	**81.82**	85	81.82	80
CART	36.36	**95**	36.36	**95**	36.36	**95**
LGP	**81.82**	90	**81.82**	90	**81.82**	90
Random Forests	63.63	90	81.81	80	72.72	85

Table 4. Lymphoma Cancer Classification Accuracies

	No of Features					
	7		28		54	
	Class 1	Class2	Class 1	Class2	Class 1	Class2
MARS	**100**	44.44	79.31	77.78	96.55	33.33
CART	86.2	88.88	**96.55**	55.55	96.55	55.55
LGP	**100**	**100**	**96.55**	**100**	**100**	**100**
Random Forests	89.65	**100**	**96.55**	88.88	96.55	88.88

to (0,1) immediately, having a zero rate of false positives, and then continue along to (1,1). Classification accuracies of the best feature set for different cancer classifications are given in Figures 2, 3, 4, and 5. Figure 2 summarizes the classification performance of classifiers for Leukemia cancer dataset using 6 features. Figure 3 summarizes the classification performance of classifiers for Prostate cancer dataset using 52 features. Figure 4 summarizes the classification performance of classifiers for Colon cancer dataset using 27 features. Figure 5 summarizes the classification performance of classifiers for Lymphoma cancer dataset using 54 features. LGP performed the best for all the datasets with different feature dimensionalities.

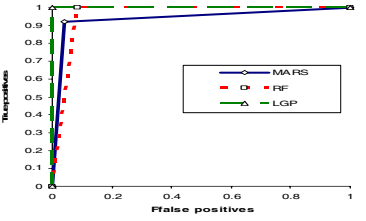


Fig. 2. Classifiers Performance on Leukemia Dataset Using 6 Features

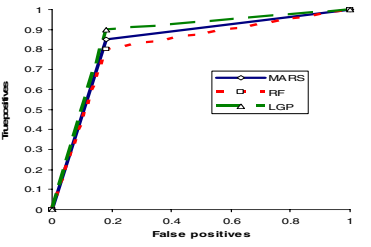


Fig. 4. Classifiers Performance on Colon Dataset Using 27 Features

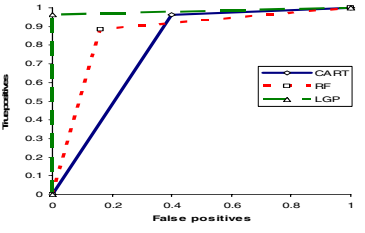


Fig. 3. Classifiers Performance on Prostate Dataset Using 52 Features

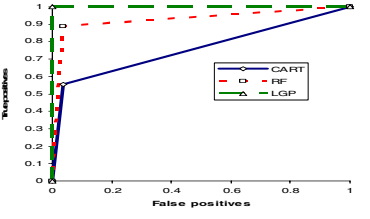


Fig. 5. Classifiers Performance on Lymphoma Dataset Using 27 Features

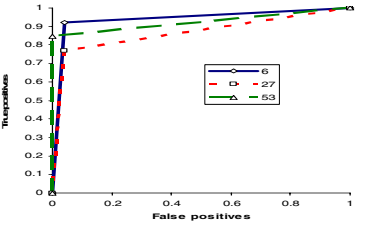


Fig. 6. Performance of MARS on Most Significant Features of Leukemia

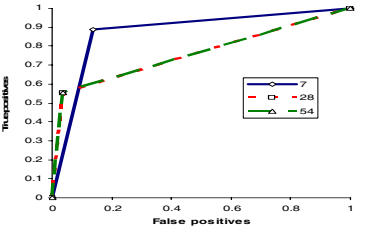


Fig. 8. Performance of CART on Most Significant Features of Lymphoma

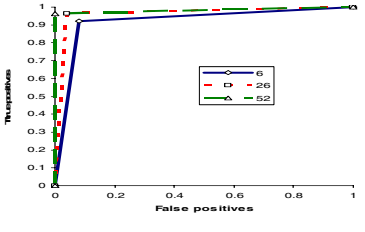


Fig. 7. Performance of LGP on Most Significant Features of Prostate Cancer

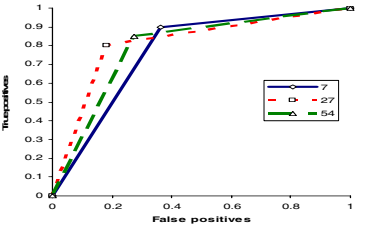


Fig. 9. Performance of RF on Most Significant Features of Colon Cancer

Comparison of classification accuracies of most significant features based on t-test is given in Figures 6-9. Performance results of MARS on most significant features of Leukemia cancer dataset are summarized in Figure 6. Performance results of LGP on most significant features of Prostate cancer dataset are summarized in Figure 7. Performance results of CART on most significant features of Lymphoma cancer dataset are summarized in Figure 8. Performance results of Random forests on most significant features of Colon cancer dataset are summarized in Figure 9.

10 Summary and Future Work

Although the performance of the four methods used is comparable in all datasets, we found that linear genetic programs and Random Trees achieved consistently the best results. MARS performs very closely to CART.

LGP performs the best using 6 features for Leukemia dataset and 7 features for Colon and Lymphoma cancer datasets. For Prostate cancer dataset LGP performs the best using 52 features.

The classifiers used in this paper showed comparable or better performance in some cases when compared to the ones reported [artificial neural networks, clustering, support vector machines, etc] in the literature using the same datasets. Our results demonstrate the potential of using learning machines in diagnosis of malignancy of a tumor. As a future work we plan to use large datasets of patients. As more inputs are added, feature selection will have to follow a more stringent scrutiny.

References

1. P. Brown, D. Botstein, "Exploring the New World of the Genome with DNA Microarrays", Nature Genetics Supplement, Vol. 21, pp. 33-37, 1999.
2. J. Quackenbush, "Computational Analysis of Microarray Data", Nature Rev. Genteics, Vol. 2, pp. 418-427, 2001.
3. S. Dudoit, J. Fridlyand, T. Speed, "Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data", J. Am. Statistical Assoc., Vol. 97, pp. 77-87, 2002.
4. C. Peterson, M. Ringner, "Analysis Tumor Gene Expression Profiles", Artificial Intelligence in Medicine, Vol. 28, no. 1, pp. 59-74, 2003.
5. M. Eisen, P. Spellman, P. Brown, D. Botstein, "Cluster Analysis and Display of Genome-Wide Expression Patterns", Proc. Nat'l Acad. Sci. USA, Vol. 95, pp. 14863-14868, 1998.
6. P. Tamyo et al. "Interpreting Patterns of Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation", Proc. Nat'l Acad. Sci. USA, Vol. 96, pp. 2907-2912, 1999.
7. P. Armitage, G. Berry, Statistical Methods in Medical Research, Blackwell 1994.
8. Salford Systems. TreeNet, CART, MARS, Random Forests Manual.
9. T. Hastie, R. Tibshirani, and J. H. Friedman, The elements of statistical learning: Data mining, inference, and prediction. Springer, 2001.
10. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and regression trees. Wadsworth and Brooks/Cole Advanced Books and Software, 1986.
11. L. Breiman. Random Forests. Journal of Machine Learning, Vol. 45, pp. 5-32, 2001.

12. J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, Cambridge, MA: The MIT Press, 1992.
13. D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Reading, MA: Addison-Wesley, 1989.
14. AIM Learning Technology, http://www.aimlearning.com.
15. T. Golub et al., "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression", Science, Vol. 286, pp. 531-537, 1999.
16. M. Shipp et al., "Diffuse Large B-Cell Lymphoma Outcome Prediction by Gene Expression Profiling and Supervised Machine Learning", Nature Medicine, Vol. 8, no. 1, pp. 68-74, 2002.
17. D. Singh et al., "Gene Expression Correlates of Clinical Prostate Cancer Behavior", Cancer Cell, Vol. 1, no. 2, pp. 227-235, 2002.
18. U. Alon et al., "Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays", Proc. Nat'l Acad. Sci., Vol. 96, pp. 6745-6750, 1999.
19. http://www.broad.mit.edu/
20. http://microarray.princetion.edu/oncology

Cognitive Analysis in Diagnostic DSS-Type IT Systems

Lidia Ogiela[1], Ryszard Tadeusiewicz[2], and Marek R. Ogiela[2]

[1] AGH University of Science and Technology, Faculty of Management,
Al. Mickiewicza 30, PL-30-059 Kraków, Poland
logiela@agh.edu.pl
[2] AGH University of Science and Technology, Institute of Automatics
{rtad, mogiela}@agh.edu.pl

Abstract. This paper demonstrates that AI methods - in particular, linguistic mechanisms of semantic meaning reasoning can be applied to the development of intelligent IT systems. They also facilitate an in-depth analysis of the meaning presented in DDS information systems.

This paper also presents the IT mechanisms of object meaning description on selected examples of spinal cord image analysis. The procedures for such semantic reasoning are based on the model of cognitive resonance. These have been applied to the task of interpreting the meaning of selected diagnostic images from the central nervous system as an intelligent analysis module in IT systems. The application presented in this paper is of a research character and it serves the preparation of efficient lesion detection methods applied to a dataset originating from magnetic and resonance examinations of the spinal cord structures.

1 Introduction

DSS systems (*Diagnostic Support Systems*) are currently very popular due to their wide diagnostic possibilities. In this paper we shall show an example of a system that was prepared not only to diagnose, but one that is also oriented towards the issues of cognitive analysis and the understanding pathological lesions taking place in the area of central nervous system. Particular attention is paid to disease lesions in the spinal cord.

Every medical image constituting a type of primary component for diagnostic IT systems is subject to analysis. The objective is to determine whether there is any important disease lesions observed in the patient's analysed organ or whether there are no such changes (i.e. the patient is healthy). If there are any such lesions, their type is analysed and the system directs its functions towards determining what disease the patient has. DSS systems operate on the basis of three main rules:

– **Image transformation** in order to obtain the best possible content quality and substance which the image carries,

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 962–971, 2006.
© Springer-Verlag Berlin Heidelberg 2006

– **Image analysis** in order to get the image properties in the form of a feature vector,
– **Image recognition** in order to classify all features of the analysed image.

DSS systems proposed in earlier research were used, among others, for pancreas as well as for kidney and heart disease diagnosis. Their functioning is based on medical image recognition methods [4] (fig. 1).

Fig. 1. Medical image recognition diagram

Due to the fact that DSS systems develop very rapidly, an attempt was made to construct a new class of such systems using in their operation the mechanisms of cognitive analysis. The said are to be directed at attempts to automatically understand the semantics of analysed images, and therefore at their content meaning interpretation.

2 Cognitive Analysis in Information Systems

Cognitive analysis used in IT systems is very often based on the syntactic approach [5]. For the purpose of meaning image interpretation it first uses a pre-processing operation usually composed of:

– Image coding by means of terminal elements of the introduced language,
– Analysed object shape approximation, as well as,
– Filtration and pre-processing of the input image.

As a result of the execution of such stages it is possible to obtain a new image representation in the form of hierarchic semantic tree structures and subsequent production steps of this representation from the initial grammar symbol [3]. An intelligent cognitive system distinguishing at the stage of pre-processing image data must, in the majority of cases, perform image segmentation, identify primitive components and determine spatial as well as semantic relations between them. An appropriate classification (also machine perception) is based on the recognition of whether a given representation of the actual image belongs to a class of images generated by languages defined by one of possible number of grammars. Such grammars can be considered to belong to sequential, tree and graph grammars while recognition with their application is made in the course of a syntactic analysis performed by the system [3,4].

In the most recent research on intelligent information systems it was observed that the recognition of an analysed image alone is insufficient since more and

more frequently there is a postulate to direct the intelligent information systems' possibilities so that they are able to perform the operation of automatically understanding image semantics. In order to enable such reasoning, the techniques of artificial intelligence are used. Apart from a simple recognition of an image they enable one that also extracts important semantic information allowing for a meaning interpretation, i.e. machine understanding.

This process relates only to cognitive information systems and it is a lot more complex than with pure recognition. This is due to the fact that in this case the flow of information goes clearly in two directions. In this model the stream of empirical data, as contained in the sub-system and aimed to register and analyse the image, interferes with the stream of generated expectations [3,5].

Between the stream expectation, generated for every hypothetical image and the data steam that is obtained by means of analysis of the currently considered image, there must be a special interference. As a result of this some coincidences (of expectations and features found on the image) gain on importance while others (both compliant and non compliant) lose their importance. This interference leads to a cognitive resonance, which confirms one of certain possible hypotheses (in the case of an image whose content can be understood) or makes it possible to determine that there is a discordance, which cannot be removed, between the currently perceived image and all other Gnostic hypotheses with an understandable interpretation. The second case stands for a failure of automatic image understanding.

Cognitive information systems function based on the cognitive resonance phenomenon which belongs only to these systems and differentiates them from other intelligent IT systems [3]. The application and use of such systems can be multiple due to wide possibilities offered to them by contemporary science. Nevertheless the greatest possibilities for the use of cognitive IT systems are currently offered by the medicine. This is due to the fact that there are more and more diseases in on-going pathological processes in individual organs and a growing number of detection cases as well as diagnosing these diseases. Medical images belong to some of the most varied data and they contain extremely deep and important (among others, for the patient's fate) meaning interpretation. Cognitive information systems could certainly also serve many other fields of science and everyday life, should an attempt be made to develop intelligent information systems in the field of economics, marketing, management, logistics, military affairs by adding the process of understanding the analysed information or data.

3 Artificial Intelligence Techniques in DSS Diagnostic Systems

In the state-of-the-art development trends of intelligent IT systems it has been noticed that the pre-processing, analysis and classification (recognition) operations on the examined data are no longer sufficient. On the other hand, there is a more and more frequent postulate to direct them at the operation of automatic meaning understanding, as carried by the analysed and transformed data, for

example the semantics of the analysed medical images. The human mind has incomparably greater perception capacities than a computer even with the best software so that it can reach such meanings appropriate for the observed objects or analysed data infinitely better than a machine. Nevertheless also machine understanding techniques are slowly being improved and with time they could be used for the performance of a more complex reasoning process, one relating to the significance of data collected rather than just for their simple analysis. In order to enable IT systems such semantic reasoning based on data, advanced IT techniques are used. These techniques, apart from simple information analysis and possible classification (recognition) of data destined for analysis, make it possible also to extract important semantic information from them, ones that point to meaning interpretation. At the current stage of development, data semantic analysis is always set in some pre-determined context. It is impossible for a computer to discover simultaneously the analysis objective and its result. This means that systems currently built can undertake an attempt at understanding data with some *a priori* pre-definition of what the understanding is supposed to serve. This must be differentiated from a situation in which a human being, coming across a new situation analyses it in many respects; the outcome of the analysis could be completely unexpected conclusions standing for a complete mental consideration of a given situation, i.e. its complete understanding. Referring to a frequently quoted example of semantic analysis of some specified medical images one can expect that the computer, after an analysis of X-ray image will 'understand' that the patient suffers from some kind of disease. This would not be achievable applying only the technique of automatic image recognition. On the other hand, a human being looking at the same image can, of course, do the same by diagnosing (the diagnosis being the same as the computer would have made or a different one). However, only a man can understand something totally unexpected, for example that an image is bad in quality because the X-ray machine was out of focus and that the examination must be repeated. The first type of understanding is well set in the context of medical examination. It is therefore available both for a medical doctor and for an appropriately programmed computer. The latter requires going outside the framework of an *a priori* defined scenario and for the time being it is available only for humans.

The main objective of the considerations presented in this paper is to focus the Reader's attention only at the first, easier way of interpreting data understanding process (for example, of images). Still even this process is a lot more complex than just data analysis and their possible recognition. Information flow in the second case is clearly two-sourced and two-directional (just like in the cognitive understanding process model, as taking place during eye perception). In the model considered here, the empirical data stream is collected and stored in a sub-system whose objective is to register and analyse the data the which the analysed IT stores and processes in accordance with its destination; this interferes with a stream of automatically generated expectations concerning some selected features and data properties. The source of this expectation stream is the knowledge resources located in the system. It is a basis for the generation

of semantic hypothesis while the knowledge source are people (experts), from whom the knowledge was obtained and adjusted appropriately for being used in automatic reasoning process.

The terms and conceptual basis of the above-defined approach is a new knowledge field, the so-called cognitive analysis. Currently it is better known in the context of psychological scientists' analyses examining human cognitive processes. It is also known in the context of hypotheses about the nature of reason and rationality, as examined by philosophers dealing with the epistemology, gnoseology and semiotics foundations as well as criteriology by D. J. Mercier and other advanced intellectual trends. To a smaller degree, however, was it so far used in science itself [3].

4 DSS System Model for Perceptual Central Nervous System Image Analysis

In this chapter we shall propose, as an example of intelligent IT system, a medical model of IT system supporting diagnosing. The selected system conducts intelligent analysis of image data relating to pathological lesions in the central nervous system, related both to selected disease units of the spinal cord [3]. This model will be based on the construction and the operating rule of DSS systems (*Diagnostic Support Systems*). Due to the fact that the issue of occurrence of disease units in the spinal cord is extremely extensive, some selected pathological phenomena, representative of central nervous system disease types will be presented.

The main element of a correctly functioning IT system supporting the medical image diagnostics is, in accordance with the concept presented in this paper, analysis preparation of a cognitive method of disease units and pathological lesions as occurring in the spinal cord. The cognitive analysis contained in the DSS-central nervous system is aimed to propose an automatic correct interpretation method of these extremely complicated medical images, ones resulting from imaging parts of the nervous system. Such images are difficult to interpret due to the fact that various patients have various morphologies of the imaged organs. This is true both of the correct state and if there are any disease lesions. The nervous system, similarly as most elements of the human body, is not always correctly built and fully developed from the birth. The anatomy and pathomorphology differentiate between a number of developmental defects of the central nervous system. It often occurs that this system for the first couple of years functions correctly and only after some time there are some troubles with its functioning, demonstrated by the child's behaviour and feeling: seen either as a single symptom or as a widespread disease. All kinds of troubles occurring in the central nervous system, identified with disease units of the spinal cord are clinically diagnosed and subject to diagnostic procedure based mainly on image diagnostics. Due to small differentiation in the absorption of X-rays by the distinguished medical structures of the brain (for example, by the while and grey substance) as well as due to the fact that the whole central nervous system is

hidden behind bones (of the scull and backbone) which strongly attenuate X-rays, the main role in image examinations of the central nervous system is customary assigned to NMR topography (*Nuclear Magnetic Resonance*) labelled also zeugmatography or most frequently the MRI method (*Magnetic Resonance Imaging*) - imaging based on the nuclear magnetic resonance phenomenon.

Magnetic resonance makes it possible to obtain maps of density distribution (the so-called topography) primarily of hydrogen atom nuclei (protons) and of these protons' relaxation time. Owing to the application of a projection corresponding to the tomography technique (computational reconstruction of the examined parameter distribution based on many multi-directional probing) the NMR image can be obtained on any cross-section of the body. Hydrogen is a constituent of water making up 60-70% of living organisms; it is also a constituent of all organic compounds. It is worth remembering that fats have an extremely high amount of hydrogen. Information obtained about its distribution inside the organism is the basis for image construction: the images differentiate tissues with regards to the degree of their hydration or fat content. Proton density and their relaxation times can be mirrored by brightness (i.e. greyness degree) of points on the given map. The method of magnetic resonance offers a lot more contrasting soft tissue images then X-ray images. In the case of many diseases it can also show more precisely the difference between a healthy tissue and one that was changed by disease.

All the analysed images of spinal cord were, before their proper recognition, subject to segmentation and filtration procedures. Their aim was to extract from among other image elements important elements of the spinal cord [3]. Structures shown in this way were then subject to cognitive analysis stages using the grammar described below.

In order to analyse disease lesions of the spinal cord, the following attributed grammar has been proposed:

$$G_{sc} = (\Sigma_N, \Sigma_T, P, ST) \tag{1}$$

where:

Σ_N - stands for a set of non-terminal symbols (intermediary in the process of image description generation),
Σ_T - stands for a set of terminal symbols (final symbols describing shape features),
P - stands for a production set,
ST - stand for the grammar start symbol.

$\Sigma_N = \{$*SPINE_LESION, SPINAL_STENOSIS, SPINAL_DILATATION, SPINAL_TUMOR, N, D, S*$\}$
$\Sigma_T = \{n, d, s\}$

Apart from these, the following meaning was given to terminal elements present in the description:

n$\in$[-11°, 11°], d$\in$(11°, 180°), s$\in$(-180°, -11°),

Table 1. Production set defining changes in the G_{sc} grammar

Pathological lesion	Grammar rules	Semantic actions
Dilation/cyst	*1. SPINE_LESION* → *SPINAL_DILATATION* *2. SPINAL_DILATATION* → *D N S* *D N* *D S*	*Lesion* = spinal dilatation
Neoplastic tumours	*3. SPINE_LESION* → *SPINAL_TUMOR* *4. SPINAL_TUMOR* → *D S D S* *S D S N* *S D S D* *D S D N*	*Lesion* = spinal tumor
Stenosis, compression	*5. SPINE_LESION* → *SPINAL_STENOSIS* *6. SPINAL_STENOSIS* → *S N D* *S D* *S N*	*Lesion* = spinal stenosis
Elements of the detected lesions	*7.* $N \rightarrow n \mid n\ N$ *8.* $D \rightarrow d \mid d\ D$ *9.* $S \rightarrow s \mid s\ S$	*Lesion features* = *location,* *length,* *diameter,* *quantity,* *severity.*

ST = *SPINE_LESION*,
P production set has been defined as in table 1 below:

The proposed grammar makes it possible to detect various kinds of spinal cord or meningeal stenoses characteristic for neoplastic lesions and inflammatory processes of the spinal cord. Figure 2a presents an image of the spinal cord with a visible deformation; figure 2b shows the spinal cord image after binarisation while figure 2c depicts the diagram of the spinal cord. The red area represents the area of occurrence of the anomalies within the structure of the spinal cord. The set of yellow chords, cross-cutting the spinal cord in subsequent points perpendicularly to its axis, as shown on figure 2c which demonstrates how the width diagram was made.

Spinal cord width diagram (figure 2c) presents, in the most concise form, the results of spinal cord morphology analysis. It is the most precious source of information when one is looking for pathological lesions and it contains all-important data about the examined fragment of central nervous system. At the same time it ignores all spinal cord image details unimportant from the diagnostic point of view, as presented on figure 2a.

To give an example, the spinal cord MR image, as presented above in figure 2 will be subject to (on figure 3) a diagnostic description of pathological lesions detected in the spinal cord. Image 3 presents an example of results obtained by the author in the course of examinations for a given disease case. The results presented here have been achieved by the application of attribute grammar and they are an example of the cognitive approach to the medical data considered

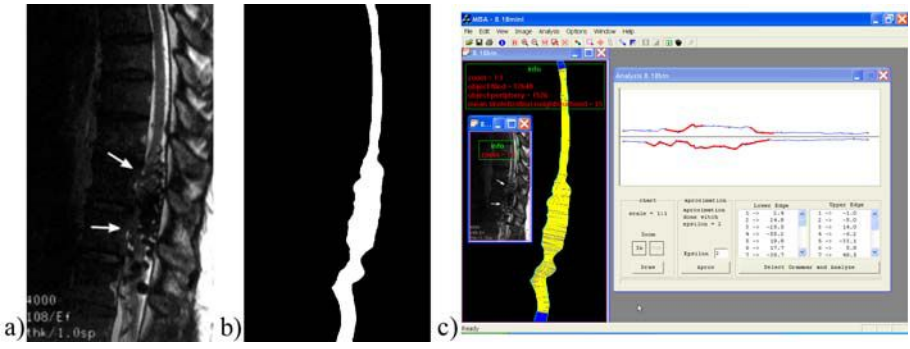


Fig. 2. Spinal cord: a) deformed, b) after binarisation, c) spinal cord width diagram

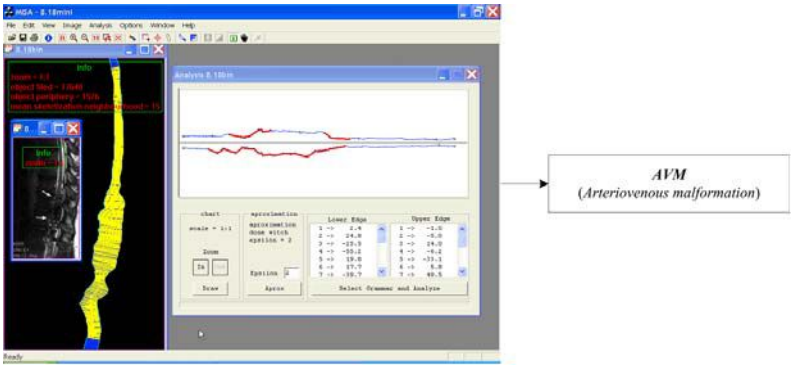


Fig. 3. Diagnostic description of spinal cord lesions with AVM syndrome detected as a result of cognitive analysis

here. The type of lesion detected here has been assigned based on its location and on morphometric parameters determined by the grammar semantic procedures.

The example above (and many others, obtained as a result of research [3]) present the results of semantic meaning interpretation of the analysed and detected pathological lesions occurring in the spinal cord.

5 Conclusions

In order to perform meaning analysis on spinal cord images with the use of a linguistic mechanism as described in this paper, the MISA (Medical Image Syntax Analyser) computer system has been developed. This enables the analysis and classification of spinal cord images analysed in this paper.

The application efficiency of cognitive analysis procedures, using this system, has been presented in a table and it is directed towards comparing the results obtained from the use of this system with those that one can consider as a correct diagnosis (table 2).

Table 2. The efficiency of cognitive analysis methods directed towards discovering and understanding selected disease phenomena in the central nervous system

The analysed disease lesion	Number of examined images	Number of correctly recognised images (lesions)	Cognitive analysis efficiency [%]
Spinal cord dilation	2	2	100
Cysts	18	17	94
Neoplastic tumours (extra- and intramedullary)	27	25	93
Stenoses (amputations) and spinal cord compression	14	12	86
Spinal cord degeneration	23	20	87
Total	84	76	90,5

These results are obtained as a result of the application of semantic analysis algorithms conducted in reasoning modules of the proposed system and based on semantic actions assigned to structural rules.

The research conducted by the author, based on the analysis of images with pathological lesions in a part of the central nervous system, the spinal cord, have demonstrated that cognitive data analysis can be a factor that significantly enriches the possibilities of contemporary information systems. In particular, the described research has demonstrated that an appropriately built image grammar enables the conduct of precise analysis and the description of medical images from which important semantic information can be gained on the nature of processes and pathological lesions as found in the patient's spinal cord. It is worth emphasising that the results described in this paper have been obtained following the cognitive process, simulating an experts' method of thinking: if one observes a deformation of the organ shown by the medical image used, then one tries to understand the pathological process that was the reason for the appearance of deformations found. One does not perform a mechanic classification for the purpose of pointing out more similar samples on the pathological image. Moreover, the research conducted has demonstrated that for cognitive analysis attempts (on the central nervous system) it is possible to apply it on sequential grammar- based linguistics.

Acknowledgement

This work was supported by the AGH University of Science and Technology under Grant No. 10.10.120.39.

References

1. Albus, J. S., Meystel, A. M.: Engineering of Mind - An Introduction to the Science of Intelligent Systems. A Wiley-Interscience Publication John Wiley & Sons Inc (2001)

2. Burgener, F. A., Meyers, S. P., Tan, R. K., Zaunbauer, W.: Differential Diagnosis in Magnetic Resonance Imaging Georg Thieme Verlag (2002)
3. Ogiela, L.: Usefulness assessment of cognitive analysis methods in selected IT systems. Ph. D. Thesis. AGH Kraków (2005)
4. Ogiela, M. R., Tadeusiewicz, R., Ogiela, L.: Inteligent Semantic Information Retrieval In Medical Pattern Cognitive Analysis. Computational Science and Its Applications - ICCSA 2005, **3483** 852–857
5. Tadeusiewicz, R., Ogiela, M. R.: Medical Image Understanding Technology, Artificial Intelligence and Soft-Computing for Image Understanding. Springer-Verlag Berling Heildelberg (2004)

Interpretability of Bayesian Decision Trees Induced from Trauma Data

Derek Partridge, Vitaly Schetinin, Dayou Li, Timothy J. Coats,
Jonathan E. Fieldsend, Wojtek J. Krzanowski,
Richard M. Everson, and Trevor C. Bailey

[1] School of Engineering, Computer Science and Mathematics, University of Exeter, EX4 4QF, UK
{D.Partridge, J.E.Fieldsend, W.J.Krzanowski, R.M.Everson, T.C.Bailey}@exeter.ac.uk
[2] Department of Computing and Information Systems, University of Luton, LU1 3JU, UK
{Vitaly.Schetinin, Dayou.Li}@luton.ac.uk
[3] Accident and Emergency Department, Leicester Royal Infirmary, LE1 5WW, UK
tc61@le.ac.uk

Abstract. Decision Tree (DT) technology with its sequential decision process composed of hard thresholds applied to individual features is often not an optimal choice. But DTs are typically credited with being predictive structures that a domain specialist can interpret as opposed to a black-box neural network, for example. There are two anticipated classes of benefit when Artificial Intelligence technology is applied to medicine: automation of decision making and enhancement of medical knowledge. In this paper we present the use of Bayes-ian averaging as a principled approach to optimal classifier systems using DT technology in which a confidence rating can be associated with every predicted result. However, averaging over an ensemble of DTs causes the problem that such an ensemble becomes uninterpretable. Thus we also present a procedure for extracting interpretable archetype DTs. We demonstrate these innovations by application to Trauma data.

1 Introduction

A ubiquitous paradigm is the need to make decisions such as clinical diagnoses or prognoses on the basis of certain observable attributes. This paradigm can be cast as a classification problem: sets of specific attribute values map individually to one of the possible classes of interest. Thus a set of attribute or feature values such as age, weight, blood pressure may be mapped to the classes hypertension or not-hypertension.

When the desired mappings cannot be analytically formulated, they are derived from examples. Statistics and Artificial Intelligence (AI) offer a variety of techniques for deriving a mapping exemplified by a set of examples [1], [2], [3], [4], [5].

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 972–981, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Apart from the difficulties that naturally arise from the conjunction of sensitive, potentially life-threatening, medical decisions and computer technology, the sought-after mappings are seldom simply and unambiguously present in the set of examples. There is a variety of reasons for this: natural person-to-person variability; the desired classification is vitally dependent upon unacknowledged feature values; the classification associated with each set of feature values is often equivocal; the feature values themselves are unlikely to be uncorrupted or even measured consistently over the period of time needed to amass enough data samples; some feature values will be expert assessments and these will vary with both time and individual experts.

Decision trees (DTs) [1], [2], [3], [4], [5] are a prime example of an inductive AI technology that also offers interpretability to the user, and in this paper we will examine this claim as well as the use of DTs as powerful medical decision-making systems in their own right. We will explore these aspects in the specific medical context triage cast in terms of the above-described paradigm: will a given person (age, weight, etc.) survive certain injuries? The sets of feature values map to the categories lived or died which will be used, in the context of an automated system, to predict likely survivability as a result of traumatic injury [6]. But in the context of medical knowledge about decision making in the face of traumatic injury we would hope to learn about how to refine and optimize human decision making, e.g. what are the important features, and what is the simplest, most powerful way to combine them in order to make a fast and accurate prediction?

In general, extracting the desired mapping from the sample data can be viewed as one of fitting a mathematical model to the sample data. The mathematical models might be neural networks such as multi-layer perceptrons (MLP) [5]. By fitting a model to the data we mean tuning the variables, or parameters, of the particular family of models chosen in order to optimize the performance of the trained model. The tuning is achieved using a training set of data samples and a learning algorithm, the goodness of fit achieved is typically measured by the accuracy of the tuned model on a test set of data samples.

This straightforward view of model fitting yields a single model, say, an MLP with a specific parameters values set. It is most likely that other parameter settings in the MLP family of models will be similarly optimal, perhaps with respect different regions of the general mapping sought. Thus, two different optimal models might be identically accurate in terms of the percentage of test samples correctly classified, but on non-identical subsets of the test samples. Additionally, as optimality of a specific fit is judged by test-set accuracy, a different specific fit might prove optimal on a different test set.

One way around this awkward problem of reliance on a potentially suboptimal spe-cific fit is to work with an average classification generated by a collection of differ-ently optimized models [5], [7], [8]. The blossoming field of Multi-Classifier Systems (MCS) can be viewed as one rich and varied attack on the problem of suboptimal models [9], but a relatively new and principled approach is through Bayesian averaging.

2 Bayesian Model Averaging

Bayes rule has long been known and used in limited ways as a basis for constructing classifier systems [5], [7], [8]. Full implementation of Bayesian systems has been hampered by the need to compute certain integrals that are analytically intractable. But now with advances in the computational power readily available coupled with appropriate progress in Markov Chain Monte Carlo (MCMC) methods, accurate approximation of the Bayesian integrals has become a practical reality [7], [8]. By this means, we are able to integrate out of the classifier system any dependence on specific settings of the model parameters, and by following Bayesian principles we can be assured that our results are truly optimal within the constraints of the model chosen and the data set available. In addition, we use the set of classification results (from which our Bayesian average will be calculated) as a basis for computing a confidence rating for the averaged result produced.

A state of the art Reversible Jump (RJ) MCMC method [10] applied to an appropriate formulation of Bayesian DTs permits a computationally expensive but efficient selection of optimal specific DT models from within the range of all DT models of the type specified (in our case binary DTs). Full details of this procedure have been presented elsewhere [7], [8], and for current purposes we need only consider the specific strategies that we imposed to facilitate this investigation and the nature of the DTs generated.

During sampling from the Markov Chain, the parameters of the chosen classifier model (binary DTs with k terminal nodes to be used for classification) are drawn from the given proposal distributions in order to provide a new candidate model [7], [8], [13]. The candidate is accepted or rejected according to a Bayes rule calculated on the given data $\mathbf{D}$. Thus, for the m-dimensional input vector $\mathbf{x}$, data $\mathbf{D}$ and parameters θ, the class posterior distribution $p(y|\mathbf{x}, \mathbf{D})$ is

$$p(y|\mathbf{x}, \mathbf{D}) = \int \mathbf{p}(\mathbf{y}|\mathbf{x}, \theta, \mathbf{D})\mathbf{p}(\theta|\mathbf{D})\mathbf{d}\theta \approx \frac{\mathbf{1}}{\mathbf{N}}\mathbf{\Sigma}_{\mathbf{i=1}}^{\mathbf{N}}\mathbf{p}(\mathbf{y}|\mathbf{x}, \theta^{\mathbf{i}}, \mathbf{D}) \tag{1}$$

where $p(\theta|\mathbf{D})$ is the posterior distribution of parameters θ conditioned on data $\mathbf{D}$, and N is the number of samples θ^i taken from the posterior distribution.

3 Averaging and Interpretability of Optimal DTs

While the Bayesian basis for our procedures guarantees the optimality of the averaged results, the concept of the average DT, which would presumably be the DT (and hence the explanation) underlying the optimal system performance, is non-simple. What DT is the average of a set of DTs? As the question appears to have no sensible answer, we face a challenge if we wish to use the power of the Bayesian average approach and also capitalize on the interpretability of DTs [11,12].

We propose and illustrate a strategy for extracting readily interpretable and maximally informative DTs from the population, the MCS, which constitutes

the optimal average. We call these archetypal DTs (ADT), i.e., DTs that cover most of the training examples classified as confident and correct.

Clearly the classification confidence is maximal, equal to 1.0, if all the classifiers assign a given input to the same class, otherwise the confidence is less than 1.0. The minimal value of confidence is equal to $1/C$ if the classifiers assign the input datum to the C classes in equal proportions. So for a given input $\mathbf{x}$ and class c, the classification confidence is the population χ of classifiers that assign $\mathbf{x}$ to c. We can define a given level of the classification confidence, $\chi_0 : 1/C \leq \chi_0 \leq 1$, for which the cost of misclassification is small enough to be accepted. Then for the given input, the outcome of the MCS is said to be $confident$ if the ratio $\chi \geq \chi_0$. Clearly, on the labeled data we can distinguish between $confidentandcorrect$ outcomes and $confidentbutincorrect$ outcomes [13]. These latter outcomes of the MCS may appear due to noise or overlapping classes in the data.

The task is to find DTs that cover a maximal number of the training samples classified as confident and correct while the number of misclassifications on the remaining samples is minimal. To find such a DT set, we can remove the conflicting examples from the training data and then select the DTs with a maximal cover of the training samples classified by the DT ensemble as confident and correct.

Thus the main steps of the archetype selection procedure are as follows:

> Amongst a given Bayesian DT ensemble find a set of DTs, S1, which cover a maximal number of the training samples classified as confident and correct with a given confidence level χ_0.
> Find the training samples which were misclassified by the Bayesian DT ensemble and then remove them from the training data. Denote the remaining training samples as D1.
> Amongst the set S1 of DTs find those which provide a minimal misclassification rate on the data D1. Denote the set found as S2.
> Amongst the set S2 of DTs select those whose size is minimal. Denote the set of such DTs as S3. The set S3 contains the desired DTs.

The above procedure finds one or more DTs and puts them in the set S3. These DTs cover a maximal number of the training samples classified as confident and correct with a given confident level χ_0. The size of these DTs is minimal and any of them can be finally selected for interpreting the confident classification.

4 Application to the Trauma Data

This section describes the experimental results obtained by applying the proposed above technique on the Trauma data collected in the Royal London Hospital.

4.1 The Trauma Data

The Trauma data used in our experiments consist of 316 labeled examples of difficult cases for the clinicians deciding on the survival rate. These data contain

Table 1. The features of Trauma data

Variable	Name	Type
x_1	Age	Continuous
x_2	Gender	(0, 1)
x_3	Injury	(0, 1)
x_4	Head injury	(0, 6)
x_5	Facial injury	(0, 4)
x_6	Chest injury	(0, 6)
x_7	Abdominal injury	(0, 5)
x_8	Limbs	(0, 5)
x_9	External injury	(0, 4)
x_{10}	Respiration rate	Continuous
x_{11}	Systolic BP	Continuous
x_{12}	GCS eye	(0, 4)
x_{13}	GCS motor	(0, 6)
x_{14}	GCS verbal	(0, 5)
x_{15}	Oximetry	Continuous
x_{16}	Heart rate	Continuous

the 16 features listed in Table 1. This table lists such features as Age, Respiration rate, Systolic Blood Pressure (BP), Oximetry (%) and Heart rate, which are continuous; the remaining features, such as Glasgow Coma Score (GCS), are nominal. 210 data points randomly selected from the original data form a training dataset and the remaining 106 form a test dataset. The survival rates are 0.47 and 0.56 for the training and test datasets, respectively.

Using our Bayesian averaging over 5000 DTs, we obtained a misclassification rate of 13.2 *by applying the procedure described in section 3 we found the set S3 which contains 21 DT involving a minimal number (9) of splitting nodes. From these 21, 10 DTs misclassified 14.1but only by 0.94can obtain a minimal misclassification rate equal to* 13.21

The resultant ADT is presented in Figure 1. Each interior, or splitting, node provides a specific question that has a YES/NO answer, and two branches. The DTs are depicted with the positive response to the specific question corresponding to the branch that is labeled with the question; the negative response corresponds to the other branch.

4.2 Interpretability of the Archetypal DTs

As an explanation of the Trauma decision process, the DT was judged to be biologi-cally plausible and a general fit with what would be expected from a clinical perspec-tive. It seems to be picking out brain injury (Head score and GCS), bleeding (Systolic BP and heart rate) and preexisting physiological reserve (age) as important factors.

The main causes of death after injury are brain damage and bleeding. The early stage of the DT seems to be saying: if you have a severe head injury, it does not matter whether you are bleeding (reflected in physiological disturbance) or

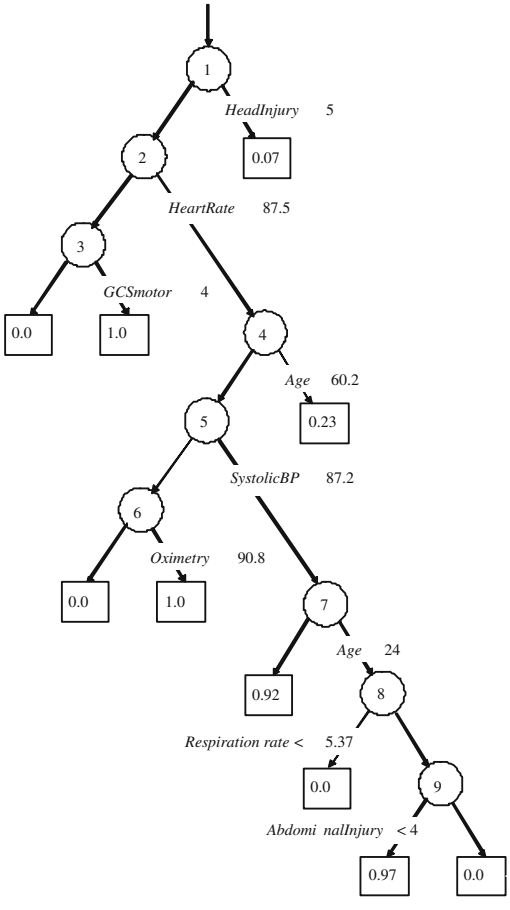


Fig. 1. An ADT with 13.2% of misclassification rate on the test data

not, you are likely to die. If you have not got a severe head injury, the amount of physiological distur-bance (bleeding or respiratory distress) and your capacity to respond to that disturbance becomes important. Head injury, the first splitting-node decision, fits with what we know about brain injury being a huge influence on the patients outcome: even if you stop the bleeding the patient will still die. It is interesting that there is a second group of patients that have a head injury score of less than 5 and a normal heart rate where GCS motor response becomes important. This decision structure is suggestive of hypoxic brain injury, and this hypothesis could be further explored by reference to the case notes of the individuals concerned.

With respect to Figure 1, we note that the way the two age nodes are used is very in-teresting because current injury models [6] use only one with a cutoff at 55 years. We know that the extremes of age are very different in almost all areas of medicine, so the fact that there are two decision points, one for old and one

for young, fits well with such preconceptions. The slight surprise is that there is not a younger cutoff, because preteen children have better outcomes, but there may be too few cases in this age range for the modeling process to identify this effect in the available dataset.

Overall, it is interesting that little seems to be contributed by the anatomical input features (i.e. the Injury Severity Scores for each body area), apart from head injury. The nationwide cost of this must run into millions of pounds, and yet this DT appears to tell us that we need only look at the head and abdomen. Currently, 150 hospitals each spend about 10000 annually on data collection - in salaries. The system will soon be enlarged to all 300 hospitals in the UK. This will result in a total cost to the NHS of about 3 million each year, so decreasing the amount of data that is required could make significant financial savings.

4.3 The Feature Importance

In Figure 2, we can see that such features as Age (x_1), Head injury (x_4), Abdominal injury (x_7), Systolic BP (x_{11}), and GCS motor (x_{13}) are used in the Bayesian DTs on average more frequently than the others. In contrast, feature External injury (x_9) is used with a less frequency.

Additionally, we can see from the error bars that the posterior weights of some features (e.g., for Head injury, x_4, Chest injury, x_6, and Heart rate, x_{16}), have a high variance. Such wide deviations may be caused by variations in the training examples within the 5 fold cross-validation.

Trauma care is an area of medicine where there is an existing predictive model and the factors influencing survival are relatively well understood [6]. The set of features ranked on their posterior weights as shown Figure 2 reveals a good correlation with those factors that clinicians regard as important for their patients. Brain injury does not heal, so outcome is directly and strongly related to the extent of brain injury, seen in the high rankings for head injury and GCS motor (which is known to be the most relable of the three components of the GCS). The ability of the body to cope with injury is directly related to age (for example, there is a rough rule of thumb in burns patients that if the percentage body area burnt plus age exceeds 100, the patient will die). The importance of age is seen in its high rank.

5 The Comparison of Performances

In this section we compare our technique of extracting an ADT with the Bayesian averaging (BA) and the maximum a posteriori (MAP) techniques described in [12]. The comparison is made in terms of misclassification within 5 fold cross-validation on the real-world medical data sets known as Pima and Wisconsin [14] as well as the Trauma data. All these data sets are 2-class problems. The numbers of the labeled examples in the Pima, Wisconsin, and Trauma data sets were 768, 683, and 316, while the numbers of features were 8, 9, and 16, respectively.

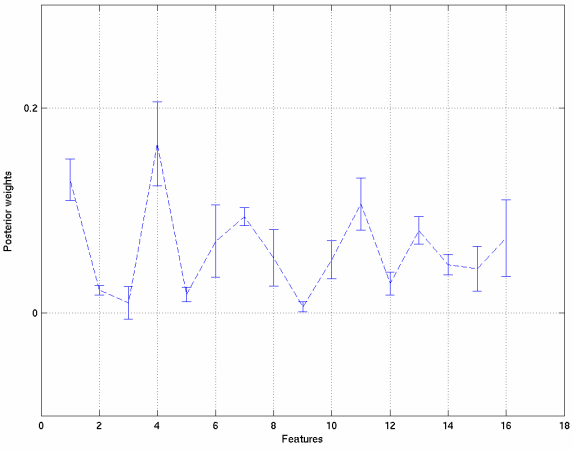


Fig. 2. Posterior weights of the features averaged within 5 fold cross-validation

Table 2. The misclassification rates on the test datasets within 5 fold cross-validation

Techniques	Pima	Wisconsin	Trauma
BA	27.7 ± 6.5	3.8 ± 4.5	13.1 ± 7.5
ADT	26.7 ± 7.5	3.2 ± 3.9	14.1 ± 7.7
MAP	27.8 ± 8.4	3.9 ± 2.8	18.5 ± 2.6

Table 3. The sizes of DTs within 5 fold cross-validation

Techniques	Pima	Wisconsin	Trauma
BA	17.8 ± 1.1	11.3 ± 1.8	12.5 ± 1.0
ADT	17.6 ± 3.9	9.6 ± 6.6	10.6 ± 3.6
MAP	21.7 ± 2.4	14.2 ± 6.5	16.8 ± 1.7

The Bayesian DT ensemble technique ran with the pruning factor set equal to 5. The number of burn-in and post burn-in samples, and sampling rates, were set to 10000, 5000, and 7, respectively. The proposal probabilities for birth, death, change-split, and change-rule were set to 0.1, 0.1, 0.2, and 0.6, respectively. The proposal distribution for the change moves was Gaussian with mean 0 and variance 1.0.

The misclassification rates of the above three techniques, BA, ADT, and MAP, are shown in Table 2.

Clearly, in the theory, the BA technique should provide fewer misclassifications than the ADT and MAP techniques. In our experiments, however, we can observe that all these techniques have nearly the same misclassification rates within 5 fold cross-validation. Nevertheless, comparing the average rates of misclassification, we can see that the BA and ADT techniques slightly outperform the MAP

technique. At the same time, the average misclassification rates of the BA and ADT techniques are almost the same.

Table 3 shows the sizes of DTs induced in our experiments within 5 fold cross-validation. From this table we can certainly conclude that the ADT technique provides shorter DTs than the MAP technique.

6 Conclusions

Decision trees, particularly when set within a framework of Bayesian averaging, prove to be powerful automatic classification systems which in the Trauma domain, at least, outperform the traditional automatic decision structures in terms of classification uncertainty [13]. In addition, a Bayesian averaging approach offers the possibility of an estimate of the confidence to be attached to every prediction.

However, perhaps more importantly DT classifiers are said to be preferred (in con-trast to, say, neural net classifiers) because they are interpretable, and this property will facilitate the use of DT classifiers to extract useful knowledge about the optimal decision processes within the application domain. The biological plausibility of DTs may well be more acceptable to clinicians than a black box.

Because an average DT does not appear to be a sensible concept with respect to interpretability of the optimal decision processes, a selection procedure for extracting ADTs from Bayesian-averaged collections was proposed and demonstrated. The selected tree was judged to be a useful explanation of the Trauma decision process. Ojective evidence for useful explanatory power was provided in terms of both a subse-quent focus of attention on specific input features (e.g. the age cutoff) that resulted in the extraction of new knowledge about the role of this feature, and objective confirmations of the roles of certain features, e.g., age as a reservoir of capacity to survive, and lack of useful value in certain injury classifications.

Acknowledgment

The work reported was largely supported by a grant from the EPSRC under the Critical Systems Program, grant GR/R24357/01.

References

1. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Bel-mont, Wadsworth (1984)
2. Galant, S.: Neural Network Learning and Expert Systems. MIT Press (1993)
3. Buntine, W.: Learning Classification Trees. Statistics and Computing 2 (1992) 63-73
4. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
5. Duda, R.O., Hart, P.E.: Pattern Classification. Wiley Interscience (2001)

6. Boyd, C.R., Tolson, M.A., Copes, W.S., Evaluating Trauma Care: The TRISS Method. J. Trauma 27 (1988) 370378
7. Chipman, H., George, E., McCullock, R.: Bayesian CART Model Search. J. American Statistics 93 (1998) 935-960
8. Denison, D., Holmes, C., Malick, B., Smith, A.: Bayesian Methods for Nonlinear Classifica-tion and Regression. Wiley (2002)
9. Kuncheva, A.: Combining Pattern Classifiers: Methods and Algorithms. Wiley (2004)
10. Green, P.: Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination. Biometrika 82 (1995) 711-732
11. Chipman, H., George, E., McCulloch, R.: Making Sense of a Forest of Trees. In: Weisberg, S., (ed.): Proceedings of the 30th Symposium on the Interface, Interface Foundation of North America (1998) 84-92
12. Domingos, P.: Knowledge Discovery via Multiple Models. Intelligent Data Analysis 2 (1998) 187-202
13. Schetinin, V., Partridge, D., Krzanowski, W.J., Everson, R.M., Fieldsend, J.E., Bailey, T.C., Hernandez, A.: Experimental Comparison of Classification Uncertainty for Randomized and Bayesian Decision Tree Ensembles. J. Math. Modeling and Algorithms 4 (2006) forthcoming
14. Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Datasets. www.ics.uci.edu/ mlearn/MLRepository. Irvine, University of California (1998)

The Greatest and the Least Eigen Fuzzy Sets in Evaluation of the Drug Effectiveness Levels

Elisabeth Rakus-Andersson

Blekinge Institute of Technology, School of Engineering, Study of Mathematics,
S-37179 Karlskrona, Sweden
Elisabeth.Andersson@bth.se

Abstract. The eigen fuzzy set of a given fuzzy relation often corresponds to an occurrence of invariability in natural sciences. By determining the fuzzy relations as connections between pairs of symptoms we utilize the greatest and the least eigen fuzzy sets in order to find the estimates of the medicine effectiveness levels.

1 Introduction

The existence of the greatest eigen fuzzy set of a fuzzy relation was confirmed in the 80-ties of the twentieth century [4, 7, 8, 9]. In the latest investigations the scientists have proved that even the least eigen fuzzy set can be generated for the given relation [1, 2, 5]. The eigen fuzzy sets have already been applied to the evaluation of medicine action levels when considering the medicine influence on clinical symptoms [3, 6].

We continue the last item by accomplishing an own proof of the existence of the least set especially, which differs from the conceptions formulated in [1, 2, 5]. The theoretical discussion, which concerns eigen sets, constitutes the contents of Section 2. In Section 3 we introduce the medical problem that involves applications of eigen fuzzy sets. Finally, a simple medical exercise is solved in Section 4 to give an image of the functional utility of the presented model.

2 Theoretical Assumptions of Eigen Fuzzy Problem

A particular result of a relation composition is known as the eigen set of a fuzzy relation [7, 8].

Assume that $X = \{x_1,...,x_n\}$ is a finite set of real numbers. The eigen fuzzy set of the fuzzy relation $R \subseteq X \times X$ is a set $A \subseteq X$, which satisfies $R \circ A = A$.

R is the fuzzy relation determined as $R \subseteq X \times X$ with the membership function $\mu_R(x,x') : X \times X \to [0,1]$, $x, x' \in X$. It is proved that the eigen fuzzy set $A \subseteq X$, $\mu_A(x) : X \to [0,1]$, which is a part of the equation $A \circ R = A$, exists [7, 8, 9].

We define the set A_0 with $\mu_{A_0}(x) = a_0$, $a_0 = \min_{x' \in X}(\max_{x \in X} \mu_R(x,x'))$ for all $x \in X$.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 982 – 987, 2006.
© Springer-Verlag Berlin Heidelberg 2006

The fuzzy connection $A_0 \circ R = A_0$ is a true equality because of $\mu_{A_0 \circ R}(x') = \max_x(\min(\mu_{A_0}(x), \mu_R(x,x')) = \max_x(\min(a_0, \mu_R(x,x')) = \min(a_0, \max_x \mu_R(x,x')) = a_0 = \mu_{A_0}(x')$, $x, x' \in X$. Hence, A_0 is an eigen fuzzy set of R.

The next introduced set A_1 is identified by its membership function given by

$$\mu_{A_1}(x') = \max_{x \in X} \mu_R(x,x') \tag{1}$$

for all $x' \in X$.

The fuzzy sets, which are members of the sequence $(A_n)_n$, in which

$$A_2 = A_1 \circ R = A_1 \circ R^1,\ A_3 = A_2 \circ R = A_1 \circ R^2, \cdots, A_{n+1} = A_n \circ R = A_1 \circ R^n, \tag{2}$$

exist for all integers $n>0$.

The sets satisfy the inclusions

$$A_0 \subseteq \cdots \subseteq A_{n+1} \subseteq A_n \subseteq \cdots \subseteq A_2 \subseteq A_1. \tag{3}$$

To prove (3) we apply the mathematical induction. On the basis of the definition of A_0 we conclude that $A_0 \subseteq A_1$ since $\mu_{A_0}(x') = \min_{x'}(\max_x \mu_R(x,x')) \le \max_x \mu_R(x,x')) = \mu_{A_1}(x')$. We deduce that even $A_2 \subseteq A_1$. By conveying, for every $x` \in X$, that $\mu_{A_2}(x') = \mu_{A_1 \circ R}(x') = \max_{x \in X}(\min(\mu_{A_1}(x), \mu_R(x,x')) \le \max_{x \in X} \mu_R(x,x') = \mu_{A_1}(x')$, we state that $A_2 \subseteq A_1$.

We shall now prove that the assumption $A_n \subseteq A_{n-1}$ induces the conclusion $A_{n+1} \subseteq A_n$, $n \in N$, since $A_n \subseteq A_{n-1} \to A_n \circ R \subseteq A_{n-1} \circ R \leftrightarrow A_{n+1} \subseteq A_n$.

The set A_0 is the eigen set of R. A_1, the other set proposed by (1), rarely is a solution of the restriction $A_1 \circ R = A_1$. If $A_n \circ R = A_n$, for A_n being a member of the sequence of sets given by (2), then we will allege that A_n is the expected greatest eigen set of the relation R, which often differs from A_0. The set A_0 is the least set in the chain of sets in (2) and all sets included between A_1 and A_n are not eigen.

Suppose that $A_0 \ne A_{n+k} = \cdots = A_{n+1} = A_n \ne \cdots \ne A_2 \ne A_1$, then the composition $A_n \circ R$ leads to $A_n \circ R = A_1 \circ R^{n-1} \circ R = A_1 \circ R^n = A_{n+1} = A_n$.

A_n is thus the greatest eigen fuzzy set (GEFS) of R provided that $A_n = A_{n+1}$.

We recall that membership degrees of A_{n+1} are calculated as

$$\mu_{A_{n+1}}(x') = \mu_{A_n \circ R}(x') = \max_{x \in X}(\min(\mu_{A_n}(x), \mu_R(x,x')) \tag{4}$$

for each $x' \in X$.

It can be desirable to find the smallest eigen fuzzy set of a given fuzzy relation as well. In spite of some accomplished investigations of the topic [1, 2] let us propose the own contribution as the following proof of the least eigen set existence.

We preserve the unchanged set A_0 with $\mu_{A_0}(x) = a_0$ for all $x \in X$, where $a_0 = \min_{x' \in X}(\max_{x \in X} \mu_R(x,x'))$. A_0 is the eigen set of R as it has been proved before.

The set A_1 gets now new membership degrees determined as

$$\mu_{A_1}(x') = \min_{x \in X} \mu_R(x, x') \tag{5}$$

for all $x' \in X$.

We propose the same sequence of fuzzy sets $(A_n)_n$, $A_2 = A_1 \circ R = A_1 \circ R^1$, $A_3 = A_2 \circ R = A_1 \circ R^2$, $\cdots, A_{n+1} = A_n \circ R = A_1 \circ R^n$, which fulfill inclusions

$$A_1 \subseteq A_2 \subseteq \cdots \subseteq A_n \subseteq A_{n+1} \subseteq \cdots \subseteq A_0. \tag{6}$$

The boundary inclusion $A_1 \subseteq A_0$ in the chain is true because of the inequality $\mu_{A_0}(x') = \min_{x'}(\max_x \mu_R(x, x')) \geq \min_x \mu_R(x, x')) = \mu_{A_1}(x')$.

To confirm the reliability of other inclusions in (6) we return to the assumptions of the mathematical induction. In order to check that $A_1 \subseteq A_2$ we thus notice that $\mu_{A_2}(x') = \mu_{A_1 \circ R}(x') = \max_{x \in X}(\min(\mu_{A_1}(x), \mu_R(x, x')) \geq \min_{x \in X} \mu_R(x, x') = \mu_{A_1}(x')$, $x' \in X$.

The last structure is equivalent to $A_1 \subseteq A_2$ since $\mu_{A_1}(x') \leq \mu_{A_2}(x')$.

The induction assumption $A_{n-1} \subseteq A_n$ is used in the proof to obtain the conclusion $A_n \subseteq A_{n+1}$. We start with $A_{n-1} \subseteq A_n$ to compose with R both sides of the inclusion as $A_{n-1} \circ R \subseteq A_n \circ R$, which is comparable to $A_n \subseteq A_{n+1}$.

The set A_0 is the eigen set of R but A_1 seldom is regarded as eigen. Let us assume that A_n is a member of the sequence listed in (6) and that it fulfils $A_n \circ R = A_n$ for $A_1 \neq A_2 \neq \cdots \neq A_n = A_{n+1} = \cdots = A_{n+k} \neq A_0$. Then A_n will be the least eigen set (LEFS) of the relation R, which is different from A_0.

In order to evaluate GEFS and LEFS we adopt a procedure, which consists of the following steps:

Algorithm 1

1. Find the set A_1,
2. Set the index n=1,
3. Calculate $A_{n+1} = A_n \circ R$,
4. $A_{n+1} \overset{?}{=} A_{n \to Yes \to A = A_{n+1},}^{\to No \to n = n+1 \to \text{Go to step 3}}$

in which A_1 is computed either by the application of (1) or by the choice of (5).

The relation R keeps the given fuzzy set invariant, which apparently fits to a medical appearance when a medicine has no more effect in the curative process.

3 Eigen Sets in Effectiveness Levels of Drugs

Let us assume that characteristic qualitative symptoms of a morbid unit are found in a sample of patients. After the treatment some symptoms should disappear entirely while the other symptoms are still present.

Let us denote a non-fuzzy set of symptoms by $S = \{S_1, \ldots, S_n\}$.

An estimation of the maximal level is possible by employing a fuzzy relation $R_{\max}$, which is created according to the formulation: "The action of the drug on the j^{th} symptom is equal or stronger than on the k^{th} one, $j, k = 1, \ldots, n$". The membership degree $\mu_{R_{\max}}(S_j, S_k)$ indicates the strength of the relationship between the j^{th} and the k^{th} symptom.

If m and p denote the number of examined patients respectively the number of patients who fit for the definition of $R_{\max}$ when comparing the medication effects for S_j and S_k, then we will compute the membership degrees $\mu_{R_{\max}}(S_j, S_k)$ as [3]

$$\mu_{R_{\max}}(S_j, S_k) = \frac{p}{m} \tag{7}$$

for $j, k = 1, \ldots, n$.

Suppose that "–" is assigned to the lack of a symptom after the treatment and "+" designates its presence in patient after the medication. The sign patterns "– –" and "– +", counted with respect to the pair S_j, S_k in the group of m patients help us to appreciate p. For the pairs (S_j, S_j), $j = 1, \ldots, n$, the value of p is computed as a number of recoveries from S_j.

The relation $R_{\max}$ has the greatest eigen fuzzy set $A_{\max}$, which is defined in the universe S as

$$A_{\max} \circ R_{\max} = A_{\max}. \tag{8}$$

$A_{\max}$ is found as the result of Algorithm 1, in which A_1 is defined by (1). The relation, designed in accordance with the statement: "The drug acts equally strongly or more strongly on the j^{th} symptom than on the k^{th} one" has its eigen set as an unchangeable component of the equation (8). We thus conclude that membership degrees of $A_{\max}$ show the level "the drug action on the considered symptoms is not stronger". Moreover, we are able to accept this level as optimal since $A_{\max}$ is the greatest solution of (8) in the sense of the greatest membership degree values.

An estimation of the minimal medicine effect is connected with forming another fuzzy relation $R_{\min}$ proposed as a clue:"The action of the drug on the j^{th} symptom is equal to or weaker than on the k^{th} one, $j, k = 1, \ldots, n$." The suggested formula of calculating membership degrees of $R_{\min}$ is Eq. (7) to which the sign configurations "– –" and "+ –" are attached.

The relation $R_{\min}$ also generates its own, this time the least, eigen fuzzy set $A_{\min}$ that constitutes a compound of an equation

$$A_{\min} \circ R_{\min} = A_{\min} \tag{9}$$

To decide $A_{\min}$ we perform the steps of Algorithm 1, which includes A_1 computed by the action of (5).

$A_{\min}$ does not change its membership degrees after the next composition with $R_{\min}$: "The drug affects equally or more weakly the j^{th} symptom compared to the k^{th} one". Then membership degrees of the least eigen set, associated with symptoms $S_1, \ldots, S_n$, indicate the minimal level of the medicine effectiveness. $A_{\min}$, as the least eigen set of

$R_{\min}$, provides us with the statement "the action of the medicine on the considered symptoms cannot be weaker".

The values of $\mu_{A_{\min}}(S_j)$ and $\mu_{A_{\max}}(S_j)$, $j = 1, \ldots, n$, constitute the borders of an interval, which is treated as the range of the medicine effectiveness for each symptom S_j. This should help us in making the judgement of the tested drug usability.

4 The Medical Example

The diagnosis D known as a throat inflammation is accompanied by the set of symptoms $S = \{S_1 =$ "*sore throat* (*pain*)", $S_2 =$ "*temperature*", $S_3 =$ "*inflammation state*"$\}$. The physician has prescribed Bayer's aspirin as a remedy that should improve the health conditions in the group of 30 patients suffering from throat inflammation.

The application of (7) with the sign pattern "– –" and "– +" gives $A_{\max}$ as

$$R_{\max} = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.8 & 0.8 & 0.8 \\ 0.6 & 0.6 & 0.6 \end{bmatrix} \end{array}$$

which has the greatest eigen fuzzy set decided as

$$A_{\max} = \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{bmatrix} 0.8 & 0.8 & 0.8 \end{bmatrix}. \end{array}$$

Equation (7), in which numbers of the associations "– –" and "+ –" constitute the basis of the p value computations, result in the relation $A_{\min}$ yielded as

$$R_{\min} = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{bmatrix} 0.5 & 0.8 & 0.6 \\ 0.5 & 0.8 & 0.6 \\ 0.5 & 0.8 & 0.6 \end{bmatrix} \end{array}$$

which possesses the least eigen fuzzy set

$$A_{\min} = \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{bmatrix} 0.5 & 0.8 & 0.6 \end{bmatrix}. \end{array}$$

By interpreting the membership degrees of $A_{\min}$ and $A_{\max}$ in the percentage scale we conclude that Bayer's aspirin removes S_1 in 50%–80% and S_2 – in 80%, while S_3 disappears for 60%–80% of the sample of patients.

5 Conclusions

As a counterpart of the discussion on the least eigen fuzzy set, we expand the own conception of the proof to confirm that the least eigen set exists.

The greatest and the least sets of fuzzy relations have been employed to approximate the optimal level of a medicine efficacious power. We have obtained the medicine action intervals evaluated for each symptom.

In constructing the relations we regard pairs of symptoms to learn about their influence on each other. Even if we appreciate effectiveness levels for individual symptoms, we will be aware of the complex dependency among symptoms, which influences single ranges. This aspect of complexity is an advantage of fuzzy research when comparing fuzzy results to computations of statistical ranges that do not consider interactions among the examined objects.

References

1. Bede, B., Nobuhara, H., Hirota, K.: Numerical Computation of Eigen Fuzzy Sets and Applications to Image Analysis. In: Dzitac, I., Maghiar, T., Popescu C. (eds.): Proc. of the International Conference on Computers and Communications, ICCC 2004. Univ. din Oradea, Romania (2004) 72-77
2. Fernández, M., Suárez, F., Gil, P.: T-eigen Fuzzy Sets, Inf. Sci. 75 (1993) 63-80
3. Gerstenkorn, T., Rakus, E.: An Application of Fuzzy Set Theory to Differentiating the Effectiveness of Drugs in Treatment of Inflammation of Genital Organs. Fuzzy Sets and Systems 68 (1994) 327-333
4. Goetschel, R., Voxman, W.: Eigen Fuzzy Number Sets. Fuzzy Sets and Systems 16 (1985) 75-85
5. Jacas, J., Recasens, J.: Fuzzy T-transitive Relations: Eigenvectors and Generators. Fuzzy Sets and Systems 72 (1995) 147-154
6. Rakus-Andersson E.: An Application of Fuzzy Numbers in Eigen Fuzzy Set Problem to Differentiating the Effectiveness of Drugs. In: Yinming Liu, Guoqing Chen (eds.): "Proc. of the International Conference on Fuzzy Information Processing – Theories and Applications. Tsinghua University Press – Springer (2003) 85-90
7. Sanchez, E.: Resolution of Eigen Fuzzy Set Equations. Fuzzy Sets and Systems 1 (1978) 69-74
8. Sanchez, E.: Eigen Fuzzy Sets and Fuzzy Relations. Journal of Mathematical Analysis and Applications 81 (1981) 399-421
9. Wagenknecht, M., Hartmann, K.: On the Construction of Fuzzy Eigen Solutions in Given Regions. Fuzzy Sets and Systems 20 (1986) 55-65

Cardiac Ventricle Contour Reconstruction in Ultrasonographic Images Using Bayesian Constrained Spectral Method

Tomasz Sołtysiński, Krzysztof Kałużynski, and Tadeusz Pałko

Institute of Precision and Biomedical Engineering, Department of Mechatronics, Warsaw University of Technology, Św. Andrzeja Boboli 8, 02-525 Warsaw, Poland
solek@mchtr.pw.edu.pl

Abstract. We propose a novel method of active spectral segmentation of cardiac ventricle by application of Bayesian inference adapted to contour detection in speckled ultrasonographic cardiological data. We discuss the advantages and limitations of this method and describe possible further developments.

1 Introduction

Ultrasonography is the most commonly used imaging modality in cardiology. The delineation of cardiac walls and cavities is important from the point of view of analysis of ventricle contractions. More recently, with the advent of elastography and cardiac strain-rate imaging, this delineation is even more of interest [2,3,4,5,7]. However, the properties of ultrasonic cardiac tissue and blood images sometimes make the delineation of cardiac cavities difficult [10]. There is a need for new methods resulting in better ventricle contour reconstruction. Below presented is a new approach to ventricle contour detection using an application of spectral method similar to that proposed by Li and Hero [9] and driven by Bayesian constraint [8]. The preliminary implementation of proposed method has been already successfuly applied to noisy CT data of the brain with aneurysm [12].

2 Methods

2.1 Application of PDE's in Shape Reconstruction

Let $g = g(\theta)$ be a noisy radial function defined in 2D spherical coordinates. The function g is called the polar edge map and can be estimated by a rough object edge detection method like basic thresholding or watershed algorithm. The method is valid only for star-like objects with circumference described by g. To find function $f(\theta)$, the smooth representation of g, serving in fact as an approximation of the desired active shape, we need to apply a method directly revealing f by minimizing the energy functional:

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 988–997, 2006.
© Springer-Verlag Berlin Heidelberg 2006

$$E(f,g) = \mu \int_S Y(f,g) d\Omega_S + \int_S Z(f) d\Omega_S \tag{1}$$

In the above equation Y denotes the distance between the function f and polar edge map g, Z measures the reconstruction smoothness and μ is responsible for the tradeoff between fidelity of reproduction of the actual contour and smoothness, $d\Omega_S$ is a differential shape element on the unit circle. Setting $Y(f,g) = (f(\theta) - g(\theta))^2$ and $Z(f) = \| \nabla \|^2$, where ∇ is the gradient operator the energy functional becomes

$$E(f,g) = \int_S \mu (f(\theta) - g(\theta))^2 d\Omega_S + \int_S \| \nabla \|^2 \, d\Omega_S \tag{2}$$

$E(f,g)$ can be further minimized over f by the usage of calculus of variation to determine an Euler-Lagrange equation for a stationary point of the above energy functional. This procedure yields the following equation:

$$\nabla^2 f - \mu(f - g) = 0 \tag{3}$$

This is an elliptic equation of Helmholtz type. Our case of spherical coordinates is also Helmholtz-like. Moving the g term to right hand side and expressing it using previous values of f yields the following linearized form:

$$\alpha \nabla^2 f_{n+1} - f_{n+1} = g_{f_n} \tag{4}$$

which can be easily solved by the fast spectral method, with $\alpha = 1/\mu$. The dependence of g_{f_n} on g and other data is the essence of our approach and is explained throughout next subsections.

2.2 Bayesian Edge Map Determination

The edge map determination is based on Bayesian approach and provides the best choice among all others taking into account the risk associated with each one and mutual relationship between choices.

Let $P(E_i/I)$ denote the required probability of the most appriopriate edge in our existing data set. This is conditional probability as it depends on the contents of I. $P(E_i/I)$ is the probability of the fact that the I point belongs to the edge E_i knowing the value of intensity of this point. Let $P(I/E_i)$ be a probability of how much the value or intensity of a point is depending on edge E_i. This term serves as a kernel. $P(E_i)$ is simply the probability of existence of the edge E_i among all other detected edges. Then the required probability can be found by solving the Bayes rule:

$$P(E_i/I) = \frac{P(I/E_i)P(E_i)}{P(I)} = \frac{P(I/E_i)P(E_i)}{\sum_i P(I/E_i)P(E_i)} \tag{5}$$

$P(I)$ is a sum of all probabilities $P(I/E_i)$ weighted by $P(E_i)$ and thus remaining constant. $P(I)$ is only a normalizing term and can be excluded from further analysis. The standard way of solving the equation is the maximization of

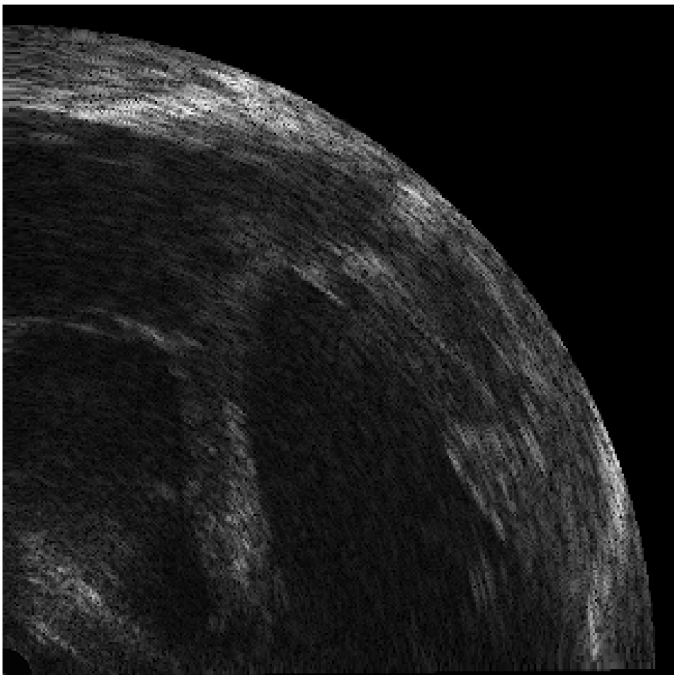


Fig. 1. Part of the ultrasonic image of ventricle and valves

the right hand side over the parameter E_i and then maximization of the found solution over all accesible data. The former procedure is known as *maximum likelihood* (ML) and the latter as *maximum-a-posteriori* (MAP). The $P(E_i)$ is a *prior* and we put our *a priori* knowledge inside it.

In practice we are estimating the $P(I/E_i)$ from the histogram of I along given radius from centroid to a point on circumference of some bounding box. The histogram is shrank in such a way that each bin is equal to the edge size assuming that each expected edge covers the same number of intensity levels. After calculation of the probability over all edges E_i we are performing ML step what allows to estimate the most probable edge in I. Then the MAP is done by searching for maximum over the data itself, and usually the first maximum in $P(I/E_i)$ is detected as an edge. The $P(E_i)$ is simply a constant value. Having this knowledge we can easily determine the position of edge in I even if the data is highly corrupted by noise.

Performing the classification of $P(E/I)$, $G = classification(P(E/I))$, we estimate G, which is Bayesian constrained edge map. Calculating it for each θ_j we find the representation of initial estimate of the polar edge map, $g(\theta) = G(\theta)$.

2.3 Fast Spectral Method

The spectral methods are widely used for all kind of problems that can be expanded into Fourier series and solved in Fourier space. For the purpose of

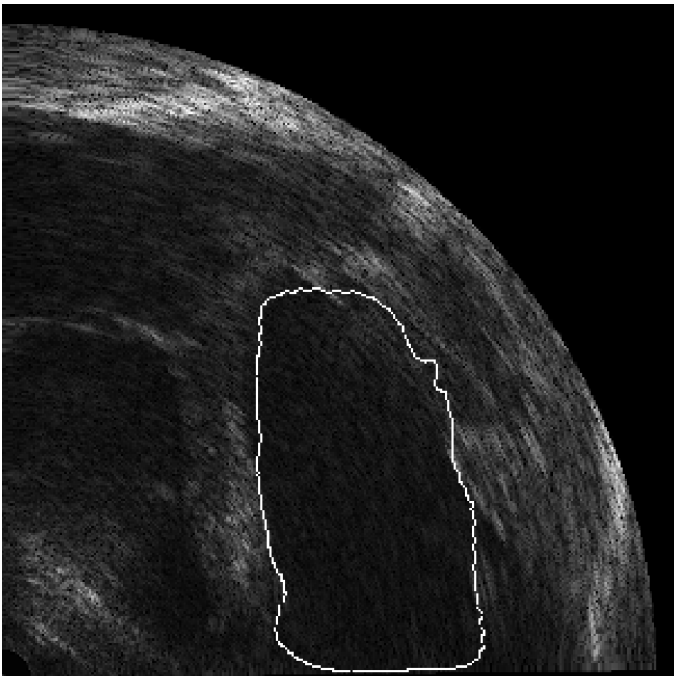


Fig. 2. The image with superimposed ventricle contour using Bayesian constrained spectral segmentation for μ=110

this study, following [9] we adapt Cheong's method [1] to solve the equation 3. We express the Laplacian operator ∇^2 in polar coordinates, assuming the unit radius:

$$\nabla^2 = \frac{1}{sin\theta}\frac{\delta}{\delta\theta}\big(sin\theta\frac{\delta}{\delta\theta}\big) \tag{6}$$

Both functions, f and g are defined on the computational grid (θ_i), where $\theta_i = \pi(j+0.5)/J$. J is the number of points along the cavity circumference high enough to engage all points covered by g. Each point in g may be now expressed by its discrete cosine transform (DCT) representation yielding

$$g(\theta_i) = \sum_{n=0}^{J-1} g_n cosn\theta_i \tag{7}$$

with g_n being simply the coefficients of discrete cosine transform. Applying 6 into 3 we can write the equation 3 as an ordinary differential equation (ODE):

$$\frac{1}{sin\theta}\frac{\delta}{\delta\theta}\big(sin\theta\frac{\delta}{\delta\theta}f(\theta)\big) = \mu[f(\theta) - g(\theta)] \tag{8}$$

which yields an algebraic system of equations in Fourier space:

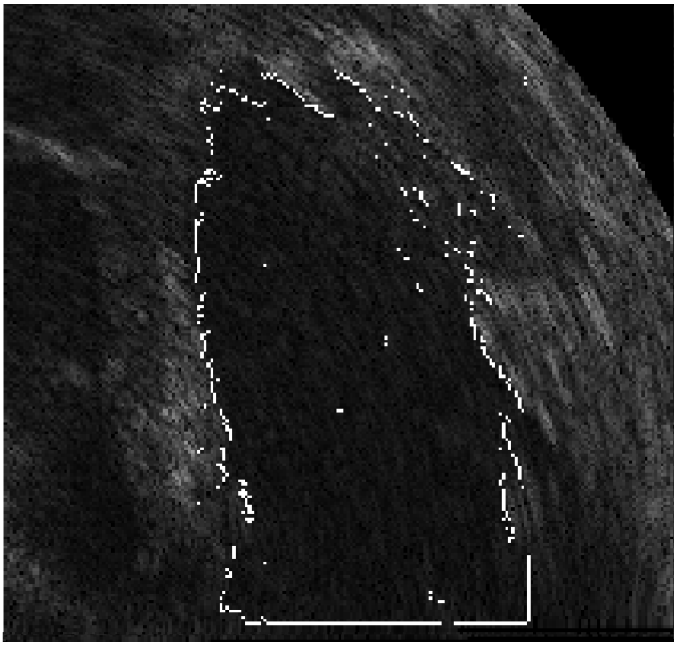


Fig. 3. Bayesian estimation of the most probable contours

$$p_{n-2}f_{n-2} - p_n f_n + p_{n+2}f_{n+2} = \mu\big[\frac{1}{4}g_{n-2} - \frac{1}{2}g_n + \frac{1}{4}g_{n+2}\big] \tag{9}$$

where

$$p_{n-2} = \frac{(n-1)(n-2)+\mu}{4}, \quad p_n = \frac{n^2+\mu}{2}, \quad p_{n+2} = \frac{(n+1)(n+2)+\mu}{4} \tag{10}$$

after substitution of 7 into 8 and expressing f in the same way as g (eq.7). The index $n = 1, 3, ..., J-1$ for odd n and $n = 0, 2, ..., J-2$ for even n. The system of equation 9 may be now expressed as a double matrix equation:

$$\mathbf{B}_e\hat{f}_e = \mathbf{A}_e\hat{g}_e, \quad \mathbf{B}_o\hat{f}_o = \mathbf{A}_o\hat{g}_o \tag{11}$$

with subscripts e for even and o for odd n, $\hat{f}$ and $\hat{g}$ denote the column vector of expansion coefficients of $f(\theta)$ and $g(\theta)$, respectively. $\mathbf{B}$ is a tridiagonal matrix containing the left hand side of equation 9 and $\mathbf{A}$ is tridiagonal matrix with constant coefficients along each diagonal corresponding to right hand side of 9.

2.4 Data Processing

The shape reconstruction is done in two steps. Firstly, we analyze the data and classify the most probable edges according to given benchmarks. This is realized

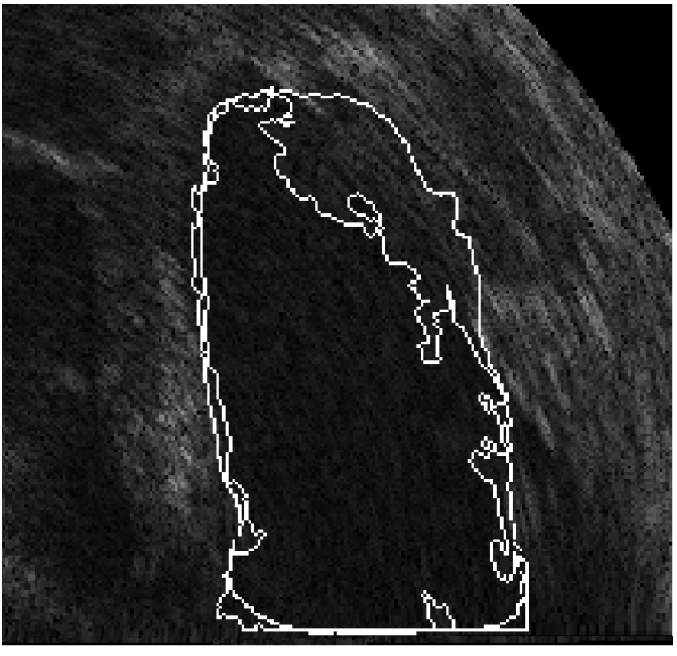


Fig. 4. The comparison of simple optimized thresholding segmentation together with the proposed one. The global character of structure is much better preserved by the new method.

by Bayesian procedure described previously. Secondly, the equation 4 is solved in an iterative way, and in each step the right hand side term is updated by its current approximation obtained by inverse DCT (IDCT) from the cosine expansion coefficients derived from the residuals remaining after substraction of the initial estimation of edge map and currently found solution, obtained by IDCT from the vector f_n. The calculated set of expansion coefficients f_{n+1} serves for the reconstruction of f_i, the representation of g on the certain level of approximation i. This function carries the information about the structure of the real edge on a given scale i. Summing all partial functions f_i we recover the required smooth approximation to g, recovering the most probable edge map.

2.5 Experimental Data

The algorithm has been applied to a number of ultrasonograhic cardiac images. A centroid was calculated for each scan and then the final analysis of contour was done. The data was collected from a healthy volunteer, using an Ultramark ATL scanner equipped with a 3.5 MHz phased array probe. The data was sampled at 20 MHz rate.

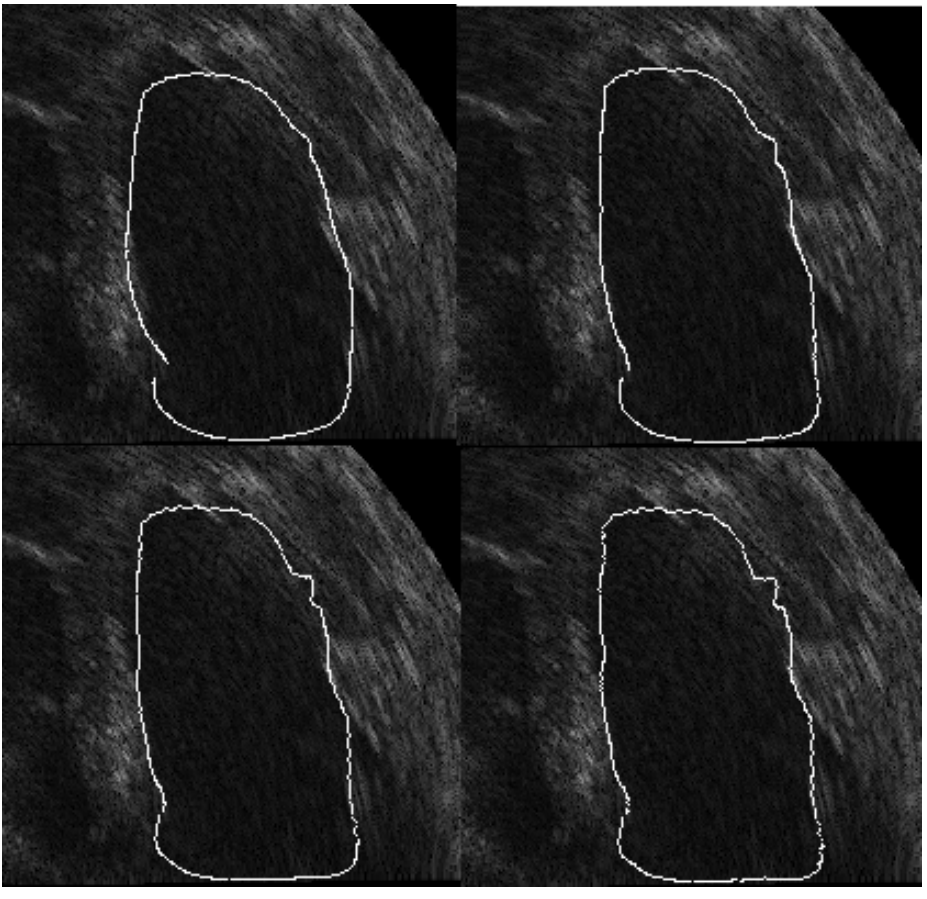


Fig. 5. The subsequent iterations of PDE approaching the final contour. From top left to bottom right there is 1st, 2nd, 3rd and 4th step of active segmentation.

3 Results

Figures 1 and 2 shows the results of application of our method to ultrasonic cardiac scans. Figure 1 shows the example raw ultrasonic data. Its Bayesian constrained spectral segmentation with low level of details is shown in Figure 2. Figure 3 serves an example of the Bayesian inference on edge detection applied to raw data. The comparison between a simple thresholding segmentation with a threshold of the best choice and the segmentation proposed throughout this paper is also shown and displayed on Figure 4.

To illustrate the algorithm performance we have imaged all subsequent steps of PDE iteration that provide the curve approaching the real contour. Figure 5 presents 4 steps of solving the PDE, the last is corresponding to the best fit of found contour to the real one and is also a final solution. Figure 6 contains

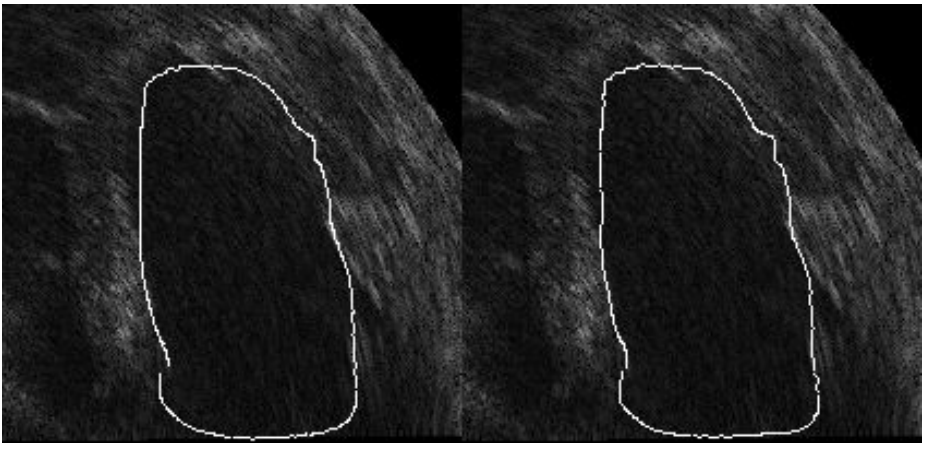


Fig. 6. The final iterations of PDE approaching the final contour. On the left there is the contour reconstructed by our method with μ=50, on the right there is the same image segmented with μ=80.

the final solutions of PDE, equivalent to the best representation of required structure, derived for different values of μ parameter responsible for the quality of reconstruction.

4 Discussion

The noisy nature of the ultrasonic images makes segmentation of such data a complex task. From anatomical study and other modalities we know the shape of explored structure [11]. This structure is usually weak in ultrasonic data but may be recovered due to its global properties [10]. Some knowledge can also be revealed through Bayesian inference. The precise tuning of such method is complicated and depends on the proper choice of priors and parameters describing the smoothness of a fit, a recovered structure and the level of noise and linear and nonlinear distortion. Some minor imperfections in the final segmentation are easily noticed, for instance those in non perfect recovering of right-upper, upper and upper-left borders of the cardiac ventricle, but they may be overcome by introduction of more complex priors into Bayesian inference on edge detection. Such research based on multiscale analysis of data content [6] is currently carried out providing promising preliminary results and serves as a clue for futher developement of the method. There is also a possibility to incorporate multimodal priors based on models derived from data acquired by complementary technique like MRI [10]. Another approach is to use knowledge from an anatomical data base.

The segmented structure corresponds to a cardiac ventricle. During the cardiac cycle the ventricle is changing its dimensions very considerably. A very active region of left atrioventricular valve (mitral valve) makes the circumference

open and hence some approximation is necessary. We treat this region artificially as a boundary between mitral valve and cardiac ventricle what results in the imperfections mentioned above.

Main advantage of our methodology is detecting a global structure instead of only the local details as in simple thresholding techniques. This directly leads to global parametrization of shape required for elastography. Another kinds of base functions, like wavelets, may be applied to represent the global structure of object.

Finally, the modifications of MAP and ML steps may be introduced. Increasing the data region over that currently used along given radius for the MAP step, or incorporating the information obtained by edge filters into the MAP step, also increase the probability of proper edge detection and decrease that of the fake one to insignificant level.

5 Conclusions

The new method presented here combines fast spectral method and Bayesian inference. The method is well suited for the analysis of noisy and disturbed data like that obtained by ultrasonography although some tuning of the algorithm performance is still required. The main limitation is the requirement of radial description of the object structure.

References

1. Cheong, H.,: Double Fourier series on a sphere: Applications to elliptic and vorticity equations, Jour. of Comp. Physics, **157, 1** (2000) 327-349
2. Dydenko I, Friboulet D, Gorce JM, D'hooge J, Bijnens B, Magnin IE.: Towards ultrasound cardiac image segmentation based on the radiofrequency signal, Medical Image Analysis **7** (2003) 353-367
3. Fleming D, Xia X, McDicken WN, Sutherland GR, Fenn L.: Myocardial velocity gradients detected by doppler imaging, Br. J. Radiol. **67** (1994) 679-688
4. Hammoude A.: Endocardial border identification in two-dimensional echocardiografic images: review of methods, Comp. Med. Imag. Graph. **32** (1998) 181-193
5. Heimdal A, Stoylen A, Torp H, Skjaerpe T.: Real-time strain rate imaging of the left ventricle by ultrasound, J. Am. Soc. Echocard. **11** (1998) 1013-1019.
6. Holschneider M., Kronland-Martinet R., Morlet J., Tchamitchian P.: A real time algorithm for signal analysis with the help of the wavelet transform, Wavelets: Time-Frequency Methods and Phase-Space, Springer-Verlag (1989) 286-297
7. Kowalski M, Kukulski T, Jamal F, D'hooge J, Weidemann F, Rademakers F, Bijnens B, Hatle L, Sutherland GR.: Can natural strain and strain rate quantify regional myocardial deformation? A study in healthy subjects, Ultrasound in Med. & Biol. **27(8)** (2001) 1087-1097
8. Laidlaw D.H., Fleischer K.W., Barr A.H.: Partial volume segmentation with voxel histograms, Handbook of Medical Imaging, Processing and Analysis, ed. Bankman, I.N., Academic Press (2000) 195-214
9. Li, J., Hero A.O.: A Fast Spectral Method for Active 3D Shape Reconstruction, Jour. of Math. Imaging and Vision, **20** (2004) 73-87

10. Mitchell, S.C., Bosch, J.G., Lelieveldt, B.P.F., van der Geest R.J., Reiber, J.H.C, Sonka, M.: 3-D Active Appearrance Models: Segmentation of Cardiac MR and Ultrasound Images, IEEE Trans. on Medical Imaging, **21, 9** (2002) 1167-1178
11. Rogowska, J.: Overview and Fundamentals of Medical Image Segmentation, Handbook of Medical Imaging, Processing and Analysis, ed. Bankman, I.N., Academic Press (2000) 69-86
12. Soltysinski T.: Novel algorithm for active surface reconstruction by Bayesian constrained spectral method, Hozman J., Kneppo P. (Editors). IFMBE Proceedings, Vol. 11. Prague: IFMBE, 2005. ISSN 1727-1983. (Proceedings of the 3rd European Medical & Biological Engineering Conference - EMBEC05. Prague, Czech Republic, 20-25.11.2005), 4191-4195

A Model of a Diagnostic Rule in the Dempster-Shafer Theory

Ewa Straszecka

Silesian University of Technology
Institute of Electronics, 16 Akademicka St., 44 -100 Gliwice, Poland
ewa.straszecka@polsl.pl

Abstract. In the paper a model of a rule in medical diagnosis is proposed. The Dempster-Shafer theory of evidence and fuzzy sets are implemented in the rule representation. The basic probability assignment describes certainty of the rule. Fuzzy sets model the rule premises. The diagnosis is indicated by the belief and the plausibility measures. Thresholds are used to adjust the significance of the rules and quality of observations. The suggested methods are verified for databases of thyroid gland diseases: the database found in the Internet and individually gathered data, as well as simulated data and the iris plants database.

1 Introduction

Rules are usual representation of medical knowledge in diagnosis support systems. They model dependence among symptoms and diagnoses. Their representation is crucial as it determines diagnostic inference of the systems. Usually, heuristic rules have the following form:

$$IF\ X^1\ is\ X_l^1,\ and\ \ldots,\ and\ X^n\ is\ X_l^n\ THEN\ D_l, \tag{1}$$

where X^i are medical parameters (e.g. laboratory tests), X_l^i are linguistic variables that describe values of the medical parameters (e.g. 'high', 'normal'), and D_l is a diagnosis. Such rules are often provided with certainty factors as it can be observed in medical indices [3] or in expert systems [5]. The IF-THEN dependence of the rule (1) is usually defined in terms of conditional probability or fuzzy implication. In case of conditional probability, the premise of rule (1) has to be either true or false, which means that linguistic variable X_l^i is replaced with a single value or an interval. Hence, instead of the premise $'X^i$ is $X_l^{i\prime}$ we have $x^i = x_0^i$ or $x^i \in [x_1^i, x_2^i]$, where x^i is a possible value of the X^i parameter, and x_0^i, x_1^i, x_2^i are values specific for a test (e.g. laboratory norms). Then, the rule (1) becomes:

$$IF\ x^1 = x_0^1,\ and\ \ldots,\ and\ x^n \in [x_1^n, x_2^n]\ THEN\ D_l. \tag{2}$$

The premise of rule (2) is either true or false, depending on the evidence, and norms of the parameter X^i are constant. Thus, the representation (2) of the rule (1) is somewhat simplified. Still, it is convenient, and for that reason often used.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 998–1007, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Certainty factor for the rule can be easily defined as the conditional probability $P(D_l/X^i, \ldots, X^n)$, yet it is difficult to calculate in practice.
If we use fuzzy rules for (1) representation, then X_l is described by the fuzzy membership function $\mu_l(x)$. The truth of the premise can be defined in the $[0, 1]$ interval, and norms can be described by fuzzy sets. Still, if a fuzzy implication resembles the link between the premise and the conclusion, then the D_l has to be also fuzzy. Thus, instead of (1) we have the following diagnostic rule:

$$IF\ X^1\ is\ X_l^1,\ and\ \ldots,\ and\ X^n\ is\ X_l^n\ THEN\ R\ is\ R_{D_l}, \tag{3}$$

where R stands for the disease risk and R_{D_l} is the linguistic value of the risk, represented by the $\mu_l(r)$. The formulation (3) is awkward because of a lack of $\mu_l(r)$ domain. An 'artificial' domain, which can be defined, is difficult to interpret for physicians.

Pure probabilistic and fuzzy approaches have many other important drawbacks that concern dependence of symptoms or significance of a symptom for a diagnosis. Thus, the aim is to propose the suitable representation of the diagnostic rule comprising the following features:

- simultaneous representation of fuzzy and crisp statements in a premise of a rule
- separate representation of premise imprecision and rule uncertainty;
- exact representation of rule conclusions;
- applicability in an inference that is intuitively clear for physicians.

Numerical calculations necessary for the diagnosis determination have to be simple, quick and reliable. The paper presents solutions of the mentioned problems that can be implemented in a diagnosis support system.

2 Materials and Methods

2.1 Rule Definition

The Dempster-Shafer theory of evidence (DST) can be particularly convenient for the diagnosis support [2]. The theory neglects dependence of symptoms, so troubles with the conditional probability calculation can be avoided. In the DST a basic probability assignment (BPA) is defined for a set of focal elements [2]. In the diagnosis the focal element corresponds to the premise of the rule, including one or several symptoms. Thus, the premise $p_j \equiv 'X\ is\ X_l'$ creates the single focal element a_j^l. The conjunction of premises $'X^j\ is\ X_l^j\ and\ X^k\ is\ X_l^{k\prime}$composes a complex focal element $a_i^l = \{p_j, p_k\}$, where $p_k \equiv 'X^k\ is\ X_l^{k\prime}$. Hence, the rule (1) can be written as:

$$IF\ a_i^l\ THEN D_l. \tag{4}$$

It must be stressed that $THEN$ in the rule (4) does not denote an implication. It denotes the fact of assigning the conclusion to the focal element. The set of rules (4) defined for the chosen diagnosis D_l makes a knowledge base of this

diagnosis. Thus, the A_l set of focal elements ($A_l = \{a_i^l\}, i = 1, ..., n_l$) refers to the D_l diagnosis. The BPA is defined for the knowledge base as [2]:

$$m_l(f) = 0, \sum_{\substack{a_i^l \in A_l, \\ i=1,...,n_l}} m_l(a_i^l) = 1, \tag{5}$$

where m_l is the BPA for the $l-th$ diagnosis, f denotes the focal element 'none of the $l-th$ diagnosis symptoms is present' and n_l is the number of rules (4). The BPA value for the $i-th$ rule, i.e. $m_l(a_i^l)$, may stand for its certainty factor. Each focal element is defined using one (in case of the single focal element) or several (for the complex focal element) membership or characteristic functions. The single symptom occurs when for the observed value x^{i*} of the X^i parameter the membership function value is greater than a threshold, i.e. $\mu_l(x^{i*}) \geq \eta$, or characteristic function $\chi_l(x^{i*})$ equals 1. The η threshold resembles a level of ignorance of the symptom formulation. If η is small (still greater than zero), even dubious symptoms are considered during the inference. For big η the inference is limited to the surest symptoms. The $m_l(a_i^l)$ can be determined as the normalized frequency of the a_i^l symptom occurrence for training data among A_l set of symptoms. Hence, BPA:

$$m_l(f) = 0, \sum_{\substack{a_i^l \in A_l, \\ \eta_i \geq \eta_{BPA}}} m_l(a) = 1, \tag{6}$$

where η_{BPA} is the threshold for the BPA calculation and η_i is the certainty level of the $i-th$ rule premise. For the single focal element the certainty level:

$$\eta_i = \mu_l\left(x^{i*}\right), \text{ or } \eta_i = \chi_l\left(x^{i*}\right), \tag{7}$$

where χ_l is the characteristic function of symptom's presence. It equals 1 when $x^{ik*} \equiv 'present'$ or 0 when $x^{ik*} \equiv 'absent'$. For the complex fuzzy focal elements

$$\eta_i = \min_j \mu_l\left(x^{ij*}\right), \tag{8}$$

where $\mu_l\left(x^{ij*}\right)$ refers to the $j-th$ condition of the $i-th$ premise. The complex elements $a_i^l = \{p_j, p_k\}$ may include fuzzy and crisp premises, for instance: $p_j \equiv 'X^j is X_{D_l}^j' \equiv 'X^1 \, is \, high'$, $p_k \equiv 'X^k \, is \, X_{D_l}^k' \equiv 'X^k \, is \, present'$. In such a case

$$\eta_i = \min\left(\mu_l(x^{ij*}),\, \chi_l(x^{ik*})\right). \tag{9}$$

It is also possible to match a fuzzy observation and the fuzzy premise. In that case the certainty level for the single focal element equals:

$$\eta_i = \max_{x^i} \min\left(\mu_l(x^i),\, \mu_l^*(x^i)\right), \tag{10}$$

where $\mu_l^*(x^i)$ denotes an evidence given by the membership function of a fuzzy set. The possibility to enter a fuzzy evidence can be an important new quality in the diagnosis support. For instance, such patient's symptoms like rapid

exacerbation of disease manifestations can be easily represented. All in all, the following generalization of (7)-(10) definitions is possible:

$$\begin{aligned}
&\eta_i = \min_j \max_{x^{ij}} \min \left(\psi_l(x^{ij}),\, \psi_l^*(x^{ij})\right), \\
&\psi_l(x^{ij}) = \mu_l(x^{ij}), \text{ or } \psi_l(x^{ij}) = \chi_l(x^{ij}), \\
&\psi_l^*(x^{ij}) = \mu_l^*(x^{ij}), \text{ or } \psi_l^*(x^{ij}) = \mu_l(x^{ij*}), \text{ or } \psi_l^*(x^{ij}) = \chi_l(x^{ij*}), \\
&\text{or } \psi_l^*(x^{ij}) = \delta_{l\,x^{ij*}},
\end{aligned} \tag{11}$$

where $\delta_{l\,x^{ij*}}$ stands for the singleton of the observed value x^{ij*}.
Hence, the rule (1) becomes finally:

$$\begin{aligned}
&IF\ a_i^l\ THEN\ D_l \ \text{ with certainty } m(a_i^l) \\
&a_i^l \equiv 'XisX_l^1\,' \text{ or } a_i^l \equiv ('XisX_l^1', \textit{ and } \ldots, \textit{and}\ 'XisX_l^{n\prime}),
\end{aligned} \tag{12}$$

and X_l^i is defined by means of $\mu_l(x^i)$ or $\chi_l(x^i)$. Thus, fuzzy or crisp conditions occur in the premise of the rule (12) and the conclusion D_l remains crisp. In this way, the rule formulation is identical with (1). Moreover, the imprecision of the symptom is modeled by $\mu_l(x^i)$ separately from the uncertainty of the rule defined by the $m(a_i^l)$. In general case $a_i^l \cap a_k^l \neq \emptyset$ for $i \neq k$ i.e. the same symptom may repeat several times in different focal elements. Despite of this feature, the BPA calculation is easy, as the DST neglects dependence of a_i^l and a_k^l. For each diagnosis a set of rules (12) has to be created.

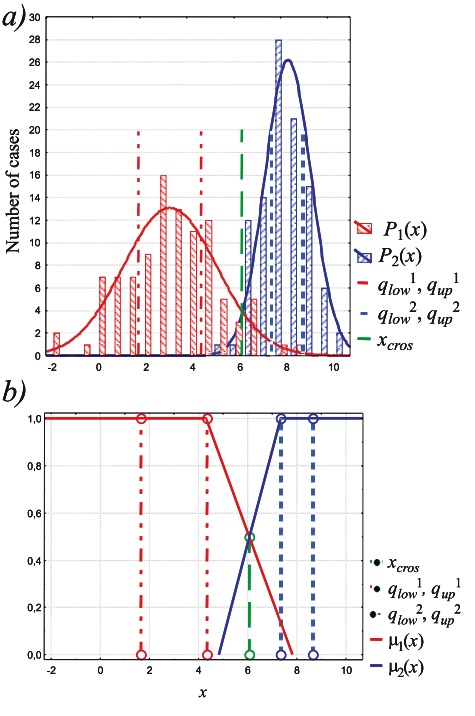


Fig. 1. Membership function determination

2.2 Membership Functions

Membership functions that are used in (12) can be defined in a dialog with an expert (a physician) or during training data investigation. The second manner is described in details in [6]. Here only a brief explanation is given. Training data for two competitive diagnoses create two samples (see Fig.1) Theoretical distributions of the samples can be found. They intersect at a point. The intersection (x_{cros}) as well as lower and upper quartiles (q^i_{low}, q^i_{up}) of the samples create points for determination of membership functions. The x_{cros} resembles the point of crossover of two membership functions. It has been assumed that membership function values in this point equal 0.5, while in quartiles they equal 1. When more than two diagnoses are considered, the process proceeds for the next two partly overlapping distributions. During calculations a modification of the membership functions has been also tried. It turned out that membership functions with slightly steeper slopes and a wider interval of maximal values worked better. Hence, finally $q^i_{low} - 0.7(q^i_{low} - x_{cros})$ and $q^i_{up} + 0.7(x_{cros} - q^i_{up})$ points have replaced quartiles.

Expert's knowledge may also influence membership functions. An expert may indicate x_{cros}, while training data determine quartiles. However, the membership function can be determined exclusively on the basis of training data, even if the data are incomplete and not numerous and an expert is not available.

2.3 Inference

Let us assume that the diagnostic knowledge base includes rules (12), membership functions are defined and BPA are calculated according to (6). Inference with the knowledge base requires matching patient's symptoms with the rules. In case of crisp premise and observation accuracy of matching equals 0 or 1, while in case of the fuzzy premise the certainty level $\eta_i \in [0, 1]$. Thus, belief of the diagnosis can be defined as:

$$Bel(D_l, \eta_T) = \sum_{\substack{\eta_i \geq \eta_T, \\ i=1,\ldots,n_l}} m_l(a^l_i), \tag{13}$$

Plausibility of the diagnosis cannot be defined by means of the $\min_j$ operator in the certainty level (11). Plausibility concerns all focal elements that are different from the f focal element. Hence, θ_i certainty level is defined:

$$\theta_i = \max_j \max_{x^{ij}} \min\left(\psi_l(x^{ij}),\, \psi^*_l(x^{ij})\right), \tag{14}$$

and the plausibility is calculated as:

$$Pl(D_l, \eta_T) = \sum_{\substack{\theta_i \geq \eta_T, \\ i=1,\ldots,n_l}} m_l(a^l_i). \tag{15}$$

The threshold η_T in (13), (15) qualifies symptoms that are sufficiently precise for diagnostic inference. The definitions are extensions of the classical measures

[2] for fuzzy focal elements. The Bel and Pl values change along with the η_T threshold modification. Therefore, the threshold should resemble a level of ignorance that is allowed during inference. The $[Bel(D_l, \eta_T), Pl(D_l, \eta_T)]$ interval determines credibility of the $l-th$ diagnosis.

2.4 Example

Let us consider diagnostic inference (Fig.2). It is performed for the diagnosis D_l which rules are listed on the left-hand side and membership functions that are represented by dashed lines in diagrams. Each rule create a focal element for which the $m_l(a_i)$ value is proposed. The focal elements are complex for the first and third rule and single for the second rule. It should be noticed that the conjunction of the focal elements, i.e. rule premises, is not empty. Observations (denoted in the diagrams by solid lines) match premises of the rules. In case of the first rule the crisp observation match the crisp premise, and the fuzzy observation ('normal') match the fuzzy premise ('low'). The certainty levels η_1 and θ_1 are calculated according to the equations (11) and (14).

Table 1. Values of Bel and Pl for different thresholds

η_T	Bel	Pl
[0, 0.5]	$m_l(a_1) + m_l(a_2) + m_l(a_3) = 1$	$m_l(a_1) + m_l(a_2) + m_l(a_3) = 1$
(0.5, 0.75]	$m_l(a_2) + m_l(a_3) = 0.7$	$m_l(a_1) + m_l(a_2) + m_l(a_3) = 1$
(0.75, 1]	0	$m_l(a_1) + m_l(a_3) = 0.6$

It is worth noticing that $\eta_i \leq \theta_i$. In the second rule the observation which is the singleton $\delta_{l\,2.25}$ is compared to the fuzzy set μ_{high}. The focal element is single in this case, so $\eta_2 = \theta_2$. In the third rule the crisp observation and the fuzzy premise as well as the fuzzy observation and the fuzzy premise are compared. Since η_i and θ_i values are smaller than 1, the Bel and Pl values depend on the η_T threshold. They are presented in Tab.1. From the example we can learn that the threshold can be either too low (in the first row of Tab.1) or too high (in the last row of Tab.1) to infer a reliable diagnosis. Thus, a criterion of the threshold choice has to be proposed.

2.5 Elaboration of the Final Diagnosis

If all focal elements are single, then $Bel(D_l, \eta_T) = Pl(D_l, \eta_T)$ and for $\eta_T = 1$ the both measures are equal to the conditional probability of the D_l, given the observed symptoms. This almost never occurs in a diagnostic routine. Belief is a measure of the diagnosis credibility and is based on the most certain information. Plausibility concerns amount of available information. Final conclusion of the inference is elaborated after comparison of belief values for several diagnoses. If $\exists_{\substack{i \neq j, \\ i,j=1,\ldots,N}} Bel(D_i, \eta_T) > Bel(D_j, \eta_T)$ (where N is the number of possible diagnoses), then the D_i diagnosis is announced as the final conclusion. When maximal Bel values occur for at least two diagnoses, the final diagnosis

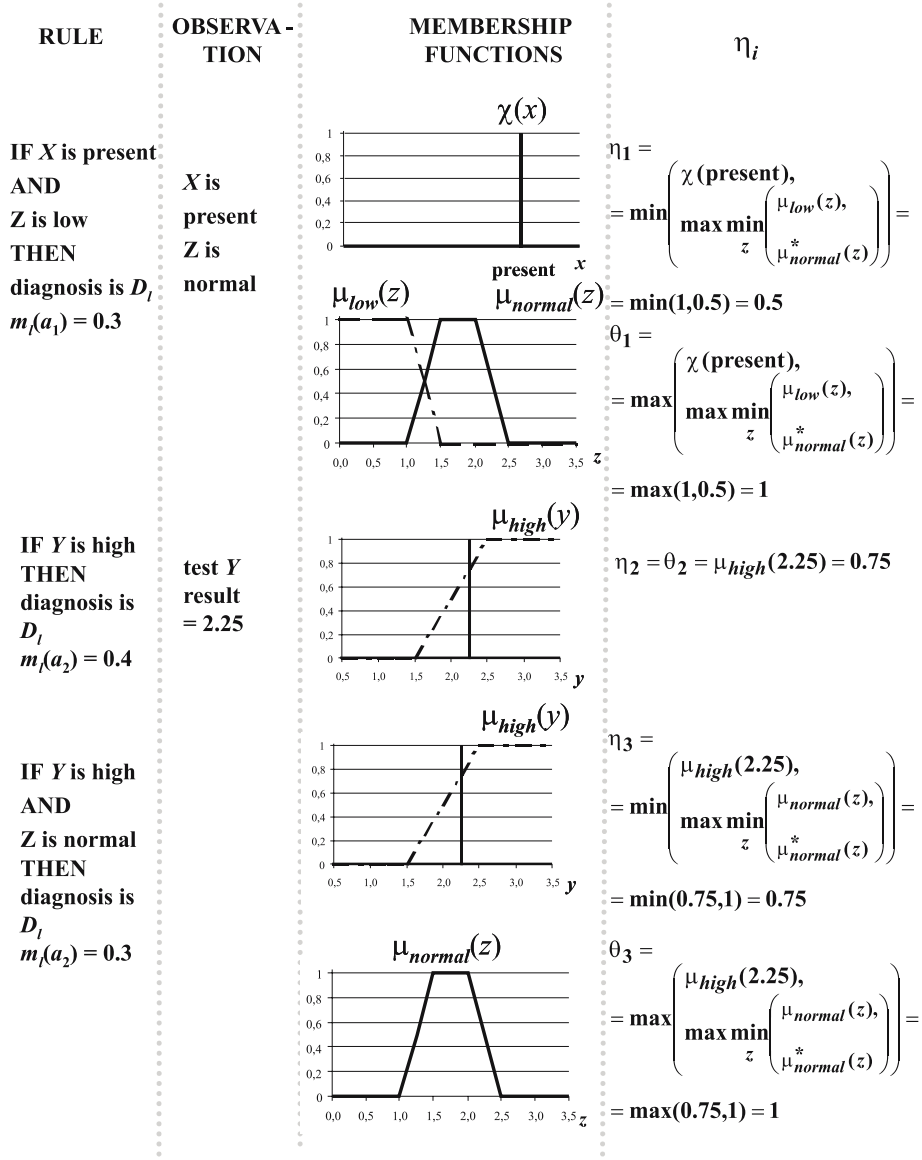


Fig. 2. Inference

cannot be stated. The problem is that for different η_T thresholds both cases (i.e. $Bel(D_i, \eta_T)$ superiority and $Bel(D_i, \eta_T)$ equality) can occur for the same patient. Thus, among values of the belief measure obtained for different thresholds only one (let us denote it by $Bel_{comp}(D_i)$)should be chosen for the comparison. The best choice is made according to the Pl measure. Namely, the $Bel(D_i, \eta_T)$

for which η_T is the greatest and Pl value is still the highest is chosen. Let t be the interval of the threshold values for which the plausibility is maximal, i.e:

$$\eta_T^* \in t \Rightarrow Pl(D_i, \eta_T^*) = \max_{\eta_T} Pl(D_i, \eta_T), \tag{16}$$

then

$$\eta_t^* = \sup t \Rightarrow Bel_{comp}(D_i) = Bel(D_i, \eta_t^*). \tag{17}$$

In practice, η_t^* is found with some precision (for instance 0.05). Generally, η_{BPA} and η_T thresholds do need to be related. Still, both the common sense and numerical experiments [6] confirm that knowledge should be more accurately formulated than the symptoms that are used during consultation.

2.6 Databases

The method described in this paper can be useful in a diagnosis support when training data are incomplete or not numerous. This has been the case of data gathered during works on the diagnosis support in thyroid gland diseases. Symptoms in this problem are crisp and fuzzy. However, the method has been also verified for numerical data available in the Internet, which are not deficient, but hard to classify. Moreover, data have been simulated to investigate the problem of designing membership functions. Thus, four kinds of data have been used during the verification. The first database is available on the Internet: ftp.ics.uci.edu/pub/machine-learning-data-bases/thyroid-disease, files new-thyr.*. The database is related to the problem of thyroid gland diseases and the reference [1] provides its comprehensive statistical analysis. The data are called the 'Internet data' from now on. Number of training/test cases for the data have been as follows: hyperthyroidism 75/75, euthyroidism 15/20, hypothyroidism 15/15. The data have been used to check whether the proposed method can be more efficient than statistics in differential diagnosis of thyroid gland diseases. The second database has been the well-known Iris Plants Database from http://www.ics.uci.edu. The data has been divided into 25/25 train/test cases for the three defined categories. The database is a benchmark in research works and so it has been used to verify correctness of the proposed method. Next, data have been simulated with comparable statistic parameters to the Internet data. Each diagnostic group has included 100/100 cases. In such a way sets for 50 simulation runs of the method have been prepared. The simulated data have been used to find out to which extent performance of the method depends on the coherence of learning and test data and to improve membership functions. Finally, individually gathered data with the following number of cases: hyperthyroidism 52/16, euthyroidism 26/26, hypothyroidism 23/23 have been used.

3 Results

Verification of the proposed method has consisted in an error investigation. The error is the percentage of wrong diagnoses indicated by the method for the test

data. Wrong diagnoses have been outputs different from that specified in the test data as well as undetermined diagnoses (because of belief equality). Classical methods (ISODATA, fuzzy ISODATA and IF-THEN rule Mamdani-like inference) have resulted in errors greater than 30% for the Internet database [6]. Three diagnostic categories have been considered. Authors of statistical investigations [1] have concluded that several methods have been acceptable for selection of two categories (e.g. 'hyperthyroidism' and 'the rest'), but a classification for the three categories has always failed. In case of the proposed method the error found for the Internet data has been reduced to 2.67% when the optimum threshold has been chosen. Nine diagnostic rules have been formulated for each of the categories. The thresholds η_{BPA} and η_T has been changed in the $[0, 1]$ interval, with 0.05 step. Generally, better results have been obtained for thresholds higher than 0.1 and lower than 0.9 and $\eta_{BPA} > \eta_T$. The error has been significantly smaller in comparison to the classical methods. Moreover, it has occurred only in case of euthyroidisim, so patient's health would not be jeopardized.

Classification of the Iris Plant Databases has followed the same algorithm. All combinations of parameters, i.e. 15 rules have been created. Calculations have resulted in the global error of 5.33% (2 cases wrongly classified in the groups of versico and virgini) when cases were divided at random for training and test sets. Still, the proposed method is sensitive for the right choice of training data. It has been suspected that sound training data may improve classification of doubtful cases of the test data. Therefore, wrongly classified cases have been attached to the training data until the error has stabilized (on the level of 6.67%). Thus, all difficult cases were gathered in one set. Then the learning and test sets has been exchanged. In this way a division of cases for sound training data and difficult test data have been made. In such circumstances, the classification of the test data has been perfect (0% error). It can be concluded that though the error is not big for random training data, it is better to split the data into two sub-sets and to select suitable training cases.

Simulations have confirmed that the exact error depends on the training sample. For the simulated data the error has varied from 1% to 24%, though half of the samples have resulted in $5\% - 13\%$ error, which could be considered as a satisfactory result. The mean error has been 9.3%.

The individually gathered data have comprised ten crisp and fuzzy parameters. Each parameter has been considered separately and additionally 11 complex rules have been formulated. The global error has equaled 7.69%. Still wrong diagnoses have regarded only one test group - the hypothyroidism, so it could be suspected that training data for this group have been inadequate. The error would probably diminish when a better database is gathered.

4 Conclusions

The proposed method of diagnostic rule interpretation makes it possible to represent a diagnostic rule in the form that is very close to its intuitive formulation

by experts. It takes advantage of the belief and plausibility measures defined in the Dempster-Shafer theory of evidence. However, the measures and the basic probability assignment are defined for fuzzy focal elements, so the the theory is extended. Uncertainty and imprecision measures in knowledge representation and inference are also separated. Consequently, all kinds of information: precise ('yes/no'), measurements and even fuzzy linguistic expressions can be used as diagnostic inputs. Certainty of the rule can be modeled by the basic probability assignment. Imprecision of the symptom is described by a fuzzy set. The assignment and the fuzzy set are determined using training data, hence rules are adapted to patients' data. It means that not only IF-THEN rules, but also training data compose a knowledge base. Therefore, training data should be selected if we want to ensure high quality of the diagnosis. Still, it not difficult to choose the suitable cases and the method works for not numerous and incomplete data. The method can be even more efficient when a medical expert will help in rule formulation and will provide 'typical' training cases. As a result an effective method of diagnosis support can be implemented.

References

1. Coomans D., Broeckaert I., Jonckheer M., Massart D. L.: Comparison of multivariate discrimination techniques for clinical data-application to the thyroid functional state. Meth. Inform. Med. **22**(1983)93–101
2. Gordon J., Shortliffe E. H.: The Dempster-Shafer Theory of Evidence, in Buchanan B.G., Shortliffe E.H. (Eds): Rule-Based Expert Systems, (Addison Wesley)(1984) 272–292
3. Górnicki T.: Choroby tarczycy. PZWL, Warszawa(1988)56–89
4. Iliad, Widows-Based Diagnostic Decision Support Tools for Internal Medicine. User Manual, Applied Medical Informatics(1994)
5. Shortliffe E.H.: Computer-based medical consultations: MYCIN. Elsevier, New York-Oxford-Amsterdam(1976)
6. Straszecka E.: Combining uncertainty and imprecision in models of medical diagnosis. Information Sciences (in print)

DNA Fragment Assembly by Ant Colony and Nearest Neighbour Heuristics

Wannasak Wetcharaporn[1], Nachol Chaiyaratana[1,2], and Sissades Tongsima[3]

[1] Research and Development Center for Intelligent Systems,
King Mongkut's Institute of Technology North Bangkok
1518 Piboolsongkram Road, Bangsue, Bangkok 10800, Thailand
w_wannasak@hotmail.com, nchl@kmitnb.ac.th
[2] Institute of Field Robotics,
King Mongkut's University of Technology Thonburi
91 Pracha u-tid Road, Bangmod, Thungkru, Bangkok 10140, Thailand
[3] National Center for Genetic Engineering and Biotechnology,
National Science and Technology Development Agency
113 Thailand Science Park, Phahonyothin Road, Pathumthani 12120, Thailand
sissades@biotec.or.th

Abstract. This paper presents the use of a combined ant colony system (ACS) and nearest neighbour heuristic (NNH) algorithm in DNA fragment assembly. The assembly process can be treated as combinatorial optimisation where the aim is to find the right order of each fragment in the ordering sequence that leads to the formation of a consensus sequence that truly reflects the original DNA strands. The assembly procedure proposed is composed of two stages: fragment assembly and contiguous sequence (contig) assembly. In the fragment assembly stage, a possible alignment between fragments is determined where the fragment ordering sequence is created using the ACS algorithm. The resulting contigs are then assembled together using the NNH rule. The results indicate that in overall the performance of the combined ACS/NNH technique is superior to that of a standard sequence assembly program (CAP3), which is widely used by many genomic institutions.

1 Introduction

To understand the whole genetic makeup of an organism, the information regarding the entire DNA (deoxyribonucleic acid) sequence is required. DNA is a double helix comprised of two complementary strands of polynucleotides. Each strand of DNA can be viewed as a character string over an alphabet of four letters: A, G, C and T. The four letters represent four bases, which are adenine (A), guanine (G), cytosine (C) and thymine (T). The two strands are complementary in the sense that at corresponding positions A's are always paired with T's and C's with G's. These pairs of complementary bases are referred to as "base pairs" (bp). With the advent of shotgun genome sequencing technique whereby the entire long DNA are broken into numbers of small fragments, which are long enough to be read by present sequencing machines. At present,

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1008–1017, 2006.
© Springer-Verlag Berlin Heidelberg 2006

strands of DNA that are longer than 600 base pairs cannot routinely be sequenced accurately [1]. At the early stage of shotgun genome sequencing, large number of overlaped but "unordered" fragments are obtained. Hence, one of major procedures in a genome discovery project is to assemble these unordered DNA fragments. The DNA fragment assembly involves finding the right order of each fragment in the fragment ordering sequence, which leads to the formation of a consensus sequence that truly reflects the original DNA strands. A number of deterministic and stochastic search techniques have been used to solve DNA fragment assembly problems [2]. For instance, Huang and Madan [3] and Green [4] have used a greedy search algorithm to solve the problem. However, a manual manipulation on the computer-generated result is required to obtain a biologically plausible final result. Other deterministic search algorithms that have been investigated include a branch-and-cut algorithm [5] and a graph-theoretic algorithm where DNA fragments are either represented by graph nodes [6,7] or graph edges [8]. The capability of stochastic search algorithms such as a simulated annealing algorithm [9], a genetic algorithm [10,11,12] and a neural network based prediction technique [13] has also been investigated. The best DNA fragment assembly results obtained from stochastic searches have been reported in the research efforts by Parsons and Johnson [11], and Kim and Mohan [12] where genetic algorithms have proven to outperform greedy search techniques in relatively small-sized problems. In addition, the need for manual intervention is also eliminated.

Although significant results have been achieved, the search efficiency could further be improved if the redundancy in the solution representation is eliminated from the search algorithms [10]. Similar to a number of combinatorial optimisation techniques, the use of a permutation representation is required to represent a DNA fragment ordering solution in a genetic algorithm search. With such representation, different ordering solutions can produce the same DNA consensus sequence. Since genetic algorithms are parallel search techniques, the representation redundancy mentioned would inevitably reduce the algorithm efficiency.

2 Ant Colony System

A search algorithm that does not suffer from the aformentioned effect is an ant colony system (ACS) algorithm [14], which was originally proposed to solve a travelling salesman problem (TSP). From literature [10], the underlying DNA fragment assembly problem can then be treated as an instance of the TSP. The natural metaphor on which ant algorithms are based is that of ant colonies. Real ants are capable of finding the shortest path between a food source and their nest without using visual clues by exploiting pheromone information. While walking, ants deposit pheromone on the ground, and probabilistically follow pheromone previously deposited by other ants.

The way ants exploit pheromone to find the shortest path between two points is portrayed in Fig. 1. Consider a situation where ants arrive at a decision point

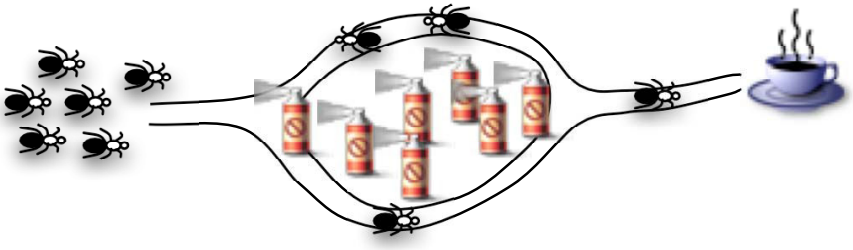


Fig. 1. Ants' beheviour in probabilistically chosen paths which lead to food source

in which they have to decide between two possible paths for both getting to and returning from their destination. Since they have no clue about which is the best choice, they have to pick the path randomly. It can be expected that on average half of the ants will decide to go on one path and the rest choose to travel on the other path. Suppose that all ants walk at approximately the same speed, the pheromone deposited will accumulate more on the shorter path.

After a short transitory period the difference between the amounts of deposited pheromone on the two paths is sufficiently large so as to influence the decision of other ants arriving at the decision point. New ants will thus prefer to choose the shorter path since at the decision point they perceive more pheromone. To solve the TSP where the best tour (minimising the travelling distance) is to be reported, the iterative cycle is performed. During each cycle, an ant (computer agent) is set to start off from a city (node). Utilising the above ants' probabilistic behaviour, accumulated pheromone appears on links between nodes. Ants shall prefer to take links that they have not traversed before. The iterative cycle will be repeated until reaching a given maximum number of cycles. At the end, all ants will use the shorter path. If the ants have to complete a circular tour covering n different destinations without visiting order preference, the emerged best tour will be a solution to the n-city TSP.

3 DNA Fragment Assembly Using ACS and NNH

During an assembly process, the fragments are aligned in order to create a consensus sequence that represents the original or parent DNA strands. An alignment between two fragments can be created if there is a portion from each fragment that together can produce a match between either the same-base ordering sequences or the complementary-base ordering sequences. The alignments of fragments are schematically displayed in Fig. 2. The number of matching bases between two aligned fragments together with penalties from mismatches and gaps are generally referred to as the overlap score. The overlap score for an alignment between two fragments can be calculated using a Smith-Waterman algorithm [15]. If the search for a possible alignment between a given fragment and other fragments returns either a relatively low or a zero overlap score, there

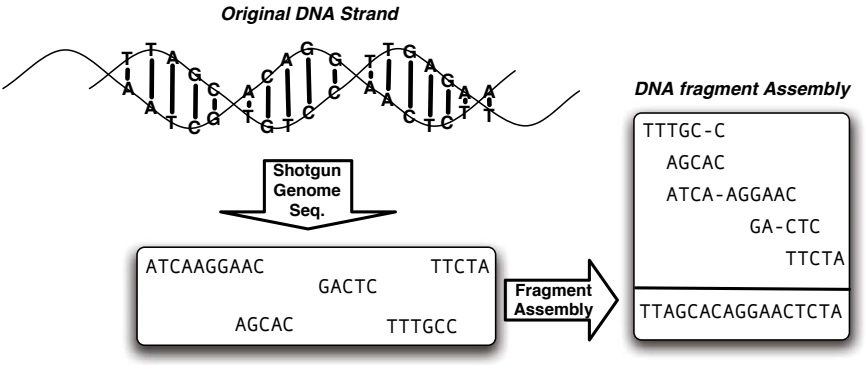


Fig. 2. DNA fragment assembly process

will be gaps in the consensus sequence. In such a case, the consensus sequence will contain multiple disjoint sequences called contiguous sequences or contigs.

From a combinatorial optimisation viewpoint, the construction of a consensus sequence is similar to that of a tour in the TSP. This is because each fragment would have to be in a *specific fragment ordering sequence* in order for the formation of a consensus sequence to take place. The explanation of the ant colony system algorithm given in section 2 can then be applied to a DNA fragment assembly problem by utilising the overlap score, which provides information regarding how well two fragments can fit together. This underlying score can be regarded as the inverse of the distance between two cities in the TSP problem. In other words, the target of a TSP is to find the shortest circular tour which links the cities together while the aim of a DNA fragment assembly problem is to maximise the sum of overlap scores between consecutive fragments in the ordering sequence.

However, a DNA fragment assembly problem is a special kind of symmetric TSPs. In brief, distances between cities r and s in the forward and backward journeys are equal in a symmetric TSP. A factor that makes a DNA fragment assembly problem a special form of symmetric TSPs is the consideration on the original parent DNA strand at which the fragment came. Each fragment used in the assembly process has an equal probability of coming from one of the two parent DNA strands. With different assumptions on the origin of the fragment, the resulting overlap score would also be different. With this factor, there are four possible configurations for obtaining the overlap score between two fragments. The summary of four alignment configurations is given in Table 1. From Table 1, during the use of the ACS algorithm if an ant is at fragment r where the fragment is assumed to come from the forward DNA strand, the only possible configurations for an alignment with fragment s are configurations 1 and 2. On the other hand, if the fragment r comes from the complementary strand where the order of base reads must always be in reverse, the feasible configurations for an alignment with fragment s are configurations 3 and 4. It remains that the only difference between a TSP and a DNA fragment assembly

Table 1. Four alignment configurations between two fragments

Configuration	Assumption about the Strand of Origin	
	Fragment r	Fragment s
1	Forward	Forward
2	Forward	Reverse complement
3	Reverse complement	Forward
4	Reverse complement	Reverse complement

problem is that there would not be a proper alignment between the first and the last fragments in the consensus sequence that is comparable to the connection between the first and the last cities in the TSP solution.

Many objective functions can be used to represent correctness of the obtained consensus sequence. The objective function investigated is a minimisation function, which is a combination between the number of contigs and the difference in length between the longest and the shortest contigs. With the use of this objective function, the solution that has the lesser number of contigs will be regarded as the better solution. The locations of the beginning and the end of each contig in the fragment ordering sequence are the locations where the overlap score between two consecutive fragments is lower than a threshold value. However, more than one solution generated may have the same number of contigs. If this is the case, the solution that is the better solution is the one where the difference between the length of the longest and the shortest contigs is minimal. This part of the objective function is derived from the desire that the ultimate goal solution is the one with either only one contig or the fewest possible number of contigs where each contig is reasonably long.

We make use of a nearest neighbour heuristic (NNH) rule to provide the initial solution to the ACS algorithm and the ACS local pheromone updating factor [14]. Let fragment r be the one which is randomly chosen by an ant from the fragment set. The origin of fragment r (forward or reverse complementary strands) is also randomly chosen. The candidate fragment s that is best aligned with fragment r is then located from the remaining fragments. This will provide both the overlap score and the alignment configuration. The process of identifying the candidate fragment for an alignment with fragment s is then carried out and the process continues until the last fragment has been used in the construction of the fragment ordering sequence.

Recall that a fragment that has already been used in the fragment ordering sequence will be removed from the remaining fragment list. During the last stage of the assembly process, the overlap score between the last and the first fragments in the ordering sequence will be the value obtained using the alignment configuration enforced by the choices on the strand of origin for both fragments. Since the consensus sequence is not circular, the circular fragment ordering sequence obtained will be split at the location where the overlap score between two fragments in the ordering sequence is minimal. From the procedure given,

the choice of the first fragment in the ordering sequence will dictate how a consensus sequence is formed. For an assembly problem with n fragments, a total of n consensus sequences can be generated using the NNH rule. Notice that the strand of origin of the first fragment has no influence on the number of maximum possible unique solutions that can be generated. This is because changing the strand of origin of the first fragment will simply result in the reversal of the strand of origin of all fragments in the ordering sequence.

4 Case Studies

In this paper, the data sets are obtained from a GenBank database at the National Center for Biotechnology Information (NCBI) [16]. The parent DNA strands are extracted from the human chromosome 3 where the strands with the sequence length ranging from 21K to 83K base pairs are utilised. Each fragment is unclipped (low quality base reads are retained) and has the total number of bases between 700 and 900. This means that the fragments contain sequencing errors generally found in any experiments. The data set is prepared such that the consensus sequence contains either one contig or multiple contigs. The summary of the data set descriptions is given in Table 2. From Table 2, coverage is the average number of fragments covering each base pair on the parent strands.

Table 2. Information on the data set

Accession Number	AC023501		AC023159		AC005903		AC026318	
Base Pair	20,824		34,680		63,949		83,181	
Case Study	1	2	3	4	5	6	7	8
Coverage	10	10	5	5	7	6	7	7
Number of Fragments	368	367	279	269	611	591	709	708
Gaps	0	1	0	6	0	1	0	1

5 Methods, Results and Discussions

The ACS algorithm, the NNH rule and a CAP3 program [3], which is a standard assembly program, have been applied to all eight case studies. In the case of the NNH search, all possible n solutions with different starting fragments are generated where n is the number of fragments. The solutions are obtained using the sum of overlap scores as the maximisation objective. The best solution is then picked where contigs are produced by assembling aligned fragments together and applying a majority-vote rule, as illustrated in Fig. 2, for the base calling purpose. Next, an attempt on DNA contig assembly is made where the NNH rule is still in use. Similar to the early assembly procedure, all possible l solutions are generated this time where l is the number of contigs from the primary assembly stage and the best solution among l solutions are chosen as the final solution. In contrast, the ACS algorithm runs with the minimisation

Table 3. Number of contigs from the solutions produced by the NNH+NNH approach, the ACS+NNH approach and the CAP3 program

Problem	Number of Contigs			
	Parent Strand	NNH+NNH	ACS+NNH	CAP3
AC023501 (21K bp)				
No gaps	1	1	1	3
With gaps	2	2	2	4
AC023159 (35K bp)				
No gaps	1	11	5	10
With gaps	7	18	11	9
AC005903 (64K bp)				
No gaps	1	14	11	3
With gaps	2	14	2	3
AC026318 (83K bp)				
No gaps	1	15	15	25
With gaps	2	15	15	25

objective described in section 3 are repeated ten times in each case study. In this investigation, the parameter setting for the ACS algorithm is the recommended setting for solving symmetric travelling salesman problems given in Dorigo and Gambardella [14]. During each ACS run, the initial solution used is randomly chosen from all n solutions produced by the NNH rule. After all ACS runs are finished, the best solution is picked and contigs are obtained by assembling fragments together. The contig assembly is then commenced where the NNH rule is applied. It is noted that since the CAP3 program is deterministic in nature, the program is executed only one time for each case study. From the assembly results obtained, two discussion topics can be given: the number of contigs in the assembly solutions and the quality of the solutions.

5.1 Number of Contigs in the Assembly Solutions

The numbers of contigs obtained from the NNH+NNH approach, the ACS+NNH approach and the CAP3 program, are reported in Table 3. From Table 3, the CAP3 program outperforms the ACS+NNH approach in cases 4 and 5 while the ACS+NNH approach is the best technique in the remaining cases. The results also indicate that as the problem size increases, the number of contigs produced by the ACS+NNH approach also increases. On the other hand, there is no correlation between the problem size and the number of contigs in the the CAP3 solutions. In overall, the performance of the ACS+NNH approach is higher than that of the CAP3 program. Furthermore, both NNH+NNH and ACS+NNH techniques have the same performance in cases 1, 2, 7 and 8 while the ACS+NNH approach has a higher performance in cases 3, 4, 5 and 6. This means there is a range on the problem size where the ACS+NNH approach is better than the NNH+NNH approach.

Table 4. Assembly errors expressed in terms of the sum of substitution and insertion/deletion errors, and the coverage error

Problem	Substitution & Indel Errors (%)			Coverage Error (%)		
	NNH+NNH	ACS+NNH	CAP3	NNH+NNH	ACS+NNH	CAP3
AC023501 (21K bp)						
No gaps	1.63	1.89	0.13	0.00	0.00	0.00
With gaps	1.62	1.43	0.10	0.00	0.19	0.00
AC023159 (35K bp)						
No gaps	1.17	1.57	0.21	8.48	8.22	6.98
With gaps	N/A	N/A	0.22	N/A	N/A	1.10
AC005903 (64K bp)						
No gaps	1.08	1.01	0.10	0.45	0.45	2.38
With gaps	1.02	0.97	0.11	0.47	1.22	2.06
AC026318 (83K bp)						
No gaps	1.19	1.11	0.39	12.42	7.44	11.08
With gaps	1.19	1.09	0.40	12.23	7.49	10.87

5.2 Quality of the Assembly Solutions

The quality of a contig is measured by the base difference between the parent DNA sequence and the contig of interest. This difference is expressed in terms of three types of assembly error: a substitution error, an insertion/deletion (indel) error and a coverage error. A substitution error appears when a base in one of two aligned sequences—a parent DNA sequence and a contig in this case—does not match its counterpart in the other sequence. When a base in one aligned sequence seems to have been deleted as the result of a divergence of the sequence from its counterpart, such absence is labelled as a deletion error in the derived sequence. On the other hand, when a base appears to have been inserted to produce a longer sequence, an insertion error is labelled in the augmented sequence. A deletion in one sequence can thus be viewed as an insertion in the other sequence. Hence, these two types of error are generally referred to together as an insertion/deletion error. In contrast to substitution and insertion/deletion errors, a coverage error is detected when there are bases in the parent DNA sequence, which are located outside the part of contig that best matches the parent sequence and thus not covered by any contigs. These assembly errors, expressed in terms of the percentage of errors out of the total number of bases in the parent sequence, are tabulated in Table 4. The sum of substitution and insertion/deletion errors from the ACS+NNH approach is higher than that of the CAP3 program in all case studies. However, in the first four case studies, the coverage errors from the CAP3 program are either lower than or equal to that from the ACS+NNH approach while the solutions that have lower coverage errors in the last four case studies are produced by the ACS+NNH approach. It is also noticeable that the errors from the NNH+NNH and ACS+NNH approaches are very similar in all case studies except for the last two cases where

the coverage errors of the solutions from the ACS+NNH approach are lower. In terms of solution quality, the right combination between the ACS algorithm and the CAP3 program may yield contigs that have even lower assembly errors.

6 Conclusions

A DNA fragment assembly problem is treated as a TSP where a fragment ordering sequence conveys a tour that covers all cities while the overlap score between two aligned fragments in the ordering sequence is regarded as the inverse of the distance between two cities. The proposed ACS+NNH procedure was compared with a CAP3 program [3]. The results suggest that the solutions produced by CAP3 contain a higher number of contigs than the solutions generated by the ACS+NNH procedure and the quality of the combined ACS/NNH solutions is higher than that of the CAP3 solutions when the problem size is large.

Acknowledgements

This work was supported by the Thailand Toray Science Foundation (TTSF) and National Science and Technology Development Agency (NSTDA) under the Thailand Graduate Institute of Science and Technology (TGIST) programme. The authors acknowledge Prof. Xiaogiu Huang at the Iowa State University for providing an access to the CAP3 program.

References

1. Applewhite, A.: Mining the genome. IEEE Spectrum **39**(4) (2002) 69–71
2. Pop, M., Salzberg, S.L., Shumway, M.: Genome sequence assembly: Algorithms and issues. Computer **35**(7) (2002) 47–54
3. Huang, X., Madan, A.: CAP3: A DNA sequence assembly program. Genome Research **9**(9) (1999) 868–877
4. Green, P.: Phrap documentation. Phred, Phrap, and Consed **www.phrap.org** (2004)
5. Ferreira, C.E., de Souza, C.C., Wakabayashi, Y.: Rearrangement of DNA fragments: A branch-and-cut algorithm. Discrete Applied Mathematics **116**(1-2) (2002) 161–177
6. Batzoglou, S., Jaffe, D., Stanley, K., Butler, J., Gnerre, S., Mauceli, E., Berger, B., Mesirov, J.P., Lander, E.S.: ARACHNE: A whole-genome shotgun assembler. Genome Research **12**(1) (2002) 177–189
7. Kececioglu, J.D., Myers, E.W.: Combinatorial algorithms for DNA sequence assembly. Algorithmica **13**(1-2) (1995) 7–51
8. Pevzner, P.A., Tang, H., Waterman, M.S.: An Eulerian path approach to DNA fragment assembly. Proceedings of the National Academy of Sciences of the United States of America **98**(17) (2001) 9748–9753
9. Burks, C., Engle, M., Forrest, S., Parsons, R., Soderlund, C., Stolorz, P.: Stochastic optimization tools for genomic sequence assembly. In: Adams, M.D., Fields, C., Venter, J.C. (eds.): Automated DNA Sequencing and Analysis. Academic Press, London, UK (1994) 249–259

10. Parsons, R.J., Forrest, S., Burks, C.: Genetic algorithms, operators, and DNA fragment assembly. Machine Learning **21**(1-2) (1995) 11–33
11. Parsons, R.J., Johnson, M.E.: A case study in experimental design applied to genetic algorithms with applications to DNA sequence assembly. American Journal of Mathematical and Management Sciences **17**(3-4) (1997) 369–396
12. Kim, K., Mohan, C.K.: Parallel hierarchical adaptive genetic algorithm for fragment assembly. In: Proceedings of the 2003 Congress on Evolutionary Computation, Canberra, Australia (2003) 600–607
13. Angeleri, E., Apolloni, B., de Falco, D., Grandi, L.: DNA fragment assembly using neural prediction techniques. International Journal of Neural Systems **9**(6) (1999) 523–544
14. Dorigo, M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation **1**(1) (1997) 53–66
15. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. Journal of Molecular Biology **147**(1) (1981) 195–197
16. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Wheeler, D.L.: GenBank. Nucleic Acids Research **33** (2005) D34–D38

Application of Bayesian Confirmation Measures for Mining Rules from Support-Confidence Pareto-Optimal Set

(Invited Paper)

Roman Slowinski[1,2], Izabela Brzezinska[1], and Salvatore Greco[3]

[1] Institute of Computing Science Poznan University of Technology, 60-965 Poznan, Poland

[2] Institute for Systems Research, Polish Academy of Sciences, 01-447 Warsaw, Poland
{Izabela.Brzezinska, Roman.Slowinski}@cs.put.poznan.pl

[3] Faculty of Economics, University of Catania, Corso Italia, 55, 95129 Catania, Italy
salgreco@unict.it

Abstract. We investigate a monotone link between Bayesian confirmation measures and rule support and confidence. In particular, we prove that two confirmation measures enjoying some desirable properties are monotonically dependent on at least one of the classic dimensions being rule support and confidence. As the confidence measure is unable to identify and eliminate non-interesting rules, for which a premise does not confirm a conclusion, we propose to substitute the confidence for one of the considered confirmation measures. We also provide general conclusions for the monotone link between any confirmation measure enjoying some desirable properties and rule support and confidence.

1 Introduction

Knowledge patterns discovered from data are usually expressed in a form of "*if..., then...*" rules. They are consequence relations representing mutual relationship, association, causation, etc. between independent and dependent attributes. Typically, the number of rules generated from massive datasets is very large, and only a small portion of them is likely to be useful. In order to measure the relevance and utility of the discovered patterns, quantitative measures, also known as attractiveness or interestingness measures (metrics), have been proposed and studied. Measures such as confidence and support, gain [10], conviction [3], etc. have been introduced to capture different characteristics of rules. Among widely studied interestingness measures, there is, moreover, a group of Bayesian confirmation measures, which quantify the degree to which a piece of evidence built of the independent attributes provides "evidence for or against" or "support for or against" the hypothesis built of the dependent attributes [9]. An important role is played by a confirmation measure denoted in [9] and other studies by f, and by a confirmation measure s proposed by [6]. Both of them have a valuable property of monotonicity (M) introduced in [12].

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1018–1026, 2006.
© Springer-Verlag Berlin Heidelberg 2006

Bayardo and Agrawal [2] have proved that for a class of rules with fixed conclusion, the upper support-confidence Pareto border (i.e. the set of non-dominated, Pareto-optimal rules with respect to both rule support and confidence) includes optimal rules according to several different interestingness measures, such as gain, Laplace [7], lift [13], conviction [3], an unnamed measure proposed by Piatetsky-Shapiro [16]. This practically useful result allows to identify, the most interesting rules according to several interestingness measures by solving an optimized rule mining problem with respect to rule support and confidence only.

As shown in [12], the semantics of the scale of confidence is not as meaningful as that of confirmation measures. Moreover, it has been analytically shown in [4] that there exist a monotone link between some confirmation measures on one side, and confidence and support, on the other side. In consequence, we propose in this paper, two alternative approaches to mining interesting rules. The first one consists in searching for a Pareto-optimal border with respect to rule support and confirmation measure f, the second concentrates on searching for a Pareto-optimal border with respect to rule support and confirmation measure s.

The paper is organized as follows. In the next section, there are preliminaries on rules and their quantitative description. In section 3, we investigate the idea and the advantages of mining rules constituting Pareto-optimal border with respect to support and confirmation measure f. Section 4 concentrates on the proposal of mining Pareto-optimal rules with respect to support and confirmation measure s. In section 5, we generalize the approaches from sections 3 and 4 to a broader class of confirmation measures. The paper ends with conclusions.

2 Preliminaries

Discovering rules from data is a domain of inductive reasoning. To start inference it uses information about a sample of larger reality. This sample is often given in a form of an information table, containing objects of interest characterized by a finite set of attributes. Let us consider information table $S = (U, A)$, where U and A are finite, non-empty sets called *universe* and *set of attributes*, respectively. One can associate a formal language L of logical formulas with every subset of attributes. Conditions for a subset $B \subseteq A$ are built up from attribute-value pairs (a,v), where $a \in B$ and $v \in V_a$ (set V_a is a domain of attribute a), using logical connectives $\neg$ (not), $\wedge$ (and), $\vee$ (or). A *decision rule* induced from S and expressed in L is denoted by $\phi \rightarrow \psi$ (read as "*if* ϕ, *then* ψ") and consists of condition and decision formulas in L, called premise and conclusion, respectively.

In this paper, similarly to [2], we only consider all minimal rules with the same conclusion, which can be induced from a dataset. Let us remind that a rule is minimal if, for a given conclusion, there is no other rule with weaker conditions.

2.1 Monotonicity of a Function in Its Argument

For x belonging to a set ordered by the relation $\succ$ and for the values of g belonging to a set ordered by the relation $\leq$, a function $g(x)$ is understood to

be *monotone* (resp. *anti-monotone*) in x, if $x_1 \prec x_2$ implies that $g(x_1) \leq g(x_2)$ (resp. $g(x_1) \geq g(x_2)$).

2.2 Support and Confidence Measures of Rules

With every rule induced from information table S, measures called *support* and *confidence* can be associated. The *support* of condition ϕ, denoted as $sup(\phi)$, is equal to the number of objects in U having property ϕ. The support of rule $\phi \rightarrow \psi$, denoted as $sup(\phi \rightarrow \psi)$, is equal to the number of objects in U having both property ϕ and ψ; for those objects, both premise ϕ and conclusion ψ evaluate to true.

The *confidence* of a rule (also called *certainty*), denoted as $conf(\phi \rightarrow \psi)$, is defined as follows:

$$conf(\phi \rightarrow \psi) = \frac{sup\,(\phi \rightarrow \psi)}{sup\,(\phi)},\ sup(\phi) > 0 \tag{1}$$

Note, that it can be regarded as a conditional probability $Pr(\psi|\phi)$ with which conclusion ψ evaluates to true, given that premise ϕ evaluates to true, however, expressed in terms of frequencies.

2.3 Bayesian Confirmation Measures *f* and *s*

In general, confirmation measures quantify the strength of confirmation that premise ϕ gives to conclusion ψ. All confirmation measures take (desired) positive values in situations where the conclusion of the rule is verified more often when its premise is verified, rather than when its premise is not verified. For the confirmation measures a desired property of monotonicity (M) was proposed in [12]. This monotonicity property says that, given an information system S, a confirmation measure is a function non-decreasing with respect to $sup(\phi \rightarrow \psi)$ and $sup(\neg\phi \rightarrow \neg\psi)$, and non-increasing with respect to $sup(\neg\phi \rightarrow \psi)$ and $sup(\phi \rightarrow \neg\psi)$. Among confirmation measures that have property (M) there is confirmation measure f [9] and confirmation measure s [6].

The confirmation measures f and s are defined as follows:

$$f(\phi \rightarrow \psi) = \frac{Pr(\phi|\psi) - Pr(\phi|\neg\psi)}{Pr(\phi|\psi) + Pr(\phi|\neg\psi)}, \tag{2}$$

$$s(\phi \rightarrow \psi) = Pr(\psi|\phi) - Pr(\psi|\neg\phi). \tag{3}$$

Taking into account that conditional probability $Pr(\circ|*) = conf(* \rightarrow \circ)$, confirmation measures f and s can be expressed as:

$$f(\phi \rightarrow \psi) = \frac{conf(\psi \rightarrow \phi) - conf(\neg\psi \rightarrow \phi)}{conf(\psi \rightarrow \phi) + conf(\neg\psi \rightarrow \phi)}, \tag{4}$$

$$s(\phi \rightarrow \psi) = conf(\phi \rightarrow \psi) - conf(\neg\phi \rightarrow \psi). \tag{5}$$

2.4 Partial Order on Rules in Terms of Two Interestingness Measures

Let us denote by $\preceq_{AB}$ a partial order on rules in terms of any two different interestingness measures A and B. The partial order $\preceq_{AB}$ can be decomposed into its asymmetric part $\prec_{AB}$ and symmetric part $\sim_{AB}$ in the following manner: given two rules r_1 and r_2, $r_1 \prec_{AB} r_2$ if and only if

$$\begin{array}{l} A(r_1) \leq A(r_2) \wedge B(r_1) < B(r_2), \text{or} \\ A(r_1) < A(r_2) \wedge B(r_1) \leq B(r_2); \end{array} \tag{6}$$

moreover, $r_1 \sim_{AB} r_2$ if and only if

$$A(r_1) = A(r_2) \wedge B(r_1) = B(r_2). \tag{7}$$

2.5 Implication of a Total Order $\preceq_t$ by Partial Order $\preceq_{AB}$

Application of some measures that quantify the interestingness of a rule induced from an information table S creates a total order, denoted as $\preceq_t$, on those rules. In particular, measures such as gain, Laplace, lift, conviction, one proposed by Piatetsky-Shapiro, or confirmation measures f and s result in such a total order on the set of rules with a fixed conclusion, ordering them according to their interestingness value.

A total order $\preceq_t$ *is implied by* partial order $\preceq_{AB}$ if:

$$\begin{array}{l} r_1 \preceq_{AB} r_2 \Rightarrow r_1 \preceq_t r_2, \quad \text{and} \\ r_1 \sim_{AB} r_2 \Rightarrow r_1 \sim_t r_2. \end{array} \tag{8}$$

It has been proved by Bayardo and Agrawal in [2] that if a total order $\preceq_t$ is implied by support-confidence partial order $\preceq_{sc}$, then the optimal rules with respect to $\preceq_t$ can be found in the set of non-dominated rules with respect to rule support and confidence. Thus, when one proves that a total order defined over a new interestingness measure is implied by $\preceq_{sc}$, one can concentrate on discovering non-dominated rules with respect to rule support and confidence. Moreover, Bayardo and Agrawal have shown in [2] that the following conditions are sufficient for proving that a total order $\preceq_t$ defined over a rule value function $g(r)$ is implied by partial order $\preceq_{AB}$:

$g(r)$ is monotone in A over rules with the same value of B, and
$g(r)$ is monotone in B over rules with the same value of A.

3 Pareto-Optimal Border with Respect to Rule Support and Confirmation Measure f

Due to the semantic importance and utility of confirmation measure f, a verification of the monotonicity of confirmation measure f in rule support and confidence has been conducted in [4]. It has been proved that rules maximizing

confirmation measure f can be found on the Pareto-optimal support-confidence border. However, the utility of confirmation measure f outranks the utility of confidence. The confidence measure has no means to show, that the rule is useless when its premise disconfirms the conclusion. Such situation is expressed by a negative value of any confirmation measure, thus useless rules can be filtered out simply by observing the confirmation measure's sign. Therefore, we find it interesting to propose a new Pareto-optimal border – with respect to rule support and confirmation measure f.

An analysis of the monotonicity of confidence in rule support for a fixed value of confirmation f, as well as in confirmation f for a fixed value of rule support, has been performed. The following theorems have been proved in [5].

Theorem 1. *Confidence is monotone in confirmation measure f.*

Theorem 2. *Confidence is independent of rule support, and therefore monotone in rule support, when the value of confirmation measure f is held fixed.*

It follows from the above results that rules optimal in confidence lie on the Pareto-optimal border with respect to rule support and confirmation measure f. Even more, the Pareto-optimal border with respect to support and onfirmation measure f is identical with the Pareto-optimal border with respect to support and confidence.

Consequently, other interestingness measures that are monotone in confidence, must also be monotone in confirmation measure f, due to the monotone link between confidence and confirmation measure f. Thus, all the interestingness measures that were found on the support-confidence Pareto-optimal border shall also reside on the Pareto-optimal border with respect to rule support and confirmation measure f. We find it valuable to combine those two measures in the border, as confirmation measure f is independent of rule support, and rules that have high values of confirmation measure f are often characterized by small values of the rule support.

A computation experiment showing rules in confirmation measure f and rule support has been conducted. A real life dataset containing information about technical state of buses was analyzed. The set consisted of 76 objects described by 8 criteria and divided into 3 decision classes. For one of those classes a set of all rules was generated. The values of confirmation measure f and rule support for those rules were placed on Fig.1. It can be easily observed that the Pareto-optimal set of rules (marked in Fig.1 by squares) includes rules maximizing such interestingness measures as confidence, Laplace, lift (marked in Fig.1 by asterisk), Piatetsky-Shapiro (marked in Fig.1 by a cross).

For rules with a fixed conclusion, mining the set of non-dominated rules with respect to rule support and confirmation measure f will identify rules optimal according to such interestingness measures as confidence, conviction, lift, Laplace, Piatetsky-Shapiro, gain, etc. However, if those non-dominated rules are characterized by a negative value of confirmation measure f, then they must be discarded because in those rules the premise just disconfirms the conclusion. A final set of rules representing "'the best"' patterns discovered from the

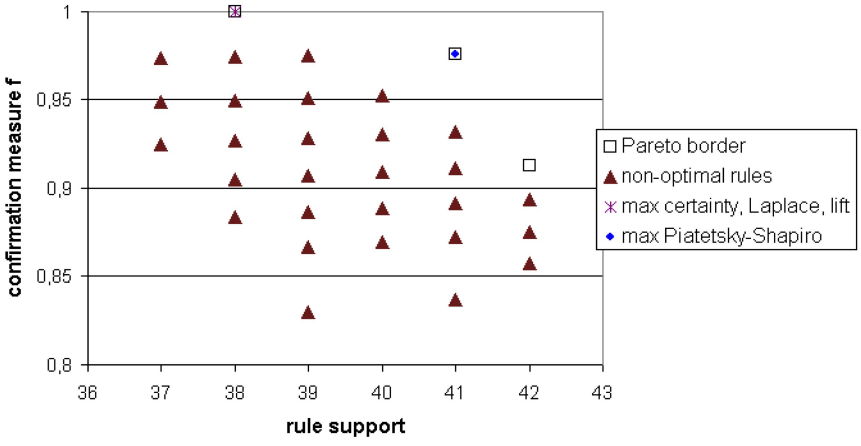


Fig. 1. Pareto-optimal border with respect to rule support and confirmation measure f includes rules being optimal in many other measures (technical state of buses dataset)

whole dataset shall be a union of all the non-negative-in-f rules from all the Pareto-optimal borders (all possible conclusions) with respect to support and confirmation measure f.

4 Rules Optimal with Respect to Confirmation Measure s

The second confirmation measure that came into the scope of our interest was confirmation measure s. Similarly to confirmation measure f, it also has the desirable property of monotonicity (M). On the contrary to confirmation measure f, however, it is dependent on both rule support and confidence. The monotonicity of confirmation measure s in confidence for a fixed value of support, as well as in rule support for a fixed value of confidence, has been analyzed. The following theorems have been proved in [5].

Theorem 3. *When the rule support value is held fixed, then confirmation measure s is monotone in confidence.*

Theorem 4. *When the confidence value is held fixed, then:*

- *confirmation measure s is monotone in rule support if and only if $s \geq 0$,*
- *confirmation measure s is anti-monotone in rule support if and only if $s < 0$.*

As rules with negative values of confirmation measure s are discarded from consideration, the result from *Theorem 4* states the monotone relationship just in the interesting subset of rules. Since confirmation measure s has the property of monotonicity (M), we propose to generate interesting rules by searching for rules maximizing confirmation measure s and support, i.e. substituting the confidence in the support-confidence Pareto-optimal border with the confirmation measure

s and obtaining in this way a support-confirmation-s Pareto-optimal border. This approach differs from the idea of finding the Pareto-optimal border according to rule support and confirmation measure f, because support-confirmation-f Pareto-optimal border contains the same rules as the support-confidence Pareto-optimal border, while, in general, the support-confirmation-s Pareto-optimal border can differ from the support-confidence Pareto-optimal border. Moreover, as measure f, unlikely to s, is a satisfying confirmation measure with respect to the *property of symmetry* verified in [8], mining the Pareto-optimal border with respect to rule support and confirmation measure f still remains a good alternative idea.

5 Rules Optimal with Respect to Any Confirmation Measure Having the Property of Monotonicity (M)

A general analysis of the monotonicity of any confirmation measure that enjoys the property of monotonicity (M) has also been conducted.

Let us use the following notation:

$$a = sup(\phi \rightarrow \psi),\ b = sup(\neg\phi \rightarrow \psi),\ c = sup(\phi \rightarrow \neg\psi),\ d = sup(\neg\phi \rightarrow \neg\psi).$$

Let us consider a Bayesian confirmation measure $F(a, b, c, d)$ being differentiable and having the property of monotonicity (M). The following theorems have been proved in [5].

Theorem 5. *When the value of rule support is held fixed, then the confirmation measure $F(a, b, c, d)$ is monotone in confidence.*

Theorem 6. *When the value of confidence is held fixed, then the confirmation measure $F(a, b, c, d)$ is monotone in rule support if:*

$$\frac{\partial F}{\partial c} = \frac{\partial F}{\partial d} = 0 \quad or \quad \frac{\frac{\partial F}{\partial a} - \frac{\partial F}{\partial b}}{\frac{\partial F}{\partial d} - \frac{\partial F}{\partial c}} \geq \frac{1}{conf(\phi \rightarrow \psi)} - 1. \tag{9}$$

It is worth noting, that, due to *Theorem 6*, all those confirmation measures that are independent of $sup(\phi \rightarrow \neg\psi)$ and $sup(\neg\phi \rightarrow \neg\psi)$ are found monotone in rule support when the value of confidence is kept unchanged.

Theorem 5 and *Theorem 6* outline an easy method of verification of existence of the monotone link between any confirmation measure with the property of monotonicity (M), and rule support and confidence. Confirmation measures that positively undergo such verification are, in our opinion, good candidates for substituting the confidence dimension in the Pareto-optimal border with respect to rule support and confidence proposed by Bayardo and Agrawal in [2]. Thanks to the monotonicity of a confirmation measure in rule support and confidence, a monotone link of that confirmation measure with other interestingness measures such as lift, gain, Laplace, etc. is assured. Therefore, the Pareto-optimal border with respect to rule support and a confirmation measure includes rules optimal

according to the same metrics as the support-confidence Pareto-optimal border. Due to the fact that the scale of confirmation measures is more useful than that of confidence, we propose searching for the non-dominated set of rules with respect to rule support and a confirmation measure with the propery of monotonicity (M). We find confirmation measure f particularly valuable for its property of monotonicity (M) and for being a satisfying measure with respect to the property of symmetry, and confirmation measure s for its property of monotonicity (M) and its simplicity.

6 Conclusions

Bayardo and Agrawal have opted in [2] for an approach to mining interesting rules based on extracting a Pareto-optimal border with respect to rule support and confidence. We have analyzed and described the monotone link between the confirmation measures f and s, and rule support and confidence. This analysis has also been extended to a more general class of all the confirmation measures that have the property of monotonicity (M). The results show that it is reasonable to propose a new approach in which we search for a Pareto-optimal border with respect to rule support and a confirmation measure, in particular, we are in favor of confirmation measure f or s. Consequently, our future research will concentrate on adapting the "APRIORI" algorithm [1], based on the frequent itemsets, for mining most interesting association rules with respect to rule support and either confirmation measure f or s.

Acknowledgements

The research of the first two authors has been supported by the Polish Ministry of Education and Science. The third author wishes to acknowledge financial support from the Italian Ministry of Education, University and Scientific Research (MIUR).

References

1. Agrawal, R., Imielinski, T. and Swami, A.: Mining associations between sets of items in massive databases. [In]: *Proc. of the 1993 ACM-SIGMOD Int'l Conf. on Management of Data*, 207-216 (1993).
2. Bayardo, R.J., Agrawal, R.: Mining the most interesting rules. [In]: *Proc. of the Fifth ACM-SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 145-154 (1999).
3. Brin, S., Motwani, R., Ullman, J. and Tsur, S.: Dynamic itemset counting and implication rules for market basket data. [In]: *Proc. of the 1997 ACM-SIGMOD Int'l Conf. on the Management of Data*, 255-264 (1997).
4. Brzezinska, I., Slowinski, R.: Monotonicity of a Bayesian confirmation measure in rule support and confidence. [In]: T. Burczynski, W. Cholewa, W. Moczulski (Eds.) *Recent Developments in Artificial Intelligence Methods*, AI-METH Series, Gliwice, 39-42 (2005).

5. Brzezinska, I., Greco, S., Slowinski, R.: *Investigation of monotone link between confirmation measures and rule support and confidence*. Research Report RA-025/05, Institute of Computing Science, Poznan University of Technology, Poznan, (2005).
6. Christensen, D.: Measuring confirmation. *Journal of Philosophy XCVI*, 437-461 (1999).
7. Clark, P. and Boswell, P.: Rule induction with CN2: Some recent improvements. [In]: *Machine Learning: Proc. of the Fifth European Conference*, 151-163 (1991).
8. Eells, E., Fitelson, B.: Symmetries and asymmetries in evidential support. *Philosophical Studies*, 107 (2): 129-142 (2002).
9. Fitelson, B.: *Studies in Bayesian Confirmation Theory*. Ph.D. Thesis, University of Wisconsin, Madison (2001).
10. Fukuda, T., Morimoto, Y., Morishita, S. and Tokuyama, T.: Data Mining using two-dimensional optimized association rules: scheme, algorithms, and visualization. [In]: *Proc. of the 1996 ACM-SIGMOD Int'l Conf. on the Management of Data*, 13-23 (1996).
11. Good, I.J.: The best explicatum for weight of evidence. *Journal of Statistical Computation and Simulation*, 19: 294-299 (1984).
12. Greco, S., Pawlak, Z. and Slowinski, R.: Can Bayesian confirmation measures be useful for rough set decision rules? *Engineering Applications of Artificial Intelligence*, 17: 345-361 (2004).
13. International Business Machines: *IBM Intelligent Miner User's Guide*, Version 1, Release 1 (1996).
14. Morimoto, Y., Fukuda, T., Matsuzawa, H., Tokuyama, T. and Yoda, K.: Algorithms for mining association rules for binary segmentation of huge categorical databases. [In]: *Proc. of the 24^{th} Very Large Data Bases Conf.*, 380-391 (1998).
15. Morishita, S.: On classification and regression. [In]: *Proc. of the First Int'l Conf. on Discovery Science – Lecture Notes in Artificial Intelligence*, 1532: 40-57 (1998).
16. Piatetsky-Shapiro, G.: Discovery, Analysis, and Presentation of Strong Rules. Chapter 12 [in]: *Knowledge Discovery in Databases*, AAAI/MIT Press (1991).

Cognitive Analysis Techniques in Business Planning and Decision Support Systems
(Invited Paper)

Ryszard Tadeusiewicz[1], Lidia Ogiela[2], and Marek R. Ogiela[1]

[1] AGH University of Science and Technology, Institute of Automatics,
Al. Mickiewicza 30, PL-30-059 Kraków, Poland
{rtad, mogiela}@agh.edu.pl
[2] AGH University of Science and Technology, Faculty of Management
logiela@agh.edu.pl

Abstract. This paper presents an entirely original concept of IT systems construction never published before. It supports the **strategic** business decision-making processes based on cognitive analysis methods. The essence of this new approach is in that the **automatic understanding** methods, tested previously in the area of medical image interpretation, enhancing the traditional set of automatic analysis and automatic classification methods, will be used to develop new generation business IT systems. This paper proposes a holistic concept of a new IT economy system, which is a significantly enhanced DSS-type system (*Decision Support Systems*). The essence of innovations introduced in this paper is in adding understandable business data to the analysis process of the said data. Such a system is essentially different from all known DSS-type systems, it is also different from all approaches to intelligent IT system construction, as described in the literature like, for example, those based on neural networks or in the use of expert systems. In this paper we therefore propose an acronym for the new system labelled UBMSS (*Understanding Based Managing Support Systems*). Cognitive methods, on which the UBMSS concept and construct are based, copy the psychological and neurophysiological processes of understanding the analysed data, as they take place in the brain of a competent and particularly gifted man.

1 Introduction

Intelligent IT systems are currently used in practically all scientific research fields as well as in technical and medical solutions and, an area not frequently mentioned, in military applications. Such systems are also functioning more and more frequently with greater success in business, which for its daily work uses earlier generation IT systems, i.e. transaction, register and settlement systems, ones supporting management and facilitating taking decisions. However, the practice of using IT systems in business process management has gone in a different direction. Modern production, service and trading companies as well as banks and dispatch companies use IT systems, often highly integrated ones (currently the

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1027–1039, 2006.
© Springer-Verlag Berlin Heidelberg 2006

norm are ERP-class systems) and data processing systems excellent from the point of view of technology but they do not take significant advantage of the achievements in such disciplines as artificial intelligence or cognitive science. It is precisely in the use of AI and the cognitive approach that one can find quite a significant scope of IT innovations that could constitute the source of competitive advantage.

The premises that led the authors of this paper to formulate some general recommendations concerning the development and implementation of cognitive analysis methods in business systems, in particular systems supporting management, are expressed in the following statements:

- Nowadays the scope of typical IT systems for management needs is extremely wide and it relates, among others, to trade, banking, production, services, logistics and many other fields. In each of these fields typical IT systems are used by **everyone**, their use therefore cannot be a factor constituting competitive advantage.
- IT systems currently in use are dedicated mainly to storing and processing information for reporting purposes. They enable therefore evaluation as to how a given company functions **currently** and what economic results it achieved in the nearer or more remote **past**. Such systems are perfect for making decisions relating to the current running and management of a company, but they are not a good tool to support more strategic decisions or ones that are required to develop drastic reorganisation plans or to change a company's mission. One could say that an attempt to shape a company strategy actively based on data supplied by most currently used IT management systems is comparable with the driving of a car in a situation where the driver can only see that section of road in his rear mirror. In other words the road that he has already travelled.
- IT tools used to support economic short and long-term planning (i.e. econometric models, simulation software and forecasting tools) are, of course, very useful but they have a major disadvantage. Each can support the **evaluation** of any idea the user presents and describes. Yet none of these tools supports the generation of innovative ideas. One could say that IT systems currently used for management needs can answer any question asked but they **cannot answer questions that have not been asked.**

The reason behind the fact that even the most modern IT systems perform the task of strategic management in such an imperfect and poor way is that computers, by storing data and playing with information, focus on the **form** and in no way get to the **content**. They do not even try to mine the **meaning** contained in the information.

However, as discussed above, this management type for which currently used IT systems (for example the ERP-class ones) are best adjusted are rather low-level systems, in particular for tactic management. This is true mainly about routine management in the conditions of stable and continuous company operations. To execute more ambitious tasks, in particular to support a **strategic idea generation**, it is necessary to have a new IT tool that practically does

not yet exist on the market of currently used IT techniques. The concept and construction rules of such new generation business IT systems require scientific research. This paper aims to initiate and inspire such research.

The details of the proposed concept will be presented in chapter 3. Nevertheless one should already suggest now that we are aiming to develop an IT system capable of using both the form and the **content** of business information stored and analysed. Every experienced board member in a large and modern company will confirm that to manage efficiently, in particular to find and implement new concepts it is necessary to **understand** the situation of one's own company as well as that of other companies (those co-operating and competing) and of the whole analysed market segment. Only a businessman who understands the micro- and macroeconomic situation well can have the courage to propose innovative changes, often even revolutionary changes. A lack of understanding weakens the will and courage to act and to take important strategic changes. However, should there be enough courage, a lack of understanding of all signals and data could still lead to disaster rather than success.

Later in this paper we shall try to outline the concept forming the basis for such an automatic economy data understanding system. Such intelligent IT systems based on cognitive science, developed for the needs of economy and management, will be the object of this article. Nevertheless, before we try to describe these systems and propose rules according to which they could operate, let us have a look at what typical business IT systems are used nowadays in almost all companies. This is to define the new concept of new generation systems using the incremental method: by means of showing differences between the new and the commonly known and used systems.

2 Business IT Systems

Currently the IT systems most frequently used in the field of economics and management are aimed mainly at enhancing the processes of information on resources flow management in order to fulfil better the needs of all business process participants. The most important innovation directions, as introduced in IT systems are based on the following requirements:

- System, data and process integration,
- Partial system functions unification,
- Improving access to data base for all organisational units,
- Promoting modern data presentation methods (visualisation) in order to support their analysis,
- Enhancing decision-making processes and decision communication,
- Aiming at module form and openness of the whole system,
- Ensuring a complex character in which the whole system functions,
- Constant improvement of the content and technological advancement,
- Aiming at functional and structural flexibility,
- Ensuring constant compliance with changing system environment elements, in particular with the current legal state, evolving in compliance with the adopted legislative procedures [2],

If we want to propose something new in this article relating to services available for the users of typical tools making up the *business intelligence* term, *i.e. data warehouse* and OLAP tools *(on-line analytical processing)*, we need to first mention the modern facilities that already exist. This is necessary to show what the difference is between them and the approach proposed in this paper. Management experts and designers of modern IT business solutions have known for along time that the more demanding system user will not normally limit his needs to a mere screening of the data obtained and subsequently processed by the system. Therefore for a long time in the proposed software there were also various options to analyse collected data. Those were for example statistical, forecasting options with trend detection features and the use of econometric model capacities. Such tools are undoubtedly very useful and they are readily applied. Nevertheless in IT, more than in any other field the saying that "better is an enemy good" holds true. Currently only statistical or econometric data processing techniques used routinely for their analysis are not enough. More and more often one looks for a deeper sense or the meaning of different data in the context of considered business strategies; this, however, requires a new generation tools.

In the next chapter we shall propose such a new generation IT tool using mathematical linguistics techniques as well as advanced artificial intelligence (in particular the *structural pattern recognition* technique). Based on an analogy with cognitive systems developed for the needs of medical diagnostics for a number of years now, we shall propose a solution leading to the development of a cognitive system capable of understanding automatically business data important for companies. The foundation for such IT systems, in this paper referred to as UBMSS systems, is cognitive data analysis based on the process of understanding and semantic reasoning. It is developed to copy the reasoning process taking place in the human brain. Two elements will be of key importance in the concept presented here:

- Linguistic description of the analysed business data properties,
- Automatic conversion (parsing) of linguistic structures into the meaning sphere, based on the considered business data analysis. An internal (located in the system) knowledge source about the meaning of some specified phenomena and business processes is involved with this process; it is being confronted with the currently observed situation and it leads to an understanding of the situation pursuant to the so-called cognitive resonance [8], whose description is to be found in the next chapter.

3 Cognitive Analysis Basis

From the viewpoint of psychological sciences, in the process of understanding any information obtained by a man, subject to cognitive analysis, there are three stages:

- ***Registration, remembering*** *and* ***coding*** the information obtained.
- ***Preserving*** — a latent stage of natural processes.
- Information ***reproduction*** — its scope covers the remembering, recognition, understanding and the learning of some skills anew [4].

The neurophysiological model of cognitive analysis that we need in this paper is based on the functioning and operation of large brain fragments that can be described by examining (among others) large neural group dynamics attractors (amounting even to millions of neurons), defined in the stimulation area of these neurons.

Even though there are no identical states of brain surface (for example examined with EEG), in the dynamics of its activities one can find constant relations between these attractors, i.e. relatively repetitive dynamic neurophysiological states. Appropriately mathematically interpreted dynamic states of the brain are characterised by some deeper relations, which could have their logical representation. Correspondence between the state of mind and brain does not refer to the electrophysiological surface phenomena of volatile nature but it points to some stability of attractor states.

Let us now try to shift these statements to a cognitive IT system model, of interest for us and used for business purposes. Its task would be to interpret facts based on understanding and reasoning conducted in connection with the semantic content of the processed data.

Every IT system supposed to perform a semantic (directed at the meaning) analysis of a selected object or the basis of the information must contain some **knowledge** necessary to make a correct meaning analysis. Confronting the obtained description of a currently analysed object (that could be, for example, a specified market situation), a description evaluated pursuant to some features characteristic for it, with a set of expectations and hypotheses relating to that state, generated by knowledge based on representation, we obtain the premise to show the real meaning and sense of the said object, that is to **understand it**. This is true for every system capable of understanding any data and information. This is due to that it is always only the knowledge held previously, i.e. the basis to generate system expectations, that can constitute the reference point for semantic analysis of features obtained as a result of the conducted analysis of every object, analysed at the system input. As a result of the combination of certain features of the analysed objects with expectations generated based on knowledge about its semantic content, we find the cognitive resonance phenomenon (Figure 1). That is the key to meaning analysis of objects or information [5].

In accordance with the concept developed by the authors over a number of years, cognitive analysis used in IT systems dedicated to automatic understanding, is based on the syntactic approach. For the purposes of meaning analysis and interpretation of the analysed object it therefore uses a linguistic description. This allow to create the basis for the automatic generation of (potentially) an infinite number of various meanings using just a small set of elements, formal rules (enabling computer application) and axioms; these would form the grounds for an automatic understanding system.

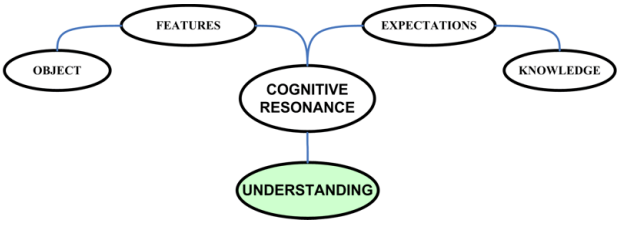


Fig. 1. The cognitive resonance phenomenon in the process of the analysed object understanding

4 UBMSS-Class System: The Concept

We shall now present a proposed structure and operational methods of the already introduced UBMSS-class system. First, in order to systematise our considerations and to establish a reference point, let us recall a traditional (nowadays applied in practice) structure of business IT system application: computers are, obviously, involved since they are the ones that store and process data as well as analyse data in various ways. Information obtained from such computer systems is entirely sufficient for an effective management of business processes at the tactical level (as marked jokingly on figure 2). On the other hand, if we talk about management at the strategic level, we find out that despite automated data collection, storage and analysis, the task of business meaning understanding of the said data is in traditional systems the unique area of people (experts). So is taking and implementation of strategic decisions: this belongs only to people holding appropriate, high positions. The structure of such traditional IT system, as presented on Figure 2 will be the starting point to propose a general structure of an UBMSS-class system.

In such system, whose structure has been presented on figure 3, the initial processes of storing and pre-processing phenomena taking place in the analysed business entity, are analogous to the one we are dealing with in the traditional systems. The only difference is that with the perspective of automatic interpretation of data analysis process results, one can compute and collect a larger number of ratios and parameters since the interpreting automaton will not be dazzled or perplexed by an excess of information. This is what happens when people, interpreting situations, are 'bombed' with hundreds of ratios among which they can hardly find the important ones and then need to make a huge effort in order to interpret them correctly. There may also be no change to the business process management at the tactical level. This was left out of figure 3 entirely since the UBMSS concept does not refer to this level at all.

The difference between a traditional system and the UBMSS one becomes visible when the computer on itself, without human participation, attempts to describe the properties and consequences of the ratios computed. The results of automatic interpretation are expressed in the categories of the applied description language for the interpreted data properties. The above-mentioned

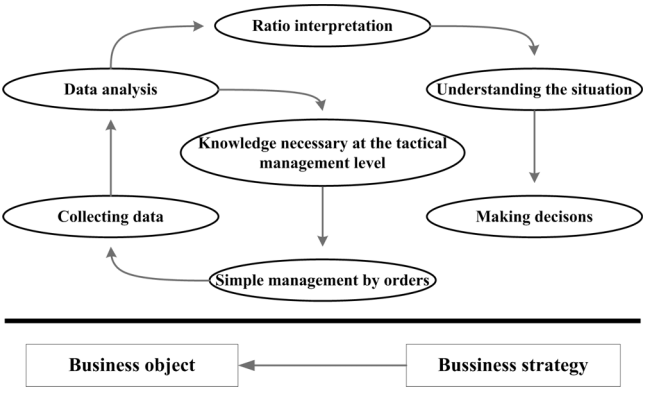


Fig. 2. Division of functions between people and computers in a traditional IT system supporting business decision-making

language is a key element at this stage. It must be designed with great know-how. Its construction must therefore be based on collecting and systematising expert knowledge. Referring to the analogy with medical image automatic understanding systems, which were earlier said to be the area of some fully successful implementations of ideas described here, one could say that just like in medical systems, the basis for the development of the language subsequently used for semantic image interpretation (and diagnosing a disease) were some specified **changes** in the shape of the analysed organs. In the UBMSS-class systems the basic constructing units of the developed language should be **changes** of some specified business indicators.

The lack of changes, both in medicine and in business is usually a factor calming one down. In accordance with the medical rule, *primum non nocere*, and with common sense "if it isn't broke don't fix it", it commands to refrain from steering activities. On the other hand, every change in the observed system and in its environment requires an interpretation in the categories of its meaning

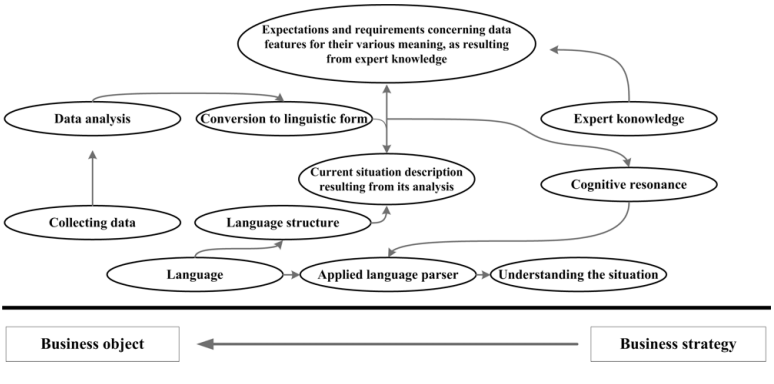


Fig. 3. General UBMSS-class system structure

since it is the disclosure of the change meaning (which is what a UBMSS-class system should mainly be used for) that makes it possible for one to understand its sense: the source of a threat or an arising opportunity. This is one of the main prerequisites for taking strategic decisions.

Of course, focusing attention on a business index and ratio changes computed in the input part of the analysed UBMSS-class IT system, as only on those elements, which should be the basic components of an artificial language, is just the first step. The listing and appropriate categorisation of changes that should be registered in linguistic business processes corresponded only to the stage at which one defines the alphabet to be later used to build words and sentences, i.e. the language main object. In order to make it possible to create from the elements of this alphabet counterparts of words and sentences for subsequent use by the UBMSS system to describe the states of business process, which require understanding and interpretation, calls for an introduction of additional mechanisms. These mechanisms would enable combining the above-mentioned sentences into larger units. Therefore, at a level superior to the above-described alphabet one must build the whole grammar of rules and transformations. This grammar can be used to create complete languages of description expressing important content, necessary to understand automatically the analysed processes.

Of course, talking about grammar in this place we mean a **formal grammar** of an appropriate type. It is similar to those successfully used to define languages, the key to understand medical images in publications specified in the bibliography [3, 7].

Analogies one can draw here are didactic and far going, therefore just to give some background information we shall mention one of them. In medical image understanding systems we constantly refer to, at our disposal were tools detecting local changes in the shape of some specified internal organs and their morphologic structures [8]. These were the above-mentioned ABC. To understand the state of a given organ correctly, one needed to add to these graphic primitives their mutual spatial relations and combine them with anatomy elements. Owing to a definition of rules and the grammar constructs connected with them, one could combine for example a graphic category "change of edge line direction of a specified contour" with a meaning category "artery stenosis anticipating a heart failure."

Similarly, by building into the proposed language grammar the ability to associate business changes detected in various parts of the managed company and its environment as well as the possibility to trace and interpret time sequences of these changes and their correlations, it will be possible, for example, to understand what are the real reasons behind poorer sales of goods or services offered. As a result it will be, for example, possible to find out about the fact that this is due to the wrong human resources policy rather than the wrong remuneration (bonus) system.

After the development of an appropriate language which will (automatically!) express semantically oriented descriptions of phenomena and business processes detected in the business unit (e.g. a company) as supervised by the information

system, a further UBMSS system operation will be very similar to the structure in which function the medical systems previously built by this system authors, as described in the previous chapter. The starting point for the business data automatic interpretation process, the process finishing with understanding their business meaning, is the description of the current state proposed in the system. It is expressed as a sentence in this artificial language, built specially for this purpose. Without going into details (described, among others, in earlier publications listed in the bibliography) one can say that the above-mentioned language description for a human being is completely illegible and utterly useless. A typical form of such notation is composed of a chain of automatically generated terminal symbols. Their meaning is well based in the mathematically expressed grammar of the language used. Yet from the human point of view this notation is completely illegible.

A condition to mine the meanings contained in this notation and to present them in a form useful, by giving the necessary knowledge necessary to develop a new strategy concept, to people (those who take decisions at appropriate levels), is to translate these symbolic notation into a notation understandable for people. For this purpose two elements, shown in figure 3, are necessary.

The first of these elements is duly represented knowledge of people (experts) who based on their theoretical knowledge and based on practical experience could supply a number of rules. Those rules state that in some circumstances, whose meaning interpretation could be described in detail, some particular features and properties should be found in the input data; those would be described with the use of a selected language. Now we have a description of the real situation, generated by tools founding their work on the results of business data analysis; it is generated with the use of the language we developed. We also have a set of hypothetical situations that carry some specified meaning connotations, which came into existence owing to the use of expert knowledge. We can therefore check in which areas these two descriptions converge (that increases the credibility of some semantic hypotheses) and in which the descriptions are contradictory. The latter case forms grounds to exclude other hypotheses and to narrow down the field of possible meanings.

The process of mutual interference between input information stream and the stream of expectations generated by an external knowledge source of the system, whose result is the development of "resonance peaks" in these areas in which the real situation "concords" with some specified expectations. These expectations result from the knowledge gathered before and in earlier works they have been called the cognitive resonance. Cognitive resonance happens in the course of the iteration process of comparison between features computed for input data and features theoretically forecast. Nevertheless we can also expect a possibility that the process will not always beconvergent and the result will not always be unique. Yet in most practical implementations researched by experiments, the authors have managed to obtain the desired convergence and cognitive resonance uniqueness. As a result it mined from the input data (in most research the data was medical images) information necessary to give the data correct interpretation

in the interpreted meanings area, that is to lead to a situation in which the system **understands** the data and that it will be able to suggest to people the correct semantic interpretation.

The second UBMSS system element specified on figure 3 whose meaning has to be explained is a **parser** translating internal description languages into a form understandable for a human being. Let us remind that the parser concept, treated as a translation automaton steered by the used grammar syntax, translating some language formulas from an encoded into an executable form, has been known in IT for a number of years in the context of programming languages.

In the UBMSS system described here the role of the parser is greater since its operations are steered to a significant extent by the cognitive resonance mechanism. In fact, in the cognitive system, the parser performs primarily the structural and meaning entry analysis. These entries were automatically generated in a special artificial language noting important semantic facts. Nevertheless the UBMSS system parser performs the above-mentioned meaning analysis as if as a side action since its basic role in the described diagram. As a translator, it receives an abstract code as its input. The code describes, in a language developed especially for this purpose, the current business situation. The output is to be the meaning of this situation specified in manner useful for men. The need for this meaning conversion from one language into another results from the fact that an artificial language developed to generate internal descriptions of the analysed business phenomena is constructed to obtain uniquely and effectively (automatically!) symbolic entries registering all important business process properties. These are obtained on the basis of the analysed data. This kind of meaning code can be built but essentially it is not understandable for people. Were its form understandable, it would not be very effective in the course of internal analysis leading to the cognitive resonance outcome.

Luckily, during the translation there is a confrontation between the current description of the analysed business situation and the model entries resulting from expert knowledge. As a result of that we obtain the above-mentioned cognitive resonance but also entries generated automatically in this artificial, not understandable language are converted into entries legible for a human being. Their interpretation is now understandable. Based on these entries, the outcome of the parser's operation, one obtains the necessary knowledge. Subsequently, when one already understands what is taking place in the business system, one can make strategic decisions. No one would dare to transfer the very last step to the machine. This is among others because there is a need to take responsibility for the decisions taken and it would be hard to sue computer software.

5 Test UBMSS-Class System Implementation

The objective of this article was to outline the concept of a new IT cognitive type system for business applications, referred to as an UBMSS-class system. In

developing this concept we combined the knowledge about IT systems currently developed for the management needs and the experience obtained by the authors developing earlier cognitive systems for medical image understanding. There are no doubts that the construction of a fully-fledged cognitive system based on the concept described in this paper will be a difficult and an expensive undertaking; its test implementation will require considerable funds and great courage. There is nothing surprising therefore in that in this paper we cannot yet boast of any description of an operational real-life system belonging to the class analysed here. This does not mean, however, that the authors have not conducted any experimental research.

The successful attempt to develop an experimental implementation of the IT system belonging to the UBMSS consisted of building a lab version of the data acquisition system originating from the supervision of health services diagnostic processes and of building for this case the necessary grammar and knowledge base. The created system was applied for interpretation tasks, i.e. processed data carrying meaning. The obtained usefulness of correct input data interpretation in the form of multi-dimensional vectors determining numeric data amounted to 90.5 % [4]. It should be noted, however, that not everything in this pilot version operated as planned. For this reason there are works conducted on enhancing the semantic reasoning algorithm on input vectors and there is also research conducted on the improvement of the cognitive analysis efficiency offered by this system.

Works are difficult and time-consuming because the UBMSS-class IT cognitive system is a completely new proposal and there is no experience that we could draw on. Still we claim that this concept has a bright future since, apart from the development of decision-taking processes, it brings in a new element to develop interpretation systems and computer understanding of the analysed data semantics. This is an extremely difficult task but should we be successful we must be aware that the UBMSS concept, apart from application to business-type data can also be adopted to interpret patterns in a different context that require automatic understanding. This could be, for example, the behaviour of people in the context of elections. This issue, however, is definitely out of the scope of this paper and it will not be discussed here.

6 Summary

This paper presented the problem of divergence of the contemporary IT systems capacity, used for management support and business system operations registration, with the needs imposed by contemporary business upon managers. It was found that typical IT systems now used (for example, of the ERP class) couldn't satisfy all the needs of modern decision-makers. This disparity between the needs and real capacity can be particularly well seen in connection with the fact that nowadays the need for computer management support are associated with a need to make strategic decisions relatively frequently. To make this term clear let us specify that in this paper we understand that all decisions taken at

various levels of company management are strategic if they are not limited to a simple regulation of stable business processes but which induce and impose changes. Therefore strategic decisions cannot be made solely on the basis of information about the current state of affairs in the on-going business processes. Their very essence is changing the state of affairs. For this reason current IT systems, focused mainly on registration and settlement functions are not a good tool to support such decision-making processes.

In the core part of this paper we have tried to outline the concept structure that could be the basis for such an UBMSS-class system dedicated for automatic business data understanding. We have tried to demonstrate briefly that a will to build such a system is an objective worth aiming at. Looking at the presented concept now, from the perspective of this summary, one could wander whether such system (assuming we shall manage to build it) will not change negatively the situation of all global economy participants. Such fears are seemingly not deprived of rational grounds: if every manager, regardless of his or her talents is able (owing to intelligent computer assistance) to make correct strategic decisions in difficult economic situations, than the capacity to take optimum decisions will no longer be a factor giving competitive advantage and many entities could face a situation more difficult than they experienced so far.

Acknowledgement

This work was supported by the AGH University of Science and Technology under Grant No. 10.10.120.39.

References

1. Albus, J. S., Meystel, A. M.: Engineering of Mind - An Introduction to the Science of Intelligent Systems. A Wiley-Interscience Publication John Wiley & Sons Inc. (2001)
2. Laudon, K. C., Laudon, J. P.: Management Information Systems - Managing the Digital Firm. Seventh Edition Prentice-Hall International Inc. (2002)
3. Ogiela, M. R., Tadeusiewicz, R.: Artificial Intelligence Structural Imaging Techniques in Visual Pattern Analysis and Medical Data Understanding. Pattern Recognition Elsevier **36/10** (2003) 2441–2452
4. Ogiela, L.: Usefulness assessment of cognitive analysis methods in selected IT systems. Ph. D. Thesis. AGH Kraków (2005)
5. Ogiela, M. R., Tadeusiewicz, R.: Picture Languages in Medical Pattern Knowledge Representation and Understanding. in V. Torra, Y. Narukawa, S. Miyamoto (Eds.) Modeling Decisions for Artificial Intelligence. Lecture Notes in Artificial Intelligence Springer-Verlag Berlin - Heidelberg - New York **3558** (2005) 442–447
6. Skomorowski, M.: A Syntactic-statistical approach to recognition of distorted patterns (in Polish). UJ Kraków (2000)
7. Tadeusiewicz, R., Flasiński, M.: Pattern Recognition (in Polish). PWN Warszawa (1991)

8. Tadeusiewicz, R., Ogiela, M. R.: Medical Image Understanding Technology, Artificial Intelligence and Soft-Computing for Image Understanding. Springer-Verlag Berling Heildelberg (2004)
9. Tadeusiewicz, R., Ogiela, M. R.: Intelligent Recognition in Medical Pattern Understanding and Cognitive Analysis. Chapter in book Muhammad Sarfraz (eds.) Computer-Aided Intelligent Recognition Techniques and Applications. John Wiley & Sons Ltd. Hoboken New Jersey (2005) 257-274.

PERT Based Approach to Performance Analysis of Multi–Agent Systems

Tomasz Babczyński and Jan Magott

Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
Tomasz.Babczynski@pwr.wroc.pl

Abstract. The following analytical approaches: queuing network models, stochastic timed Petri nets, stochastic process algebra, Markov chains are used in performance eevaluation of multi–agent systems. In this paper, new approach which is based on PERT networks is proposed. This approach is applied in performance evaluation of layered multi–agent system. These layers are associated with the following types of agents: manager, bidder, and searcher ones. Our method is based on approximation using Erlang distribution. Accuracy of our approximation method is verified using simulation experiments.

1 Introduction

Information retrieval systems in heterogeneous distributed environment are ones of the most popular examples of multi–agent systems (*MASs*). When the information retrieval system is developed, performance metrics of information distribution, retrieval and recovery are taken into account.

In order to get better values of performance metrics for MAS, these systems are combined from agents of different types. There are agents with: complete knowledge (*Fat Agents*) and limited knowledge (*Thin Agent*), mobile agents that migrate through a net. Additionally, there are hierarchical *MAS* with different agents at different hierarchy levels. An appropriate *MAS* organisation can reduce communicating complexity. *MAS* communication is realised concurrently in distributed environment.

For *MAS* , time of delivering a response on client request is significant metric. This time is combined from the agent activity times and inter–agent communication times. If new *MAS* is designed, then according to performance engineering of software systems [12], performance requirements have to be considered at each phase of life cycle.

Performance analysis of systems at phases before implementation can be based on performance models of: components and inter–component communication. Parameters of these models can be obtained from existing components or on the base of software engineers intuition. The performance measures for the models can be found by: analysis or simulation. Now, multi–agent technologies, e.g., [5], [7], are often based on Unified Modeling Lanuage (UML) [4] or its modifications. Therefore, approaches that are applied in performance analysis of systems

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1040–1049, 2006.
© Springer-Verlag Berlin Heidelberg 2006

designed using UML can be applied in performance evaluation of multi–agent systems. Hence, the following analytical approaches: queuing network models [8], stochastic Petri nets [9], stochastic process algebra [11], Markov chains can be used in performance evaluation of multi–agent systems.

In this paper, the approach, which is based on PERT networks, is proposed. This approach is applied in performance evaluation of layered multi–agent system. These layers are associated with the following types of agents: manager, bidder, and searcher ones. Our method is based on approximation using Erlang distribution. Erlang distribution is one of probability distributions that are used in evaluation of completion times in PERT networks. Erlang distributions create the family of distributions with different number of stages. Accuracy of our approximation method is verified using simulator. This simulator has been previously used in simulation experiments with the following multi–agent systems: personalized information system [1], industrial system [2], system with static agents and system with mobile agent [3]. These systems have been expressed in standard FIPA [6] which the JADE technology [7] is complied with.

The paper is organized as follows. In section 2, the multi–agent system is described. Then our approximation method is presented. In section 4, accuracy of our approximation method is verified by comparison with simulation results. Finally, there are conclusions.

2 Layered Multi–Agent System

We consider layered multi–agent information retrieval (MAS) system given at Fig. 1.

The MAS includes: one manager type agent (MTA) as Fat Agent, two bidder type agents($BTAs$) as Thin Agents, Searcher type agents ($STAs$) as Thin Agents. One BTA co–operates with a number of $STAs$.

After receiving a request from an user, the MTA sends messages to the $BTAs$ in order to inform them about the user request. Then the MTA is waiting for two responses from the $BTAs$. Having responses from the $BTAs$, the MTA prepares the response for the user.

After receiving a request from the MTA, the BTA sends messages to all $STAs$ co–operating with this BTA. Then the BTA is waiting for responses from all its $STAs$. Having responses from the $STAs$, the BTA prepares the response for the MTA.

The STA prepares the response by searching in Data Base (DB). Each STA is associated with one DB. The probability of finding the response in the DB is denoted by f_rate. Time unit is second, and it will be omitted. Searching time is expressed by uniform distribution over the time interval $[0, b)$. Hence, the expected searching time, provided there is the required information in the DB, is equal to $b/2$. With the probability $1 - f_rate$, the response is not found in the DB. In this case the searching time is equal to b.

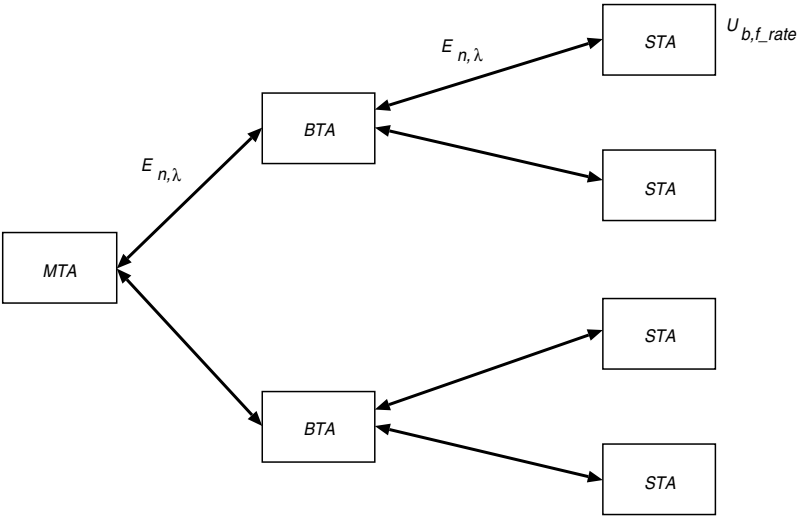


Fig. 1. Layered multi–agent information retrieval system

Message transmition times between the *MTA* and the *BTA*, and between the *BTA* and the *STA* are given by two stage Erlang distributions E_2 with parameter $\lambda = 1$ for each stage. Each stage is described by exponential distribution with this parameter.

3 Erlang Distribution Based Approximation Method

We will explain how the expected value of time of receiving of a response by the user is approximated. Because of the lack of space some derivations will be omitted.

Probability distributions of times are approximated by Erlang distribution.

Probability density function and cumulated distribution function of Erlang distribution with n stages and with parameter λ are given (following [10]) by expressions:

$$f_{E_{n,\lambda}}(t) = \frac{\lambda^n t^{n-1} e^{-\lambda t}}{(n-1)!}$$

$$F_{E_{n,\lambda}}(t) = 1 - \sum_{i=0}^{n-1} \frac{\lambda^i t^i e^{-\lambda t}}{i!}$$

Random variable (RV) with this distribution will be denoted by $E_{n,\lambda}$. This RV can be interpreted as sum of n RVs with exponential distribution and each with parameter λ. Expected value and variance for this RV are equal to:

$$E(E_{n,\lambda}) = n/\lambda, \qquad Var(E_{n,\lambda}) = n/\lambda^2.$$

For the *RV* T, the squared coefficient of variation (SCV) of the T is defined by the formula:

$$SCV(T) = \frac{Var(T)}{E(T)^2}$$

where: $E(T)$ is the expected value of T, $Var(T)$ is the variance of T.

The SCV for the $E_{n,\lambda}$ is equal to $SCV(E_{n,\lambda}) = 1/n$.

The *RV* of the *STA* searching time in the *DB* will be denoted by U_b. This *RV* has the probability density function:

$$f_{U_b}(t) = \begin{cases} f_rate \cdot 1/b & \text{for } t \in [0, b) \\ (1 - f_rate) \cdot \delta(t - b) & \text{for } t = b \\ 0 & \text{otherwise} \end{cases}$$

Expected value, variance, and SCV for this *RV* are given by the following expressions:

$$E(U_b) = \frac{b\,(2 - f_rate)}{2}$$

$$Var(U_b) = \frac{b^2\, f_rate\,(4 - 3\, f_rate)}{12}$$

$$SCV(U_b) = \frac{f_rate\ (4 - 3\, f_rate)}{3\ (2 - f_rate)^2}$$

Let us consider the approximation of the probability distribution of the *RV* X of the length of the time interval between the time instant when the *BTA* sends the request to given *STA* and the time instant when the *BTA* receives the response from this STA. This *RV* is given by the expression: $X = E_{n,\lambda} + U_b + E_{n,\lambda}$. We suppose that *RVs* of transmission times between agents and *RVs* of searching processes in the *DBs* are independent. Hence, expected value, variance, and SCV for the *RV* X are expressed by the following formulae:

$$E(X) = 2\,\frac{n}{\lambda} + b - 1/2\, b\, f_rate$$

$$Var(X) = 2\,\frac{n}{\lambda^2} + 1/3\, b^2 f_rate - 1/4\, b^2 f_rate^2$$

$$SCV(X) = \frac{24\, n + \left(4\, f_rate - 3\, f_rate^2\right) \lambda^2 b^2}{3\ \left(4\, n + (2 - f_rate)\, \lambda\, b\right)^2}$$

For multi–agent system described in section 2, the *RVs* of transmission times between agents are two stage Erlang distributions with parameter $\lambda = 1$ for each stage, and will be denoted by $E_{2,1}$.

The *RV* X is approximated by the *RV* $E_{n,\lambda}$ (with n stages and the parameter λ), and with the $SCV = 1/n$ such that $|SCV(X) - 1/n|$ is minimal. The expected values of the *RVs* X and $E_{n,\lambda}$ are equal. Hence, the parameter λ is

selected according to the equality $\lambda = n/E(X)$. The variance of the *RV* $E_{n,\lambda}$ is equal to $Var(E_{n,\lambda}) = n/\lambda^2$.

Let m be the number of *STAs* associated with one *BTA*. Let $E_{n,\lambda}(i)$ be such a *RV* $E_{n,\lambda}$ that approximates the length of the time interval between the time instant when the *BTA* sends the request to i^{th} *STA* and the time instant when the *BTA* receives the response from this *STA*. In this case, the *RV* Y of the *BTA* waiting time for all responses from *STAs* is $Y = \max_{i\in\{1,\ldots,m\}} E_{n,\lambda}(i)$. Probability density function and cumulated distribution function of the *RV* Y are given by the expressions:

$$f_Y(t) = \frac{m\, t^{n-1} e^{-\lambda t} \lambda^n \left((n-1)! - \Gamma(n, \lambda t)\right)^{m-1}}{((n-1)!)^m}$$

$$F_Y(t) = \left(1 - \sum_{i=0}^{n-1} \frac{\lambda^i t^i e^{-\lambda t}}{i!}\right)^m$$

where: $\Gamma(n, \lambda t) = \int_{\lambda t}^{\infty} x^{n-1}\, e^{-x} dx$ is upper incomplete Γ function [10].

The k^{th} moment (noncentral) of the *RV* Y is obtained by numeric integration of the following formula:

$$\mu^{(k)}(Y) = k \int_0^{\infty} t^{k-1} \left(1 - F_Y(t)\right) dt$$

Then the *RV* Y is approximated by *RV* $E_{n1,\lambda 1}$ in the same way as the *RV* X has been approximated by the *RV* $E_{n,\lambda}$.

The *RV* of the length of the time interval between the time instant when the *MTA* sends the request to given *BTA* and the time instant when the *MTA* receives the response from this *BTA* is approximated by the *RV*: $Z = E_{2,1} + E_{n1,\lambda 1} + E_{2,1}$.

The expected value of time of receiving of a response by the *MTA* (or user), i.e. response time, is approximated in the similar way as the expected value of the *RV* Y has been approximated.

4 Accuracy of the Approximation Method

In the prevoius section, the method of approximation has been shown. In order to evaluate the accuracy of the approximation method, the simulation experiments for: the *MAS* containing m *STAs* for each *BTA*, where $m = 3, 10$ have been performed. For each *MAS*, the following values of the probabilities of finding the response in the *DB* $f_rate = 0.1, 0.3, 0.6,$ and 0.9 have been considered. Additionally, the experiments for the following values of the maximal searching time in the *DB* $b = 2, 4, 8, 16, 32,$ and 64 have been executed. The transmission time between agents is given by *RV* $E_{2,1}$. Hence, the mean transmission time between the agents is equal to $E(E_{2,1}) = 2$, and the mean transmission time in

both directions, e.g., from the BTA to the STA and from the STA to the BTA, is equal to $E(tr) = 4$.

Let the symbol $E(b, f_rate)$ denote the percentage errors of the mean response time approximation for the b and for the f_rate.

In Table 1, the percentage errors of the mean response time approximation $E(b, f_rate)$ for the $m = 3$, for the $b = 2, 4, 8, 16, 32,$ and 64, $f_rate = 0.1, 0.3, 0.6,$ and 0.9 are given.

Table 1. The percentage errors of the mean response time approximation $E(b, f_rate)$ for the $m = 3$, for the $b = 2, 4, 8, 16, 32,$ and 64, for $f_rate = 0.1, 0.3, 0.6,$ and 0.9, for $n_{max} = 5, 10, 15, 20, 25$

f_rate \ b	2	4	8	16	32	64
0.1	0.1%	−0.5%	0.5%	5.5%	11.0%	15.0%
0.3	−0.2%	1.4%	2.7%	6.5%	11.7%	15.9%
0.6	−0.7%	0.1%	2.4%	6.0%	9.7%	13.6%
0.9	−0.1%	−0.2%	1.3%	1.8%	6.8%	4.7%

From Table 1 we can derive the following conclusions. The approximation is better for smaller values of the b because the influence of the $RV\ U_b$ which is not the Erlang distribution is smaller. For greater values of the $b = 16, 32, 64$ the approximation error is the smallest for the $f_rate = 0.9$. The distribution of the $RV\ U_b$ is combined form: the uniform distribution and the Dirac's delta. For the $f_rate = 0.9$, the influence of the Dirac's delta in this distribution is the smallest.

Let the percentage error of the mean response time approximation $E(b)$ for the b be defined by the expression:

$$E(b) = \frac{|E(b, 0.1)| + |E(b, 0.3)| + |E(b, 0.6)| + |E(b, 0.9)|}{4}.$$

In Table 2, the percentage errors of the mean response time approximation $E(b)$ for the $m = 3$, for the $b = 2, 4, 8, 16, 32,$ and 64 are given.

Table 2. The percentage errors of the mean response time approximation $E(b)$ for the $m = 3$, for the $b = 2, 4, 8, 16, 32,$ and 64

b	2	4	8	16	32	64
$E(b)$	0.27%	0.54%	1.71%	4.93%	9.81%	12.31%

Now we will concentrate on the case when the $b = 16$. Let the symbol U_{b,f_rate} denote the $RV\ U_b$ with the probability f_rate. The $RV\ X$ of the length of time interval between the time instant when the BTA sends the message to given

STA and time instant when the *BTA* receives the response from this *STA* is $X_{16,f_rate} = E_{2,1} + U_{16,f_rate} + E_{2,1}$ The mean of the random variable U_{b,f_rate} is equal:

$$E(U_{b,f_rate}) = f_rate \cdot b/2 + (1 - f_rate) \cdot b.$$

Hence, $E(U_{16,0.9}) = 8.8$, $E(U_{16,0.1}) = 15.2$, and as a result $E(X_{16,0.9}) = 12.8$, $E(X_{16,0.1}) = 19.2$. In spite of $b/E(E_{2,1}) = 8$ and $2 \cdot E(tr) < E(U_{16,0.9})$, i.e., the uniform distribution of the *RV* of the searching time in the *DB* is dominating the Erlang distribution of the *RV* of the transmission times, the Erlang distribution based approximation is good.

The greater the ratio $b/E(E_{2,1})$ is, the distribution of the $RV\,U_b$ of the searching time stronger dominates the Erlang distribution of the *RV* of the transmission times. Hence, for the greater ratios $b/E(E_{2,1})$, the approximation error $E(b)$ is greater.

In Table 3, the percentage errors of the mean response time approximation $E(b)$ for the $m = 10$, for the $b = 2, 4, 8, 16, 32,$ and 64 are given.

Table 3. The percentage errors of the mean response time approximation $E(b)$ for the $m = 10$, for the $b = 2, 4, 8, 16, 32,$ and 64

b	2	4	8	16	32	64
$E(b)$	0.77%	1.06%	4.11%	11.71%	21.71%	28.12%

It can be seen that the percentage errors of the mean response time approximation $E(b)$ for the $m = 10$ are greater than these errors for the $m = 3$.

Let us now analyse the accuracy of the approximation as a function of maximal number of stages of Erlang distributions used in the approximation. In Section 3, it has been stated that in order to approximate the *RV* X by the *RV* $E_{n,\lambda}$, the number n of stages is such that $|SCV(X) - 1/n|$ is minimal. When the maximal number of stages of Erlang distributions used in the approximation is equal to e.g. $n_{max} = 15$ then the approximation is executed in the following way. If the n obtained from the formula $|SCV(X) - 1/n|$ is such that $n \leq 15$ then the *RV* used in the approximation has n stages. If otherwise, then the *RV* used in the approximation has 15 stages. Sometimes, it is impossible to get the n satisfying the requirement $|SCV(X) - 1/n|$ is minimal because the calculations for the great n cannot be executed. From the other point of view, it is worth to know whether the calculations for greater number of stages of Erlang distributions are worth to execute. In Tables 4 and 4, the approximation error as a function of the maximal number n_{max} of stages of Erlang distributions used in the approximation have been presented.

Let us consider the row of Table 4 for $f_rate = 0.3$. For the $n_{max} = 15, 20, 25$ the percentage errors of the approximation are equal, while for the $n_{max} = 5, 10$ these errors are greater. It means that $SCV(X)$ for the approximated *RV* X satisfies the inequality $10 \leq 1/SCV(X) \leq 15$ and the $n = 10$ is not sufficient.

Table 4. The percentage errors of the mean response time approximation $E(b, f_rate)$ for the $m = 3$, for the $b = 16$, for $f_rate = 0.1, 0.3, 0.6$, and 0.9, for $n_{max} = 5, 10, 15, 20, 25$

f_rate \ n_{max}	5	10	15	20	25
0.1	30.1%	17.1%	11.3%	8.1%	5.7%
0.3	22.6%	10.5%	6.8%	6.8%	6.8%
0.6	12.2%	6.1%	6.1%	6.1%	6.1%
0.9	5.7%	2.4%	2.4%	2.3%	2.3%

Table 5. The percentage errors of the mean response time approximation $E(b, f_rate)$ for the $m = 10$, for the $b = 16$, for $f_rate = 0.1, 0.3, 0.6$, and 0.9, for $n_{max} = 5, 10, 15, 20, 25$

f_rate \ n_{max}	5	10	15	20	25
0.1	47.7%	25.9%	17.4%	12.9%	10.0%
0.3	38.3%	18.3%	12.8%	12.8%	12.8%
0.6	23.3%	13.9%	13.9%	13.9%	13.9%
0.9	13.5%	8.6%	8.6%	8.6%	8.6%

Let us now analyse the row of Table 4 for the $f_rate = 0.1$. In this case, for an approximated $RV\ S$, the relation $20 < 1/SCV(S)$ holds. Hence, 20 stages of Erlang distribution is not sufficient. The reason is as follows. For the $f_rate = 0.1$, in the probability distribution of the searching time, the Dirac's delta δ is selected with the probability 0.9. The SCV for the RV with the probability distribution expressed by the δ is equal to 0. Hence, and in order to approximate this RV exactly, the Erlang distribution with infinite number of stages have to be used. Because of $X_{16,f_rate} = E_{2,1} + U_{16,f_rate} + E_{2,1}$, Erlang distribution with finite number of stages can be used in the approximation of the $RV\ X_{16,f_rate}$. However, for the $f_rate = 0.1$, the influence of the δ is the greatest, and Erlang distribution with the greatest number of stages have to be used in the approximation.

We can conclude that if $n_{max1} < n_{max2}$ then the approximation error for the n_{max1} is not smaller than the approximation error for the n_{max2}.

5 Conclusions

The approximation method of the mean response time for the layered multi–agent information system has been presented. This system has three layers of agents, namely, manager, bidder, and searcher type ones denoted by the abreviations MTA, BTA, STA. After receiving a request from an user, the MTA sends

the messages to the *BTAs* in order to inform them about the user request. After receiving a request from the *MTA*, the *BTA* sends the messages to the *STAs*. The *STAs* prepare the responses for the *BTA* by searching in the Data Base (*DB*). The probability of finding the response in one *DB* is denoted by f_rate. Searching time is expressed by the *RV* U_{b,f_rate} with the distribution that is combination of the uniform distribution over the time interval $[0, b)$ (selected with the probability f_rate) and Dirac's delta in b (selected with the probability 1-f_rate). Message transmission times between the *MTA* and the *BTA* and between the *BTA* and the *STA* are given by the random variables (*RV*) of n stage Erlang distribution. In verification of accuracy of the method by simulation experiments, the transmission times are expressed by *RV* of two–stage Erlang distribution.

In the approximation method, the *RV* with n stage Erlang distribution is used. It has been obtained from the simulation, that the sum of the *RV* of the Erlang distribution (representing the transmission time) and the *RV* U_{b,f_rate} of searching time can be approximated by the other Erlang distribution with suitable number of stages. This is true even if the expected value of the *RV* U_{b,f_rate} is clearly greater than the expected value of the *RVs* of the transmission times.

The following conclusions have been obtained.

The approximation is better for smaller values of the b because the influence of the *RV* U_{b,f_rate} which is not the Erlang distribution is smaller. Let the mean transmission time between the agents in both directions, e.g., from the *BTA* to the *STA* and from the *STA* to the *BTA*, be denoted by $E(tr)$. The greater the ratio $b/E(tr)$ is, the distribution of the *RV* U_{b,f_rate} of the searching time stronger dominates the Erlang distribution of the *RV* of the transmission times. Hence, for the greater ratios $b/E(tr)$, the approximation error is greater.

For greater values of the b, the approximation error is the smallest for the greatest f_rate because the influence of the Dirac's delta in the distribution of the *RV* U_{b,f_rate} is the smallest.

It can be seen that the percentage errors of the mean response time approximation for the $m = 10$ are greater than these errors for the $m = 3$.

Accuracy of the approximation as a function of maximal number of stages of Erlang distributions used in the approximation has been studied too.

Many multi–agent systems have layered structure with the following agents: client assistant, brokers, and execution agents. The presented approximation method can be used for finding the mean time of response on client request for this class of systems.

References

1. T. Babczyński, Z. Kruczkiewicz, J. Magott, Performance evaluation of multiagent personalized information system, in: Proc. 7th Int. Conf. Artificial Intelligence and Soft Computing – ICAISC, Zakopane, 2004, Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence (LNCS/LNAI), Springer–Verlag, Vol. 3070, 810-815.

2. T. Babczyński, Z. Kruczkiewicz, J. Magott, Performance analysis of multiagent industrial system, in: Proc. 8th Int. Workshop Cooperative Information Agents – CIA, Erfurth, 2004, LNCS/LNAI, Springer–Verlag, Vol. 3191, 242-256.
3. T. Babczyński, Z. Kruczkiewicz, J. Magott, Performance comparison of multiagent systems, in: Proc. Central and Eastern European Conference on Multiagent Systems – CEEMAS, 2005, LNCS/LNAI, Springer–Verlag, Vol. 3690, 612-615.
4. G. Booch, J. Rumbaugh, I. Jacobson, The Unified Modeling Language, User Guide, Addison Wesley Longman, 1999.
5. S.A. Deloach, M.F. Wood, C.H. Sparkman, Multiagents systems engineering, International Journal of Software Engineering and Knowledge Engineering, Vol. 11, No. 3, 2001,231-258.
6. Foundation for Intelligent Physical Agents, `http://www.fipa.org/specs/`
7. JADE, `http://jade.tilab.com/`
8. P. Kahkipuro, UML based performance modelling framework for object–oriented distributed systems, in: Proc. UML 99, The Unified Modeling Language: Beyond the Standard, LNCS, Vol. 1723, 1999.
9. P. King, R. Pooley, Using UML to derive stochastic Petri net models, in: Proc. 15th Annual UK Performance Engineering Workshop, University of Bristol, 1999, 45-56.
10. MathWorld, Wolfram Research, Inc., `http://mathworld.wolfram.com/ErlangDistribution.html`, `http://mathworld.wolfram.com/topics/GammaFunctions.html`.
11. R. Pooley, Using UML to derive stochastic process algebra models, in: Proc. 15th Annual UK Performance Engineering Workshop, University of Bristol, 1999, 23-33.
12. C. U. Smith, Performance Engineering of Software Systems, Addison – Wesley, 1990.

Rule-Based Automated Price Negotiation: Overview and Experiment

Costin Bădică[1], Maria Ganzha[2], and Marcin Paprzycki[3]

[1] University of Craiova, Software Engineering Department
Bvd.Decebal 107, Craiova, RO-200440, Romania
badica_costin@software.ucv.ro
[2] Elbląg University of Humanities and Economy, ul Lotnicza 2, 82-300 Elbląg, Poland
ganzha@euh-e.edu.pl
[3] Computer Science Institute, Warsaw School of Social Psychology, 03-815 Warsaw, Poland
Marcin.Paprzycki@swps.edu.pl

Abstract. The idea of automating e-commerce transactions attracted a lot of interest during the last years. Multi-agent systems are claimed to be one of promising software technologies for achieving this goal. In this paper we summarize state-of-the-art in rule-based approaches to automated negotiations and present initial experimental results with our own implementation of a rule-based price negotiation mechanism in a model e-commerce multi-agent system. The experimental scenario considers multiple buyer agents involved in multiple English auctions that are performed in parallel.

1 Introduction

During last years, interest in e-commerce has shifted from simple Web presence of a business to advanced use of e-commerce technologies in order to support growth of business itself — by improving its efficacy and profitability. Therefore, the idea of automating e-commerce transactions attracted a lot of research interest ([15]).

Most of currently existing e-commerce systems involve humans that make most important decisions in various activities along the lifeline of an e-commerce transaction. At the same time, software agents are claimed to be one of the best technologies for automating e-commerce processes. It is expected that intelligent agents will be able to substantially reduce (if not eliminate) need for human involvement — in all but most crucial decisions. In this context, we have set up a project to contribute development of such an agent system [11]. In particular our project has two main goals: (1) to build a large-scale implementation approximating an e-commerce environment; (2) to develop a tool that we will be able to use for modeling various e-commerce scenarios.

E-commerce research proposes that when digital technologies are utilized to mediate commercial transactions, then the complete process can be conceptualized as consisting of four phases: (i) *pre-contractual phase* including activities like need identification, product brokering, merchant brokering, and matchmaking; (ii) *negotiation* where negotiation participants negotiate according to the rules of a particular market mechanism and using their private negotiation strategies; (iii) *contract execution* including activities like: order submission, logistics, and payment; and (iv) *post-contractual phase*

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1050–1059, 2006.
© Springer-Verlag Berlin Heidelberg 2006

that includes activities like collecting managerial information and product or service evaluation.

Focus of this paper is on the *negotiation phase* of a transaction taking place in a multi-agent e-commerce system that we have started to develop ([11]). As a part of this work we are interested in endowing our agents with flexibility necessary to engage in unknown in advance forms of negotiations using rule-based approaches. We start our presentation with an overview of state-of-the-art of rule representations in automated negotiations. We follow with a brief summary of the architecture of our proposed system that uses a rule-based framework for enforcing specific negotiation mechanisms, together with a sample scenario. Finally, we discuss some experimental results of our implementation.

2 Background on Rule-Based Negotiation

Rules have been indicated as a very promising technique for formalizing multi-agent negotiations ([4,7,8,10,16,18,22,27,28]). When considering design of systems for automated negotiations it is typically the case that *negotiation protocols* (or *mechanisms*) that define "rules of encounter" between participants and *negotiation strategies* that define behaviors aiming at achieving a desired outcome are distinguished. However, rule representations were proposed for both negotiation mechanisms ([7,24,16,18,27,28]) and strategies ([10,22,8]). Let us now summarize most important developments in the area of rule based approaches to automated negotiations.

In our work we follow a rule-based framework for enforcing specific negotiation mechanisms inspired by [7]. Its authors sketched a complete framework for implementing portable agent negotiations that comprises: (1) negotiation infrastructure, (2) generic negotiation protocol and (3) taxonomy of declarative rules. The *negotiation infrastructure* defines roles of negotiation participants and of a host. Participants exchange proposals within a negotiation locale managed by the host. The *generic negotiation protocol* defines three phases of a negotiation: admission, exchange of proposals and formation of an agreement, in terms of how and when messages should be exchanged between the host and participants. *Negotiation rules* are used for enforcing the negotiation mechanism. Rules are organized into a taxonomy: rules for participants admission to negotiations, rules for checking the validity of proposals, rules for protocol enforcement, rules for updating the negotiation status and informing participants, rules for agreement formation and rules for controlling the negotiation termination.

The proposal for formalizing negotiations introduced in [24] goes beyond the framework of [7]. Its authors suggest to use an ontology for expressing negotiation protocols. Whenever an agent is admitted to negotiation it also obtains a specification of the negotiation rules in terms of the shared ontology. In some sense, the negotiation template used by our implementation (see [4]) is a "simplified" negotiation ontology and the participants must be able to "understand" the slots defined in the template. This approach is exemplified with a sample scenario. The ontology approach introduced in [24] is taken further in [23] by investigating how the ontology can be used to tune the negotiation strategy of participant agents. However, paper [24] contains neither implementation

details, nor experimental results. Furthermore, we were not able to obtain a complete version of the ontology described in the paper.

In [27,28] a mathematical characterization of auction rules for parameterizing the auction design space is introduced. The proposed parametrization is organized along three axes: i) *bidding rules* – state when bids may be posted, updated or withdrawn; ii) *clearing policy* – states how the auction commands resource allocation (including auctioned items and money) between auction participants (this corresponds roughly to agreement making in our approach); iii) *information revelation policy* – states how and what intermediate auction information is supplied to participating agents.

In [18] authors developed a special declarative language CLP ("Courteous Logic Programs as KR") for expressing and reasoning about contracts and negotiations. This project was a continuation of the Michigan AuctionBot project ([26]), and its authors focused on the automatic configuration of negotiations based on a contract and showed how rules generated during the negotiation process can be combined with the partial contract to form an executable final contract. Background knowledge supporting this infrastructure was embodied in three CLP rule sets: *Auction-Configuration*, *Auction-Space*, and *Auctionbot-Mapping*. *Auction-Configuration* supports reasoning about alternative negotiation structures and also specifies how to split contract into an array of auctions. *Auction-Space* implements a cleaner, more general parameterization of the auction design space, imposes constraints and conditional defaults on parameters, and infers auction parameters from higher-level knowledge about the negotiation. *AuctionBot-Mapping* maps the *Auction-Space* parameterization to the existing set of *AuctionBot* parameters. Unfortunately, we were not able to find any information about continuation of this interesting project.

In [16] an implementation of a new rule-based scripting language (*AB3D*) for expressing auction mechanisms is reported. The design and implementation of AB3D were primarily influenced by the parametrization of the auction design space defined in [27,28] and the previous experiences with the Michigan Internet AuctionBot ([26]). According to [16], A3BD allows the initialization of auction parameters, the definition of rules for triggering auction events, the declaration of user variables and the definition of rules for controlling bid admissibility.

A formal executable approach for defining the strategy of agents participating in negotiations using defeasible logic programs is reported in [12] and [10]. The approach is demonstrated on English auctions and bargaining with multiple parties by indicating sets of rules for describing strategies of participating agents. However, paper [12] contains neither implementation details, nor experimental results.

In [22] a preliminary implementation of a system of agents that negotiate using strategies expressed in defeasible logic is described. The implementation is demonstrated with a bargaining scenario involving one buyer and one seller agent. The buyer strategy is defined by a defeasible logic program. Note that the implementation reported in [22] builds on the architecture of negotiating agents previously introduced in [10]. Note also that defeasible logic programs are able to express courteous logic programs proposed in [18] and yet to support efficient reasoning, which suggest that they might be the appropriate representation formalism of negotiation strategies.

The CONSENSUS system that enables agents to engage in combined negotiations was presented in [8]. CONSENSUS allows agents to negotiate different complementary items on separate servers on behalf of human users. Each CONSENSUS agent uses a rule base partitioned into: i) *basic rules* that determine the negotiation protocol, ii) *strategy rules* that determine the negotiation strategy and iii) *coordination rules* that determine the knowledge for assuring that either all of the complementary items or none are purchased. Note that in CONSENSUS the rule-based approach is taken beyond mechanism and strategy representation to capture also coordination knowledge.

Another interesting work is the open environment for automated negotiations specifically targeted to auctions – auction reference model (ARM, [20]) and its associated declarative auction specification language (DAL, [19]). It should be noted that, while not explicitly using rules, a DAL specification actually models the flow of an auction using a rule-based approach. DAL constructs comprise the following: views, validations, transitions and agreement generators. Views are analogous to visibility rules, validations are analogous to validity and protocol enforcement rules, transitions are analogous to update rules and agreement generators are analogous to agreement formation and negotiation life-cycle rules.

Before proceeding further, let us make the following remark. E-commerce is seen as one of the key services of modern information society and therefore, the ability of software agents to discover remote markets and engage in commercial transactions governed by market mechanisms unknown in advance, is of primary importance. Rules constitute a very promising approach to describing negotiation processes (see for instance all references cited in this section). However, a key aspect for success of automated negotiations, that already generated some interest in the research community, is the development of a truly open rule-based semantic description of the market mechanism [7,4,19,20,24,23]. As our research indicates, we are still quite far from that vision of software agents needing only minimal pre-compiled knowledge to enable them to ”sense” the negotiation mechanism and ”tune” the negotiation strategy accordingly. It is exactly this issue that catalyzes our work and that differentiates it from previous works, making us to proceed further.

3 System Description and Experiment

3.1 Conceptual Architecture

Our system acts as a distributed marketplace in which agents perform functions typically observed in e-commerce ([11]). E-shops are represented by shop and seller agents, while e-buyers are represented by client and buyer agents. In Figure 1 we present Use-Case diagram of the complete system. Outside o bounds of the system we can see *User-Client* who will attempt at buying products from one of the e-shops and *User-Seller* who tries to sell products in her e-store. Let us now briefly summarize the most important agents appearing in the system and their functionalities (for a complete discussion of the system see [3,5,6]). *User-Client* is represented by the *Client Agent* (*CA*). The *CA* is completely autonomous and as soon as the decision to purchase product *P* is communicated by the *User-Client*, it will work until either the product is purchased or, due to the market circumstances (e.g. prices are to high) purchase is abandoned.

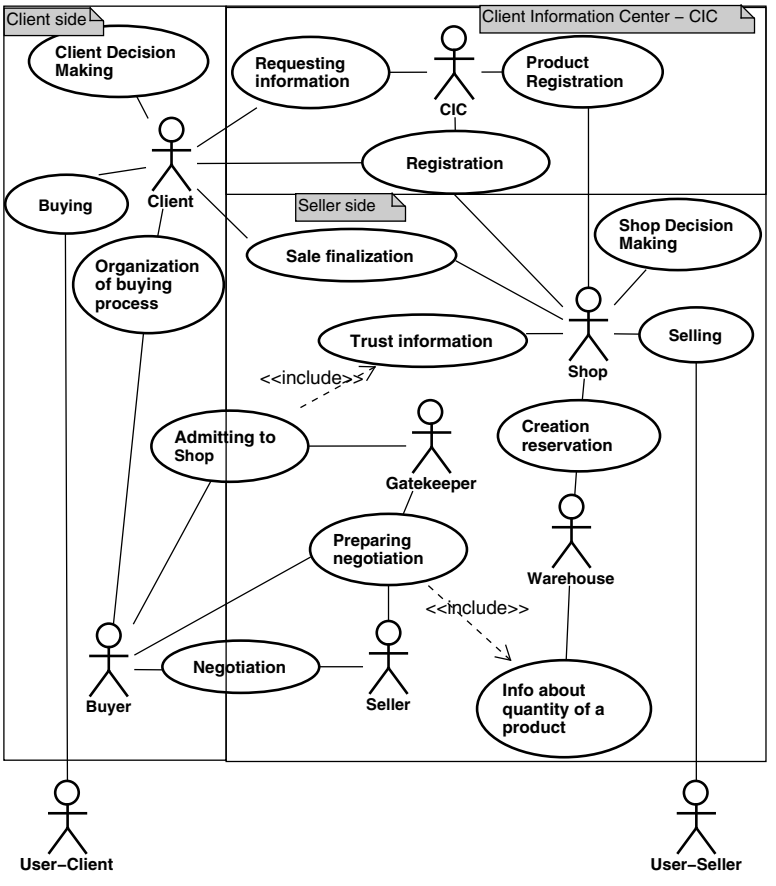


Fig. 1. Use Case diagram of the proposed agent-based e-commerce system

The *CA* communicates with the *Client Information Center* (*CIC*) agent which contains complete information which e-stores sell which products. For each store that sells the requested product, the *CA* delegates a single *Buyer Agent* (*BA*) with a mission to be involved in price negotiations and if successful, possibly attempt at making a purchase (successful price negotiations result in a product reservation for a specific time period; after which products that have not been actually purchased are returned to the pool of products available for sale). Since multiple *BA*s representing the same *CA* can win price negotiations and report to the *CA*, it is the *CA* that makes the decision if either of available offers is good enough to make a purchase. *Buyer Agents* either migrate to the negotiation host or are created locally [6]. They can participate in negotiations only if the *Gatekeeper Agent* (*GA*) allows this. The *GA* utilizes *trust information* to evaluate if a given *BA* should be admitted (*BA*s that win price negotiations but do not make a purchase may be barred from subsequent negotiations). The *GA* is one of agents that represent that e-store and is created by the *Shop Agent* (*SA*). The *SA* is the central manager of the e-shop. Facilitating the selling process, the *SA* utilizes the (*GA*), as well as

a *Warehouse Agent* (*WA*) that is responsible for inventory and reservation management; and a set of *Seller Agents* (*SeA*) that negotiate price with incoming *BA*s.

In our experiments we considered simplified version of this scenario that involves a single *Shop Agent* and n *Client Agent*s CA_i, $1 \leq i \leq n$. The *SA* is selling m products $\mathcal{P} = \{1, 2, \ldots, m\}$. We assume that each client agent CA_i, $1 \leq i \leq n$, is seeking a set $\mathcal{P}_i \subseteq \mathcal{P}$ of products (we therefore restrict our attention to the case where all sought products are available through the SA). The SA is using m *Seller Agents* SeA_j, $1 \leq j \leq m$ and each SeA_j is responsible for selling a single product j. Each CA_i is using buyer agents BA_{ik} to purchase products from the set $\mathcal{P}_i$. Each BA_{ik} is responsible for negotiating and buying exactly one product $k \in \mathcal{P}_i$, $1 \leq i \leq n$. To attempt purchase *Buyer Agent*s BA_{ik} migrate to the SA and engage in negotiations; a BA_{ik}, that was spawned by the *Client Agent* CA_i, will engage in negotiation with seller SeA_k, to purchase product k. This simple scenario is sufficient for the purpose of our paper, i.e. to illustrate our rule-based system and show how a number of rule-based automated negotiations can be performed concurrently. In this setting, each *Seller Agent* SeA_j plays the role of a negotiation host defined in [7]. Therefore, in our system, we have exactly m instances of the framework described in [7]. Each instance is managing a separate "negotiation locale", while all instances are linked to the *Shop Agent*. For each instance we have one separate set of rules together with a negotiation template that describes the negotiation mechanism implemented by that host. Note that each seller may use a different negotiation mechanism (different form of an auction, or an auction characterized by different parameters, such as the starting price or the bidding increment). See [4] for the details of our implementation of this conceptual architecture using JADE ([13]) and JESS ([14]).

3.2 Rule-Based Representation of English Auctions

For the purpose of this paper we have set up our system for a particular negotiation scenario involving English auctions. Technically, English auctions are single-item, first-price, open-cry, ascending auctions ([15],[25]). In an English auction there is a single item (or a collection of products treated as a single item) sold by a single seller and many buyers bidding against each other for buying that item. Usually, there is a time limit for ending the auction, a seller reservation price that must be met by the winning bid for the item to be sold and a minimum value of the bid increment. A new bid must be higher than the currently highest bid plus a minimal bid increment in order to be accepted. All the bids are visible to all the auction participants.

The constraints describing English auctions were encoded as a modularized set of JESS rules. The rules were then used to initialize rule inference engines encapsulated by the negotiation hosts [4]).

Let us now consider a few sample rules for representing English auctions. These rules are described informally using a pseudo-code notation that is independent of any implementation-level language (like JESS).

POSTING-BUYER rule specifies that a $buyer$ participant can post a proposal whenever there is an offer already posted by a $seller$ participant.

POSTING-BUYER
IF
There is a valid proposal Pr of a participant with role $buyer$ $\wedge$
There is an active proposal of a participant with role $seller$
THEN
Proposal Pr is posted

IMPROVEMENT-BUYER rule specifies that a $buyer$ participant must post a proposal with a price that must overbid the currently highest id with at least a given increment)that is a parameter of the auction).

IMPROVEMENT-BUYER
IF
Negotiation is on goods A $\wedge$
Bid increment is Inc $\wedge$
Currently highest bid is B $\wedge$
Proposal Pr on goods A with price P was posted by a $buyer$ $\wedge$
$P > B + Inc$
THEN
Proposal Pr is active

AGREEMENT-FORMATION rule specifies that whenever agreement formation is triggered, if the currently highest bid is greater than the $seller$ reservation price (that it is not disclosed to the participants), an agreement is formed between the submitter of the highest bid and the $seller$.

AGREEMENT-FORMATION
IF
The currently highest bid is B and was submitted by $buyer$ $S1$ $\wedge$
There is an active proposal of $seller$ $S2$ with price P $\wedge$
Negotiation is on goods A $\wedge$
$B \geq P$
THEN
An agreement of $S1$ with $S2$ to transact goods A at price $P1$ is formed

3.3 Participants Strategy

Strategies of participant agents are defined in accordance with the negotiation protocol (i.e. English auctions in this particular setting). Basically, the strategy defines if and when a participant will submit a proposal depending on what are the values of its parameters. For the time being we opted for a simple solution: the participant submits a first bid immediately after it was granted admission and whenever it gets a notification that another participant issued an accepted proposal. The value of the bid is equal the sum of the currently highest bid and an increment value that is private to the participant. Each participant has its own valuation of the negotiated product. If the value of the new bid exceeds this value then the proposal submission is canceled (given product became "too expensive" for a given *BA*). Note that in the case of an English auction there is no particular strategy for the *Seller Agent* as it plays only a passive role.

Agent strategies were implemented in Java as participant agent behaviors ([13]). In the future we plan to design the system in such a way that strategies will also be represented in the rule-based form ([10,12]). This will provide us with the required flexibility to easily add multiple strategies to our implementation. Obviously, in practice, this form of strategy representation is required only for more involved forms of price negotiations (where utilization of complicated strategies makes much more sense).

3.4 Experiment

In the experiment we considered $m = 10$ products and $n = 12$ clients seeking all of them, i.e. $\mathcal{P}_i = \mathcal{P}$ for all $1 \leq i \leq 10$. The auction parameters were the same for all auctions: reservation price 50 and minimum bid increment 5. Clients reservation prices were randomly selected from the interval [50,72] and their bid increments were randomly selected from the interval [7,17].

In this experiment 143 agents were created: 1 shop SA, 10 sellers SeA_i, $1 \leq i \leq 10$, 12 clients CA_i, $1 \leq i \leq 12$, and 120 buyers BA_{ik}, $1 \leq i \leq 12$, $1 \leq k \leq 10$, and 10 English auctions were run concurrently. One separate JESS rule engine was also created for each English auction (therefore a total of 10 JESS rule engines were run in parallel). The average number of messages exchanged per negotiation was approximately 100 and all the auctions finished successfully. This means that a total of more than 1000 messages was exchanged during negotiations. While the total number of agents and messages is still small (for instance in comparison with these reported in [9], this experiments indicates that the proposed approach has good potential for supporting experiments on large-scale.

Figure 2 shows messages exchanged between the seller SeA_1 and buyers BA_{i1}, $1 \leq i \leq 12$ that were captured with the help of the JADE sniffer agent.

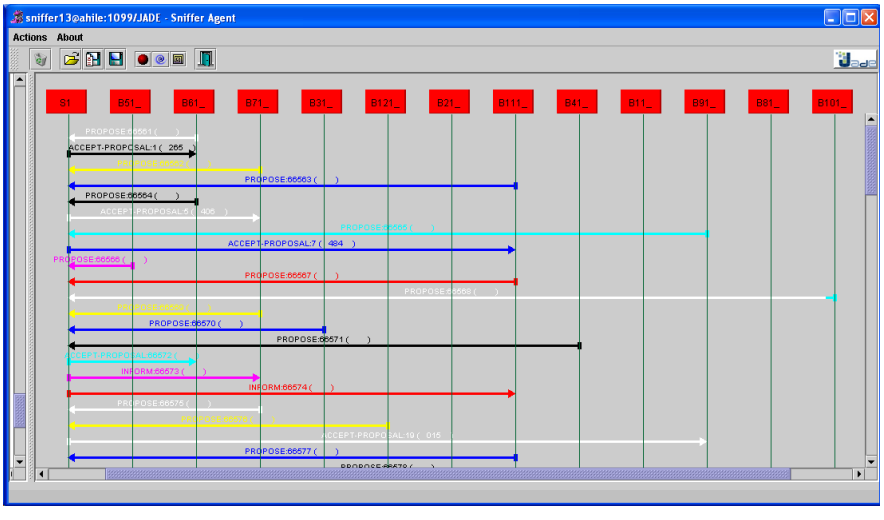


Fig. 2. Negotiation of a seller with 12 buyers in an English auction

4 Conclusions and Future Work

In this paper we discussed rule-based approaches for automated negotiation in a model multi-agent e-commerce systems. Our discussion was supplemented by providing experimental results obtained using our own implementation of a rule-based price automated negotiation framework. The results support the claim that rules are a feasible and scalable technology for approaching flexible automated negotiation in e-commerce.

As future work we plan to: (i) complete the integration of the rule-based framework into our agent-based model e-commerce system; (ii) to asses the generality of our implementation by extending it to include other price negotiation mechanisms; (iii) to conceptualize representation and ways to efficiently implement multiple strategy modules; iv) to investigate the applicability of rule-markup languages ([21]) for devising an open rule-representation of negotiation mechanisms. We will report on our progress in subsequent papers.

References

1. Bădică, C., Ganzha, M., Paprzycki, M., Pîrvănescu, A.: Combining Rule-Based and Plug-in Components in Agents for Flexible Dynamic Negotiations. In: M. Pěchouček, P. Petta, L.Z. Varga (eds.): *Proceedings of CEEMAS'05*, Budapest, Hungary. LNAI 3690, Springer Verlag (2005) 555–558.
2. Bădică, C., Ganzha, M., Paprzycki, M., Pîrvănescu, A.: Experimenting With a Multi-Agent E-Commerce Environment. In: *Proceedings of PaCT'2005*, Krasnoyarsk, Russia. LNCS 3606 Springer Verlag, (2005) 393–401.
3. Bădică, C., Ganzha, M., Paprzycki, M.: Mobile Agents in a Multi-Agent E-Commerce System. In: *Proceedings of the SYNASC 2005 Conference*, Timişoara, Romania. IEEE Computer Society Press, Los Alamitos, CA, (2005) 207–214.
4. Bădică, C., Bădiţă, A., Ganzha, M., Iordache, A., Paprzycki, M.: Rule-Based Framework for Automated Negotiation: Initial Implementation. In: A. Adi, S. Stoutenburg, S. Tabet (eds.): *Proceedings RuleML'2005*, Galway, Ireland. LNCS 3791, Springer Verlag (2005) 193–198.
5. Bădică, C., Ganzha, M., Paprzycki, M.: UML Models of Agents in a Multi-Agent E-Commerce System. In: *Proc. ICEBE'2005*, Beijing, China. IEEE Computer Society Press, Los Alamitos, CA, (2005) 56–61.
6. Bădică, C., Ganzha, M., Paprzycki, M.: Two Approaches to Code Mobility in an Agent-based E-commerce System. In: C. Ardil (ed.), *Enformatika*, Volume 7, (2005) 101–107.
7. Bartolini, C., Preist, C., Jennings, N.R.: A Software Framework for Automated Negotiation. In: *Proceedings of SELMAS'2004*, LNCS 3390, Springer Verlag (2005) 213–235.
8. Benyoucef, M., Alj, H., Levy, K., Keller, R.K.: A Rule-Driven Approach for Defining the Behaviour of Negotiating Software Agents. In: J.Plaice et al. (eds.): *Proceedings of DCW'2002*, LNCS 2468, Springer verlag (2002) 165–181.
9. Chmiel, K., Tomiak, D., Gawinecki, M., Karczmarek, P., Szymczak, Paprzycki, M.: Testing the Efficiency of JADE Agent Platform. In: *Proceedings of the* 3^{rd} *International Symposium on Parallel and Distributed Computing*, Cork, Ireland. IEEE Computer Society Press, Los Alamitos, CA, USA, (2004), 49–57.
10. Dumas, M., Governatori, G., ter Hofstede, A.H.M., Oaks, P.: A Formal Approach to Negotiating Agents Development. In: *Electronic Commerce Research and Applications*, Vol.1, Issue 2 Summer, Elsevier Science, (2002) 193–207.

11. Ganzha, M., Paprzycki, M., Pîrvănescu, A., Bădică, C., Abraham, A.: JADE-based Multi-Agent E-commerce Environment: Initial Implementation, In: *Analele Universităţii din Timişoara, Seria Matematică-Informatică* Vol. XLII, Fasc. special, (2004), 79–100.
12. Governatori, G., Dumas, M., ter Hofstede, A.H.M., and Oaks, P.: A formal approach to protocols and strategies for (legal) negotiation. In: Henry Prakken, editor, *Procedings of the 8th International Conference on Artificial Intelligence and Law*, IAAIL, ACM Press, (2001) 168–177.
13. JADE: Java Agent Development Framework. See `http://jade.cselt.it`.
14. JESS: Java Expert System Shell. See `http://herzberg.ca.sandia.gov/jess/`.
15. Laudon, K.C., Traver, C.G.: *E-commerce. business. technology. society* (2^{nd} ed.). Pearson Addison-Wesley, (2004).
16. Lochner, K.M., Wellman, M.P.: Rule-Based Specification of Auction Mechanisms. In: *Proc. AAMAS'04*, ACM Press, New York, USA, (2004).
17. Lomuscio, A.R., Wooldridge, M., Jennings, N.R.: A classification scheme for negotiation in electronic commerce. In: F. Dignum, C. Sierra (Eds.): *Agent Mediated Electronic Commerce: The European AgentLink Perspective*, LNCS 1991, Springer Verlag (2002) 19–33.
18. Reeves, D. M.; Grosof, B. N.; Wellman, M. P.; and Chan, H. Y.: *Toward a declarative language for negotiating executable contracts.* In: AAAI-99 Workshop on ArtiïñĄcial Intelligence in Electronic Commerce (AIEC-1999).
19. Rolli, D., Eberhart, A.: An Auction Reference Model for Describing and Running Auctions. In: *Electronic Markets – The International Journal* (2005).
20. Rolli, D., Luckner, S., Gimpel, A.: A Descriptive Auction Language. Electronic Markets. In: *7 Internationale Tagung Wirtschaftsinformatik*, Bamberg, Germany, (2005).
21. RuleML Initiative. See `http://www.ruleml.org`.
22. Skylogiannis, T., Antoniou, G., Bassiliades, N.: A System for Automated Agent Negotiation with Defeasible Logic-Based Strategies – Preliminary Report. In: Boley, H., Antoniou, G. (eds): *Proceedings RuleML'04*, Hiroshima, Japan. LNCS 3323 Springer-Verlag (2004) 205–213.
23. Tamma, V., Phelps, S., Dickinson, I., Wooldridge, M.: Ontologies for Supporting Negotiation in E-Commerce. In: *Engineering Applications of Artificial Intelligence*, 18, Elsevier (2005) 223–238.
24. Tamma, V., Wooldridge, M., Dickinson, I: An Ontology Based Approach to Automated Negotiation. In: *Proceedings AMEC'02: Agent Mediated Electronic Commerce*, LNAI 2531, Springer-Verlag (2002) 219–237.
25. Wooldridge, M.: *An Introduction to MultiAgent Systems*, John Wiley & Sons, (2002).
26. Wurman, P.R., Wellman, M.P., Walsh, W.E.: The Michigan Internet AcutionBot: A Configuarable Auction Server for Human and Software Agents. In: *Proceedings of the second international conference on Autonomous agents. Agents'98*, Minneapolis, USA. ACM Press, New York, USA, (1998), 301-308.
27. Wurman, P.R., Wellman, M.P., Walsh, W.E.: A Parameterization of the Auction Design Space. In: *Games and Economic Behavior*, 35, Vol. 1/2 (2001), 271–303.
28. Wurman, P.R., Wellman, M.P., Walsh, W.E.: Specifying Rules for Electronic Auctions. In: *AI Magazine* (2002), 23(3), 15–23.

A New Version of the Fuzzy-ID3 Algorithm

Łukasz Bartczuk[1] and Danuta Rutkowska[1,2]

[1] Department of Computer Science
Czestochowa University of Technology, Poland
[2] Department of Knowledge Engineering and Computer Intelligence
Academy of Humanities and Economics in Lodz, Poland
{bartczuk, drutko}@kik.pcz.czest.pl

Abstract. In this paper, a new version of the Fuzzy-ID3 algorithm is presented. The new algorithm allows to construct decision trees with smaller number of nodes. This is because of the modification that many different attributes and their values can be assigned to single leaves of the tree. The performance of the algorithm was checked on three typical benchmarks data available on the Internet.

1 Introduction

Decision trees are commonly used as knowledge representation and an approach to classification. They are appreciated for their clarity and high accuracy. Many algorithms designed for buiding decision trees have been proposed. The most popular are ID3 (Interactive Dichotimizer 3) introduced by Quinlan in 1986 [8], and its modifications, e.g. C4.5 [9]. Those algorithms allow to create decision trees from symbolic data, in an easy and effective way. Numerical data, when applied, must be splitted into limited number of disjoint intervals. The data present values of the attributes, i.e. features of objects to be classified.

In some classification problems, determination of crisp values of attributes is not possible or not fully correct. The solution of that problem is the use of the theory of fuzzy sets and fuzzy logic, introduced in 1965 by Lofti Zadeh [12]. Fuzzy sets may describe uncertain or imprecise phenomenona. The Fuzzy-ID3 algorithm [6],[7], created by Janikow in 1995, combines simplicity and clarity of decision trees with fuzzy sets which can define linguistic values and allow to use fuzzy intervals.

There are two main problems related to decisions trees and fuzzy decision trees. The first is the large size of the tree (number of nodes) in high dimensional classification tasks. The second problem concerns the structure of the tree. The tree created according to the ID3 or Fuzzy-ID3 algorithm is a proper structure for the data representation. However, it is not always the best solution. The main reason of these two problems is the manner in which the attribute (represented by a node) to be the best split is chosen; see Fig.1.

In this paper, a modified Fuzzy-ID3 algorithm is presented. According to this modification, more than one attribute, and more than one linguistic value of these attributes, may be assigned to single leaves (decision nodes). This

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1060–1070, 2006.
© Springer-Verlag Berlin Heidelberg 2006

modification results in obtaining trees with smaller number of nodes. An example of such a tree is illustrated in Fig. 4.

The paper is organized as follows. Section 2 presents the ID3 and Fuzzy-ID3 algorithms. The modified version of the Fuzzy-ID3 algorithm is proposed in Section 3. Experimantal results are illustrated in Section 4. Final conclusions are included in Section 5.

In this paper, capital letters denote sets, for instance A, C, E - sets of attributes, classes, and examples, respectively. Cardinalities of the sets are denoted as $|A|$, $|C|$, $|E|$. Specific attributes, for $k = 1, \ldots, |A|$, are denoted as A^k, and A^k is a set of values of attribute A^k, so $A^k = \{a_l^k\}$, for $l = 1, \ldots, |A^k|$, and $A^k \in A$. Specific classes are denoted as c^j, for $j = 1, \ldots, |C|$, and $c^j \in C$. Examples are denoted as e^i, for $i = 1, \ldots, |E|$ and $e^i \in E$. Thus, every example is described as $e^i = [a_i^1, \ldots, a_i^{|A|}, c_i^j]$, where $c_i^j \in C$ is the class associated with e^i.

2 ID3 and Fuzzy-ID3 Algorithms

This section presents classical and fuzzy versions of the ID3 algorithm introduced in [8] and [6], respectively. Decision trees are techniques for partitioning examples into sets corresponding to decision rules.

2.1 ID3 Algorithm

The purpose of the ID3 algorithm is to create a tree structure from an example set, E, which contains values of attributes, A^k, for $k = 1, \ldots, |A|$, that characterize objects to be classified. In addition, every example includes the class, c^j, to which the object belongs. These examples are called training examples, and E is a training set. The tree structure can further be used for classification, data analysis or knowledge representation.

This algorithm employs the entrophy for determining the discriminatory power of each attribute. This is applied in order to determine the attribute that should be chosen to split the node associated with this attribute. The ID3 algorithm is based on the following assumptions [8]:

(1) The root node of the decision tree contains all training examples. Each node is reqursively split by partitioning its examples.
(2) Every training example belongs to class c^j with probability (the relative frequency):

$$p_j = \frac{\left|E_j^N\right|}{|E^N|} \tag{1}$$

where E^N - set of examples in node N, and E_j^N - set of examples that belong to class c^j in node N; $E_j^N \subset E^N \subset E$.
(3) For the data set in current node, N, we compute the information content:

$$I^N = -\sum_{j=1}^{|C|} p_j \log_2 p_j \tag{2}$$

(4) If an attribute, A^k, is chosen as a node, N, of the decision tree, the information to be supplied to the subtree corresponding to the node's branch, i.e. the path from parent (root) node N,$A^k = a_l^k$, to a child node, is denoted as $I^{N|a_l^k}$. The expected information required for the subtree with the attribute A^k in node N is determined as follows:

$$I^{N|A^k} = \sum_{l=1}^{|A^k|} \frac{\left|E_{a_l^k}^N\right|}{|E^N|} I^{N|a_l^k} \tag{3}$$

where $I^{N|A^k}$ is called the weighted entrophy, $E_{a_l^k}^N$ denotes the set of examples whose attribute value a_l^k corresponds to the node's branch.

(5) The information gained by branching on the attribute A^k at node N is:

$$G = I^N - I^{N|A^k} \tag{4}$$

The node is split using the most discriminatory attribute, whose information gain, determined using (4), is maximal.

The process of splitting tree nodes starts from the root node (as node N), then repeats, and the algorithm ends when all attributes appear on the path from the root node to the current node or when all examples in the node come from a unique class. The fulfillment of the second criterion can leads to overlearning effect. The threshold $\tau \in [0, 1]$ can be used to prevent that situation. If the ratio of the number of examples with the same class to all examples in node N is equal or greater than this threshold, the node became a leaf. The ID3 algorithm is presented, in many publications, e.g in [5],[6],[8],[9].

2.2 Fuzzy-ID3 Algorithm

In classical decision trees, created by the ID3 algorithm, atrributes can have only symbolic or discrete numerical values. In case of fuzzy decision trees attributes can also have linguistic values (eg. small, warm, low) represented by fuzzy sets. Fuzzy decision trees have been obtained as a generalisation of classical decision trees through application of fuzzy sets and fuzzy logic. The Fuzzy-ID3 algorithm is an extension of ID3 algorithm. The difference between these two algorithms is in the method of computing the example count in node N. In the Fuzzy-ID3 algorithm the total examples count, P^N, in node N are expressed as [6]:

$$P^N = \sum_{j=1}^{|D_C|} P_j^N \tag{5}$$

where: D_C - set of linguistic values for the decision attribute, $\mathbf{x}_i$ and y_i are input vector and output value, which correspond to attributes and class, respectively, and the examples count, P_j^N, for decision j (class c_j) is determined as follows:

$$P_j^N = \sum_{i=1}^{|E^N|} \mathsf{f}\left(\mu_s(\mathbf{x}_i), \mu_j(c_i)\right) \tag{6}$$

where: f - function employed to compute the value of fuzzy relation (e.g. min, prod) [2],[10],[11],[13], μ_s - membership function of Cartesian product of fuzzy sets that appear on the path from the root node to node N, and μ_j - membership function of fuzzy set that determines class c^j, for $j = 1 \ldots, |C|$.

Equations (2) and (3) in the Fuzzy-ID3 algorithm takes the forms:

$$I^N = -\sum_{j=1}^{|D_C|} \frac{P_j^N}{P^N} \log_2 \frac{P_j^N}{P^N} \tag{7}$$

$$I^{N|A^k} = \frac{\sum_{l=1}^{|A^k|} P^{N|a_l^k} I^{N|a_l^k}}{\sum_{l=1}^{|A^k|} P^{N|a_l^k}} \tag{8}$$

where: $P^{N|a_l^k}$ - total examples count in node N containing value a_l^k, assuming that attribute A^k is used to split the node N. Stopping criteria are the same as in the ID3 algorithm.

2.3 Illustration of Fuzzy Decision Trees

Suppose we want to build a tree to solve a binary classification task with two attributes (*attr1* and *attr2*), and three fuzzy sets (*low*, *medium*, *high*) defined for each attribute. We know that attribute *attr2* is relevant for the solution of this problem because all examples with value *low* for this atrribute belong to class *0* and with value *hi* belong to class *1*; see Fig. 1b.

Suppose also that the proportion of examples with these values in the data set is small. When we compute examples count for each class and total examples count, for each fuzzy set (attribute value) that can be associated with the child node of the *Root* node, we get the values shown in Table 1, where *lv* stands for "linguistic value" that is the attribute value.

If we compute the weighted entrophy (8) and information gain (4), for each attribute, we see that according to the algorithm the best attribute to split is *attr1*. The result is a correct representation of the data set, but this is not the

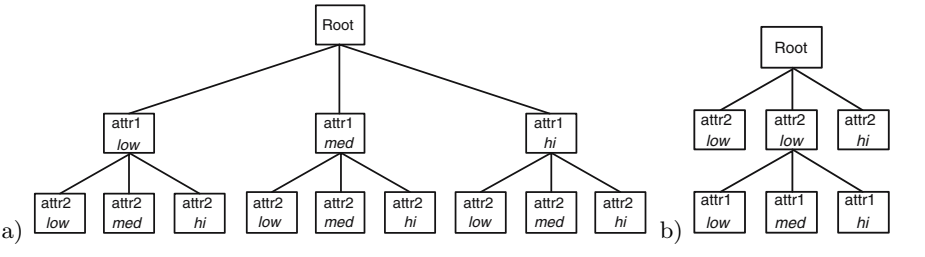


Fig. 1. Two possible trees: a) created by Fuzzy-ID3 algorithm, b) the better tree that can be created for the same problem

Table 1. The examples count for each class and total examples count in each fuzzy set that can be associated with the child node of the *Root* node

lv	$P_0^{Root\|lv}$	$P_1^{Root\|lv}$	$P^{Root\|lv}$
a_{low}^{attr1}	14.55	4.20	18.75
a_{med}^{attr1}	19.27	3.98	23.25
a_{hi}^{attr1}	4.58	49.95	54.53
a_{low}^{attr2}	13.23	0	13.23
a_{med}^{attr2}	23.39	49.85	73.24
a_{hi}^{attr2}	0	9.24	9.24

best solution that can be achieved for this problem. This is because the algorithm chooses the best split on average.

The tree created according to the Fuzzy-ID3 algorithm contains thirteen nodes and is depicted in Fig. 1a. The better tree that can be created for the same problem contains only seven nodes and is shown in Fig. 1b.

The solution of the problem mentioned above for the crisp ID3 algorithm has been presented by Friedman et. al.[4]. This algorithm, which is called the Lazy Decision Tree is very interesting for symbolic or numerical values of attributes, but it requires a process of creating of a new decision tree for every new example. In case of fuzzy values, many branches of the tree can be activated. This causes that number of computations that have to be performed may be too big to build a new tree for every new example. Therefore, this solution can be inefficient for fuzzy decision trees.

3 Fuzzy Decision Trees with Multi-Attribute Leaves

In this section, a new version of Fuzzy ID3 algorithm is proposed. The classical algorithms, described in Section 2 have been designed to create decision trees with nodes that represent only one attribute value. This algorithm allows to use more than one attribute value in leaves, so the decision trees contain less number of the nodes. This algorithm can be called MAL Fuzzy ID3, or MAL FID3, for short, where MAL stands for Multi-Attribute Leaves.

3.1 MAL Fuzzy ID3 Algorithm

The proposed algorithm introduces some modifications to the tree structure and to the procedure of creating the tree. We assume that there can be more than one linguistic value in the leaves of the tree, and also that there can be values of different attributes. This modification allows the use of all values of attributes that give unambiguous classification as a child of the current node. The membership of the example, in such a node, can be computed as the maximum values of the membership functions describing fuzzy sets in this node. We can also use other

s-norms [10],[11],[2],[13], than the maximum. However, we will achieve better results when we apply an arithmetic mean value of membership functions.

Let F^N denotes a set of values of attributes which can be used to split node N, and E^N - set of examples that have nonzero membership in node N.

The proposed algorithm is shown below:

Step 1: In the root of the tree, assume: $F^N = A$ and $E^N = E$

Step 2: From set F^N choose these linguistic values that give unambiguous classification, i.e. for $a_l^k \in F^N$; let us define

$$\Theta_j^N = \left\{ a_l^k : \frac{P_j^{a_l^k}}{\sum\limits_{m=1,..,|C|} P_m^{a_l^k}} > \tau \right\} \quad \text{for} \quad j = 1, \ldots, |C| \tag{9}$$

where $P_j^{a_l^k}$ - total example count for class c^j, and attribute value a_l^k. For each nonempty set Θ_j^N create a new node. Linguistic values from sets Θ^N will not be taken into consideration for further spliting of the nodes.

Step 3: From set E^N, choose those examples, $e^i \in E^N$ for which the arithmetic mean value of membership of fuzzy sets describing linguistic values from Θ_j^N is smaller than threshold $\sigma \in [0, 1]$, that is

$$\Psi^N = \left\{ e^i : \frac{\sum\limits_{a_l^k \in \Theta_j^N} \mu_{a_l^k}(e^i)}{|\Theta_j^N|} < \sigma \right\} \quad \text{for} \quad j = 1, \ldots, |C| \tag{10}$$

Step 4: For examples from set Ψ^N, compute information content according to (7).

Step 5: Compute weighted entrophy (8) for all attributes from F^N, and theirs values which are not included in Θ_j^N; $\quad j = 1, \ldots, |C|$.

Step 6: Select the attribute maximizing the information gain, G, and split the node N, using this attribute.

Step 7: For the nodes created in step 6, set $F^{N+1} = F^N \setminus \bigcup\limits_{j=1,\ldots,|C|} \Theta_j^N$ and $E^{N+1} = \Psi^N$

Step 8: Repeat steps from 2 to 8 until the stopping criteria are not fulfilled.

3.2 Illustration of MAL FID3 Algorithm on IRIS Data

The application of this algorithm will be presented on the Iris classification problem [1],[3]. We have to split the iris flowers into three classes representing iris spieces. Each example is described by four attributes (width and length of petal and width and length of sepal).

The data set E consists of 150 examples spliting into three classes: Setosa, Versicolour and Virginica (50 examples from each class). Distribution of the

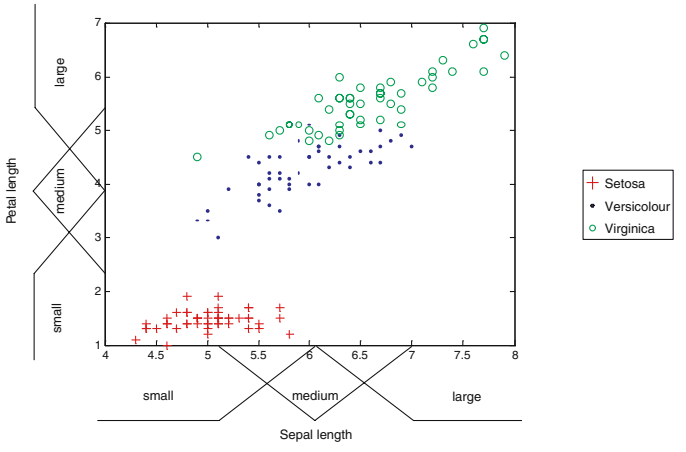


Fig. 2. Distribution of examples for Iris classification problem; for attributes: Sepal length and Petal length

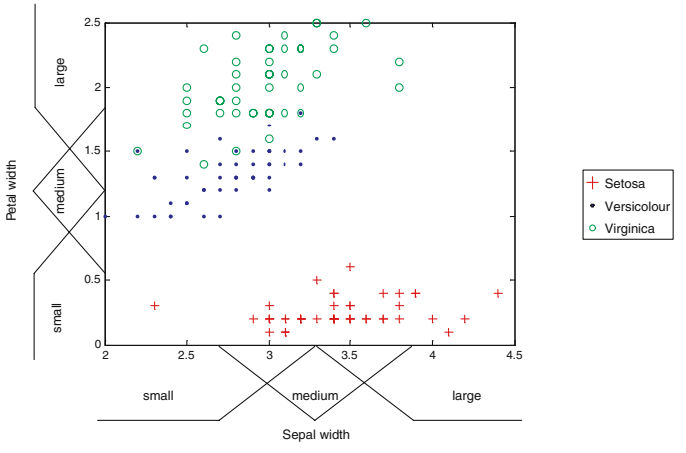


Fig. 3. Distribution of examples for Iris classification problem; for attributes: Sepal width and Petal width

examples in the attribute space is presented in Figs. 2 and 3. For each attribute, three fuzzy sets (representing values: small, medium, large) are defined.

In the begining the set of attribute values contains the following elements:

$$F^{Root} = \left\{ \begin{array}{lll} a^{PL}_{small}; & a^{PL}_{medium}; & a^{PL}_{large}; \\ a^{PW}_{small}; & a^{PW}_{medium}; & a^{PW}_{large}; \\ a^{SL}_{small}; & a^{SL}_{medium}; & a^{SL}_{large}; \\ a^{SW}_{small}; & a^{SW}_{medium}; & a^{SW}_{large}; \end{array} \right\}$$

where PL, PW, SL, SW stands for petal length, petal width, sepal lenght, sepal width, respectively.

Set E^{Root} includes all 150 examples from set E. According to equations (5) and (6), we compute the ratio of examples count for each class to total examples count, for fuzzy sets describing liguistic values from F^{Root}. The results are presented in Table 2.

Table 2. The ratio of examples count from each class to total examples count for all linguistic values

Sepal length (SL)				Sepal width (SW)			
Value	*Setosa*	*Versicolour*	*Virginica*	*Value*	*Setosa*	*Versicolour*	*Virginica*
small	0.704	0.2432	0.0523	small	0.0766	0.5457	0.3775
medium	0.070	0.5053	0.4239	medium	0.3695	0.2705	0.3599
large	0	0.2405	0.7594	large	0.8231	0.0204	0.1564
Petal length (PL)				**Petal width (PW)**			
Value	*Setosa*	*Versicolour*	*Virginica*	*Value*	*Setosa*	*Versicolour*	*Virginica*
small	**0.948**	0.0520	0	***small***	**0.9036**	0.0964	0
medium	0	0.8616	0.1384	***medium***	0	**0.8849**	0.1151
large	0	0.2293	0.7707	large	0	0.1424	0.8576

Assuming the threshold $\tau = 0.87$ (values for which the ratio exceed this threshold are marked as bold in Table 2.), we can create the following sets:

$$\Theta^{Root}_{Setosa} = \left\{a^{PL}_{small}; a^{PW}_{small}\right\}, \quad \Theta^{Root}_{Versicolour} = \left\{a^{PW}_{medium}\right\}$$

For $\sigma = 0.5$, set Ψ^N contains 102 examples. By executing steps 4 and 5, the algorithm chooses attribute *petal width* for splitting the root node. Repeating the algorithm for each new nonleaf node, we obtain the tree shown in Fig. 4.

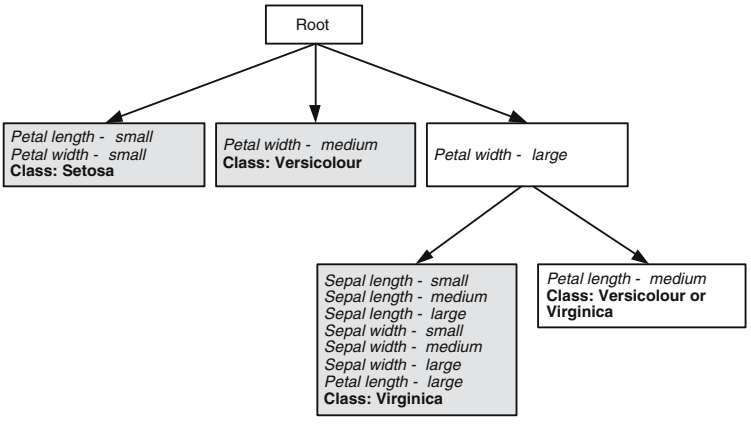


Fig. 4. Decision tree created by MAL FID3 algorithm, for Iris classification problem

It contains 6 nodes and gives 96% correct classifications. The tree created by the classical Fuzzy-ID3 algorithm contains 26 nodes and reaches 93% correct classifications.

4 Other Experimental Results

Three popular data sets, which are available at the UCI Machine Learning Repository [1], were used in the experiments. Every experiment was repeated twenty times to get the error ratio, shown in Tables 3-5. These tables present results for the classical Fuzzy-ID3 algorithm and the MAL FID3. Table 3 concerns the wine classification problem, with 13 attributes, 3 classes, 128 examples in the training set and 32 examples in the testing set. Table 4 includes results for the glass classification problem with 9 attributes, 6 classes, 174 examples in the training set and 40 examples in the testing set. Table 5 presents results for the heart desease (medical diagnosis problem), with 10 attributes, 2 classes, 221 examples in the training set and 40 examples in the testing set. The threshold value, τ, and the average number of nodes obtained from 20 experiments are included in the tables.

As can be noticed for all presented problems, the trees obtained by the MAL FID3 algorithm, proposed in this paper, are smaller (less number of nodes) than those obtained by the classical Fuzzy-ID3 algorithm. The error ratios are comparable for both kinds of the trees.

Table 3. Results for the wine classification problem

	Fuzzy-ID3		MAL FID3	
τ	Average number of nodes	Error[%]	Average number of nodes	Error[%]
0.65	13.1	22.81	9	11.40
0.7	44.85	17.65	13.4	9.68
0.75	111.45	12.50	13.8	8.75
0.8	251.15	10.78	17.2	9.37
0.85	502.55	8.59	33.65	7.03
0.9	974	8.75	80.9	5.31

Table 4. Results for the glass classification problem

	Fuzzy-ID3		MAL FID3	
τ	Average number of nodes	Error[%]	Average number of nodes	Error[%]
0.65	280.45	41	98.95	42.62
0.7	342.95	40.87	192.55	41.25
0.75	386.35	40.87	203.35	40.87
0.8	443.9	41.25	291.6	40.62
0.85	515.6	41.12	313.65	40.12
0.9	591.25	40.62	354.35	41.50

Table 5. Results for the heart classification problem

	Fuzzy-ID3		MAL FID3	
τ	Average number of nodes	Error[%]	Average number of nodes	Error[%]
0.65	4.57	25	3	17.3
0.7	64.92	23.05	3	18.75
0.75	122.85	22.67	6.78	18.75
0.8	240.14	22.67	39.28	20.70
0.85	374.78	20.52	84	21.77
0.9	477	20.70	167.28	23.92

5 Conclusions

In this paper, a new version of the Fuzzy-ID3 algorithm, called MAL FID3, is presented. The modification, introduced to the classical Fuzzy-ID3, makes possible the use of many values of different attributes in the leaves of a tree. The trees build according to the proposed algorithm are smaller (less number of nodes) than those created by the classical Fuzzy-ID3 method. For some problems these trees can produce better classification results.

The purpose of the future works is a further reduction of the size of fuzzy decision trees and elimination of those fuzzy sets from the leaves that have no influence on the classification process.

References

1. Blake C., Keogh E., Merz C.: UCI repository of machine learning databases, [http://www.ics.uci.edu/∼mlearn/MLRepository.html], Irvine, CA: University of California, Dept. of Computer Science (1998).
2. Dubois D., Prade H.: Fuzzy Sets and Systems: Theory and Applications, Academic Press, San Diego (1980).
3. Fisher R.A. The Use of Multiple Measurements in Taxonomic Problems, Annals of Eugenics (1936) Part II 179-188
4. Friedman J.H., Kohavi R., Yun Y.: Lazy Decision Trees, in Proc. of the Thirteenth National Conference on Artificial Intelligence, (1996), 717-724.
5. Ichihashi H., Shirai T., Nagasaka K., Miyoshi T., Neuro-fuzzy ID3: a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning, Fuzzy Sets and Systems, **81**, Issue 1, (1996), 157-167.
6. Janikow C.Z.: Fuzzy Decision Trees: Issues and Methods, IEEE Transactions on Systems, Man, and Cybernetics, **28**, Issue 3, (1998), 1-14.
7. Janikow C.Z.: Exemplar Learning in Fuzzy Decision Trees, in Proc. of IEEE International Conference on Fuzzy Systems, Piscataway, NJ, (1996), 1500-1505.
8. Quinlan J.R.: Induction of decision trees, Machine Learning, **1**, (1986), 81-106.
9. Quinlan J.R.: C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, Inc., Los Altos, California, (1993).

10. Rutkowska D.: Neuro-Fuzzy Architectures and Hybrid Learning, Physica-Verlag, Springer-Verlag Company, Heidelberg, New York, (2002)
11. Rutkowski L.: Artificial intelligence methods and techniques, PWN, in Polish, (2005)
12. Zadeh L.A.: Fuzzy Sets, Information and Control, **8**, (1965), 338-353.
13. Zimmermann H.-J.: Fuzzy Set Theory, Kluwer Academic Publishers, Boston, Dordrecht, London (1994).

New Interpolation Method with Fractal Curves

Andrzej Cader and Marcin Krupski

Academy of Humanities and Economics, Lodz, Poland
Institute of Informatics, Managementand and Transport,
Department of Management Systems
acader@wshe.lodz.pl, mkrupski@wshe.lodz.pl

Abstract. Classic interpolation methods using polynominals or other functions for reconstruction complex dependences or contours perform their role well in the case of smooth and relatively regular curves. However, many shapes found in nature or dynamic relations corresponding to real process are of very irregular character and the appropriate characteristics are rough and demonstrate a complex structure at difference scales. This type of curves are numbered among fractals or stochastic fractals multifractals. In practice it is impossible to approximate them with the help of classic methods. It is necessary to use fractal methods for the interpolation. At present the only group of this type of methods are the ones based on fractal interpolation functions (FIFs) suggested by Barnsley [1]. However, these methods are burdened with numerous inadequacies making it difficult to use them in practice. The study presents another alternative method of using fractal curves for complex curves approximation. This method is more adequate than FIF for multifractal structures interpolation. It generalizes classic notion of an interpolation knot and introduces non-local values for its description, as for instance fractal dimension. It also suggests continuous, as regards fractal dimension, family of fractal curves as a set of base elements of approximation an equivalent of base splines. In this aspect the method is similar to the classic B-splines method and does not use Iterated Function Systems (IFS), as Barnseys method does. It may be determined as a hard interpolation method aiming at working out an algorithm providing its effective application in practice, whereas to a lesser degree attention is paid to mathematical elegance.

1 Introduction

One-dimensional interpolation consists in reconstruction of the given curve or functional dependence with the help of point pattern (interpolation knots), which represent this curve or dependence. In majority of the most common methods polynominals are used for this purpose. The methods of this type are reconstructed well by smooth curves, not torn, for which the complexity of their internal structure may be neglected the complexity which becomes apparent in subsequent magnification. The quality of reconstruction depends on the frequency of sampling the interpolation step and proper selection of interpolating functions.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1071–1081, 2006.
© Springer-Verlag Berlin Heidelberg 2006

However, in many fields of current studies we face the necessity of reconstruction of more realistic structures of the surrounding world or complex dependences describing the processes taking place in it. These structures, or characteristics of real phenomena in majority of cases are not objects or cannot be introduced with the help of objects derived from classic geometry [2],[3]. In accordance to the known saying [4]: As Euclidon geometry is the natural way of representing man-made objects, fractal geometry is the natural way of representing objects that occur in nature. Even if it is not true, the observation of nature all the time confirms how right this saying is. Majority of objects found in nature morphological structures in biology, geology, chemistry and physics, the processes taking place in our surrounding climatic, hydrologic, biological and even social and economic are of irregular character fractal [5]-[13], more precisely they demonstrate the properties of stochastic fractals [14]-[16]. This means that they present a peculiar combination of the activity of deterministic chaos and random behaviour. Such complex dependences cannot be interpolated with the help of idealised functional relations and even if classic methods are applied for their approximation, the obtained results far differ from reality. In extreme cases, due to complex fractal structure of the reconstructed curve, the notion of interpolation knot loses sense because one unambiguous value cannot be attributed to it. It is clearly seen, that fractal structures can be only reconstructed with the help of other fractal structures or methods generating fractals. The second solution is in FIF method, in which known way of fractals formation is applied with Iterated Function System (IFS) [17]-[18].

From among huge number of studies concerning fractal interpolation, nearly all of them are directly associated with the conception presented by Barnsley or they are its elaboration [1]. Briefly, in this conception Iterated Function Systems wn, n =1, 2...N consists of pairs of maps:

$$w_n(x,y) = (L_n(x), F_n(x,y)) \tag{1}$$

such, that for the set of interpolation knots $(x_0,y_0),(x_1,y_1),.....(x_N,y_N)$ and closed interval $M=[x_0,x_N]\in\mathbf{R}$ containing $\{x_0,x_1,...x_N\}$ $L_n(x)$ are contractive homeomorphisms: $M\to M_n$ $(M_n=[x_{n-1},x_n])$:

$$L_n(x_0)=x_{n-1}, L_n(x_N)=x_n \tag{2}$$

$$|L_n(x_i)-L_n(x_j)|\le c\,|x_i-x_j|\,\forall x_i,x_j\in M \tag{3}$$

for certain $0\le c<1$. Whereas $F_n: F\to\mathbf{R}$ $(F=M\times\mathbf{R})$ fulfils:

$$F_n(x_0,y_0)=y_{n-1}, F_n(x_N,y_N)=y_N \tag{4}$$

$$|F_n(x,y)-F_n(x,z)|\le\beta_n|x-z| \tag{5}$$

for $x\in M$ and $y,z\in\mathbf{R}$ when $-1<\beta_n<1$.

In accordance with Bransley theorem, the above IFS (1) generates a unique attractor which is a graph of a continuous function $f: M\to\mathbf{R}$ fulfilling $f(x_n)=y_n$. There comes to the relation:

$$f(L_n(x))=F_n(x,f(x)) \tag{6}$$

FIF corresponding to IFS form was investigated most thoroughly:

$$L_n(x) = a_n x + b_n \qquad F_n(x,y) = \beta_n y + q_n(x) \tag{7}$$

$$a_n = \frac{x_n - x_{n-1}}{x_N - x_0} \qquad b_n = \frac{x_N x_{n-1} - x_0 x_n}{x_N - x_0} \tag{8}$$

where: $q_n(x)$ is an affine map, β_n is called a vertical scaling factor. Barnsley demonstrated that with $q_n(x)$ form:

$$q_n(x) = f \circ L_n(x) - \beta_n g(x) \tag{9}$$

where: f , g - continuous functions $(f \neq g)$, $g(x_0) = y_0$, $g(x_N) = y_N$ FIF can be generalized with optional continuous functions. In this way splines can be obtained in the form of polynominals [19]-[21] or trigonometric functions [22]. The method was extended to enable interpolation of fractal surfaces and three dimensional objects [23],[24]. Despite success and huge number of studies, both theoretical and visualizing the possibilities of application in various fields of research, FIF has not become a routinely applied method in interpolation. There are a few reasons. First of all there are no clear criteria of selection of FIFs and their coefficients for the given interpolation problems - FIFs can be constructed in infinitely many ways, each time obtaining different shape of an interpolating curve. Moreover, the method strongly depends on the choice of interpolation knots - control points, and its quality is conditioned by large number of such points. There is one more very important element - FIF is a monofractal, which significantly limits the possibilities of multifractal curves interpolation. An interpolating curve has one fractal dimension and thus is not able to reflect to the full a subtle structure of majority of objects found in nature.

It seems that in this situation other, more adequate methods should be urgently searched for. In this study a different approach to the problem of fractal interpolation has been suggested, based on the application of existing fractal curves as base splines. Such an approach certainly gives rise to series of problems and unfortunately not all of them have been solved. Nevertheless, this method seems to be a perspective one and remains open for further modifications.

2 An Idea of the Suggested Method

The suggested method uses, similarly to classic methods, a principle of sampling with constant or variable step - further on, to make it simple, a constant interpolation step is accepted. However, the term interpolation knot is generalized and replaced by a value further on called Local Interpolation Window (LIW). It is an area comprising a fragment of interpolated curve in the range $x_i - \frac{1}{2}\Delta x_i \leq x \leq x_i + \frac{1}{2}\Delta x_i$ around interpolation point x_i ; where Δx_i - the window width (Fig.1). Although the window width need not be equal to the value of interpolation step Δk , it should be adjusted to it - at even step the window width should be the same for all interpolation points. Whereas, the window height must be adjusted to the span of the curve in the window, so that the window would comprise the whole fragment of the curve.

Local Interpolation Window plays the role of a zone from which information is obtained for local reconstruction of a curve. Quantitative characteristics from LIW area, such as for example local fractal dimension complete the set of the stored date, on the basis of which the curve is reconstructed with better or worse approximation. These values are macrocharacteristics of the window limiting the quantity of data indispensable for curve reconstruction. Their selection determines not only the quality of interpolation but also the compression degree. With such a general formulation of the idea of LIW there are not any restrictions as regards the choice of values characterizing the curve in the window. Their selection is conditioned only by the accuracy of the method and inventiveness.

Fractal curves interpolation extorts a priori consideration of the most basic characteristics of such curves, that is their fractal dimension and in the case of stochastic fractals - multifractal spectrum. Such a spectrum can be determined basing on appropriately defined measure enabling to investigate heterogeneity of the distribution of points on a curve. Let random variable ζ be given. Probability P, that this variable is in the interval δx around point x determines measure:

$$\mu\left(B_{\delta x}\left(x\right)\right) = P\left(x - \frac{\delta}{2} < \zeta < x + \frac{\delta}{2}\right) \tag{10}$$

It may be e.g. probability that randomly selected curve point is in the area $B_{\delta x}\left(x\right)$. The appropriate measure, the so called Billingsley's measure [25] for each point x is defined by the expression:

$$G_d\left(x\right) = \lim_{\delta x \to 0} \mu\left(B_{\delta x}\left(x\right)\right)\left(\delta x\right)^{-d} \tag{11}$$

In generality $G_d\left(x\right)$, for an arbitrary value of d can take the values zero and infinite, there is only one value $d = \alpha\left(x\right)$ for which Billingseley's measure is finite. This value is called Holder exponent of the point x:

$$\begin{gathered} G_d\left(x\right) = 0 \;\; when \;\; d < \alpha\;\left(x\right) \\ G_d\left(x\right) = is \;\; finite \;\; if \;\; d = \alpha\;\left(x\right) \\ G_d\left(x\right) = \infty \;\; when \;\; d > \alpha\;\left(x\right) \end{gathered} \tag{12}$$

In the case when points distribution is singular and it occurs in the case of fractal sets, Holder exponent $\alpha\left(x\right)$ determines singularity strength of $G_d\left(x\right)$. Selecting the set of curve points, for which Holder exponent has the same value $H_{\alpha_i} = \left\{x : \alpha\left(x\right) = \alpha_i\right\}$, fractal dimension of this set $D\left(\alpha\right) = Dim\left(H_\alpha\right)$ can be determined (Hausedorff dimension). This dependence is called fractal spectrum.

In practice, the interpolated curves have a discrete structure with determined resolution e.g. in computer algorithms conditioned by the pixel size. In such a case coarse Holder exponent a point x $\alpha\left(x, \delta x\right)$ is defined (for the interval δx around point x):

$$G_{\alpha(x,\delta x)}\left(x\right) = \mu\left(B_{\delta x}\left(x\right)\right)\left(\delta x\right)^{-\alpha(x,\delta x)} \tag{13}$$

$\alpha(x, \delta x)$ can be determined finding the logarithm for the dependence:

$$\ln \mu (B_{\delta x}(x)) = \alpha(x, \delta x) \ln \delta x + \ln G_{\alpha(x,\delta x)}(x) \tag{14}$$

and applying a known procedure of determining points in coordinates $(\ln \mu (B_{\delta x}(x)), \ln(\delta x))$ for intervals of different size. Coarse Holder exponent determines the slope of a straight line fitted using least squares. The procedure should be applied for each pixel.

Applying approximate approach coarse Holder exponent $\alpha(x, \delta x)$ can be averaged with intervals representing LIW width:

$$\alpha(x) = \sum_{i=1}^{N} \langle \alpha(x, \delta x) \rangle_{\Delta x_i} I_{\Delta x_i}(x) \tag{15}$$

where $I_{\Delta x_i}(x)$ is the characteristic function of the set Δx_i defined as:

$$I_{\Delta x_i}(x) = 1 \; when \; x \in \Delta x_i$$

$$I_{\Delta x_i}(x) = 0 \; when \; x \notin \Delta x_i \tag{16}$$

$\langle \alpha(x, \delta x) \rangle_{\Delta x_i}$ - mean value $\alpha(x, \delta x)$ in interval Δx_i

As a result, multifractal curve disintegrates, with some approximation dependent on LIW width, into local (in the intervals Δx_i) monofractal sets of one unique value of fractal dimension $D(\Delta x_i) = D_i$ in each interval. Prime curve can be then reconstructed locally with the given fractal curves of appropriate dimension. In practice the dimension in this case may be calculated for each window like box dimension, which significantly simplifies calculations.

To increase the accuracy of interpolation a larger number of Holder exponent discrete values can be assumed in the interpolation window.

Let $\Delta\alpha_i = \frac{(\alpha_i(x)_{\max} - \alpha_i(x)_{\min})}{L-1}$, (i - window number, $j = 1, ..., L-1$). Then, the fragment of the curve belonging to the given window disintegrates into $L-1$ subsets S_{ij} so that $S_{ij} = \{x : \alpha(x) \in [\alpha_i(x)_{\min}, \alpha_i(x)_{\min} + j\Delta\alpha_i)\}$ for $j = 1, ..., L-1$. Each determined in this way subset has its unique fractal dimension $D_{ij} = Dim(S_{ij})$. In this way fragment of a curve from the given LIW was treated as discrete multifractal which can be reproduced with $L-1$ curves of appropriate dimension.

It is easy to notice, that in the above presented conception LIW width need not be equal to the interpolation step. As smaller width, the dead zone between the windows, becomes reconstructed with the help of base curves from the neighbouring windows. However, the information from this zone, which could increase the accuracy of the interpolation is needlessly lost. In the case of the window width greater than the interpolation step in the zone of window overlapping, there comes to another agreement on the shape and dimension which is definitely an additional asset in interpolation and wider LIW ensures more information and of better quality. However, the window cannot be too wide because

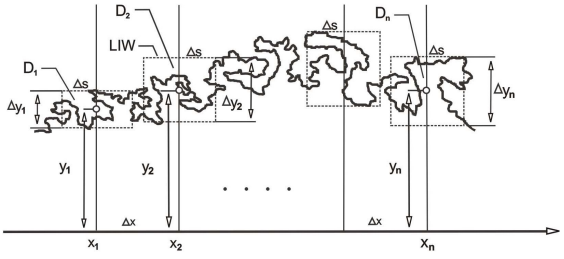


Fig. 1. Interpolation problem for fractal contour - complex structure of curve at difference scales and the lack of one unambiguous value y. Here: LIW - Local Interpolation Window and its Δs - width, Δy_i - height; (x_i, y_i) - coordinates of knots, D_i - fractal dimensions of curve in different windows.

then it loses its local character, what may have a negative effect on the accuracy of reconstruction.

3 Interpolation Base Curves

The above suggested method requires for the reconstruction of the prime curve a set of fractal curves which play the role of base elements in reconstruction. These curves must meet several conditions. Let $S, T \in \mathbf{R}$, $S = [0, 1]$, $T = [1, 2)$ and let $\mathbf{K}$ set of continuous in $U = S \times \mathbf{R}$ fractal curves $K_D(x, y)$, $x \in S, y \in \mathbf{R}$ of the dimension D such that $\forall\ D \in T\ \exists\ K_D(x, y)$ and $sK_D(x, y) = K_{D'}(sx, sy) \rightarrow D = D'$, where $s \in \mathbf{R}$ - scaling factor. The last dependence means that the scaling operation does not change the fractal dimension. Set $\mathbf{K}$ defined like that fulfils the requirements the base curves should meet.

From among numerous possible to define sets of fractal curves meeting the above conditions a family of curves based on well known von Koch curve was chosen. These curves will be called generalized von Koch curve (GKC). A similar idea to these variations of the von Koch curve is attributed to J. Lighthill by Mandelbrot [26]. Their fractal dimension changes in a continuous manner in the interval [1, 2]. Furthermore, when in the case of von Koch curves shrinking factor remains constant and equals r = 1/3 for GKC it is $r = \frac{1}{2(1+\cos(\varphi))}$, where φ is a parameter geometrical sense of which can be seen in fig.2 presenting modified generator GKC.

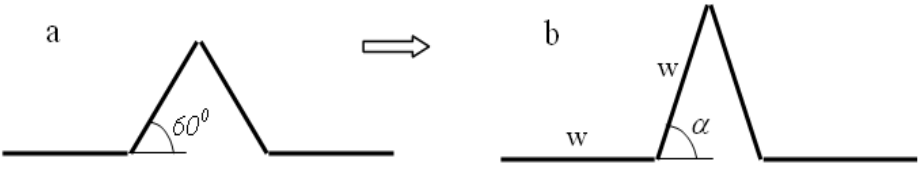


Fig. 2. Modification of the generator of von Koch curve. $w = \frac{1}{2(1+\cos(\alpha))}$

Appropriate contraction maps for GKC have the form:

$$\begin{aligned} w_1(z) &= rz \\ w_2(z) &= rz\exp(i\varphi) + r \\ w_3(z) &= r(z-1)\exp(-i\varphi) + 1 - r \\ w_4(z) &= r(z-1) + 1 \end{aligned} \tag{17}$$

where $z = x + iy$ and related fractal dimension:

$$D = \frac{\ln 4}{\ln\left[2\left(1+\cos\varphi\right)\right]} \tag{18}$$

It is worth noticing that maps (17) are contracting for $|\varphi| < \frac{2}{3}\pi$ then, for $\frac{\pi}{2} < |\varphi| < \frac{2}{3}\pi$ fractal dimension $D > 2$ and it plays the known role of latent dimension [27]. Among characteristic cases of GKC (Fig.3) it is worth paying attention to the case $\varphi = 0$ then the curve has a form of an elementary segment; $\varphi = \frac{\pi}{3}$ - the curve becomes a classic von Koch curve; $\varphi = \frac{\pi}{2}$ - the curve fills the plane (equilateral triangle).

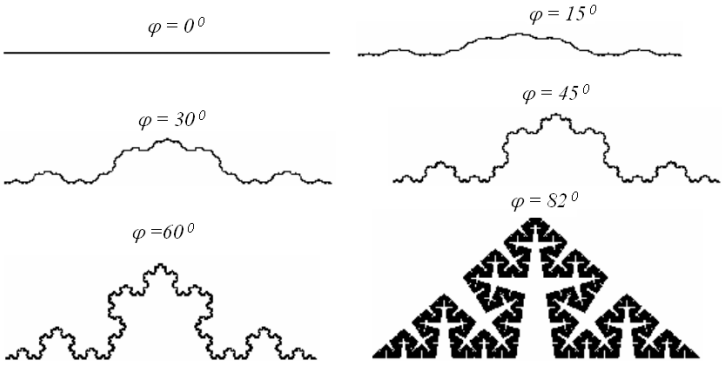


Fig. 3. Examples of GKC for different φ

The above discussed set of curves GKC perfectly meets the requirements determined for fractal base splines, moreover it enables to determine easily fractal dimension basing on the dependence (18). This set is further used in the suggested interpolation algorithm.

4 Principles of Interpolation

In the suggested method the set of base B-splines, which in classic interpolation method is formed of third degree polynominals, is substituted with the above defined set of fractal curves **K** in the form of GKC - $K_i(x, y)$. This set will be called FB-splines.

Let have a given set $N+1$ of interpolation knots (x_0, y_0, D_0, W_0^p), (x_1, y_1, D_1, W_1^p),....., (x_N, y_N, D_N, W_N^p) corresponding to particular interpolation windows LIW. To simplify, it is further assumed that the windows are spaced with uniform step size $x_{i+1} = x_i + \Delta x$; y_i determines the position of the window midpoint, D_i - fractal dimension of the interpolated curve L in i-window - further on box dimension was accepted for calculations; W_i^p - vector of additional parameters $(p = 1, ..., Z)$ characterizing the structure of a curve in i-window - at this stage of considerations the vector was assumed to be $W_i^p = 0$. Generally, curves $K_i(x, y)$ may subject W_i^p to any operations not changing the fractal dimension, and thus to affine transformations. Finally:

$$L \approx \sum_{i=0}^{N} s_i K_i(x - i\Delta x, y, D_i, W_i^p) \tag{19}$$

where fractal curves $K_i(x, y)$ are placed centrally (with maximal value) at interpolation points.

In the above expression attention should be paid to the fact that unlike the classic B-splines method, in which scaling occurs only regarding y, here the curves are scaled in all directions. Moreover, the summation was determined symbolically as summation of fractal curves, whereas the way of this summation must be defined precisely. Investigation of the interpolation problem in the Cartesian coordinate system (x, y) imposes certain rules - the summation must concern co-ordinate y treating simultaneously coordinate x as an independent variable. Furthermore, a curve resulting from summation must preserve topological properties of component curves, particularly continuity. Summation algorithm is presented in fig.4. It is based on a strict principle of summation curve shifting in the successive summation step from the starting point to the next one, only to the nearest neighbour on the left or right side connected topologically with the starting point. For instance (Fig.4), from the point (9,g) the shift in the successive step takes place to point (8,f). On the other hand, the above principle may be interpreted as the summation of each point of one curve with each point of the other curve at the established value x. In the algorithm applied in practice GKC curves were computer generated to pixel level, which enabled to control strictly the ordering of points in the course of summation.

The scale coefficient s_j^n in n-step were calculated iteratively basing on:

$$y_i = \sum_{j=0}^{N} s_j^{n+1} y_j \left(\frac{(i-j)\Delta x}{s_j^n}, D_j \right)^* \tag{20}$$

where: $s_j^0 = 1$, y_i- the given values of interpolation knots, $y_j(x, D_j)$ is a set of y value of curve $K_j(x, y, D_j)$ of the number j for the given value x, where $K_j(x, y, D_j)$ - starting FB-splines are located centrally at each interpolation point. The span of base curves $6\Delta x$ was accepted for the calculations. In the expression (20) the possibility of the occurrence of many values y of FB-splines were taken into account with the established x. Hence:

$$y_j^n(x, D_j)^* = \frac{1}{2}\left[\sup y_j^n(x, D_j) + \inf y_j^n(x, D_j)\right] \tag{21}$$

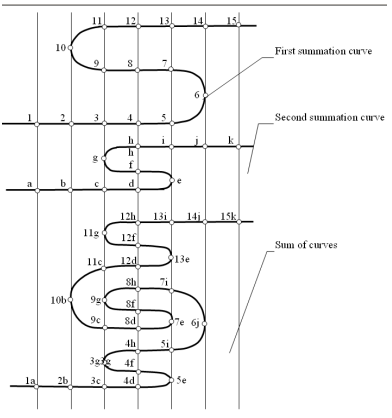


Fig. 4. Summation algorithm of points of curves in the proposed interpolation method

Iteration average quadratic error is a measure of the calculations accuracy:

$$\delta^n = \sum_{i=0}^{N} \left(y_i - \sum_{j=0}^{N} y_j \left((i-j)\, \Delta x, D_j \right)^* \right)^2 \tag{22}$$

A well known problem of the lack of data at the ends of interpolation curves should be taken into account in the interpolation. These values should be additionally defined.

In the carried out studies, the shape of the interpolating curve between the knots was found to be strongly dependent on the base line location - x axis, relative to which FB-splines are generated. The proper choice of this location ensures the convergence of the iterative process and optimal matching of the shape. This property can be used creating an additional iterative loop in the matching algorithm. Selection of appropriate position of x axis results from minimization of average square error calculated for all curve pixels:

$$\delta(z_i) = \sum_{m} \left(\mathbf{y}_m^{z_i} - y_m^{z_i} \right)^2 \tag{23}$$

where: $\mathbf{y}_m^z$, y_m^z - values y of interpolated and interpolating curve, respectively, where the summation concerns all the values (pixels) x; z_i determines different shift in the successive iteration step of axis x in relation to the initial position.

Attention should be paid to the case when the interpolated curve rapidly changes values y or fractal dimension in neighbouring LIW then, the iterations determined by the dependence (20) may not be convergent. In such a case it is necessary to break interpolation curve into smaller segments and interpolate each of them independently. The obtained in this way interpolating curves should be stitched together taking advantage of the fact that the values of these curves on their ends are not determined - they may be given in optional way.

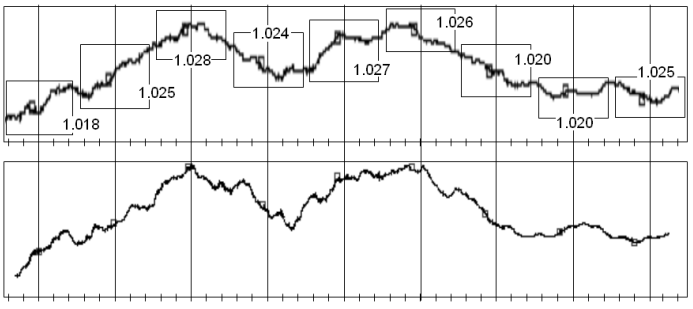


Fig. 5. Example of interpolation. On top there is showed digitalized contour of terrain landscape, which was interpolated. LIW are drowned around each interpolation knots and there are given values of fractal dimensions. At the bottom there is showed suitable interpolation curve.

Above an example is presented of an interpolation carried out with the above described method for digitalized contour of mountain elevation (fig.5). Nine interpolation knots were applied and two not shown which completed the information on the curve ends. Initiating GKC had the span $6\Delta x$. The values of fractal dimension were estimated with standard method [28] as box dimension in indicated LIW.

5 Conclusion

The initial analysis demonstrated that the presented method of interpolation can be applied very effectively for reconstruction of natural, fractal profiles or irregular series of data. The compression coefficient in this case can be extremely high because loss of information - the result of high compression - is compensated by more precise determination of non-local characteristics in a wider interpolation window around the interpolation point. The method can be improved taking into account greater quantity of additional characteristics in LIW and more precise analysis of the reproduced curve spectrum.

References

1. Barnsley M. F.: Fractal Functions and Interpolation. Constructive Approximation. 2, (1986), 303 - 332
2. Mandelbrot B.: The Fractal Geometry of Nature. Freeman, San Francisco, CA, (1982)
3. Barnsley M. F.: Fractals Everywhere. 2nd ed., Academic Press, Inc., Boston, (1993)
4. Hearn D., Baker M. P.: Computer Graphics. Prentice Hall, (1997).
5. Stanley H. E. and Ostrosky N.: On Growth and Form: Fractal and Non-Fractal Patterns in Physics. Nijhoff, Boston, Mass., (1986)
6. Nittman J., Daccord G., and Stanley H. E.: Fractal growth of viscous fingers: quantative characterization of a fluid instability phenomenon. Nature, London, 314, 141 (1985).

7. Lovejoy S.: Area-perimeter relation for rain and cloud areas, Science, 216, (1982), 185-187
8. Russ J. C.: Fractal Surfaces. New York, Plenum Press, (1984)
9. Peitgen H., and Saupe D.: The science of f ractal images. New York, Springer-Verlag, (1988)
10. Frisch U., and Parisi G.: Fully developed turbulence and intermittency. Turbulence and Predictability in Geophysical Fluid Dynamics and Climate Dynamics. M. Ghil et al. Eds., North-Holland, (1985)
11. Kawaguchi Y.: A morphological study of the form of nature. Comput. Graph. 16, 3, (1982)
12. Matsushita M.: Experimental Observation of Aggregations. In: The Fractal Approach to Heterogeneous Chemistry: Surfaces, Colloids, Polymers. D. Avnir (ed.), J. Wiley and Sons, Chichester (1989)
13. West B.: Fractal Physiology and Chaos in Medicine. World Scientific Publ. Co., Singapore (1990)
14. Feder J.: Fractals, Plenum Press, New York, (1988)
15. Voss R.: Random fractals: characterisation and measurement. In Scaling Phenomena in Disordered Systems, edited by R. Pynn and A. Skjeltorp, New York, Plenum Press, (1986)
16. Calvet L. and Fisher A.: Multifractality in Asset Returns: Theory and Evidence. The Review of Economics and Statistics, 84, 3, (2002), 381-406
17. Hutchinson J. E.: fractals and Self Similarity. Ind. Univ. J. Math. 30, (1981), 713-747
18. Barnsley M.F. and Harrington A. N.: The calculus of fractal interpolation functions. Journal of Appoximation Theory, 57, (1989), 14-34
19. Navascues M.A., Sebastian M.V.: Some Results of Convergence of Cubic Spline Fractal Interpolation Functions. Fractals, 11, 1, (2003), 1-7
20. Barnsley M.F., Elton J., Hardin D., and Massopust P.: Hidden Variable Fractal Interpolation Functions. SIAM J. Math. Anal. 20, (1989), 1218-1242
21. Massopust P. R.: Fractal Functions, Fractal Surfaces and Wavelets. Academic Press, San Diego, CA, (1994)
22. Navascues M.A.: Fractal Trigonometric Interpolation. Electronic Transactions on Numerical Analysis. 20, (2005), 64-74
23. Wittenbrink C.M.: IFS fractal interpolation for 2D and 3D visualization. In Proc. IEEE Visualation '94, (1995), 77-83
24. Zhao N.: Construction and application of fractal interpolation surfaces. The Visual Computer, 12, (1996), 132-146
25. Billingsley P.: Ergodic Theory and Information. John Wiley and Sons, Inc. (1965)
26. Mandelbrot B., Les Objects Fractals: Forme, Hasard et Dimension, Flammarion, Paris, 1975
27. Mandelbrot B.B., Frame M.: Fractals. Encyclopedia of Physical Science and Technology, June 28, 2001, Volume 6, Academic Press, Yale University, (2002)
28. Theiler J.: Estimating fractal dimension. J. Opt. Soc. Am. A, 7, 6, (1990)

Integrating Lookahead and Post Processing Procedures with ACO for Solving Set Partitioning and Covering Problems

Broderick Crawford[1,2] and Carlos Castro[2]

[1] Engineering Informatic School, Pontifical Catholic University of Valparaíso, Chile
[2] Informatic Department, Federico Santa María Technical University, Chile
broderick.crawford@ucv.cl, Carlos.Castro@inf.utfsm.cl

Abstract. Set Covering Problems and Set Partitioning Problems can model several real life situations. In this paper, we solve some benchmarks of them with Ant Colony Optimization algorithms and some hybridizations of Ant Colony Optimization with Constraint Programming techniques. A lookahead mechanism allows the incorporation of information on the anticipated decisions that are beyond the immediate choice horizon. The ants solutions may contain redundant components which can be eliminated by a fine tuning after the solution, then we explore Post Processing procedures too, which consist in the identification and replacement of the columns of the ACO solution in each iteration by more effective columns. Computational results are presented showing the advantages to use additional mechanisms to Ant Colony Optimization.

1 Introduction

Set Covering Problems (SCP) and Set Partitioning Problems (SPP) are two types of problems that can model several real life situations [10]. In this work, we solve some benchmarks of them with Ant Colony Optimization (ACO) algorithms and some hybridizations of ACO with Constraint Programming techniques like Forward Checking and Full Lookahead, and Post Processing procedures too.

There exist problems for which ACO is of limited effectiveness. Among them a prominent role is played by very strongly constrained problems. They are problems for which neighborhoods contain few solutions, or none at all, and local search is of very limited use. Probably, the most significant of such problems is the SPP. Currently, no ACO algorithm has been proposed for SPP, and a direct implementation of the basic ACO framework is incapable of obtaining feasible solutions for many standard tested instances [16]. The best performing metaheuristic for SPP is a genetic algorithm due to Chu and Beasley [6,4]. There exist already some first approaches applying ACO to the SCP. In [1,14] ACO has been used only as a construction algorithm and the approach has only been tested on some small SCP instances. More recent works [13,15,12] apply Ant Systems to the SCP and related problems using techniques to remove redundant columns and local search to improve solutions.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1082–1090, 2006.
© Springer-Verlag Berlin Heidelberg 2006

In this paper, we explore the addition to the ACO algorithm of a lookahead mechanism usually used in complete techniques. Trying to solve larger instances of SPP with the original AS or ACS implementation derives in a lot of unfeasible labeling of variables, and the ants can not obtain complete solutions using the classic transition rule when they move in their neighborhood. In this paper we propose the addition of a lookahead mechanism in the construction phase of ACO in order to that only feasible solutions are generated. The lookahead mechanism allows the incorporation of information about the instantation of variables after the current decision. The idea differs from that proposed by [18] and [11], these authors proposed a look ahead function evaluating the pheromone in the Shortest Common Supersequence Problem and esimating the quality of a partial solution of a Industrial Scheduling Problem respectively.

Additionally, solving SCP ants may contain redundant solution components which can be eliminated by a fine tuning after the solution. In order to achieve this post procedure, we try to implement an hybrid outline based on morphing procedures [5], which consist in the identification and replacement of the columns of the ACO solution in each iteration by morphs or more effective columns: minimum cost and same cover.

This paper is organized as follows: In Section 2, we formally describe SCP and SPP using mathematical programming models. In Section 3, we present experimental results obtained when applying two basic ACO algorithms for solving some standard benchmakrs. Sections 4 and 5 present results obtained when adding a postprocessing procedure and Constraint Programming techniques, respectively, to the two basic ACO algorithms. In Section 6, we present results when adding both postprocessing procedure and Constraint Programming techniques to the two basic ACO algorithms. Finally, in Section 7 we conclude the paper and give some perspectives for future research.

2 Problem Description

SPP is the problem of partitioning a given set into mutually independent subsets while minimizing a cost function defined as the sum of the costs associated to each of the eligible subsets. Its importance derives from the fact that many real life situations can be modeled as SPP, and in fact many combinatorial optimization problems (such as, crew scheduling, vehicle routing, project scheduling, and warehouse location problems, to name a few) can be modeled as SPP with maybe some additional constraints.

In SPP we are given a $m \times n$ matrix $A = (a_{ij})$ in which all the matrix elements are either zero or one. Additionally, each column is given a non-negative cost c_j. We say that a column j covers a row i if $a_{ij} = 1$. Let J denotes a subset of the columns and x_j a binary variable which is one if column j is chosen and zero otherwise. The SPP can be defined formally as follows.

$$Min \quad f(x) = \sum_{j=1}^{n} c_j \times x_j \tag{1}$$

Subject to

$$\sum_{j=1}^{n} a_{ij} \times x_j = 1; \quad \forall i = 1, \ldots, m \tag{2}$$

These constraints enforce that each row is covered by exactly one column. The SCP is a SPP relaxation. The goal in the SCP is to choose a subset of the columns of minimal weight which covers every row.

The SCP can be defined formally using constraints to enforce that each row is covered by at least one column as follows.

$$\sum_{j=1}^{n} a_{ij} \times x_j \geq 1; \forall i = 1, \ldots, m \tag{3}$$

3 Ant Colony Optimization

Although in the following description the reader is expected to be familiar with ACO algorithm proposed in [8,7], a brief introduction to ACO may be in place.

ACO is a paradigm for designing constructive metaheuristic algorithms for combinatorial optimization problems that is inspired by the shortest path searching behavior of real ant colonies. ACO algorithms are based on a colony of artificial computational agents that work cooperatively and communicate through artificial pheromone trails. ACO algorithms are essentially construction algorithms: in each algorithm iteration, every ant constructs a solution to the problem applying a transition rule labeling variables. Once every ant has generated a solution, it can deposit an amount of pheromone in the solution components. The pheromone deposited is a function of the solution quality. This information will guide the search of the remaining ants of the colony in the future.

ACO can be applied directly to the SCP and SPP. The columns are chosen as the solution components and have associated a cost and a pheromone trail. Constraints say that each column can be visited by an ant once and only once and that a final solution has to cover all rows.

A walk of an ant over the graph representation corresponds to the iterative addition of columns to the partial solution obtained so far. Each ant starts with an empty solution and adds columns until a cover is completed. A pheromone trail τ_j and a heuristic information η_j are associated to each eligible column j. A column to be added is chosen with a probability that depends of pheromone trail and the heuristic information. The most common form of the ACO decision policy when ants work with components is [9]:

$$p_j^k(t) = \frac{\tau_j * \eta_j^\beta}{\sum_{l \in N^k} \tau_l [\eta_l]^\beta} \quad \text{if } j \in N^k \tag{4}$$

where N^k is the feasible neighborhood of the ant k. The β parameter controls how important is η in the probabilistic decision.

Pheromone trail τ_j. One of the most crucial design decisions to be made in ACO algorithms is the modeling of the set of pheromones. In the original ACO implementation for TSP the choice was to put a pheromone value on every link between a pair of cities, but for other combinatorial problems often can be assigned pheromone values to the decision variables (first order pheromone values) [9]. In this work the pheromone trail is put on the problems componentes (each elegible column j) instead of the problems connections. And setting good pheromone quantity is a non trivial task too. The quantity of pheromone trail laid on columns is based on the idea: *the more pheromone trail on a particular item, the more profitable that item is* [14]. Then, the pheromone deposited in each component will be in relation to its frequency in the ants solutions. In this work we divided this frequency by the number of ants obtaining better results.

Heuristic information η_j. In this paper we use a dynamic heuristic information that depends on the partial solution of an ant. It can be defined as $\eta_j = \frac{e_j}{c_j}$, where e_j is the so called cover value, that is, the number of additional rows covered when adding column j to the current partial solution, and c_j is the cost of column j. In other words, the heuristic information measures the unit cost of covering one additional row. An ant ends the solution construction when all rows are covered. Figure 1 describe two basic ACO algorithms to solve SCP and SPP.

```
1  Procedure ACO_for_SCP_and_SPP
2   Begin
3    InitParameters();
4     While (remain iterations) do
5      For k := 1 to nants do
6       While (solution is not completed)
7        AddColumnToSolution(election)
8        AddToTabuList(k);
9       EndWhile
10     EndFor
11     UpdateOptimum();
12     UpdatePheromone();
13    EndWhile
14  Return best_solution_founded
15 End.
```

Fig. 1. ACO algorithm for SCP and SPP

Figure 1 describes the basic structure of ACO algorithm to solve SCP and SPP. In this work, we use two instances of ACO: Ant System (AS) and Ant Colony System (ACS) algorithms, the original and most famous algorithms in the ACO family [8]. ACS improves the search of AS using: a different transition rule in the constructive phase, exploting the heuristic information in a more rude form, using a list of candidates to future labeling and using a different treatment of pheromone. ACS has demostrated better performance than AS in a wide range of problems, Table 1 shows that it occurs in SPP and SCP too.

Table 1 presents results obtained when applying these algorithms for solving standard benchmarks taken from ORLIB[3]. The first four columns present the

Table 1. ACO

Problem	Rows	Columns	Optimum	AS	ACS
sppnw39	25	677	10080	11670	10758
sppnw34	20	899	10488	13341	11289
sppnw26	23	771	6796	6976	6956
sppnw23	19	711	12534	14304	14604
scp41	200	1000	429	473	463
scp42	200	1000	512	594	590
scp48	200	1000	492	524	522
scp51	200	1000	253	289	280
scp61	200	1000	138	157	154
scp62	200	1000	146	169	163
scp63	200	1000	145	161	157

problem code, the number of rows, the number of columns, and the best known solution for each instance, respectively. The last two columns present the best cost obtained when appyling AS and ACS, respectively, to solve the benchmarks.

The algorithms has been run with the following parameters setting: influence of pheromone (alpha)=1.0, influence of heuristic information (beta)=0.5 and evaporation rate (rho)=0.4 as suggested in [14,15,9]. The number of ants has been set to 120 and the maximum number of iterations to 160, so that the number of generated candidate solutions is limited to 19.200. For ACS the list size was 500 and Qo=0.5.

Algorithms were implemented using ANSI C, GCC 3.3.6, under Microsoft Windows XP Professional version 2002.

4 ACO with Post Processing

In a postoptimization step, an ant can remove redundant columns that only cover rows which are also covered by a subset of other columns in the final solution or apply some additional local search to improve solutions. Solving SCP and SPP each ant starts with an empty solution and constructs a complete solution by iteratively adding columns until all rows are covered.

In the application of ACO to other problems, such as the TSP, there are some differences. For example, the solution construction of the individual ants does not necessarily end after the same number of steps of each ant, but only when a cover is completed. Moreover, the order in which columns are added to a SCP solution does not matter, and that in a solution constructed by the ants may contain redundant solution components which can be eliminated by a fine tuning after the solution. In order to achieve this post procedure, we try to implement different hybrid outlines based on morphing procedures, which consist in the identification and replacement of the columns of the ACO solution in each iteration by morphs or more effective columns [5]. Good results are obtained on a set of SCP benchmark instances with this mechanism. Post processing locally optimizes the ants solutions and these locally optimized solutions in each iteration are used in the pheromone update. Table 2 presents results when adding this postprocessing step to the basic ACO algorithms.

Table 2. ACO with Post Processing

Problem	AS + PP	ACS + PP
scp41	438	435
scp42	536	530
scp48	516	499
scp51	264	265
scp61	143	143
scp62	155	152
scp63	150	149

The proposed post processing procedure applied to SCP shows better results than classical ACO. Better costs obtained are more expensive in time consuming but never pass the 900 seconds of execution in the worst case. The trade off between cost and execution time is solved in benefit of the cost.

5 ACO with Constraint Programming

In the original ACO implementation the SPP solving derives in a lot of unfeasible labeling of variables, and the ants can not complete solutions. Forward Checking seems to be the easiest way to prevent future conflicts in a constructive metaheuristic.

Instead of performing arc consistency to the instantiated variables, it performs a restricted form of arc consistency to the not yet instantiated variables. We speak about restricted arc consistency because Forward Checking checks only the constraints between the current variable and the future variables. When a value is assigned to the current variable, any value in the domain of a "future" variable which conflicts with this assignment is (temporarily) removed from the domain. The advantage of this is that if the domain of a future variable becomes empty, it is known immediately that the current partial solution is inconsistent.

Forward Checking therefore allows branches of the search tree that will lead to failure to be pruned earlier than with simple backtracking. Note that whenever a new variable is considered, all its remaining values are guaranteed to be consistent with the past variables, so checking an assignment against the past assignments is no longer necessary. This reduces the search tree and (hopefully) the overall amount of work done. But it should be noted that Forward Checking does more work when each assignment is added to the current partial solution. Adding Forward Checking to ACO means that columns are chosen if they do not produce any conflict with respect to the next column to be chosen. Recently, some efforts have been done in order to integrate Constraint Programming techniques to ACO algorithms for the Job Scheduling Problem [17].

Forward checking checks only the constraints between the current variable and the future variables. So, why not to perform full arc-consistency that will further reduces the domains and removes possible conflicts? This approach is called (Full) Look Ahead or maintaining arc-consistency (MAC). The advantage of look

ahead is that it detects also the conflicts between future variables and therefore allows branches of the search tree that will lead to failure to be pruned earlier than with Forward Checking. Also as with Forward Checking, whenever a new variable is considered, all its remaining values are guaranteed to be consistent with the past variables, so checking an assignment against the past assignments is not necessary. Look ahead prunes the search tree further more than Forward Checking but, again, it should be noted that look ahead does even more work when each assignment is added to the current partial solution than Forward Checking. Adding Full Look Ahead to ACO means that columns are chosen using recursively the same ides that Forward Checking and so we detect conflicts before a solution is completed. Table 3 presents results when adding Forward Checking and Full Look Ahead techniques to the basic ACO algorithms.

Table 3. ACO with Constraint Programming

Problem	AS + FC	AS + FLA	ACS + FC	ACS + FLA
sppnw39	11322	10722	10545	11322
sppnw34	10713	10713	10797	10713
sppnw26	6880	7192	6880	6850
sppnw23	13932	13254	12880	13400
scp41	458	2115	683	842
scp42	574	1990	740	752
scp48	537	1952	731	752
scp51	289	1975	464	526
scp61	155	1081	276	332
scp62	170	1004	280	352
scp63	161	763	209	267

The effectiveness of Constraint Programming is showed to the SPP, the strongly constrained problem characteristic of this problem does the stochastic behavior of ACO improved with lookahead techniques in the construction phase, so that almost only feasible solutions are induced, solving the drawback of pure ACO for the SPP deriving in a lot of unfeasible labeling of variables, and then can not obtain complete solutions.

The concept of Arc Consistency plays an essential role in Constraint Programming as a problem simplification operation and as a tree pruning technique during search through the detection of local inconsistencies among the uninstantiated variables. We have shown that it is possible to add Arc Consistency to any ACO algorithms and the computational results confirm that the performance of ACO is possible to improve with this type of hibridization. This integration improves the process, mainly with respect to success costs instead running times. Obviously, in ACS a labeling step is simply a value assignment and therefore cheap. In CP much more is involved in a single labeling step, as it triggers propagation, i.e. a potentially more expensive computational process. But the trade off in all cases considered is very convenient.

For SCP, the huge size of the search space and the relaxation of the constraints does ACO algorithms work better than ACO with Constraint Programming considering the same execution conditions.

6 ACO with Constraint Programming and Post Processing

A complementary possibility of implementation is to merge the Constraint Programming approach and Post Processing with ACO for the SCP. Although for SCP the Constraint Programming we showed that not improved ACO, is interesting know the behavior and potential of Post Processing with both. Table 4 presents results when adding Constraint Programming techniques and a post-processing step to the basic ACO algorithms.

Table 4. ACO with Constraint Programming and Post Processing

Problem	AS + FC + PP	AS + FLA + PP	ACS + FC + PP	ACS + FLA + PP
scp41	957	999	873	741
scp42	959	1007	763	776
scp48	1004	797	809	830
scp51	768	733	678	446
scp61	456	422	302	254
scp62	452	504	262	339
scp63	334	484	329	

The Post Processing procedure proposed applied to SCP after Constraint Programming techniques can not improve the results than classical ACO. Considering the same execution conditions the implementation of ACO with Post Processing showed the better results with a very appropriate trade off in relation to the execution time.

7 Conclusions and Future Directions

SCP and SPP problems had been discussed, different solving strategies for these problems has been presented. It based on Ant Colony Optimization algorithms. Computational results confirm that the performance of ACO is possible to improve with some classes of hibridization. We have successfully combined Constraint Programming and ACO for the problem of Set Partitioning solving benchmarks of data sets. Our main conclusion from this work is that we can improve ACO with CP. And in the SCP problem, improved ACO with Post Processing procedures showed better results than the original ACO proposal.

Future versions of the algorithm will study the pheromone treatment representation and the incorporation of available techniques in order to reduce the input problem (Pre Processing) and improve the solutions given by the ants (Post Processing). The ants solutions may contain redundant components which can be eliminated by a fine tuning after the solution, then we will explore Post Processing procedures, which consist in the identification and replacement of the columns of the ACO solution in each iteration by more effective columns. Besides, the ants solutions can be improved by other local search methods like Simulated Annealing or Tabu Search.

References

1. D. Alexandrov and Y. Kochetov. Behavior of the Ant Colony Algorithm for the Set Covering Problem. In *Proc. of Symp. Operations Research*, pp 255–260. Springer Verlag, 2000.
2. E. Balas and M. Padberg. Set Partitioning: A Survey. *SIAM Review*, 18:710–760, 1976.
3. J. E. Beasley. OR-Library:Distributing test problem by electronic mail. *Journal of Operational Research Society*, 41(11):1069–1072, 1990.
4. J. E. Beasley and P. C. Chu. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94(2):392–404, 1996.
5. M. J. Brusco, L. W. Jacobs, and G. M. Thompson. A morphing procedure to supplement a simulated annealing heuristic for cost and coverage correlated set covering problems. *Annals of Operations Research*, 86:611–627, January 1999.
6. P. C. Chu and J. E. Beasley. Constraint handling in genetic algorithms: the set partitoning problem. *Journal of Heuristics*, 4:323–357, 1998.
7. M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant Algorithms for Discrete Optimization. *Artificial Life*, 5:137–172, 1999.
8. M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
9. M. Dorigo and T. Stutzle. *Ant Colony Optimization*. MIT Press, USA, 2004.
10. A. Feo, G. Mauricio, and A. Resende. A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem. *OR Letters*, 8:67–71, 1989.
11. C. Gagne, M. Gravel and W.L. Price. A Look-Ahead Addition to the Ant Colony Optimization Metaheuristic and its Application to an Industrial Scheduling Problem. In J.P. Sousa et al., eds., *Proceedings of the fourth Metaheuristics International Conference MIC'01*, July 16-20, 2001. Pages 79-84.
12. X. Gandibleux, X. Delorme and V. T'Kindt. An Ant Colony Algorithm for the Set Packing Problem. In M. Dorigo et al., editor, *ANTS 2004*, vol 3172 of *LNCS*, pp 49–60. SV, 2004.
13. R. Hadji, M. Rahoual, E. Talbi, and V. Bachelet. Ant colonies for the set covering problem. In M. Dorigo et al., editor, *ANTS 2000*, pp 63–66, 2000.
14. G. Leguizamón and Z. Michalewicz. A new version of Ant System for subset problems. In *Congress on Evolutionary Computation, CEC'99*, pp 1459–1464, Piscataway, NJ, USA, 1999. IEEE Press.
15. L. Lessing, I. Dumitrescu, and T. Stutzle. A Comparison Between ACO Algorithms for the Set Covering Problem. In M. Dorigo et al., editor, *ANTS 2004*, vol 3172 of *LNCS*, pp 1–12. SV, 2004.
16. V. Maniezzo and M. Milandri. An Ant-Based Framework for Very Strongly Constrained Problems. In M. Dorigo et al., editor, *ANTS 2002*, vol 2463 of *LNCS*, pp 222–227. SV, 2002.
17. B. Meyer and A. Ernst. Integrating ACO and Constraint Propagation. In M. Dorigo et al., editor, *ANTS 2004*, vol 3172 of *LNCS*, pp 166–177. SV, 2004.
18. R. Michel and M. Middendorf. An Island model based Ant system with lookahead for the shortest supersequence problem. *Lecture notes in Computer Science, Springer Verlag*, 1498:692–701, 1998.

Learning Algorithms for Scheduling Using Knowledge Based Model

Ewa Dudek-Dyduch[1] and Tadeusz Dyduch[2]

[1] Institute of Automatics
edd@ia.agh.edu.pl
[2] Institute of Computer Science
tdyduch@agh.edu.pl
AGH University of Science and Technology
al. Mickiewicza 30, PL 30-059 Krakow

Abstract. The aim of the paper is to present a conception of learning algorithms for discrete manufacturing processes control. A general knowledge based model of a vast class of discrete manufacturing processes (DMP) is given. The model is a basis for the method of the synthesis of intelligent, learning algorithms that use information on the process gained in previous iterations as well as an expert knowledge. To illustrate the presented ideas, the scheduling algorithm for a special NP-hard problem is given.

1 Introduction

Methods of knowledge representation (KR) are methods for representation of real world in computer. A lot of the methods have been worked out. They must be suitable for the modelled area, character of pieces of information (e.g. uncertain, probabilistic or deterministic) and first of all an aim for which the knowledge is to be used. Recently, a lot of investigations referring to knowledge based control method are carried out because models and knowledge about controlled processes are the vital parts of the controlling systems. In the knowledge-based intelligent process planning systems, knowledge acquisition plays significant role. In order to discover association rules under uncertainty, fuzzy decision techniques and entropy-based analysis methods as well as fuzzy clustering integrated with variable precision rough set are used [16]. On the other hand e.g. for autonomous unmanned vehicle the system is required being able to dynamically construct a knowledge structure representing a process under control, meeting the constraints associated with a particular process. The system should be able to manage and monitor changes in the structure and derive knowledge about it. Usually process constraints are specified with temporal logic formulas and monitored using appropriate execution monitor [1]. At the same time KR methods for computer-aided manufacturing are developed [10], [9]. They are vital for information system for manufacturing management such as MRPII or/and ERPII. Each of the system contains components so called *shop floor control*, especially referring to control of discrete manufacturing processes and scheduling. However,

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1091–1100, 2006.
© Springer-Verlag Berlin Heidelberg 2006

there are no proper optimal control algorithms implemented there. The implemented algorithms use only simple control rules. On the other hand a variety of discrete processes are described and their optimisation algorithms have been presented in the scientific literature. The question arises: why there are no these optimisation algorithms implemented in information management systems, produced even by the best computer firms such as SAP, Oracle or IFS? According to the authors one of the reasons is lack of common knowledge representation method for manufacturing processes control.

The paper deals with knowledge based modelling of discrete manufacturing/production processes and its applications for manufacturing process planning algorithms. It presents developing of ideas given in [2]. Its aim is 3-fold:

- to present a general, knowledge based model of a vast class of discrete deterministic processes that is a class of discrete manufacturing/production processes (DMP),
- to discuss the model applications, especially for designing algorithms with gathering information,
- to present exemplary intelligent search algorithm for scheduling some NP-hard problem.

The control of DMP (scheduling DMP) lies in determining a manner of performing some set of jobs under restrictions referring to machines/devices, resources, energy, time, transportation possibilities, order of operation performing and others. Most of control algorithms are approximate (heuristic) due to NP-hardness of the optimisation problems. Within the frame of artificial intelligence, one attempts both formal elucidation of heuristic algorithm ideas and giving some rules for creating them (metaheuristics) [4], [5], [11], [12]. The paper is connected with this direction of the research. It uses formal model based on a special kind of the multistage decision process given below.

2 Knowledge Based Model of DMP

Simulation aimed at scheduling of any DMP consists in determining a sequence of process states and the related time instances. The new state and its time instant depend on the previous state and the decision that has been realised (taken) then. Decision determines the job to be performed, resources, transport unit etc. Manufacturing processes belong to the larger class of discrete processes, namely discrete deterministic processes (DDP). The formal model of DDP given in [3], [6], [7] will be here adopted for DMP.

Definition 1. *A discrete manufacturing/production process DMP is a process that is defined by the sextuple* $DMP = \left(U, S, s_0, f, S_N, S_G\right)$ *where* U *is a set of control decisions or control signals,* $S = X \times T$ *is a set named a set of generalized states,* X *is a set of proper states,* $T \subset \mathbb{R}^+ \cup \{0\}$ *is a subset of non negative real numbers representing the time instants,* $f : U \times S \to S$ *is a partial function called a transition function, (it has not to be determined for all elements of the set*

$U \times S$), $s_0 = (x_0, t_0)$, $S_N \subset S$, $S_G \subset S$ are respectively: an initial generalized state, a set of not admissible generalized states, and a set of goal generalized states, i.e. the states in which we want the process to take place at the end.

The transition function is defined by means of two functions, $f = (f_x, f_t)$ where $f_x : U \times X \times T \to X$ determines the next state $f_t : U \times X \times T \to T$ determines the next time instant. It is assumed that the difference $\Delta t = f_t(u, x, t) - t$ has a value that is both finite and positive.

Thus, as a result of the decision u that is taken or realised at the proper state x and the moment t, the state of the process changes for $x' = f_x(u, x, t)$ that is observed at the moment $t' = f_t(u, x, t) = t + \Delta t$.

Because not all decisions defined formally make sense in certain situations, the transition function f is defined as a partial one. Thanks to it, all limitations concerning the control decisions in a given state s can be defined in a convenient way by means of so-called sets of possible decisions $U_p(s)$, and defined as: $U_p(s) = \{u \in U : \ (u, s) \in \text{Dom } f\}$.

At the same time a DMP is represented by a set of its trajectories that start from the initial state s_0. It is assumed that no state of a trajectory, apart from the last one, may belong to the set S_N or has an empty set of possible decisions. Only a trajectory that ends in the set of goal states is admissible. The control sequence determining an admissible trajectory is an admissible control sequence (decision sequence). The task of optimisation lies in the fact of finding such an admissible decision sequence $\tilde{u}$ that would minimize a certain criterion Q.

In the most general case, sets U and X may be presented as a cartesian product $\mathbf{U} = U^1 \times U^2 \times \ldots U^m$, $X = X^1 \times X^2 \times \ldots X^n$ i.e. $u = (u^1, u^2, \ldots u^m)$, $x = (x^1, x^2, \ldots, x^n)$. There are no limitations imposed on the sets, in particular they have not to be numerical. Thus values of particular co-ordinates of a state may be names of elements (symbols) as well as some objects (e.g. finite set, sequence etc.). Particular u^i represent separate decisions that must or may be taken at the same time. The sets S_N, S_F, and U_p are formally defined with use of logical formulae. Therefore, the complete model constitutes a specialised form of a knowledge-based model (logic-algebraic model). According to it's structure the knowledge on DMP is represented by coded information on U, S, s_0, f, S_N, S_G. Function f may be defined by means of procedure or by means of rules of type IF..THEN.

The presented paradigm of knowledge based model consists of the following main procedures realising rules IF..THEN, utilizes by control algorithms: procedure that generates and examines subsets of possible decisions $U_p(s)$, procedures that realize the function f (in the most cases it is a vector function), i.e. determine the next state $(x', t') = f(u, x, t)$, procedures that examine if the state belongs to the set S_N or S_G. All the procedures are based on information acquired from three sources: description of manufacturing process that take into account all its limitation, expert knowledge referring to control rules, results of computer simulation experiments. The basic structure of DMP (def.1) is usually created on a basis of process technology description. Basing on additional expert knowledge (or analysis of DMP) subsets of states can be differentiated, for which

best decisions or some decision choice rules R (control rules) are known. Similarly, some subsets of *advantageous* or *disadvantageous* states for the controlled process can be determined. Formally, the knowledge allow us restrict sets of possible decisions U_p. Knowledge represented by the basic knowledge structure DMP (def.1) enriched by expert knowledge creates the knowledge-based model KBM of DMP. The knowledge can be enriched further as a result of simulation experiments.

Basing on the model of DMP different classes of algorithms can be formally defined and analysed. For example in [6], [14], classes of branch & bound algorithms for DMP control optimisation have been differentiated as well as some rules of automatic creation of lower bounds have been given. In the next section application of KBM of DMP for intelligent algorithm is presented.

3 Search Algorithms with Gathering Information

The most popular search algorithms consist in generating consecutive, possibly better and better, trajectories. They use a specially created function or local optimisation task for the choice of the *best* decision at each state of the generated trajectory. The criterion for local optimisation is called a preference function or simply heuristics. In this section we present a conception of algorithms that gain information on the process and also use expert knowledge.

In the author's earlier paper [4], [7], a certain general 3-stage method for designing the heuristic algorithms of this type is proposed. Let us recall it briefly. At the first stage, one formulates some conditions for the optimal (suboptimal) solution. They refer directly to subsets of decisions, or/and determine the state sets that are advantageous (or disadvantageous) from the criterion point of view or for a possibility of generating an admissible trajectory. At the second stage, one determines a local optimisation task. In order to do it, the information about the distinguished, at the first stage, *advantageous* or *disadvantageous* states as well as information on S_G, S_N and sets of possible decisions is used. As we need the generated trajectory to run only through the advantageous states and to avoid the disadvantageous ones, it seems most natural to introduce any measure of distance in the state space, and to assume some local criterions. It was explained in [7] that different semimetrics can be used as approximate measures of distance. Basing on the local change of the global criterion Q and maximization (minimization) of the mentioned distances, we obtain the substitute local problem, usually a multicriteria one. At the third stage, one should determine the manner of solving the local multicriteria optimisation task. The basic ideas of multicriteria decision approach [3], [4], [15], can be applied here. For learning algorithms, however, the most useful are these solving manners that assume priority or weight coefficients for the particular criterions because these priorities may be modified during consecutive simulation experiments. Each new generated trajectory is analysed. If it is not admissible, the reasons of the failure are examined. For example, it is examined through which subsets of not advantageous states the trajectory has passed. A role of the criterions connected with this

subsets should be strengthened for the next trajectory i.e. the weights (priorities) of these criterions should increase. When the generated trajectory is admissible, the role of the criterions responsible for the trajectory quality can be strengthened, i.e. their weights can be increased. Basing on the gained information, the local optimisation task is being improved during simulation experiments. This process is treated as learning or intelligent searching algorithm. This conception has been examined and is presented in [2].

If one posses additional expert knowledge, then a better algorithm would be proposed. If some state subsets S_{di}, $i = 1, 2, \ldots$ are distinguished and for these states some rules for decision choice R_i, $i = 1, 2, \ldots$ are given by an expert then algorithm should verify additionally to which subset the new generated state belongs and should compute the suitable rule R_i. If rules given by expert excludes some decisions then the suitable sets of possible decision $U_p(s)$ would be decreased.

Another idea of learning algorithm for some scheduling problem is given in [8], [13].

4 Example

To illustrate the application of the presented method, let us consider the following scheduling problem that takes place when scheduling preparatory works in mines.

The set of headings in a mine must be driven in order to render the exploitation field accessible. The headings form a net formally, represented by a nonoriented multigraph $G = (I, J, P)$ where the set of branches J and the set of nodes I represent the set of headings and the set of heading crossings respectively, and relation $P \subset (I \times J \times I)$ determines connections between the headings (a partial order between the headings). There are two kinds of driving machines, that differ in efficiency, cost of driving and necessity of transport. The first kind machines (set $M1$) are more effective but a cost of driving by means of them is much higher than for the second kind (set $M2$). Additionally, the first kind machines must be transported when driving starts from another heading crossing than the one in which the machine is, while the second type machines need no transport. Driving a heading cannot be interrupted before its completion and can be done only by one machine a time. There are given due dates for some of the headings. They result from the formerly prepared plan of fields' exploitation. One must determine the order of headings' driving and the machine by means of which each heading should be driven so that the total cost of driving should be minimal and each of headings should be complete before its due date.

There are given: lengths of the headings, efficiency of both kinds of machines (driving length per time unit), cost of a length unit driven for both kinds of machines, cost of the time unit waiting for both kinds of machines, speed of machine transport and transport cost per a length unit.

The problem is NP-hard [2]. NP-hardness of the problem justifies the application of approximate (heuristic) algorithms. A role of a machine transport

corresponds to retooling during a manufacturing process, but the time needed for a transport of a machine depends on a process state while retooling does not.

The process state at any instant t is defined as a vector $x = (x^1, x^2, \ldots, x^n)$, $n = |J|$. A coordinate x^j describes the state of the j-th heading (branch), $x^j = (m, \Delta, i, s)$ where m denotes the number of the machine that is assigned to the j-th heading, Δ denotes the time after which the machine will be accessible, s is a parameter that defines whether the machine is driving the heading ($s = 1$), whether it is transported in the heading ($s = 2$) whether it is waiting in one of the heading ends (the nearest heading crossings) ($s = 3$), i denotes the number of the heading crossing (node) to which the machine is moving or in which it is waiting. If there is no machine assigned to the heading then $m = 0$ and $s = 0$. If the driving of a heading has been not started yet, then $\Delta = \infty$ and when it is complete, then $\Delta = 0$. The initial state $x^0 = (0, \infty, 0, 0)$. For any state (x, t) one can determine a set of headings that are being driven (J_1), the driving of which is complete (J_2), and not started yet (J_3). A state (x, t) belongs to the set of not admissible states if there is a heading whose driving is not complete yet and its due date is earlier than t. Formally, $S_N = \{(x, t) :$ there exists $j \notin J_2$ such that $d(j) < t\}$ where $d(j)$ denotes the due date for the j-th heading. A state (x, t) is a goal one if all the headings have been driven, i.e. $S_G = \{(x, t) : \ \forall j \in J, \ j \in J_2\}$.

A decision determines the headings that should be started at the moment t, machines which should drive, machines that should be transported, headings along which machines are to be transported and machines that should wait. Thus, the decision $u = (u^1, u^2, \ldots, u^n)$ where the co-ordinate u^j refers to the j-th heading and is of the form: $u^j = (m, q)$. The symbol m denotes the number of a machine that is assigned to the heading. The parameter $q \in \{0, 1, 2, 3\}$ and denotes respectively: waiting, driving, transport and withdrawing of the machine. When a machine $m \in M1$ is in the node i and should drive the k-th heading that is not adjacent to the i-th node, then the machine is transported in the nearest way accessible in the considered state. This way is computed by the Ford's algorithm (a polynomial one).

Obviously, not all the pairs (m, q) constitute possible decisions in the state (x, t). For example, a decision $u^j = (m_k, 1)$ is possible only when the j-th heading is both being neither driven nor complete and the machine m_k is in the one of the heading crossing adjacent to the j-th heading. The complete definition of the set of the possible decision $U_p(x, t)$ will be omitted here because it is not necessary to explain the idea of the algorithm. The detailed description of the formal model for the considered problem is given in [2].

The algorithm for the solution of the problem consists in generating consecutive trajectories. Each of them is generated with the use of the specially designed local optimisation task and then is analysed. The information gained as a result of the analysis is used in order to modify the local optimisation task for the next trajectory, i.e. for the next simulation experiment. This approach is treated as learning without a teacher.

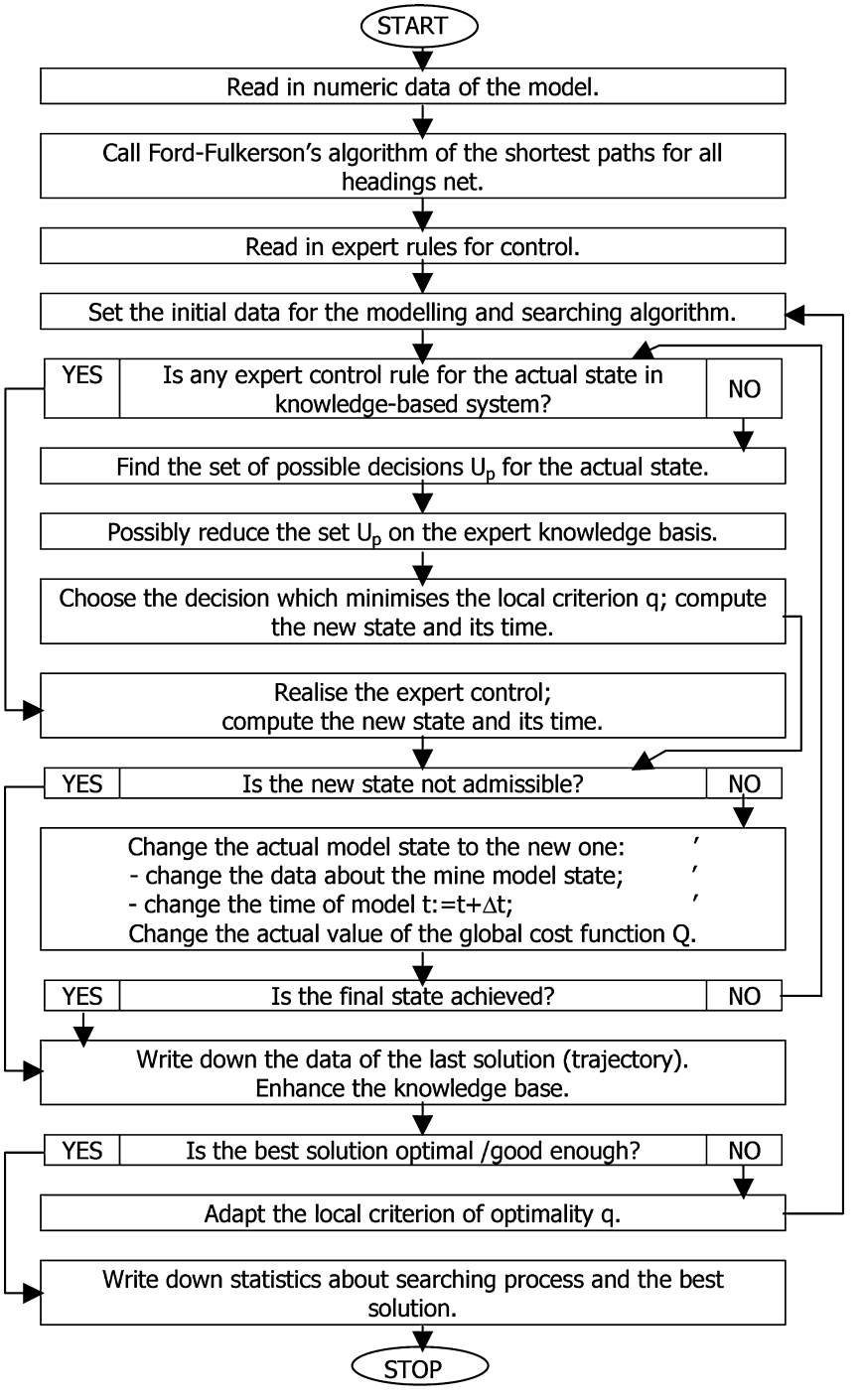


Fig. 1. Block-schema of the algorithm

The construction of the local optimisation criterion q for the presented example is based on two criteria. The first one, denoted as q_1 refers to the minimal increment of the global criterion value, computed from the current to the last state of a generated trajectory. The second one denoted as q_2 takes into account that the trajectory should pass possibly far from the not admissible states, or from the states from which there is a little chance to accomplish a goal state. Thus the local criterion $q = q_1 + bq_2$, where b is a parameter that is being changed during simulation experiments.

$$q_1 = \Delta Q(u, x, t) + Q'(x', t')$$

where ΔQ denotes the increment of the global criterion Q during a simulation step, i.e. when the current state (x, t) is changed, as a result of the decision u, to the state $(x', t') = f(u, x, t)$. $Q'(x', t')$ is a lower estimation of the global criterion value for the latter part of the trajectory, i.e. for the part that starts from the state (x', t'). This estimation is equal to the lowest cost of the driving of the remaining headings when the limitations referring to the due dates are neglected. The lowest cost is computed under assumption that only machines of the second type (set $M2$) are applied.

Criterion q_2 takes into account consequences of the decision u from the due date limitations point of view. $L(x', t')$ estimates a minimal relative distance between the new state (x', t') and the set of not admissible states S_N.

$$q_2 = \frac{1}{L(x', t')}$$

$L(x', t')$ is computed as follows. For the state x', the set of accessible headings' crossings is determined, i.e. the crossings from which driving can be started. There is also determined the set of headings J' whose driving has been not started yet and which have the due dates. For each of the headings, the shortest time needed for performing it is computed. It is denoted as $st(j)$ where j is the number of the heading. The time is needed for driving all the headings that constitute the shortest way from an accessible heading crossing and for performing the considered heading. It is assumed that the driving is performed by means of the first kind machines (i.e. the more effective ones) and their transport to the accessible crossing is neglected. Then, for the each of the headings the difference $d(j) - st(j) - t'$ is computed. If any of the differences has negative value, the generated trajectory cannot be admissible and is rejected. If all the differences are nonnegative, $L(x', t')$ is given by formula:

$$L(x', t') = \min_{j \in J'} \frac{d(j) - st(j) - t'}{d(j) - t'} = \min_{j \in J'} \left(1 - \frac{st(j)}{d(j) - t'}\right) \quad .$$

The formula is not applied for the headings that had been determined earlier on a basis of the expert knowledge or bottom up analysis. Finally the local criterion q consists of 3 components:

$$q = \Delta Q(u, x, t) + Q'(x', t') + b \ \frac{1}{L(x', t')}$$

The value of the criterion q is computed for each $u \in U_p(x,t)$. This decision u^* for which the criterion value is minimal is chosen. Then the next state $(x',t') = f(u^*,x,t)$ is generated and the new best decision $u \in U_p(x',t')$ is chosen. If a newly generated trajectory is admissible and for most of its states the distance to the set of not admissible states is relatively big, the parameter b can be decreased. In such a situation the role of the optimisation compound is enlarged. On the contrary, when the generated trajectory is not admissible, the parameter b should be increased because then the greater emphasis should be put to the due date limitations.

The presented conception is an essential extension of the one given in [2]. Computer experiments that have been carried out for the simpler algorithm are presented in [2]. They confirmed that learning-based approach is very efficient. Basing on those results one may be sure that the presented algorithm that use additionally an expert knowledge will be very efficient too.

5 Conclusions

The paper presents a conception of intelligent search algorithms (learning-based algorithms) for scheduling. A large number of difficult scheduling problems in manufacturing can be efficiently solved by means of these algorithms. A basis for the algorithms is a special kind of knowledge-based model of DMP that is given in the paper. A structure of the presented knowledge representation for DMP can be used also for another class of optimisation algorithms for DMP, e.g. for branch & bound ones [6], [14].

It should be pointed out that the presented KR structure is also useful for creating simulation packages for a large class of discrete processes because the special form of the model enables one to create the simulation package of a modular form. Such simulation package of a mixed structure, combining KR and multiagent system can be used for testing and developing strategies, prepared for crisis management.

To illustrate the conception, the learning-based algorithm for preparatory work in a mine is presented.

Acknowledgment

The research was supported in part by grant No 3 T11C 025 27 from Polish Committee of Scientific Research.

References

1. Doherty P.: *Knowledge Representation and Unmanned Aerial Vehicles*. Proc. of the IEEE/WIC/ACM Int. Conf. on Web Intelligence, IEEE (2005).
2. Dudek-Dyduch E.: *Learning based algorithm in scheduling*. Journal of Intelligent Manufacturing, Kluwer Academic Publishers vol.11, no 2 (2000), 135–143.

3. Dudek-Dyduch E.: *Formalization and Analysis of Problems of Discrete Manufacturing Processes*. Scientific bull. of AGH Academy of Science and Tech., Automatics vol.54, (in Polish), Cracow (1990).
4. Dudek-Dyduch E., Fuchs-Seliger S.: *Approximate algorithms for some tasks in management and economy*. System, Modelling, Control, no 7, vol.1, (1993).
5. Dudek-Dyduch E., Dyduch T.: *Scheduling some class of discrete processes*. Proc. of 12th IMACS World Congress, Paris (1988).
6. Dudek-Dyduch E.: *Control of discrete event processes - branch and bound method*. Proc. of IFAC/Ifors/Imacs Symposium Large Scale Systems: Theory and Applications, Chinese Association of Automation, vol.2, (1992), 573–578.
7. Dudek-Dyduch E., Dyduch T.: *Formal approach to optimization of discrete manufacturing processes*. in: Hamza M.H. (ed), Proc. of the Twelfth IASTED Int. Conference Modelling, Identification and Control, Acta Press Zurich (1993).
8. Kolish R., Drexel A.: *Adaptive Search for Solving Hard Project Scheduling Problems*. Naval Research Logistics, vol.42, (1995).
9. Liebowitz J.(ed.): *The Handbook of Applied Expert Systems*. CRC Press (1998).
10. McGuinness D.L., Patel-Schneider P.F.: *Usability Issues in Knowledge Representation Systems*. Proc. Of XV National Conf. on Artificial Intelligence, Madison, Wisconsin, (1998).
11. Pearl J.: *Heuristics : Intelligent search strategies for computer problem solving*. Addison-Wesley Comp. Menlo Park, (1988).
12. Rajedran C.: *A no wait flow shop scheduling heuristic to minimize makespan*. Journal of Operational Research Society 45, (1994), 472–478.
13. Sprecher A., Kolish R., Drexel A.: *Semiactive, active and not delay schedules for the resource constrained project scheduling problem*. European Journal of Operational Research 80, (1995), 94–102.
14. Tadeusiewicz R., Dudek-Dyduch E.: *Construction of Branch & Bound Decision Tree in Optimization of Discrete Manufacturing Processes*, in: Computer Science in Management, Poldex, Krakow, (1998), 169–180.
15. Vincke P.: *Multicriteria decision-aid*. John Wiley & Sons (1992).
16. Zhonghao W., Xinyu S,. Guojun Z,. Haiping Z.: *Integration of Variable Precision Rough Set and Fuzzy Clustering: An Application to Knowledge Acquisition for Manufacturing Process Planning*. Lecture Notes in Computer Science vol. 3642, Springer-Verlag (2005).

Knowledge Representation of Pedestrian Dynamics in Crowd: Formalism of Cellular Automata

Ewa Dudek–Dyduch and Jarosław Wąs

AGH University of Sciences and Technology, Institute of Automatics,
al. Mickiewicza 30, 30-059 Kraków, Poland
edd@ia.agh.edu.pl, jarek@agh.edu.pl

Abstract. The aim of the article is to suggest knowledge representation in modeling pedestrian traffic dynamics using the method of Cellular Automata. The article also proposes a modified formalization of Cellular Automata. The formalization enables the introduction of a new automata class with automata exceeding their classic formula, under which the condition of a cell depends only on local relations. The extended idea of the Cellular Automaton makes considering global relations in decision-making processes possible. The automata class presented here is used in modeling pedestrian dynamics in crowd.

1 Introduction

The article deals with modeling dynamic phenomena using Cellular Automata. The idea of a Cellular Automaton is used in many research areas. They are used, among others, in research on natural phenomena, such as flowing lava [11], forest fires or modeling water flows, e.g. in the case of an anastomosis river [13]. Other uses include modeling flora and fauna populations, e.g. growing of foraminiferane [12], simulation of fish shoal movements or simulation of birds flock movement [10]. Since the beginning of the 90-ties, Cellular Automata have been used in modeling people's behavior. At the beginning, the research covered modeling car traffic, then, the modeling of pedestrian traffic dynamics became more and more common [1]–[9], [14]–[16].

The Cellular Automaton is characterized by the fact that global processes can be modeled with the use of adequately defined local relation. In order to make a model using the CA, it is necessary to define a series of ideas such as a set of cells with its topology, the idea of a cell neighborhood necessary to define a local transformation as well as the set of cell states and rules which define local transformation.

The paper aim is three–fold. The first goal is to present a formal description of CA in such a way that the formalization would cover all the applications of CA done so far in different phenomena modeling. Simultaneously it would be the base for creating an extended version of Cellular Automata which are suitable for new applications. The second goal is to propose a concept of the so-called

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1101–1110, 2006.
© Springer-Verlag Berlin Heidelberg 2006

extended Cellular Automata. The third goal is the presentation of using the extended Cellular Automaton in modeling intelligent behavior of people in the crowd.

The authors propose a formal description of Cellular Automata, allowing to broaden the idea of Cellular Automata for many specific applications. The idea of a new definition of the Cellular Automaton is given in the paper. In this definition, the state of a cell depends not only on its direct surrounding. The extended definition of a Cellular Automaton is used, for instance, in the problems of pedestrian traffic dynamics modeling.

2 Formal Models of Cellular Automata

Two definitions of Cellular Automaton can be found in publications. The first of them, known as Ferber's definition says [7]:

Ferber's definition. *Cellular automata are discrete dynamical systems, whose behavior is completely specified in terms of a local relation.*

Another popular definition of the Cellular Automaton according to Weimar, similarly describes only local relations [17]:

Weimar's definition. Let us take into consideration four elements: (L, S, N, f), where: L - Set of cells of the lattice, S - Set of states, N - Set of neighbors, f - transition function. Additionally, a configuration $C_t: L \rightarrow S$ is defined as a function, which associates each state with a grid cell. An equation of change of a configuration is shown by the equation (1) with the supplement (2).

$$C_{t+1}(r) = f\left(\{C_t(i) | i \in N(r)\}\right) \tag{1}$$

where:

$N(r)$ - set of neighbors of cells r,
r - current cell number,
t - discrete time step $t = t + 1$,
i - single cell.

$$N(r) = \{i \in L | r - i \in N\} \tag{2}$$

Table 1. Weimar's definition of Cellular Automata

Definition	Description
L	– consist of discrete lattice of cells
$t \rightarrow t + 1$	– evolution takes place in discrete time steps
S	– set of "finite" states
$f : S^n \rightarrow S$	– each cell evolves according to the same rule
$N : \forall c \in N, \forall r \in L : r + c \in L$	– the neighborhood relation is local and uniform

Most models based on Cellular Automata, especially those ones which concern pedestrian traffic dynamics, exceed the CA definitions presented above. Often the models' authors point to the lack of adequate formalization. The formalism presented here assumes the following:

- model uniformity based on an identical definition of neighborhood for all cells
- grid regularity
- identical for each cell local transformation which assigns state of next cell, depending on the neighborhood state.

This uniformity makes modeling more complicated processes impossible. A good example of exceeding the formalism mentioned above can be uniting Cellular Automata with the Multi-Agent System used in pedestrian traffic simulation in a shopping center [4], or the use of static potential fields in the evacuation of passenger liners [6]. In papers [14], [15], [16], which are devoted to the strategic abilities of pedestrians, both local and global dependence of individual automaton cells are considered. The latter cover, for example, the evacuation process of a group of people – the problem of choosing the exit form a set of accessible ones, where the parameters influencing the decision are: distance from the exit, densification in exit neighborhood (neighborhood having a specific radius) and dynamics of densification changes in exit neighborhood [15].

Therefore, there is a need to define a new class of automata which enables the consideration of global relations in the whole grid. The starting point is made up of classical Cellular Automata. Hence, it is necessary to define them in such a way, as to make the definition of a new class of Cellular Automata possible on their base.

2.1 Suggested Formal Model

Here we are going to specify a formal definition of a Cellular Automaton which will be called by the authors *a Cellular Automaton with a constant grid*. The authors suggest a formal model of a Cellular Automaton slightly different from the most commonly used definitions.

The authors suggest such a formal form that will cover a broader class of automata than the automaton defined above. Additionally, this form enables, by making some parts of the definition more precise, to differentiate subclasses of Cellular Automata. Since the introduced definition is slightly different from others published so far, a name of a Cellular Automaton *with constant grid* has been introduced to make it differ. Also, the name should stress the fact that this automaton will be used in modeling the systems where the set of cells and their mutual position will not be changed. Such grid is used when spatial structures with the topology not changing in time are modeled (roads, compartments etc.).

In order to define automata with a static grid, ideas and denotations introduced for the definition needs will be presented:
The set of cells will be marked by C. In the C set we will define the adjacent relation $R_{\mathrm{p}} \subset C \times C$ which has a symmetry property and is an anti-reflexive

relation, namely $c_i R_{\mathrm{p}} c_j \Leftrightarrow c_j R_{\mathrm{p}} c_i$ and $\forall(ci \in C)\colon (ci,\, ci) \in R_{\mathrm{p}}$. This relation will be used to define different grid formats (different topologies in the set of cells). Especially, it allows to construct irregular grids. By L grid, the set of cells C with its adjacent relation R_{p} will be understood, therefore $L = (C,\, R_{\mathrm{p}})$.

Each cell has a defined neighborhood. In the model it is assumed that the definition of neighborhood need not be identical for all the cells of the C set. It will give a possibility of different defining the neighborhood for some discriminated cell subsets (e.g. for the cells which will be defined as boundary ones in a model). The connectivity of a neighborhood set will also be assumed. To define this connectivity the adjacent relation and the so-called indirect adjacent relation R_{pp} defined on its base will be used. Let us say that the cell c_j is indirectly adjacent to c_i only, when there is a sequence of cells from c_j to c_i which are adjacent cells. It can be formally recorded as equation 3:

$$\begin{aligned} c_j R_{pp} c_i \Leftrightarrow \exists(c_{j1}, c_{j2}, \ldots, c_{jn}) \quad \text{such that} \quad c_{j1} = c_k, c_{jn} = c_i \\ \text{and} \quad c_{jk} R_p c_{jk+1} \quad \text{for} \quad j = 1, 2, \ldots, n-1 \end{aligned} \tag{3}$$

By cell c_i neighborhood we mean a discriminated set of cells in which each cell c_k is adjacent or indirectly adjacent to the cell c_i i.e. $c_k R_{pp} c_i$. Naturally, $c_k R_p c_i \Rightarrow c_k R_{pp} c_i$ and it is easy to show that the indirect adjacent relation is symmetrical, anti-reflexive and transitive.

Neighborhood will be defined by the use of a function $\eta\colon C \to 2^C$, which assigns a subset of cells being its neighborhood $\eta(c)$ to each cell $c \in C$. (2^C denotes a set of all subsets of the set C). The function η will be called a neighborhood function. Let us notice that the introduction of the neighborhood function enables independent defining neighborhoods having an irregular shape. Of course, it does not exclude a situation when the neighborhood is defined identically for all the cells and has a regular shape.

Let us denote a set of cell states as S. The function $con\colon C \to S$ which assigns to each cell, its state $con(c) = s \in S$ c will be called a configuration. Since the configuration denotes the states of all the cells, it will be identified with the state of the whole grid. The set of configurations will be denoted as CON. This set will correspond to the set of automaton states. We will also use the notation $s(c)$ to determine the state of the cell [5].

Configuration changes will be defined with the transition function $f\colon CON \to CON$ which assigns the next configuration to each configuration $f(con_t) = con_{t+1}$.

The function f is the function of a global change of states. As we we know, the essence of the cellular automaton lies in the fact that the determination of a next configuration is done with the locally defined transformation which is calculated successively for each cell belonging to the grid or for cells from some subsets (if the state of the remaining cells does not change). In order to define the transition function, local rules which allow modeling behavior of the whole system will be introduced.

Let us denote the restriction of function con to the set $\eta(c_i)$ as $con/\eta(c_i)$, i.e. $con/\eta(c_i)$ is determined only for arguments from the set $\eta(c_i)$ thus determining states of all the cells belonging to the neighborhood of the cell c_i. Let

r denotes a local transformation that on a base of the $i-th$cell′s state at the moment t – $s_t(c_i)$ and its neighborhood configuration $con_t/\eta(c_i)$ determines the state of this cell in the next moment, i.e.: $r(s_t(c_i), con_t/\eta(c_i)) = s_{t+1}(c_i)$. This local transformation will be called a local rule.

Now, the transition function f can be defined by the repeated calculation of the function of the local rule r for consecutive cells as shown in (4).

$$f(con_t) = con_{t+1}, \text{ such that } \forall c_i \in C: con_{t+1}(c_i) = s_{t+1}(c_i) = r(s_t(c_i),\ con_t/\eta(c_i)) \tag{4}$$

A Cellular Automaton with a constant grid denoted as $\mathrm{CAL}_{\mathrm{const}}$, will be defined as the seventuple (5).

$$\mathrm{CAL}_{\mathrm{const}} = (C,\ R_{\mathrm{p}},\ \eta,\ S,\ CON,\ r,\ f) \tag{5}$$

Using generally accepted terminology in the automata theory we can say that the Cellular Automaton $\mathrm{CAL}_{\mathrm{const}}$ is a particular kind of automaton without input $\mathrm{CAL}_{\mathrm{const}} = (X,\ f)$, for which the set of states is equivalent to the configuration set $X \equiv CON$ where the configuration *con* is defined as a function $con: C \rightarrow S$, and f is a transition function $f: CON \rightarrow CON$ determined by means of a notion of the local rule and the idea of the neighborhood.

3 Knowledge Representation in Pedestrian Traffic Dynamics Modeling

It should be stressed that the Cellular Automaton can also be a model of knowledge representation about dynamic processes.

3.1 Extended CA Concept in Pedestrian Traffic Dynamics Modeling

We are now going to present an extended model of a Cellular Automaton $\mathrm{CAL}_{\mathrm{const}}$, which is a schema for knowledge representation about certain behavior of relocating people.

This representation has been suggested to model *intelligent* behavior of individuals in a relocating crowd. In order to illustrate the formalism introduced, a model of intelligent behavior of people, when leaving a room with several exits by a large group of people, will be presented.

An example of rules which prove intelligent behavior of individual people in this case is:

- estimating the distance from the exits,
- analyzing the crowd at each exit.

On the base of such analysis, an individual decides to relocate towards this exit which is best for them. Thus, in the model, the local rule r must regard the information individual people make their decisions on and represent their intelligence.

Let us consider two properties of the introduced the definition of Cellular Automata which was presented in the earlier chapter. Firstly, the local rule assignees the next state of each cell depending on its present state and the state of the neighborhood. Moreover, it changes only the state of this cell. Secondly, the transition function is determined by the local rule which is calculated for successive cells and the calculation order does not matter. For the needs of pedestrian traffic modeling such a model is not sufficient.

Let us introduce the following extension in the model. The local rule will be defined as a function changing not only the state of one cell but also selected cells from its surrounding. The local rule will be defined as a function assigning the next state of the cell not only on the base of its state and the states of the neighboring cells but also on the base of the states of other discriminated cells.

The local rule, denoted now as r_{ext} can be shown as follows: $r_{\text{ext}}: CON \rightarrow CON$ such that: $r_{\text{ext}}(con_t/\eta^*(c), con_t/A)) = con_{t+1}/\eta^*(c)$, where A is an extra subset of cells, the state of which influences a future state of the cells from the surrounding, whereas the symbol $\eta^*(c)$ denotes the so-called full surrounding, i.e. $\eta^*(c) = \eta(c) \cup \{c_i\}$.

Let us slightly change the definition of the transition function by assuming that its calculation algorithm needs to fix the sequence of cell review. Such a model of the Cellular Automaton will be called *Extended Cellular Automaton* and will be denoted as $\text{ECAL}_{\text{const}}$.

A simulation model of pedestrian traffic dynamics is a good example of the use of the above formalization. The presented situation is leaving a large cinema room by a group of pedestrians [15].

Let us present the room topology (Fig. 1). The room is represented by a square grid $L = (C, R_{\text{p}})$ so each cell c is indexed with the use of two indexes $c_{i,j}$ where $1 \leq i \leq i_{\max}$, $1 \leq j \leq j_{\max}$, and the values $i_{\max}$ and $j_{\max}$ result from the room and cell sizes. The adjacent relation R_{p} coincides with the natural adjacent relation in square grids. For all the cell, Moore neighborhood was assumed as $\eta(c_i)$.

The following cell subsets are discriminated in the set C:

- exit set denoted as E. Exits can be defined differently. In the discussed model, the exit corresponds to several cells and is denoted as e. Each exit consists of a subset of adjacent cells. Thus, e.g. $e_1 = \{c_{5,6}, c_{6,6}, c_{7,6}\}$.
- the set of cells corresponding to four walls $W = W_1 \cup W_2 \cup W_3 \cup W_4$ where W_i represents the cells of the $i - th$wall. For example, the first wall W_1 is defined as $W_1 = \{c_{ij} \subset C: j = 1 \text{ and } 1 \leq i \leq i_{\max}\} \backslash e_1$.
- the set of cells representing the room interior: this set consists of a subset of cells representing chairs which are obstacles, denoted as O and a subset of cells being a room for pedestrian traffic denoted as MS.

The set of cell states $S = \{0, 1\}$ where $s(c) = 0$ denotes an empty cell and $s(c) = 1$ denotes a cell occupied by one person [6]. For all the cell from the set O $s(c) = 1$ has been assumed. The size of the cell was calculated on the base of a field which is averagely taken by a person in a dense crowd, namely 40cm by 40cm.

In the room a population of people is generated at random. Pedestrian traffic is registered at discrete time intervals. These people leave the room in discrete time intervals, through accessible exits.

The pedestrians make a decision about the direction of the movement on the base of global space information, namely about their direct surrounding (neighborhood) as well as the whole room, especially about the density of people near the exits and their distance from them.

People are represented in the model by the cell occupancy $s(c) = 1$, for $c \in MS$ (MS - *Movement Space*). Their movement depends directly on the state of the neighborhood (cell occupancy).

Assuming homogeneity of the set of people, the movement is represented by an identical local rule r. This rule depends on both the state of neighboring cells $con(\eta(c))$ and the state of the cells near the exits.

The subset of cells making up the surroundings of each exit which are distinguished in the model corresponds to set A in the rule definition. A new state of c_i belonging to its full surrounding is computed on the base of the rule r_{ext}. The rule for the individual in $c_i \in MS$, denoted as $r(c_i)$ is computed as follows.

1. The cost function $cost(c_i, e_j)$ for each of the exits $e_j \in E$:

$$cost(c_i, e_j) = w_1 dist(c_i, e_j) + w_2 dens(e_j) \tag{6}$$

where:
w_1, w_2 – fixed determined coefficients
$dist(c_i, e_j)$ – the smallest number of cells which make up the way from c_i cell to any exit cell e_j (i.e. the length of such a shortest way between c_i, c_k that $c_k \in e_j$, where the way $w(c_i, c_k)$ from c_i to c_k is a sequence of adjacent cells linking c_i with c_k).
$dens(e_j)$ – the function characterising crowd density in surrounding $A(e_j)$ around an exit e_j, expressed by (7).

$$dens(e_j) = \frac{\sum_{c_i \in A(e_j)} s(c_i)}{|A(e_j)|} \tag{7}$$

where:
$|A(e_j)|$ – number of cells belonging to the surrounding e_j

2. This exit e^* is selected for which $cost(c_i, e^*)$ has the lowest value.
3. The state of the cell surrounding c_i, $con_t/\eta(c_i)$ is analyzed and an empty cell c_j adjacent to c_i and belonging to the way $w(c_i, e^*)$ is selected. If there is no such an empty cell, then an empty cell which is nearest the way $w(c_i, e^*)$ towards the exit e^* is selected.
4. The state of the cell c_j is modified : $s_{t+1}(c_j) = 1$.
5. The state of the cell c_i is modified : $s(c_i) = 0$.

The transition function modifies cell states iteratively in the determined sequence using the rule r_{ext}, and:

– if the cell state changes from 0 to 1,then such a state cannot change during the time-step-slice (despite later cell reviewing in the time-step-slice).
– if the cell state c_i changes from 1 to 0, then it can change again to 1 within the same time-step-slice.

The modification direction of cell state is defined in the model starting from the exits inside the room using the "wave propagation" method [15].

3.2 Model Implementation

Figure 1 shows a topology of the presented model. The topology is described in previous sections.

A computer application based on the described model has been made. On Fig. 2 there is shown a room being left by a group of people. People, represented

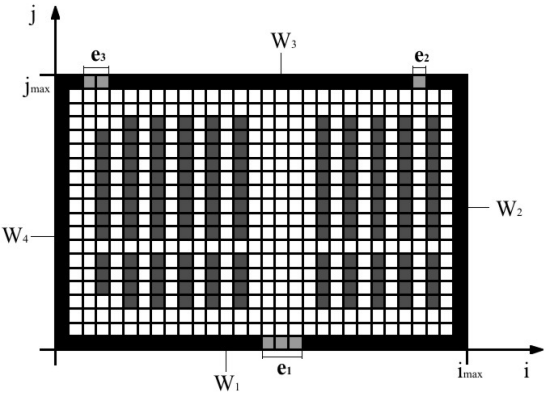


Fig. 1. Topology of the room described in the model

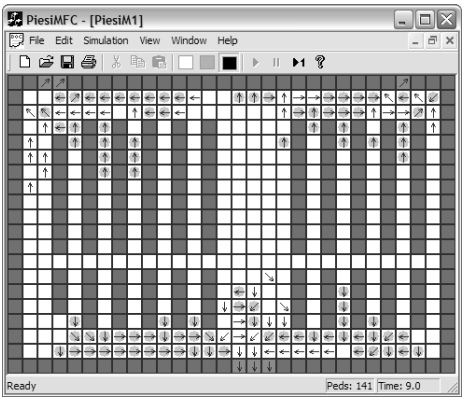


Fig. 2. An application based on the model. The room is a cinema hall being left by a group of pedestrians, which are represented by black arrows.

by arrows are leaving the cinema hall, through three accessible exits. Black cells inside the room represent chairs. The situation, when the pedestrian cannot move in the preferred direction is described by an arrow in the red field. A detailed implementation is described in [15].

4 Conclusions

The article is connected with knowledge representation about dynamic spatial processes. The knowledge representation uses the idea of a Cellular Automaton. The first aim of the article was to suggest a new formal definition of the Cellular Automaton. This definition was to be the base for making different dynamic models of space problems. The suggested formalization encompass a wider range of automata than definitions discussed so far. Moreover, the presented formalization makes it possible to classify models more rigorously, to compare them and also, enables to differentiate subclasses of cellular automata by making some parts of definitions more detailed.

Since the presented definition is slightly different than others met so far, the name Cellular Automaton with a *constant grid* has been introduced to make it differ. The name should additionally stress the fact that the automaton will be used in modeling systems with constant space typology.

The next aim of the paper was to present an extended idea and formal definition of a Cellular Automaton $\mathrm{ECAL}_{\mathrm{const}}$ making modeling of a broader class of dynamic processes possible. This idea enables a rigorous presentation of a model of a broad group of problems connected with pedestrian traffic. The presented modified formalization of cellular automata enables defining both the basic idea of a Cellular Automaton, in which the state of the cell and its change depend only on local relations, and an extension of an idea of a Cellular Automaton by global properties.

The presented formalism was illustrated with a model for pedestrian traffic simulation and testing potential strategies. The idea of an extended Cellular Automaton has been used to present the model of intelligent behavior of individuals in a crowd.

References

[1] Fukui M., Ishibashi Y.: Self-organized Phase Transitions in CA-models for Pedestrians. J. Phys. Soc. Japan (1999) 2861–2863

[2] Blue V., Adler J.: Bi-directional Emergent Fundamental Flows from Cellular Automata Microsimulation. Proceedings of ISTTT, Amsterdam (1999) 235–254

[3] Burstedde C.K., Klauck K., Schadschneider A., Zittartz J.: Simulation of Pedestrian Dynamics using a 2-dimensional Cellular Automaton. Phys. Rev. A 295 (2001) 507–525

[4] Dijkstra J., Jessurun A.J., Timmermans H.: A Multi-agent Cellular Automata Model of Pedestrian Movement. Pedestrian and evacuation dynamics, Springer-Verlag Berlin (2000) 173–181

[5] Dudek-Dyduch E., Wąs J.: Formalisation of Cellular Automata in Issues of Pedestrians Dynamic Simulation. Automatyka Journal AGH-UST Vol. 9 no. 3 Kraków (2005) 383–391 (In Polish)
[6] Klüpfel H., Meyer-Köning T., Wahle J., Schreckenberg M.: Microscopic Simulation of Evacuation Processes on Passenger Ships. Proceedings of ACRI (2000) 63–71
[7] Ferber J.: Multi-agent systems: An Introduction to Distributed Artificial Inteligence. Addison-Wesley (1999)
[8] Gloor C., Stucki P., Nagel K.: Hybrid Techniques for Pedestrian Simulations. Proceedings of 6th International Conference on Cellular Automata for Research and Industry Amsterdam (2004) 581–590
[9] Narimatsu K., Shiraishi T., Morishita S.: Acquisiting of Local Neighbour Rules in the Simulation of Pedestrian Flow by Cellular Automata. Proceedings of 6th International Conference on Cellular Automata for Research and Industry, Amsterdam (2004) 211–219
[10] Reynolds C. W.: Flocks, Herds, and Schools: A Distributed Behavioral Model. Computer Graphics 21(4) SIGGRAPH '87 Conference Proceedings (1987) 25–34
[11] Spataro W., D'Ambrosio D., Rongo R., Trunfio G.A.: An Evolutionary Approach for Modelling Lava Flows through Cellular Automata. LNCS Vol. 3305 Proceedings ACRI 2004 University of Amsterdam Science Park Amsterdam The Netherlands (2004) 725–734
[12] Tyszka J., Topa P., Łabaj P., Alda W.: Theoretical Morphology of Foraminiferal Shells. Proposal of a New Modeling Method. Geological Review Vol. 52, no. 1 (2004) 80–83 (In Polish)
[13] Topa P., Paszkowski M.: Anastomosing Transportation Networks. LNCS Vol. 2328 Proceedings of PPAM 2001 Nałęczów Berlin (2001), 904–911
[14] Wąs J., Gudowski B.: The Application of Cellular Automata for Pedestrian Dynamic Simulation. Automatyka Journal AGH-UST Kraków (2004) 303–313
[15] Wąs J., Gudowski B.: Simulation of Strategical Abilities in Pedestrian Movement using Cellular Automata. Proceedings of 24th IASTED MIC Conference Innsbruck (2005) 549–553
[16] Wąs J.: Cellular Automata Model of Pedestrian Dynamics for Normal and Evacuation Conditions. Proceedings of Intelligent Systems Design and Applications Wrocław IEEE CS Washington Brussels Tokyo (2005) 154–159
[17] Weimar J.R.: Simulation with Cellular Automata. Logos–Verlag (1998)

Algorithm for Generating Fuzzy Rules for WWW Document Classification

Piotr Dziwiński[1] and Danuta Rutkowska[1,2]

[1] Department of Computer Engineering
Faculty of Mechanical Engineering and Computer Science
Czestochowa University of Technology, Poland
dziwinski@kik.pcz.czest.pl
[2] Department of Knowledge Engineering and Computer Intelligence
Academy of Humanities and Economics in Lodz, Poland
drutko@kik.pcz.czest.pl

Abstract. This paper presents the dense areas based algorithm for generating fuzzy rules for classification WWW documents. Description document clusters in the form of fuzzy rules (FR) make possible the presentation of information in the form fuzzy granules. Moreover, each cluster might be described by several fuzzy rules. These fuzzy rules can be used as the knowledge base for searching new information from WWW resources with regard to specific topics and users' requirements.

1 Introduction

An enormous growth of the Internet and significant development of telecommunication techniques enable users to access WWW resources. About 11.5 billion WWW pages have been created until January 2005 [9]. Changeability and large amount of WWW pages is a big challenge for modern crawlers and search engines. They should reflect WWW resources as accurately as possible and also hold information about WWW resources as freshly as possible. Complete crawling entire Web is impossible in reasonable time, no matter which technology is available at the site where the search engines operate. The ideal crawler should be able to recognize relevance and importance of Web pages. The crawlers can order new links extracted from downloaded WWW pages. This can be accomplished by using different methods (e.g. measurement similarity between pages and a current query, amount of the links to point out WWW pages or the most popular Page Rank). Precise description of these methods as well others may be found in [6]. To deal with enormous WWW resources these methods are not enough efficient. One of the possible solutions of this problem can be the use of methods that are based on focused crawling. This approach makes possible to avoid areas of WWW resources that are not relevant to the information requirements of the user. Hersovici et al., in paper [11], proposed the shark-search algorithm of WWW resources. This approach searches WWW resources based on the assumption that relevant WWW pages usually are in relevant neighbourhood. Chakrabarti et al., in paper [4], proposed the focused crawling approach in

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1111–1119, 2006.
© Springer-Verlag Berlin Heidelberg 2006

which a crawler selectively discovers pages that are relevant to a pre-defined set of topics. The topics are specified by using exemplary documents. Rungsawang et al. [17] proposed the consecutive crawling to take advantage of experience from earlier crawling process. They build some knowledge base, which are used to produce better result for the next crawling. The classification process plays an important role in the scope of information retrieval. In order to classify WWW pages, different methods are used, such as: k-nearest neighbors algorithm (k-NN), Naive Bayes, support vector machines (SVM), decision trees, neural networks or induction of classification rules. Accurate description of these and other methods may be found in [15], [7], [13], [21]. A classifier can be used to distinguish between relevant and irrelevant WWW pages or resources, to help in a semi-automatic construction of large knowledge bases or to classify unknown Web pages to some predefined categories.

This paper presents an algorithm for fuzzy rules determination, based on dense areas, for WWW documents classification. This paper is organized as follows. Section 1 presents an introduction concerning information retrieval. The next section shortly describes a vector space model for text classification, preprocessing of WWW pages (parsing, stemming, tokenization of WWW pages), and most important methods to calculate importance of tokens of WWW pages and used normalization. A fuzzy inference system, simple fuzzy rules and membership functions are presented in Section 3. An algorithm for determination of the fuzzy rules based on dense areas is presented in Section 4. Experiments and their results are described in Section 5. Final remarks and directions of further works are outlined in Section 6.

2 Preprocessing of WWW Pages

Preprocessing of WWW pages is performed at the first phase. First of all, the WWW pages are parsed. The WWW pages are reduced to an unstructured representation. This can be achieved by retaining the text enclosed between *<html>* and *</html>* tags. The html tags, part of the text between script tags, some characters (e.g. %, #, $), numbers and other elements can be removed. The conversion of strings that encode international characters to standard representation should be done, if necessary. A well-designed parser ought to have a simple rule base (including e.g. html tags, script tags), which can be easily modified.

Then, stemming is applied to WWW pages. The stemming process reduces words to their root form, which become an actual index of terms. One of first method of this kind is Porter stemming algorithm [16] introduced in 1978. A new version of this algorithm may be found in [12].

Next, the stop-words are removed from WWW pages. The next phase is the process of tokenization. In this phase, a list of keywords (terms), in the form of pairs *(term, number of appearance in document)*, is created. If we have a keyword dictionary for some languages, we perform reduction of terms. As a result of the initial process with regard to WWW pages, we obtain a vector space model of terms in the form

$$\boldsymbol{x}_i = [x_{i0}, x_{i1}, \ldots, x_{iK-1}]^T, \quad i = 0, 1, \ldots, N-1$$

where: N – number of WWW pages, K – number of keywords, x_{ij} – number of appearance of keyword j in document i, for $j = 0, 1, \ldots, K-1$.

The vectors of keywords that represent WWW pages reflect a degree of association between keywords and WWW pages in the form of term weights. Most of methods determine these weights based on statistics. One of the simplest methods calculates the term frequency (TF) of WWW pages as follows [3]

$$TF_{ij} = \frac{x_{ij}}{k_i} \tag{1}$$

where k_i – number of keywords in document i.

This method is a very poor way to determine the degree of association between keywords and documents. The term weights can be inversed according to the number of occurrences in different documents. In order to achieve this relationship, we calculate the inverse document frequency by the following formula [3]

$$IDF_j = \log \frac{N}{n_j} \tag{2}$$

where n_j – number of documents in the collection of N documents in which keyword j occurs.

The most popular method determining term weights is the product of TF and IDF, in the form [3]

$$TF_{ij} \cdot IDF_j = \frac{x_{ij}}{k_i} \cdot \log\left(\frac{N}{n_j}\right). \tag{3}$$

WWW documents have different sizes. For a short document, possibility of use a keyword is smaller that for large documents. For large documents, TF achieves large values. In order to neutralize this unfavorable effect, we can use one of normalization methods. The most popular method is the cosine normalization, which gives the following equations, when applied to (1) and (3), respectively

$$Norm \quad TF_{ij} = \frac{\frac{x_{ij}}{k_i}}{\sqrt{\sum_{m=0}^{K-1}\left(\frac{x_{im}}{k_i}\right)^2}} \tag{4}$$

$$Norm \quad TF_{ij} \cdot IDF_j = \frac{\frac{x_{ij}}{k_i} \cdot \log\left(\frac{N}{n_j}\right)}{\sqrt{\sum_{m=0}^{K-1}\left(\frac{x_{im}}{k_i} \cdot \log\left(\frac{N}{n_m}\right)\right)^2}}. \tag{5}$$

The number of keywords to represent WWW pages can be restricted by selecting those keywords for which the number of occurrences in different documents is bigger than a certain threshold, P, which depends on the number of WWW pages. Some of the keywords, for which the frequency of occurrence is too high, can be removed.

3 Fuzzy Rules and Fuzzy Inference System

The key point of the issue of focused crawling is searching new information from WWW resources related to some specific topic or a set of topics. The information requirements by users, and topic classification of the WWW documents, are uncertain. We cannot describe these needs by use of the classical set theory and Boolean logic. In order to catch this non-crisp information, we should apply the fuzzy logic [23], [24]. In paper [14], Kraft et al., presents different approaches to fuzzy rules construction. In this paper, we are focusing exclusively on fuzzy description of classes of WWW documents that belong to specific topics, in the form of fuzzy rules. These rules can be a part of a fuzzy inference system (FIS). Precise description of FIS can be found e.g. in [20], [18], [19]. Standard FIS consists of the following elements: rule base, fuzzyfication unit, inference unit, defuzzyfication unit. The rules, in the rule base, are of the following form

$$R^{(l)}: \textbf{ IF } x_0 \text{ is } A_0^l \textbf{ AND } x_1 \text{ is } A_1^l \textbf{ AND } \ldots \textbf{ AND } x_{K-1} \text{ is } A_{K-1}^l \quad (6)$$

$$\textbf{THEN } y \text{ is } B^l$$

where: $l = 0, 1, \ldots, L-1$, so L – number of fuzzy rules; $\boldsymbol{x} = [x_0, x_1, \ldots, x_{K-1}]^T$ – linguistic variables that corresponds to inputs of FIS; A_i^l – fuzzy set for input linguistic variable, for $i = 0, 1, \ldots, K-1$, and B^l – fuzzy set that corresponds to the output linguistic variable y.

For classification tasks, we apply the simpler form of the rules, as follows

$$R^{(l)}: \textbf{ IF } x_0 \text{ is } A_0^l \textbf{ AND } x_1 \text{ is } A_1^l \textbf{ AND } \ldots \textbf{ AND } x_{K-1} \text{ is } A_{K-1}^l \quad (7)$$

$$\textbf{THEN } \quad y \in \text{class } \quad \mathbf{c}$$

where c is the number associated with the class corresponding to this rule, $c = 0, 1, \ldots, C-1$, and C – number of classes; see Section 4.

The inference based on the rule of this type determines the rule activation degree. Fuzzy set A_i^l is defined by Gaussian membership function [20]

$$\mu_A(x_i) = \exp\left(\frac{-(x_i - v_i^l)^2}{(\sigma_i^l)^2}\right) \quad (8)$$

or asymmetrical Gaussian membership function

$$\mu_{A_i^l}(x_i) = \begin{cases} exp\left(\frac{-(x_i - v_i^l)^2}{(\sigma_{l_i}^l)^2}\right) & \text{if } x_i < v_i^l \\ exp\left(\frac{-(x_i - v_i^l)^2}{(\sigma_{r_i}^l)^2}\right) & \text{if } x_i > v_i^l \\ 1 & \text{if } x_i = v_i^l \end{cases} \quad (9)$$

where: v_i^l – center, σ^l – width, σ_r^l, σ_l^l – right and left parts of width of the membership functions.

To determine fuzzy rules (8) and (9), it is necessary to calculate the parameters of fuzzy sets A_i^l .

4 Algorithm for Fuzzy Rules Generation

In [22] Tao et al. provide unsupervised fuzzy clustering algorithm to cluster pixels in a color image. This algorithm is based on the subtractive clustering algorithm, proposed in [5], which uses the density function, similar to that expressed by Eq. (10). The "density" is understood as *number of points in the neighborhood*. We modify this algorithm in order to determine fuzzy rules for classification WWW documents. In this way, we obtain the following algorithm.

Let us denote:
$\boldsymbol{x}^c = [\boldsymbol{x}_0^c, \boldsymbol{x}_1^c, \ldots, \boldsymbol{x}_{N^c-1}^c]^T$, where N^c – number of vectors which describe WWW pages that belong to cluster c, for $c = 0, 1, \ldots, C-1$; C – number of classes, $\boldsymbol{x}_i^c = [x_{i0}^c, x_{i1}^c, ..., x_{iK-1}^c]^T$, where K – number of keywords (vector length); this vector includes term weights, $\boldsymbol{d}^c = [d_0^c, d_1^c, ..., d_{N^c-1}^c]^T$ – density vector that corresponds to individual vectors $\boldsymbol{x}_i^c$ which includes information about the number of other vectors in radius r_k^c, $\boldsymbol{D}^l = [D^{c0}, D^{c1}, \ldots, D^{cL^c-1}]^T$ – density vector that corresponds to membership functions $\mu_{A^{cl}}$, where $l = 0, 1, \ldots, L^c - 1$, and L^c – number of generated rules for class c.

The basic function that describes the density of vectors, also called the density function, is expressed as follows

$$d_i^c = \sum_{j=0}^{N^c-1} \prod_{k=0}^{K-1} exp\left(-\frac{(x_{jk}^c - x_{ik}^c)^2}{(r_k^c)^2}\right) - \sum_{l=0}^{L^c-1} D^{cl} \cdot \prod_{k=0}^{K-1} \mu_{A_k^{cl}}(x_{ik}^c) \qquad (10)$$

where membership function $\mu_{A_k^{cl}}$ is defined by (8) and (9).

The algorithm that determines fuzzy rules can be presented in the following steps:

1. Let $L = 0$;
2. Determine the parameter value, r_k^c, depending on the domain of the input values
$$r_k^c = \frac{\max_{j=0,1,\ldots,N^c-1}(x_{jk}^c) - \min_{j=0,1,\ldots,N^c-1}(x_{jk}^c)}{R} \qquad (11)$$
where R – multiple factor domain of input values (this factor influences on the number of fuzzy rules and accuracy of classification).
3. Calculate the density function for each vector x_i^c, by use of Eq. (10).
4. Determine m such that
$$d_m^c = \max_{i=0,1,\ldots,N^c-1}(d_i^c)$$
5. If the density value $d_m^c > 0$, then go to step (6), else stop the algorithm.
6. Set initial values of parameters of membership functions $\mu_{A_k'^c}$
$$v_k'^c = x_{mk}^c, \quad \sigma_k'^c = r_k^c,$$

7. Refresh center and width parameters of membership functions $\mu_{A_k'^c}$, using Eqs. (12), (13) [8]

$$v_k^{'c} = \sqrt{\frac{\sum_{j=0}^{N^c-1} \mu_{A_k'^c}(x_{jk}^c) \cdot x_{jk}^c}{\sum_{j=0}^{N^c-1} \mu_{A_k'^c}(x_{jk}^c)}} \tag{12}$$

$$\sigma_k^{'c} = \sqrt{\frac{\sum_{j=0}^{N^c-1} \mu_{A_k'^c}(x_{jk}^c) \cdot (x_{jk}^c - v_k^{'c})^2}{\sum_{j=0}^{N^c-1} \mu_{A_k'^c}(x_{jk}^c)}} \tag{13}$$

where $k = 0, 1, \ldots, K-1$ and $c = 0, 1, \ldots, C-1$.
Equation (12) is used to calculate the arithmetic weight-mean of the vectors, in radius r_k^c, by means of membership function $\mu_{A_k'^c}$.

8. Refresh the value of density d_m^c using the following formula

$$d_m^c = \sum_{j=0}^{N^c-1} \prod_{k=0}^{K-1} \mu_k^{'c}(x_{jk}^c) - \sum_{l=0}^{L^c-1} D^{cl} \cdot \prod_{k=0}^{K-1} \mu_{A_k^{cl}}(v_k^{'c}) \tag{14}$$

9. Increase the number of rules $L^c = L^c + 1$
Then, add the new membership function, determined in step 7, as follows

$$\mu_{A_k^{cL^c-1}} = \mu_{A_k'^c}$$

and the density value obtained in step 8,

$$D^{cL^c-1} = d_m^c$$

where $k = 0, 1, \ldots, K-1$, and K – number of keywords,

10. Go to step 3.

5 Experimental Results

In this paper, the algorithm for determination of fuzzy rules for WWW document classification, based on dense areas is introduced. The algorithm was tested on the set of 568 abstracts of documents that belong to four classes: Artificial Intelligence (116), Robotics and Vision (92), Systems (202) and Theory (158), available at WWW server [25]. In the initial process, we obtained 3819 keywords.

Experiments have been performed for selected keyword factor values P from 1 to 60. The number of keywords was determined by selecting these keywords for which the number of occurrences in different documents was bigger than threshold P. For each set of keywords, the cosine normalization $TF_{ij} \cdot IDF_j$ was calculated in order to create vectors that represent WWW pages. This set of

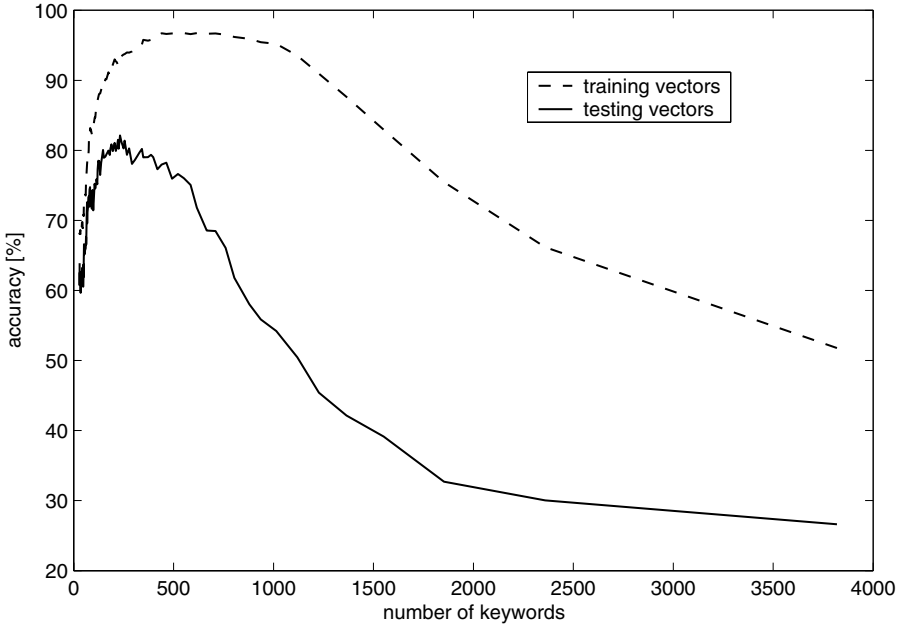


Fig. 1. Average accuracy of determination of fuzzy rules for different number of keywords for a constant value of training and testing vectors

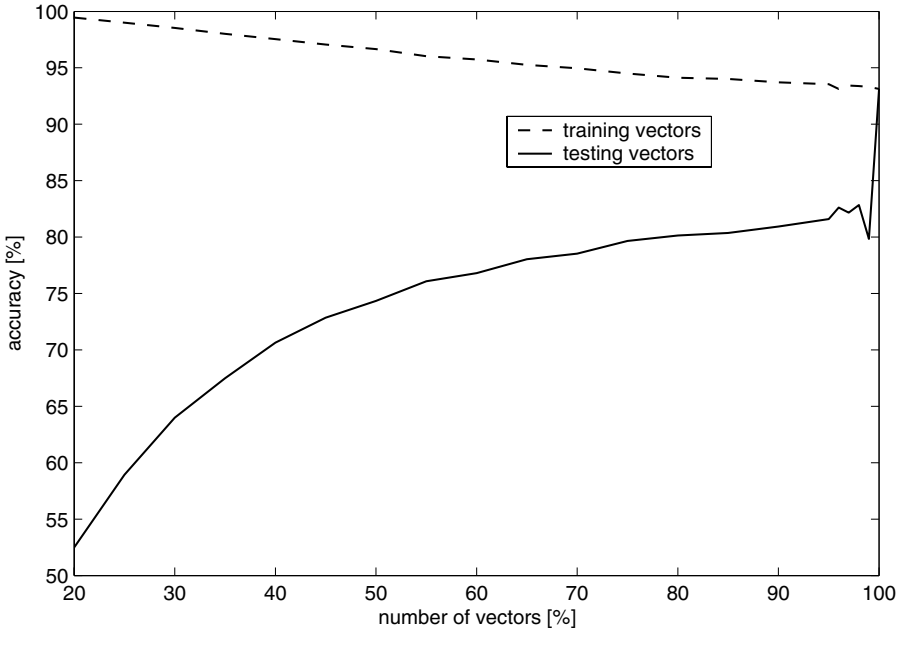


Fig. 2. Average accuracy of determination of fuzzy rules for different number of training and testing vectors for a constant value of keywords

vectors have been divided randomly into a training set (95% of all vectors), and a testing set (5% of all vectors). The vectors from the testing set are not included in the training set. The process of determination of fuzzy rules has been executed 100 times for different training sets. The results are presented in Fig. 1. The best average result for training vectors is 96.75% for 551 keywords and 82.14% for 232 keywords for testing vectors. Then, for 232 keywords (the best set of keywords for the testing vectors set) the experiments have been done 100 times. We determined ability of the fuzzy inference systems to correct classification, depending on different length of the learning and testing sets. This relationship is shown in Fig. 2. The experiments have been performed for other methods of calculation of degree of association between keywords and WWW pages, such as: TF, IDF, normalization TF, $TF \cdot IDF$.

6 Final Remarks

Correctness of the classification weakly depends on methods of calculation of term weights, for the algorithm presented in this paper. The results for the testing vectors are worse than for the training vectors because of huge number of keywords and the fact that some WWW pages contain only a small subset of the keywords. Another reason is that some keywords from the testing sets may not occur in the training sets or may occur very rarely. The effectiveness of the algorithm for generating fuzzy rules is good even for very small training sets (63.9% for 30% of all vectors, 52.5% for 20% of all vectors).

In further work, we plan to apply a two-phase algorithm. The first phase can be used to obtain centers of fuzzy rules in high dimensional keyword space. These centers can further be used for reduction of the keyword space. In this purpose, we plan to employ a method based on the Orthogonal Basis of Centroids [2]. The algorithm of generating the fuzzy rules will be applied once more for vectors in a low dimensional space at the second phase. We expect that this two-phase algorithm may improve the results. Some experiments have already been done, and the improvements observed.

References

1. Baldi P., Frasconi P., Smyth P., Modeling the Internet and the Web, Probabilistic Methods and Algorithms. Wiley. 2003.
2. Berry M. W., Survey of Text Mining, Clustering, Classification, and Retrieval. Springer-Verlag, New York, 2004.
3. Bo-Yeong K., Dae-Won K., Sang-Jo L., Exploiting concept clusters for content-based information retrieval, Information Sciences 170 (2005) 443-462.
4. Chakrabarti S., Van den Berg M., Dom B., Focused crawling: a new approach to topic-specific Web resource discovery. Computer Networks 31 (1999) 1623-1640.
5. Chiu S.L., Fuzzy model identification based on cluster estimation, J. Intell. Fuzzy Systems 2(3) (1994) 267–278.
6. Cho J., Garcia-Molina H., Page L., Efficient crawling through URL ordering, Computer Networks and ISDN Systems 30 (1998) 161-172.

7. Cortes C., Vapnik V.N., Support vector networks, Machine Learning 20 (1995) 1-25.
8. Euntai K., Minkee P., Seunghwan J., Mignon P., A new approach to fuzzy modeling, IEEE Transaction on Fuzzy Systems, Vol.5, No.3 (1997) 328-337.
9. Gulli A., Signorini A., The Indexable Web is More than 11.5 bilion pages, http://www.cs.uiowa.edu/~asignori/pubs/web-size.
10. Hastie T., Tibshirani R., Friedman J., Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer (2001).
11. Hersovici M., Jacovi M., Maarek Y. S., Pelleg D., Shtalhaim M., Ur S., The shark-search algorithm - an application: tailored web site mapping. In: Proceedings of the Seventh International World Wide Web Conference, Brisbane, Australia (1998) 317-326.
12. Jones S.K., Willet P., Readings in Information Retrieval, San Francisco: Morgan Kaufmann (1997).
13. Kłopotek A. M., Intelligent Search Engines. EXIT, Warszawa (2001), (in Polish).
14. Kraft D.H., Martin-Bautista M.J., Chen J., Sanchez D., Rules and fuzzy rules in text concept, extraction and usage, International Journal of Approximate Reasoning 34 (2003) 145-161.
15. Lam W., Ho C.Y., Using a generalized instance set for automatic text categorization. In Proc. SIGIR-98, 21stACM Int. Conf. on Research and Development in Information Retrieval (1998) 81-89.
16. Porter M., An algorithm for suffix stripping, Program, Vol.14, No.3 (1978) 130-137.
17. Rungsawang A., Angkawattanawit N., Learnable topic-specific web crawler, Journal of Network and Computer Application 28 (2005) 97-114.
18. Rutkowska D., Piliński M., Rutkowski L., Neural Networks, Genetic Algorithms and Fuzzy Systems. PWN, Warszawa, 1999, (in Polish).
19. Rutkowska D., Neuro–Fuzzy Architectures and Hybrid Learning. Springer–Verlag, Heidelberg, (2002).
20. Rutkowski L. Artificial Inteligence Methods and Techniques. PWN, Warszawa, 2005, (in Polish).
21. Sebastiani F., Machine learning in automated text categorization, ACM Computing Surveys (2002) 1-47.
22. Tao C.W., Unsupervised fuzzy clustering with multi-center clusters, Fuzzy Sets and Systems, Vol.128 (2002) 305-322.
23. Zadeh L. A., Fuzzy logic = Computing with words, IEEE Transactions of Fuzzy Systems, Vol.4, No.2 (1996).
24. Zadeh L. A., Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets and Systems 90 (1997) 111-127.
25. Source test data: http://www.cs.rochester.edu/trs.

A Possibilistic-Logic-Based Information Retrieval Model with Various Term-Weighting Approaches

Janusz Kacprzyk[1], Katarzyna Nowacka[2], and Sławomir Zadrożny[1]

[1] Systems Research Institute PAS, ul. Newelska 6, 01-447 Warsaw, Poland
{Janusz.Kacprzyk, Slawomir.Zadrozny}@ibspan.waw.pl
[2] Doctoral Studies (SRI PAS), ul. Newelska 6, 01-447 Warsaw, Poland
Katarzyna.Nowacka@orange.pl

Abstract. A new possibilistic-logic-based information retrieval model is presented. Its main feature is an explicit representation of both vagueness and uncertainty pervading the textual information representation and processing. The weights of index terms in documents and queries are directly interpreted as quantifying this vagueness and uncertainty. The classical approaches to the term-weighting are tested on a standard data set in order to validate their appropriateness for expressing vagueness and uncertainty[1].

1 Introduction

A *model of information retrieval* comprises the representation of both documents and queries as well as a mechanism for their *relevance* (*matching*) assessment. Usually both documents and queries are indexed with some *keywords* to obtain required representations. This can be done either manually or automatically. The words present in a document/query may play the role of keywords or a controlled vocabulary may be employed.

Whichever indexing mode is assumed *vagueness* and *uncertainty* is present in the process. The former manifests itself in a *gradual* nature of the relationship between a document/query and a keyword used to represent it. This is not a binary relationship: a keyword may be *important* for the representation of a document or query to a *degree*. Such an assumption seems to be quite obvious. It is now widely accepted and is the point of departure for the *vector space model*. Moreover one cannot be completely sure about the exact assessment of such a importance degree. This *uncertainty* should be also somehow represented. The vagueness and uncertainty of the representation in a natural way transfers to the notion of the *relevance* of a document for a query. It is a highly subjective and vague notion. Its automatic assessment definitely adds some uncertainty.

In the logical models [1,2] logical formulae are used for the representation purposes, and a kind of the logical entailment plays the role of the matching mechanism. Many approaches of this type are known in the literature that use various types of logic as their theoretical foundations (for the reviews cf., e.g.,

[1] Research supported by the MEiN Grant 3 T11C 052 27.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1120–1129, 2006.
© Springer-Verlag Berlin Heidelberg 2006

[3,4]). A basic logical model may be defined as follows. The queries are represented as simple formulae of the *classical propositional logic* built of propositional variables corresponding to the *keywords*. The documents are treated as the valuations of the propositional variables. A query and a document are recognized as matching if the formula representing the query is true under the valuation associated with the document.

Such a basic logical model is well-founded theoretically and provides the user with a rich formalism for the query expression. Its main shortcomings are implied by the binary nature of the classical logic employed. First of all, it is not possible to express in a gradual way the *importance* of particular keywords for the representation of the documents and queries. Secondly, also *the relevance* of a document for a query is here a yes-no notion. On the other hand, as argued above, both concepts of the importance and the relevance are intrinsically *imprecise* and that should be taken into account.

These phenomena are usually modelled using probabilistic approaches. However, in the context of logical models *possibility theory*, and in particular *possibilistic logic* might offer an interesting alternative. In this paper we propose a *possibilistic logic* [5,6] based approach that makes possible an explicit representation of *vagueness* and *uncertainty* pervading the information processing. In particular we use an extended version of the possibilistic logic due to Lehmke [7]. Our main goal in this paper is to check how the well-known *term-weighting* approaches of the vector space model are applicable in the context of such a logical model.

The related research comprises works on fuzzy logic based information retrieval approaches (cf., e.g., [8,9,10,11,12,13,14]) and on other non-classical logics based approaches (cf., e.g., [1,2,4,3,15]). Some preliminary ideas of the approach presented in this paper may be found in [16].

The organization of this paper is following: section 2 provides basics of the possibilistic logic and its extension. Section 3 presents proposed model of information retrieval and section 4 shows the results of some experiments.

2 Basics of the Possibilistic Logic and Its Extension

The possibilistic logic has been introduced by Dubois, Lang and Prade [5,6]. Here we recall only its basic concepts and limit our discussion to its propositional version. Moreover we are interested mainly in its semantics.

The starting point is the classical propositional logic which is extended by the introduction of *weighted formulae*. The motivation for the introduction of weights is to make it possible to directly express the uncertainty as to the *validity* of a formula: the higher the weight the more certain we are as to the validity of the formula. This (un)certainty is modeled in the framework of the possibility theory [17,18] and might be conveniently explained referring to the classical logical notions of the *interpretation* and *model*.

Let us assume an alphabet of our language, i.e., a set $\mathcal{A}$ of the propositional variables. Then, an *interpretation* (*valuation*) $\omega \in \Omega$ is a function:

$$\omega : \mathcal{A} \rightarrow \{0, 1\}$$

assigning the truth values 1 ("true") or 0 ("false") to all propositional variables of the alphabet $\mathcal{A}$. In the classical logic, for a given formula p we distinguish those interpretations Ω^p that make formula p true and we call $\omega \in \Omega^p$ *models* of p. The *characteristic function* of the Ω^p set may be then written as:

$$\chi_{\Omega^p}(\omega) = \omega(r) \tag{1}$$

Thus assuming p is true makes the interpretations belonging to Ω^p *possible* and those belonging to $\Omega \setminus \Omega^p$ *impossible*. Using the language of the *possibility theory* [18,19] we will say that p induces the following *possibility distribution* on the set of interpretations Ω:

$$\pi_p(\omega) = \chi_{\Omega^p}(\omega) = \begin{cases} 1 & \text{if } \omega \in \Omega^p \\ 0 & \text{if } \omega \notin \Omega^p \end{cases} \tag{2}$$

In the *propositional possibilistic logic* the formulae take the following form:

$$(p, \alpha) \tag{3}$$

where $\alpha \in [0, 1]$ expresses the lower bound on the *degree of certainty* of p while p is still assumed to be either "true (1)" or "false (0)". The *set* $\Omega^{(p,\alpha)}$ *of models of the weighted formula* (3) is *fuzzy* and its membership function is given by:

$$\mu_{\Omega^{(p,\alpha)}}(\omega) = \begin{cases} 1 & \text{if } \omega \in \Omega^p \\ 1 - \alpha & \text{if } \omega \notin \Omega^p \end{cases} \tag{4}$$

Thus, analogously to (2), a weighted formula (p, α) induces, by definition, the following possibility distribution on the set of interpretations Ω (cf. (2)):

$$\pi_{(p,\alpha)}(\omega) = \mu_{\Omega^{(p,\alpha)}}(\omega) = \begin{cases} 1 & \text{if } \omega \in \Omega^p \\ 1 - \alpha & \text{if } \omega \notin \Omega^p \end{cases} \tag{5}$$

If the state of knowledge is expressed with a formula (p, α) then the plausibility of any formula r is defined as the pair of the *possibility* and *necessity* measures (induced by the distribution $\pi_{(p,\alpha)}$ (5)) of the set of r's models, i.e., as:

$$(\Pi(\Omega^r), \mathcal{N}(\Omega^r)) \tag{6}$$

where Π and $\mathcal{N}$ denote the possibility and necessity measures, respectively, given by

$$\Pi(\Omega^r) = \max_{\omega \in \Omega^r} \pi_{(p,\alpha)}(\omega) \tag{7}$$

$$\mathcal{N}(\Omega^r) = \min_{\omega \notin \Omega^r} (1 - \pi_{(p,\alpha)}(\omega)) \tag{8}$$

It may be easily checked that for the formula p, using (5) and (7)-(8) one obtains for $\Pi(\Omega^p) = 1$ and $\mathcal{N}(\Omega^r) = \alpha$), i.e., it is to the degree 1 *possible* that p is true

and to the degree α necessary that it is true - what is in the accordance with the postulated interpretation of the weighted formula (3).

If a set of weighted formulae $P = \{(p_i, \alpha_i)\}$ is considered then jointly they induce the following possibility distribution on Ω:

$$\pi_P(\omega) = \min_i \pi_{(p_i,\alpha_i)}(\omega) \tag{9}$$

We will use the weighted formulae (3) to express the uncertainty as to the *importance* of a keyword for a document or query representation. As we assume the importance to be a *vague* concept we will use an extension of the possibilistic logic for the *many valued* case as proposed by Lehmke [7]. In this approach the formulae are still weighted formulae, but the weights are interpreted differently and referred to as *labels*. The notation of (3) changes to:

$$(p, l) \tag{10}$$

Now the interpretation $\omega \in \Omega$ is a function:

$$\omega : \mathcal{A} \to [0, 1]$$

A label l is interpreted as a fuzzy set of *truth values*:

$$\mu_l : [0, 1] \to [0, 1]$$

and (p, l) induces a possibility distribution $\pi_{(p,l)}$ on the set of interpretations Ω such that:

$$\pi_{(p,l)}(\omega) = \mu_l(\omega(p)) \tag{11}$$

For a set of formulae $\{(p_i, l_i\})$ (*the possibilistic knowledge base*) a possibility distribution induced by all of them jointly, is calculated as previously using (9).

Now the set of models Ω^r of any formula r is, in general, *fuzzy*:

$$\mu_{\Omega^r}(\omega) = \omega(r) \tag{12}$$

Thus if the state of knowledge is expressed with a formula (p, l) then the possibility that "r is true to degree α" [19] should be expressed as the fuzzy set τ such that:

$$\mu_\tau(x) = \sup_{\omega:\omega(r)=x} \pi(\omega)$$

This directly refers to the Zadeh's concept of the *fuzzy truth-value*. However it is inconvenient for our purposes and we use again the pair of the possibility and necessity values (6) to express the plausibility of the formula r – as suggested, e.g., in [19]. However this time he formulae (7)-(8) have to be adapted, taking into account that Ω^r is, in general, a fuzzy set:

$$\Pi(\Omega^r) = \max_{\omega\in\Omega} \min(\pi_{(p,\alpha)}(\omega), \mu_{\Omega^r}(\omega)) \tag{13}$$

$$\mathcal{N}(\Omega^r) = \min_{\omega\in\Omega} \max(1 - \pi_{(p,\alpha)}(\omega), \mu_{\Omega^r}(\omega)) \tag{14}$$

Some special labels l of interest here are as follows [7]. The label l^T referred to as "TRUE" has the membership function:

$$\mu_{l^T}(x) = x \tag{15}$$

Thus assuming (p, l^T) is expressing the state of knowledge means that more possible are interpretations giving p a higher truth value. A slightly modified version of l^T referred to as "TRUE with doubt δ" and denoted as l^T_δ is defined by this membership function:

$$\mu_{l^T_\delta}(x) = \begin{cases} \delta & \forall x \leq \delta \\ x & \forall x > \delta \end{cases} \tag{16}$$

Assuming (p, l^T_δ) is expressing the state of knowledge means that the certainty of p is somehow limited, thus the possibility of interpretations assigning to p the truth value lower than δ is equal δ instead of reducing towards 0 as it is the case of l^T.

3 The Model

The following notation will be used:

$D = \{d_i\}_{i \in [1,N]}$	– is the set of documents,
$T = \{t_j\}_{j \in [1,M]}$	– is the set of index terms (keywords),
q	– is a query,
$d(t_j)$	– is a weight assigned to the index term t_j in a document d,
$q(t_j)$	– is a weight assigned to the index term t_j in a query q.

It is assumed that the documents and queries are indexed using usual term-weighting approaches (c.f., e.g. [20]). Thus, each index term is assigned a *weight* in each document and query. This weight expresses its importance for the representation of this document or query. Assuming the weights are normalized we can readily interpret them as the *fuzzy set membership function values* expressing *vagueness* of the relationship between index terms and documents/queries. We can further interpret them as the *truth values* of a *many valued logic* what is convenient from the viewpoint of a logical approach. On the other hand, the results of the indexing, either manual or automatic, cannot be treated as completely certain. Thus in the proposed logical model we want to represent the above mentioned vagueness and uncertainty in a uniform way. In order to achieve that we propose to employ the extended possibilistic logic in the following way.

As in the basic logical approach a *propositional variable* p_j is associated with each index term t_j. Then, each document and query are represented as sets of *weighted propositions* of the extended possibilistic logic:

$$d_i = \{(p_k, l_k)\}_{k \in K_i}$$
$$q = \{(p_k, l_k)\}_{k \in K}$$

which may be interpreted as the possibilistic knowledge bases (cf. previous section). Now we will describe how the labels l are constructed.

Documents. Firstly, each document is represented as in the vector space model using selected term-weighting approach [20]. Thus, for each document $d \in D$ and each index term $t_j \in T$ a weight $d(t_j)$ is computed. Then the weights are normalized so as to obtain the maximum weight equal 1. Finally, the document d is represented as a set of weighted formulae (17):

$$\{(p_j, l(d(t_j))\}_{j=1,\ldots,M} \tag{17}$$

where p_j is a propositional variable corresponding to the index term t_j and $l(u)$ is a label with the following membership function (parametrized with u):

$$\mu_{l(u)}(x) = 1 - \mid x - u \mid \tag{18}$$

Thus, recalling the semantics of weighted formulae, the induced possibility distribution π_d^j on the set Ω "favors" those interpretations $\omega \in \Omega$ that are assigning p_j the truth value close to $d(t_j)$. The larger the difference $\mid \omega(p_j) - d(t_j) \mid$ is the less possible the interpretation ω is. The overall possibility distribution induced by the document d represented with the possibilistic knowledge base (17) is computed using (9).

The motivation for (17)-(18) is such that it is assumed that the weight $d(t_j)$ expresses the importance of the index term t_j for the representation of d. However due to the uncertainty related to the indexing process other levels of importance are also possible to a degree quantified by (18).

Queries. For a query q and each index term $t_j \in T$ the normalized weight $q(t_j)$ is computed as in case of documents, although a different term-weighting approach might be applied. Finally the query q is represented as the following possibilistic knowledge base:

$$\{(p_j, l^T_{1-q(t_j)})\}_{j=1,\ldots,M} \tag{19}$$

where p_j is a propositional variable corresponding to index term t_j and $l^T_{1-q(t_j)}$ is a label of the l^T_δ type (cf. (16)) with $\delta = 1 - q(t_j)$.

The possibility distribution on $\omega \in \Omega$ induced by the possibilistic knowledge base (19) representing the query q favors those ω that assign to p_j as high truth value as possible. However, those ω that assign low truth values are still possible to some degree which is the higher the lower $q(t_j)$ is.

The motivation for (19) is such that it is assumed that the index terms used in the query are treated by the user as very important (to the degree 1). It may be argued that in case of usually short queries (typical for, e.g., search engines) there is no need to use less important terms. However, the user might be uncertain if they are really important (at all!) and thus $q(t_j)$ is interpreted as an expression of this (un)certainty: when $q(t_j) = 1$ the user is completely sure and l^T_δ in (19) becomes l^T making term t_j highly desired, while small $q(t_j)$, close to 0, expresses high uncertainty as to the relevance of t_j and thus limiting its influence on the resulting possibility distribution over Ω.

Matching degree. In order to compute the matching degree between a document and a query we assume the following logical setting. A document is represented as a possibilistic knowledge base (17) and induces the possibility distribution π_d over the set of interpretations Ω defined with (18) and (9). Similarly a query is represented as a possibilistic knowledge base (19) and induces a corresponding possibility distribution π_q that in turn is interpreted as defining a fuzzy set Ω^q of the models of the query q such that $\mu_{\Omega^q}(\omega) = \pi_q(\omega)$. Then, the matching degree of a document d against a query q is computed as the pair of possibility and necessity measures (induced by possibility distribution π_d) of the fuzzy set Ω^q. Formally using the formulae (13)-(14) that might be expressed as follows:

$$\Pi_d^j(\Omega^q) = \begin{cases} 1 - q(t_j) & \text{for } q(t_j) \leq \frac{1-d(t_j)}{2} \\ \frac{1+d(t_j)}{2} & \text{for } q(t_j) > \frac{1-d(t_j)}{2} \end{cases} \tag{20}$$

$$\mathcal{N}_d^j(\Omega^q) = \begin{cases} 1 - q(t_j) & \text{for } q(t_j) \leq 1 - \frac{d(t_j)}{2} \\ \frac{d(t_j)}{2} & \text{for } q(t_j) > 1 - \frac{d(t_j)}{2} \end{cases} \tag{21}$$

where $d(t_j)$ and $q(t_j)$ are weights of index term t_j in a document d and a query q, respectively; Π_d^j and $\mathcal{N}_d^j$ are possibility and necessity measures induced by π_d^j related to an index term t_j.

The overall matching is expressed as a pair:

$$(\Pi_d(q), \mathcal{N}_d(q)) \tag{22}$$

such that

$$\Pi_d(q) = \min_j \Pi_d^j(\Omega^q)$$

and

$$\mathcal{N}_d(q) = \min_j \mathcal{N}_d^j(\Omega^q)$$

Having such pairs of numbers it has to be decided how to order the documents in response to the query. In the experiments reported in the next section we assume the lexicographic order on the pairs (22). However in the tests we check the lexicographic order based on the original pairs $(\Pi_d(q), \mathcal{N}_d(q))$ as well as on the pairs $(\mathcal{N}_d(q), \Pi_d(q))$.

The apparently simple formulae (20)-(21) are in fact quite intuitive. Let us take a closer look at them considering particular extreme cases of a term t_j weights in a document and in a query.

Case 1: $d(t_j)$ and $q(t_j)$ are *low*.
Then both possibility and necessity of match are *high*. Thus due to (9) the index term t_j has a rather limited influence on the matching. This is definitely an appropriate behaviour .

Case 2: $d(t_j)$ is *low* and $q(t_j)$ is *high*.
Then the possibility is around 0.5 and the necessity is *low*. This is surely a proper indication of the mismatch between the query and the document with *respect to the index term* t_j. Due to (9) this indication extends to the *whole*

query q. This is an effect of the non-compensatory nature of the minimum operator.

Case 3: $d(t_j)$ is *high* and $q(t_j)$ is *low*.
Then both possibility and necessity of match are *high*. Thus due to (9) the index term t_j has a rather limited influence on the matching. This is definitely an appropriate behaviour.

Case 2: $d(t_j)$ is *high* and $q(t_j)$ is *high*.
Then the possibility is *high* and the necessity is around 0.5. This is an appropriate behavior. The relatively low value of the necessity stems from the fact of a rather strongly fuzzified values of the index terms weights, both in the documents and in the queries, as defined by (17) and (19).

4 Experiments

Ideally the labels l in the representations of both documents (17) and queries (19) should be directly determined by a human user, possibly supported by a suitable user interface. However in case of documents it is a rather infeasible solution while in case of queries might require an advanced user interface. Thus in our preliminary experiments with the proposed possibilistic information retrieval model we use the weights computed using some well known term-weighting approaches [20] as a basis for the labels appearing in (18) and (19).

In our experiments we are using the Cranfield test collection (cf. [21] for a description) comprising of 1398 documents and 225 queries. We evaluate the effectiveness of the retrieval using the R-precision measure for each query, i.e., computing the precision of the results at the k-th position of the results list, where k is equal to the number of relevant documents for given query (the list of relevant documents for each query is given as a part of the Cranfield collection). Finally, the R-precision is averaged over all queries. Other measures of the effectiveness such as 11-point AVP or the Average Precision at Seen Relevant Documents is also applicable here.

Table 1. R-precision for the Cranfield test collection and various term-weighting approaches

		R-Precision	
Documents	Queries	(P,N)	(N,P)
tfx	tfx	0.1782	0.1904
tfc	nfx	0.1097	0.1106
nxx	bpc	0.0581	0.0542

Table 1 lists the results of our experiments for a few selected combinations of term-weighting approaches for documents/queries. The first and second columns indicate term-weighting approaches used for docs/queries representation (coding

of approaches as in [20]). The third column shows R-precision when resulting documents are ordered first according to the possibility measure and then to the necessity (cf.(22)). Results in column four are obtained for ordering first according to the necessity and then the possibility.

The best results has been obtained for the most popular tf × IDF approach (coded tfx × tfx). For a comparison, using the same term-weighting approach and the classical vector space model matching via the cosinus measure we have obtained slightly better results (0.2404). The ordering of the results first according to the necessity and then possibility gives slightly better results.

5 Concluding Remarks

We have proposed a new possibilistic model of information retrieval. Its main feature is an explicit representation of vagueness and uncertainty pervading the information retrieval process. The results of some preliminary computational experiments using various term-weighting approaches are reported. The results are not conclusive and definitely a further research is needed. In particular a further analysis of the interpretation of the weights as indicators of the uncertainty and vagueness of the importance of the index terms for the representation of documents and queries is required.

References

1. van Rijsbergen C. J.: A new theoretical framework for information retrieval. In Rabitti F., ed.: Proc. of ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy (1986) 194–200
2. van Rijsbergen C.J.: A non-classical logic for information retrieval. The Computer Journal **29**(6) (1986) 481–485
3. Lalmas M.: Logical models in information retrieval: Introduction and overview. Information Processing & Management **34**(1) (1998) 19–33
4. Sebastiani F.: A note on logic and information retrieval. In: MIRO'95 Proc. of the Final Workshop on Multimedia Information Retrieval, Glasgow, Scotland, Springer (1995)
5. Dubois D., Lang J., Prade H.: Possibilistic logic. In Gabbay D.M. et al., ed.: Handbook of Logic in Artificial Intelligence and Logic Programming. Volume 3. Oxford University Press, Oxford, UK (1994) 439–513
6. Dubois D. and Prade H.: Possibilistic logic: a retrospective and prospecive view. Fuzzy Sets and Systems **144** (2004) 3–23
7. Lehmke S.: Degrees of truth and degrees of validity. In Novak V., Perfilieva I., eds.: Discovering the World with Fuzzy Logic. Physica-Verlag, Heidelberg New York (2000) 192–236
8. Radecki T.: Fuzzy set theoretical approach to document retrieval. Information Processing and Management **15**(5) (1979) 247–260
9. Buell D. and Kraft D.H.: Threshold values and Boolean retrieval systems. Information Processing & Management **17** (1981) 127–136
10. Kraft D.H. and Buell D.A.: Fuzzy sets and generalized Boolean retrieval systems. International Journal on Man-Machine Studies **19** (1983) 45–56

11. Bordogna G., Pasi G.: Application of fuzzy sets theory to extend Boolean information retrieval. In Crestani F., Pasi G., eds.: Soft Computing in Information Retrieval. Physica Verlag, Heidelberg New York (2000) 21–47
12. Herrera-Viedma E.: Modeling the retrieval process of an information retrieval system using an ordinal fuzzy linguistic approach. JASIST **52**(6) (2001) 460–475
13. Yager R.R.: A note on weighted queries in information retrieval systems. JASIST **38** (1987) 23–24
14. Zadrożny S., Kacprzyk J.: An extended fuzzy boolean model of information retrieval revisited. In: Proc. of FUZZ-IEEE 2005, Reno, NV, USA, IEEE (May 22-25, 2005) 1020–1025
15. Brini A.H., Boughanem M., Dubois D.: Towards a possibilistic model for information retrieval. In De Baets B., De Caluwe R., De Tre G., Fodor J., Kacprzyk J., Zadrożny S., eds.: Current Issues in Data and Knowledge Engineering. EXIT, Warszawa (2004) 92–101
16. Bieniek K., Gola M., Kacprzyk J., Zadrożny S.: An approach to use possibility theory in information retrieval. In: Proc. of the 12th Zittau East-West Fuzzy Colloquium, Zittau, Germany (2005)
17. Zadeh L.A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems **1** (1978) 3–28
18. Dubois D., Prade H.: Possibility Theory. Series D: System Theory, Knowledge Engineering and Problem Solving. Plenum Press, New York (1988)
19. Dubois D. and Prade H.: Possibility theory, probability theory and multiple-valued logics: A clarification. Annals of Mathematics & Artificial Intelligence **32** (2001) 35–66
20. Salton G. and Buckley C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management **24** (1988) 513–523
21. Sparck Jones K., Bates R. G.: Research on automatic indexing 1974- 1976 (2 volumes). Technical report, Computer Laboratory. University of Cambridge (1977)

Sketch of Autopoietic Essence of *Computing* and *KnowledgeWorking*

Leonid Kompanets

Czestochowa University of Technology, Institute of Computer & Information Sciences
Dabrowski str., 73, 42-200 Czestochowa, Poland
leonid.kompanets@icis.pcz.pl

Abstract. The purpose of this article is to provide a precise better understanding of the *Computing-KnowledgeWorking* essence and an overview of the current/future state of its research. The main attention is concentrated on an autopoietic mechanism of the both *Computing*, and *NaturalKnowledgeWorking* techniques. The essence of theories of artificial intelligence and soft computing as well as *Computing-NaturalKnowledgeWorking* phenomenon understanding has been analysed. The innovative approach called "Three information translations", by means of which it is possible to realize the information chain "The task of end user ... computer program" in automatic mode, is presented. It is proposed to define precisely the vocabulary in the forenamed field.

1 Background, Fundamental Notion Definitions and *Computing-KnowledgeWorking* Problem Statement

In his last book [1], P. Drucker – the guru of modern management science and practice – separates (amongst some revolutionary ideas and perspectives) emergence of new category of worker – Knowledge Worker. He says that "the most important, and indeed the truly unique, contribution of management in the 20th century was the fifty-fold increase in the productivity of the "manual worker" in manufacturing. The most important contribution management that needs to be made in the 21st century is similarly to increase the productivity of "Knowledge Worker"; (here complex processes of their work called "*KnowledgeWorking*"; it concerns both natural, and computational components of knowledge). If the most valuable assets of a 20th-century company were its production equipment, then the most valuable asset of a 21st-century institution will be its knowledge workers and their productivity". These workers work their mind and character with the use of their own private production means – their own head to process the special knowledge in circumstances of common scientific/engineering applications!

In the tutorial [2], it is reminded that "In 1936 Alan Turing laid the theoretical groundwork for modern computing science (along with others, including A. Church, E. Post, A. Markov), by defining what later became known as a universal Turing machine (UTM). Apart from the UTM, various other computational formalisms exist". Consequently, it was established that the basic theoretical notion of computing is the notion of *algorithm*, and this means that in *automatic*

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1130–1139, 2006.
© Springer-Verlag Berlin Heidelberg 2006

way ***only*** *an algorithmic procedure can be executed by computer*. In the theory of artificial intelligence (AI) the Computational Theory of the Mind (CTM) [11] has been accepted as a basic labor hypothesis, and this means that the mental processes in live systems have been brought down to simulations of artifacts formally obtained by computer, which are the "artificial limbs" of relevant live systems functions. As practice confirmed, Turing test, executed to establish the level of artificial system intelligence, is controversial [11]. Today, computation is more than ever ubiquitous, embraces a wide range of topics, and gives rise to many *controversies* [2], [3], [6], [7], [8], [15], [16], and [17].

At present, scientists thinking about thinking [3], [8], [11], and [15] seek to understand, extract, use and abuse the organizational principles of information processing in natural living system. But live organism architecture is based on *Tensegrity* principle [5], which is diametrically opposed to the architecture of artificial devices, in which the silicon base was usually used. Nevertheless, researchers in this *trans-disciplinary* field are thus all directly or indirectly concerned with integration of enumerated approaches. On account of the fact that the problem base-point in understanding the *NaturalKnowledgeWorking-Computing* mechanisms phenomenon is automatic computational procedures, it may well be worthwhile to have the understanding of the following questions. What models of computation exists? What are their limits? Which problems are solvable, which are unsolvable *by computers* and *in nature*? Which computational models are good for which problems? How relevant is the concept of universal computation for the design of intelligent machines? Does a UTM really capture the essence of any and all forms of computing? What is the Church-Turing thesis? What are the relevant differences between information processing *in nature* and *in computer science*? By the end of the paper, we shall know *what computation is, what it isn't, and what its limits and applications* are. This paper shall provide the right toolbox of ideas for researchers' thinking.

2 *Algorithm* and *Computing* Fundamentals

Conventional computing is based on 18 discoveries, of which the main one is the notion of algorithm [12]. It was used in relatively non-complicated program systems. To higher achievements in modern programming and applications one can ascribe the development of *C++* language (which allows for the design of systems containing near some millions of code lines [4]) and MPEG-7 (MPEG-21) standard for multimedia metadata manipulation.

We will count over the basic notions of conventional computing: Logic and Formal Systems; Alphabets and Words, Grammars, Languages; Functions; State Machines; Automata; Turing-Neumann computer architecture; Computability, Unsolvability, Undecidability; Gödel's Theorem; Church-Turing Theorem; Complexity Theory; Randomization; Church-Turing Thesis; Computation beyond the Turing-Neumann Computers; Computation and the Brain; Unconventional Computing Mediums; and *many more*.

The basic moment is to relate the principles of artificial neural networks functioning [7] to the field of computing (Fig.3) [7], networks obviously do not contain the processor, or memory in traditional sense and are not traditionally programmable but learned. However, the mere fact of artificial neural networks emulation by means of conventional programming confirms that those networks are programmable in principle, but by means of structure formalization. This way, the artificial neural network, is the compound – structured information unit, by means of which specific mathematical procedures are being realized.

But in author's opinion, as an effect of joining two computers of Turing-Neumann architecture, there arises the emergency effect, called the new feature emergence system function that non-strictly expressed by "$1 + 1 > 2$ formula". In technical self-organizing systems and in natural living organisms, there can arise synergistic effect ("formula: $1 + 1 >>> 2$"). *Let's focus our attention on the basic fact that autopoietic computing mechanisms in broad sense are realized on the algorithmic level (formal data/knowledge structures manipulation) and/or specific manipulations of structures providing systems to synergistic (self) organization.*

Natural computing methodologies are based on DNA, amorphous, membrane, quantum, cellular, molecular, neural, evolutionary computing and various other proposals of biological/psychological-inspired types for computations that go beyond the Turing model. For example, membrane computing is a branch of natural computing which investigates computing models abstracted from the *structure* and functioning of living cells and from their interactions *in tissues* or *higher-order biological structures*. The models considered, called membrane systems, are parallel, and distributed computing models, processing multi-sets of symbols in cell-like compartmental architectures. In many applications membrane systems have considerable advantages – among these are their *inherently discrete nature, parallelism, transparency, scalability and non-determinism*.

Live nature does not need traditional programmers. It "programs" by evolution right, which is referred to by term *Natural Selection*. "I have called this principle, by which each slight variation, *if useful, is preserved*" (Charles Darwin, *The Origin of Species*, 1859). In that way, *Computing-NaturalKnowledgeWorking* phenomenon in broad sense integrates dialectically completely antagonist conceptions of computing/development. But, in author's opinion, the two mentioned aspects (*Computing and NaturalKnowledgeWorking*) cannot be univocally brought together. The concern is for *NaturalKnowledgeWorking* rationally aided by *Computing*, the autopoietic possibilities of which grow drastically.

3 Contents of Artificial Intelligence and Soft Computing

At the moment, there exist many branches of science concerning the problem. They are, for example, AI, Artificial Life (AL), Soft Computing (SC), Fuzzy Logics (FL), Computational Intelligence (CI), Genetic Computing, Natural Computing, and so forth. Their fields interweave and the names do not reflect the content such the state of the matter in fields AI, AL, SC, CI are given below.

In accordance with [10], "AI is a field of science and engineering concerned with computational understanding of what is commonly called *intelligent behavior*, and with the creation of artifacts that exhibit such behavior." As AI-complete tasks are called: Natural Language, Problem Solving and Search, Knowledge Representation and Reasoning, Learning, Vision, Robotics. In accordance with the international *Journal of Applied Soft Computing*, "SC differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation. In result, the role model for SC is the human mind. The *guiding principle* of SC is: Exploit the tolerance for imprecision, uncertainty, partial truth, and approximation to achieve tractability, robustness and low solution cost. The *basic ideas* underlying SC in its current incarnation have links to many earlier influences, among them Zadeh's 1965 paper on fuzzy sets; the 1973 paper on the analysis of complex systems and decision processes; and the 1979 report (1981 paper) on possibility theory and soft data analysis. The inclusion of neural computing and genetic computing in SC came at a later point".

At this juncture, the principal constituents of SC are FL, Neural Computing (NC), Evolutionary Computation (EC), Machine Learning (ML) and Probabilistic Reasoning (PR), with the latter subsuming belief networks, chaos theory and parts of learning theory. What is important to note is that SC is not a *mèlange*. Rather, it is a partnership in which each of the partners contributes a distinct methodology for addressing problems in its domain. In this perspective, the principal constituent methodologies in SC are complementary rather than competitive. Furthermore, SC may be viewed as a *foundation component* for the emerging field of conceptual intelligence that conclude: Fuzzy Systems; Neural Networks; EC; ML; PR. The complementarity of FL, NC, EC, and PR has an important consequence: in many cases a problem can be solved most effectively by using FL, NC, EC and PR in combination rather than exclusively. A striking example of a particularly effective combination is what has come to be known as "neurofuzzy systems". Such systems are becoming increasingly visible as consumer products ranging from air conditioners and washing machines to photocopiers and camcorders. Less visible but perhaps even more important are neurofuzzy systems in industrial applications. What is particularly significant is that in both consumer products and industrial systems, the employment of SC techniques leads to systems which have high MIQ (Machine Intelligence Quotient). In large measure, it is the high MIQ of SC-based systems that accounts for the rapid growth in the number and variety of applications of SC. The conceptual structure of SC suggests that users should be trained not just in FL, neurocomputing, genetic programming, or PR but in all of the associated methodologies, though not necessarily to the same degree.

Topics of the International Journal of Applied Soft Computing are a journal promoting an integrated view of SC to solve real life problems. "SC is a collection of methodologies, which aim to exploit tolerance for imprecision, uncertainty and partial truth to achieve *tractability, robustness and low solution cost*. The focus is to publish the highest quality research in application and

convergence of the areas of FL, NN, EC, Rough Sets and other similar techniques to address real world complexities. Major topics of *Applied Soft Computing* cover the following SC and *related techniques*, interactions between several SC, and their industrial applications: Fuzzy Computing (C.), Neuro C., Evolutionary C., Probabilistic C., Immunological C., Hybrid Methods, Intelligent Agents and Agent Theory, Causal Models, Case-based Reasoning, Chaos Theory, Interactive Computational Models. The application areas of interest include *but are not limited to*: Decision Support; Process and System Control; System Identification and Modeling; Optimization; Signal or Image Processing; Vision or Pattern Recognition; Condition Monitoring; Fault Diagnosis; Systems Integration; Internet Tools; Human-Machine Interface; Time Series Prediction; Robotics; Motion Control and Power Electronics; Biomedical Engineering; Virtual Reality; Reactive Distributed AI; Telecommunications; Consumer Electronics; Industrial Electronics; Manufacturing Systems; Power and Energy; Data Mining; Data Visualization; Intelligent Information Retrieval; Bio-inspired Systems; Autonomous Reasoning; Intelligent Agents. (Author's remark: All-embracing!?.)

4 Understanding *Computing-NaturalKnowledgeWorking*

Effective human-computer cooperation is an indispensable phenomenon. However, human feels and thinks in analogue way, and computer – in digital. An *active constituent* of computer is the formal program written by human mind in a computer language; human differs from artificial system or animal in that he has his own speech, feels pain, possesses consciousness, has motivation, and *behaves rationally* in real world. Human achieves it all by means of mind functioning and knowledge processing in his brain. Besides, computer data processing realizes in algorithmic way on syntax level. In accordance with the CTM, in human mind knowledge processing the semantic and pragmatic information

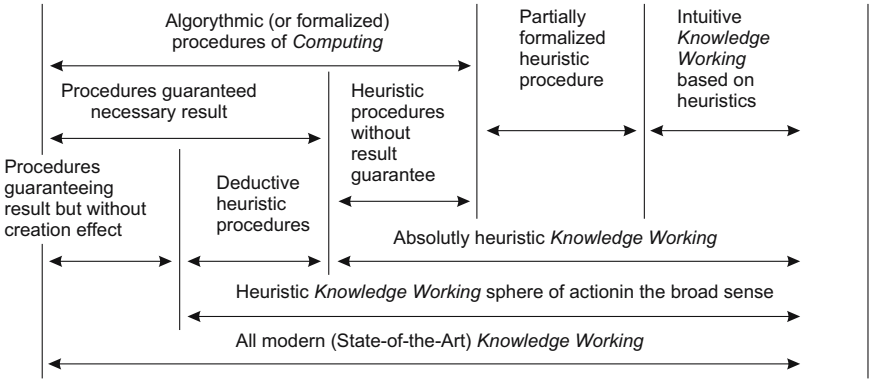


Fig. 1. Structure of *Computing-NaturalKnowledgeWorking* phenomenon with more detailed accentuation on autopoietic part of the problem

levels are additionally used. Data structures used in computer are sufficiently primitive; structures describing natural reality are extremely complicated.

In Fig. 1, 2, 3 the structure of *Computing-NaturalKnowledgeWorking* phenomenon (based on Z. Rabinovitch idea), computer Metaphor evolution [6], and computing

Paradigms [7] are given accordingly. Because of a lockage of a paper place, the contents of the points of the matter are presented in graphical notation (here and below).

It is interesting to notice three facts. Firstly, Fig.1 – structure does not contain a rigorous limit between computing phenomenon (realized in artificial environment) and *NaturalKnowledgeWorking* (in natural); it has not been said any about the possibilities of "brute force" approach to solving problems lying on borderline of the two phenomena. Secondly, there has been noted the tendency to computer metaphor change and given their generalized characteristics (Fig.2)

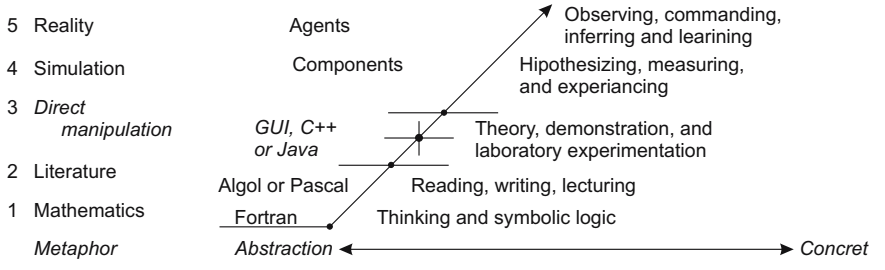


Fig. 2. Input-Output generalized scheme of computing metaphor evolution in the coordination system "Abstraction/Concrete" (based on T. Lewis's view [6])

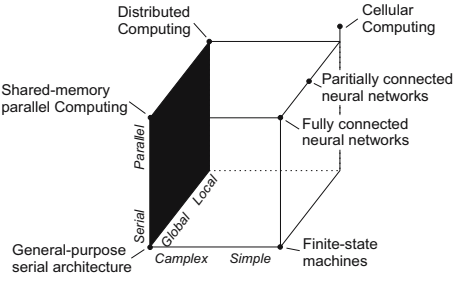


Fig. 3. Contents of interaction procedure between knowledge engineer (KE) and expert (E) from a concrete problem domain (PD) knowledge

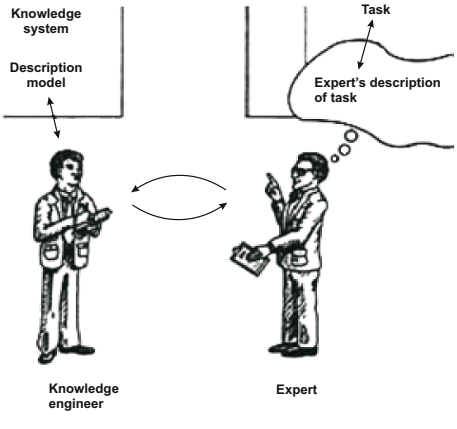


Fig. 4. Classification of computing paradigms in the coordination system "Complex/Simple - Serial/Parallel - Global/ Local" (M. Sipper's view [7])

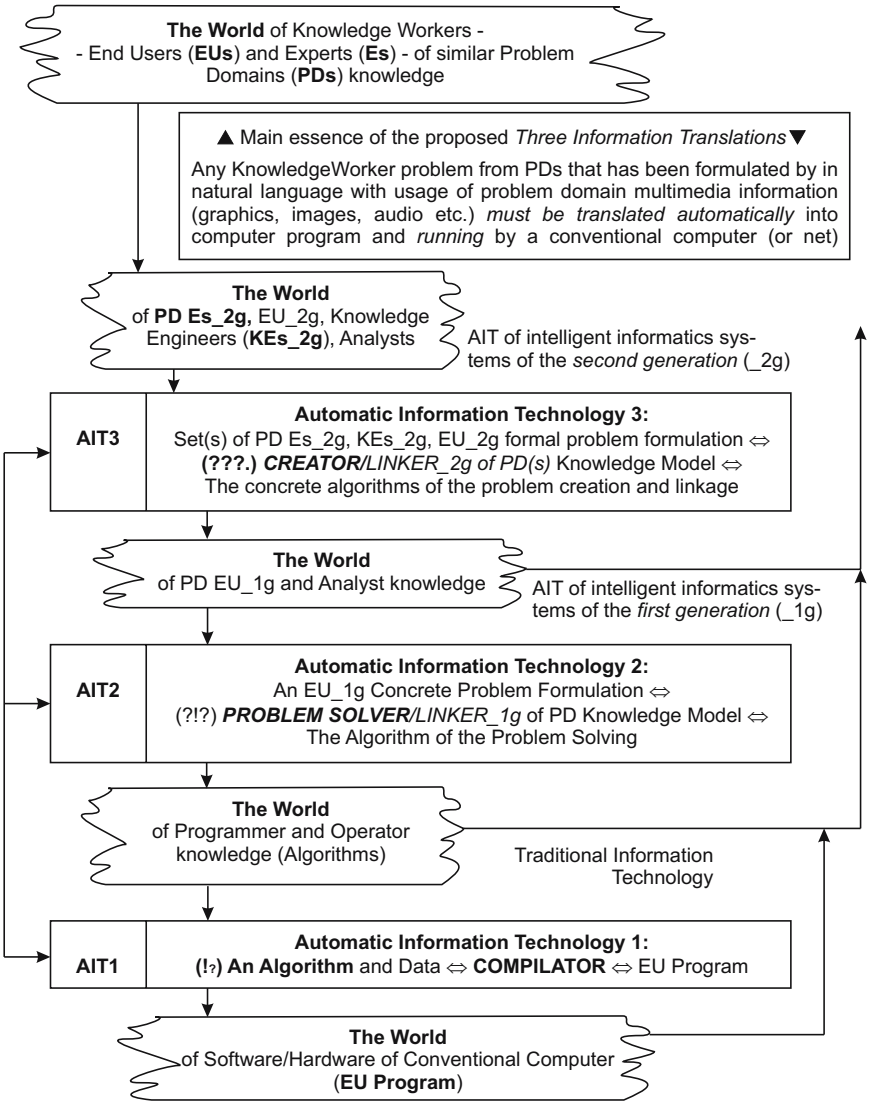


Fig. 5. Sketch of the relationship between *NaturalKnowledgeWorking* and *Computing* procedures during their automatic computer realization by means of proposed "Tree Information Transformations". (!?, ?!?, ??? – Marks correspond to author's state-of-the-art (subjective) estimation)

[7]. What deserves attention is a deeper analysis of the effect of C++ language elaboration [4], [15]. Thirdly, the phenomenon of cellular computing, connected with vast parallelism is not strictly located in the accepted coordinate system (see Fig.3).

5 Autopoietic Essence of the Three Information Transformations and the Proposed Terminology

In Fig.5, the author's original view of organization principles of conventional intelligent information system is given. A very rough sketch of the basic idea can be found in [13, Rys.1.2, p. 24]. To realize the End User (EU) of Problem Domain (PD) of knowledge approach in a (nearly, practically) automatic way by computer, we need to provide the possibility of realization of three necessary information translations *AIT1-AIT3* (Fig.5). Respectively for each translation, we need to create active program blocs: COMPILATOR, *PROBLEM SOLVER_1g, and CREATOR_2g* respectively. Functions of enumerated blocs are given in Fig.5. Commercial intelligent informatics systems of the first generation (*_1g*) exist. In Fig.4, contents of interaction procedure between knowledge engineer (KI) and expert (E) from PD of knowledge is illustrated.

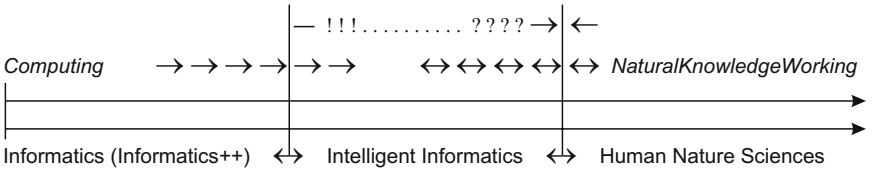


Fig. 6. Proposed modern classification of trans-disciplinary sciences based on essence of the *Computing-NaturalKnowledgeWorking* phenomenon

In Fig.6, the proposed precise classification of *Computing-NaturalKnowledge-Working* phenomenon is adduced as the alternative to existed a great deal of non-orthogonal notions (for example, such as AI, SC, CI, AL, computation with words and so forth).

6 Main Study Results

On the base of the study, the following suggestions may be accepted.

6.1. The basic scientific feature is precise vocabulary as well as precise thinking. To fulfill John von Neumann's aphorism "There is no sense being precise when you don't even know what you're talking about" in *Computing - NaturalKnowledgeWorking* field the use of notions *Informatics* (or *Informatics++*), *Intelligent Informatics, Human Nature Sciences* has been proposed.

Today, in Intelligent Informatics, methods are used that provide so much "true intelligence as there is teleportation in Formula-1 races" (A. Plachov phrase). The main thorough research track must be attributed to the elaboration of active blocs of the type of problem *SOLVER*, CREATOR, especially taking into account the linguistic aspects of strict description of intelligent problems.

6.2. Object-oriented programming and design of algorithmic language *C++/Java* type is a quality event because it gives the possibility of designing of reliable systems with the complexity of some millions code rows.

Apart from that in C++, there has been realized: a) the possibility of creating abstract data types for creating complex data structures (for example, like ones that are AI-formalisms), b) communication between modules by means of interfaces and not in procedural way, c) programmer by analogy to *End User* became *End Programmer*, that is, he obtained the possibility of programming independently of hardware features.

6.3. Natural intelligence and artificial one as well as network computing are things that may be used in common *only* for effective (systemic and/or synergistic) cooperation with computer. Computer cannot *until* think, have consciousness, motivation, feel pain, replicate, etc. And what are more computers can not do: "The most important thing in communication - to hear *what isn't being said* (Peter F. Drucker' citation)".

And this deaf angle in mind functioning simulation by means of simulation of function of their constituents (neurons) will exist until we create the model of human cell biological life, and then the method and device for SENSE detecting and processing will not be invented. Unfortunately, the situation in intelligent informatics is similar to that in P. Drucker's citation "We know nothing about motivation. All we can do is writing books about it".

References

1. Drucker, Peter F.: *Management Challenges for the 21st Century.* Butterworth-Heinemann, London, 1999
2. Teuscher, Christof: The Complete Idiot's Tutorial to In Theorio, In Silico, and In Vivo Computation. The abstract of tutorial of 8th European Conf. on *Artificial Life – ecol 2005*, 5-9 Sept. 2005, University of Kent, Canterbury, UK. Available at the official website of the Conf. http://www.ecal2005.org
3. de Kerckchove, Derick: *The Skin of Culture.* Somerville House Books Limited, Toronto, Ontario, Canada, 1995
4. Stroustrup, Bjarne: *The Design and Evolution of C++.* Addison Wesley Longman, Inc., 1994
5. Ingber, Donald E.: The Architecture of Life. *Scientific American*, Jan. 1998, 30-39
6. Lewis, Ted: Absorb and Extend: Resistance is Futile! *Computer. Innovative Technology for Computer Professionals.* May 1997, 109-113
7. Sipper, Moshe: The Emergence of Cellular Computing. *Computer. Innovative Technology for Computer Professionals.* July 1999, 18-26
8. Penrose, Rodger: *The Emperor's New Mind Concerning Computer, Minds, and the Law of Physics.* Oxford University Press, 1989
9. Kacprzyk, Janusz: Fuzzy Sets and Fuzzy Logic. In: Shapiro, Stuart C. (ed.), *Encyclopedia of Artificial Intelligence.* Sec. Ed., vol. 1, John Willey & Sons, Inc., 537-542
10. Shapiro, Stuart C.: Artificial Intelligence. In: Shapiro, Stuart C. (ed.), *Encyclopedia of Artificial Intelligence.* Sec. Ed., vol. 1, John Willey & Sons, Inc., 54-57
11. Van Gulick, R.: Philosophical Questions. In: Shapiro, Stuart C. (ed.), *Encyclopedia of Artificial Intelligence.* Sec. Ed., vol. 2, John Willey & Sons, Inc., 1137-1147
12. Uspenski V. A., Semenov A. L.: *Theory of Algorithms: Main Discoveries and Applications*, Science Academic Publishing House, Moscow, 1987 (In Russian)

13. Pospelov, Germogen S.: *Artificial Intelligence - Base of Novel Information Technology*, Science Academic Publishing House, Moscow, 1988 (In Russian)
14. Kompanets, Leonid: Giving Birth to Biometrics of Asymmetrical Face. In: Kompanets Leonid (issue ed.): *Computing, Multimedia and Intelligent Techniques. Special Issue on Live Biometrics and Security*, vol. 1(2005), Czestochowa University of Technology, Poland, June 2005, 19-36
15. Kompanets, Leonid: Biometrics of Asymmetrical Face. Proc. 1st Intern. Conf. on *Biometric Authentication - ICBA 2004*. Hong Kong, China, July 2004. In.: Zhang, David, Jain, Anil K. (Eds.), Lecture Notes in Computer Science "*Biometric Authentication*", vol. 3072, Springer-Verlag 2004, 67-73
16. Kompanets, Leonid: Counterterrorism-Oriented Psychology and Biometrics Techniques: Part 2/2. Based on Brain Asymmetry, Eyes "Fingerprints", Face Asymmetry, and Person Psyche Type Identification Information Techniques. 7th Intern. Multi-Conf. on *Systemics, Cybernetics and Informatics - SCI'03*. July 29, 2003 – Orlando, Florida, USA. In: Callaos, Nagib, Kompanets Leonid, Takeno Junishi, Wej Huaqiang (Eds.). Pre-Proceedings of Symposium on *The Role of Academy in the War on Terrorism*, 18-21
17. Kompanets, Leonid, Bobulski Janusz, Wyrzykowski Roman: Pseudo-Entropy Similarity for Human Biometrics. Post-ECCV02 (7th European Conf. on Computer Vision 2002), Workshop on *Biometric Authentication*. 1st June 2002, Copenhagen, Denmark. In: Lecture Notes in Computer Science, vol. 2359, Springer-Verlag, 2002, 67-73

Apply the Particle Swarm Optimization to the Multidimensional Knapsack Problem

Min Kong and Peng Tian

Shanghai Jiaotong University, Shanghai 200052, China
kongmin@sjtu.edu.cn, ptian@sjtu.edu.cn

Abstract. This paper proposes a new heuristic approach based on the Particle Swarm Optimization (PSO) for the Multidimensional Knapsack Problem (MKP). Instead of the penalty function technique usually used to deal with the constrained problem, a heuristic repair operator utilizing problem-specific knowledge is incorporated into the modified algorithm. Computational results show that the new PSO based algorithm is capable of quickly obtaining high-quality solutions for problems of various characteristics.

1 Introduction

Particle Swarm Optimization (PSO) is a recently developed meta-heuristic for NP-hard optimization problems. Based on the simulation of both the movement of individual of bird flocks or fish schools and their collective behavior as a swarm, Kennedy and Eberhart[5] introduced the method of PSO in 1995. Applications to various nonlinear optimization problems have shown the success of PSO[7]. To solve constrained problems, PSO usually makes use of penalty function technique in order to reduce the constrained problem to an unconstrained problem by penalizing the objective function despite ill-conditioning[4,9].

This paper deals with the application of PSO in the field of combinatorial optimization (CO) problems, which is a quite rare field tackled by PSO. The constrained problem discussed in this paper is the well-known NP-hard CO problem, the multidimensional knapsack problem (MKP), which can be formulated as:

$$\text{maximize} \quad f = \sum_{j=1}^{n} p_j x_j \tag{1}$$

$$\text{subject to} \quad \sum_{j=1}^{n} r_{ij} x_j \leq b_i, i = 1, ..., m \tag{2}$$

$$x_j \in \{0, 1\}, j = 1, ..., n \tag{3}$$

Equation (1) describes the objective function for the MKP. Each of the m constraints described in condition (2) is called a knapsack constraint, so the MKP is also called the m-dimensional knapsack problem. Let $I = \{1, 2, ..., m\}$ and $J = \{1, 2, ..., n\}$, with $b_i > 0$ for all $i \in I$ and $r_{ij} \geq 0$ for all $i \in I, j \in J$, a well-stated MKP assumes that $p_j > 0$ and $r_{ij} \leq b_i < \sum_{j=1}^{n} r_{ij}$ for all $i \in I, j \in J$.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1140–1149, 2006.
© Springer-Verlag Berlin Heidelberg 2006

MKP can be regarded as a resource allocation problem of m resources and n objects. Each resource $i \in I$ has a burget b_i, each object $j \in J$ has a profit p_j and consumes r_{ij} of resource i. The problem is to maximize the profit within a limited budget.

MKP is one of the most intensively studied discrete programming problems, mainly because its simple structure which, on the one hand allows exploitation of a number of combinatorial properties and, on the other, more complex optimization problems to be solved through a series of knapsack-type subproblems. Meanwhile, many practical problems can be formulated as a MKP, such as the capital budgeting problem, allocating processors and databases in a distributed computer system, project selection and cargo loading, and cutting stock problems.

This paper utilizes the structure of the binary PSO[6] and combines this method with a problem-specific repair operator instead of the penalty function technique to avoid the violations to problem constraints. Experimental results show that the modified PSO is good at dealing with the specific CO problem.

This paper is organized as follows, the binary PSO algorithm to MKP is briefly introduced in Section 2. In section 3, the modified PSO algorithm applied to MKP is proposed, experimental results are shown in the following Section 4, and a short discussion is presented in Section 5. We end with some conclusions in Section 6.

2 The Binary PSO Model

2.1 Solution Representation and Fitness Function

In the binary PSO model[6], a potential solution to a problem is represented as a particle having binary coordinates $x = \{x_1, \ldots, x_n\}$, $x_j \in \{0, 1\}$ in a n-dimensional space as illustrated in Fig.1.

j	1	2	3	4	5	...	$n-1$	n
x_j	0	1	0	0	1	...	0	1

Fig. 1. Solution Struction of the Binary PSO

For the application to MKP, $x_j = 0$ means that object j is not selected, while $x_j = 1$ means that the object is selected. By this solution representation, we can see that such a solution might not be feasible for MKP. An infeasible solution is one for which at least one of the knapsack constraints is violated, i.e. $\sum_{j=1}^{n} r_{ij} x_j > b_i$ for some $i \in I$.

A penalty function technique is normally incorporated to solve the constrained problem in PSO. For the MKP problem, the fitness function is modified as:

$$f = \sum_{j=1}^{n} p_j x_j - \sum_{i=1}^{m} poslin\left(M_i\left(\sum_{j=1}^{n} r_{ij} x_j - b_i\right)\right) \tag{4}$$

where M_i are some big penalty parameters and *poslin* is a positive linear transform function, which is defined as:

$$poslin(s) = \begin{cases} s & s > 0 \\ 0 & s \leq 0 \end{cases} \tag{5}$$

2.2 The Standard PSO with Penalty Function Technique

The standard PSO with penalty function technique (denoted as PSO-P) does not take care of the feasibility of the solutions generated during the iterations. The knapsack constraints are totally manifested by the penalty function. By automatically moving to the coordinates with bigger objective function value during the iterations, PSO-P is able to find good solutions observing the knapsack constraints.

In PSO-P, a number of particles move stochastically among the binary solution space by flipping various numbers of bits. The position of each particle forms a solution to MKP, which can be represented as a n-dimensional binary string: $x_i = \{x_{i1}, \ldots, x_{in}\}$. The velocity of the movement of each particle is defined as the changes of probabilities that a bit will be in one state or the other, which is represented as $v_i = \{v_{i1}, \ldots, v_{in}\}$, where v_{id} represents the probability for particle i to select 1 at bit d. The velocity of each particle is determined by three kinds of information. One is its velocity value at last iteration, the second is the record of the position of its previous best performance, denoted as $p_i = \{p_{i1}, \ldots, p_{in}\}$, which represents the experience of the particle during the search, the last is the record of the position of the best performance among its topological neighborhood, denoted as $g_i = \{g_{i1}, \ldots, g_{in}\}$, which represents the social experiences of the particles during the search. In conclusion, the velocity for particle i at bit d can be summarized as:

$$v_{id}^{n+1} = v_{id}^{n} + \varphi_1 r_1 (p_{id}^{n} - x_{id}^{n}) + \varphi_2 r_2 (g_{id}^{n} - x_{id}^{n}) \tag{6}$$

where the superscript represents the number of iterations, φ_1 and φ_2 are two positive parameters, r_1 and r_2 are two randomly generated numbers uniformly distributed in $[0, 1]$. Normally, a bound limit V_{max} is incorporated to guarantee the value of the velocity be forced into a boundary $[-V_{max}, V_{max}]$ for the purpose of divergence avoidance.

To represent the velocity as the probability for selection of 1, a sigmoid transform function is incorporated to transform the velocity to the range of $(0, 1)$:

$$S(v_{id}) = \frac{1}{1 + \exp(-v_{id})} \tag{7}$$

The resulting change in position of a particle then is defined by the following rule:

$$\begin{aligned} \text{if } rand() < S(v_{id}) \text{ then } x_{id} &= 1 \\ \text{else } x_{id} &= 0 \end{aligned} \tag{8}$$

The algorithm skeleton of PSO-P is described in Fig.2. An iteration of PSO-P comprises evaluation of each particle using the modified fitness function of (4),

and calculations of the p_i and g_i for every particle, then the velocity is derived from (6) and particles move to their new positions according to (7) and (8). The iteration repeated until some termination condition is met, such as a maximum amount of cycles performed or a satisfied solution is found.

```
Procedure PSO-P
/*Initialization*/
    Input data
    Randomly generate initial particles positions and their velocities
    Parameter setting
/*Main Iteration Loop*/
While (end condition not met) do
    Solution evaluation according to (4)
    Calculate p_id and g_id for every particle
    Calculate velocities for every particle according to (6)
    Generate new particle positions according to (7) and (8)
End
```

Fig. 2. Algorithm Skeleton of PSO-P

3 The Modified PSO with Repair Operator

Although penalty function technique works well for most of the applications of PSO to the constrained problems, it contains some parameter setting problem. If the penalty parameter values are too high, the optimization algorithms usually get trapped in local minima. On the other hand, if penalty values are too low, they can hardly detect feasible optimal solutions. Furthermore, since the penalty function technique does not use the problem specific information, the final results are often not satisfied in dealing with CO problems.

This paper proposes a modified PSO with repair operator specially designed for MKP. The modified algorithm, denoted as PSO-R, is based on the structure of the binary PSO model described in the previous section, in combination with a problem-specific repair operator to guarantee feasible solutions. Fig.3 describes the pseudo code of PSO-R.

Instead of using the penalty function technique, PSO-R incorporates a repair operator to repair the solutions found by the particles. This idea comes from Chu and Beasley[1]. The general idea behind this method is described very briefly as follows.

The repair operator utilizes the notion of the pseudo-utility ratios derived from the surrogate duality approach. The surrogate relaxation problem of the MKP can be defined as:

$$\text{maximize } f = \sum_{j=1}^{n} p_j x_j \tag{9}$$

```
Procedure PSO-R
/*Initialization*/
    Input data
    Calculate surrogate multipliers and pseudo-utility
    Sort and renumber data according to decreasing order of pseudo-utility
    Generate initial particles positions and their velocities
    Parameter setting
/*Main Iteration Loop*/
While (end condition not met) do
    Solution repair
    Solution evaluation
    Calculate p_id and g_id for every particle
    Calculate velocities for every particle
    Generate new particle positions
End
```

Fig. 3. Algorithm Skeleton of PSO-R

$$\text{subject to } \sum_{j=1}^{n}(\sum_{i=1}^{m}\omega_i r_{ij})x_j \leq \sum_{i=1}^{m}\omega_i b_i \tag{10}$$

$$x_j \in \{0,1\}, j = 1,2,...,n \tag{11}$$

where $\omega = \{\omega_1, \ldots, \omega_m\}$ is a set of surrogate multipliers (or weights) of some positive real numbers. One of the simplest methods to obtain reasonably good surrogate weights is to solve the LP relaxation of the original MKP and to use the values of the dual variables as the weights. In other words, ω_i is set equal to the shadow price of the ith constraint in the LP relaxation of the MKP.

After calculating the surrogate weights, the pseudo-utility is then defined as:

$$u_j = \frac{p_j}{\sum_{i=1}^{m}\omega_i r_{ij}} \tag{12}$$

The repair operator consists of two phases that is based on the value of u_j. The first phase, which is called DROP phase, examines each bit of the solution in increasing order of u_j and changes the value of the bit from one to zero if feasibility is violated. The second phase, which is called ADD phase, reverses the process by examining each bit in decreasing order of u_j and changes the value of the bit from zero to one as long as feasibility is not violated. To achieve an efficient implementation of the repair operator, at the initialization step, we sort and renumber variables of the original MKP problem according to the decreasing order of their u_j's. The pseudo-code for the repair operator is given in fig.4.

Although the repair operator takes some extra time at each iteration, from the description of the procedure of PSO-R, we can see that the computational complexity of the repair operator, as well as each iteration of PSO-R, is $\mathcal{O}(mn)$,

Repair Operator for PSO-R
Let: R_i = the accumulated resources of constraint i in S
Initialize $R_i = \sum_{j=1}^{n} r_{ij} S[j], \forall i \in I$
$j = n$
While ($R_i > b_i$, for any $i \in I$) do /* DROP phase */
if $S[j] = 1$ then
$S[j] = 0; R_i = R_i - r_{ij}, \forall i \in I$
endif
$j = j - 1$;
endwhile
for $j = 1$ to n do /* ADD phase */
if $S[j] = 0$ and $R_i + r_{ij} < b_i, \forall i \in I$ then
$S[j] = 1; R_i = R_i + r_{ij}, \forall i \in I$;
endif
end for

Fig. 4. Pseudo Code of the Repair Operator

which is the same as that of PSO-P. So, PSO-R takes just a little computational time over PSO-P.

4 Experimental Results

Since we have not found any literature concerning the PSO algorithm applied to the MKP problems, we select some benchmarks of MKP from OR-Library to test PSO-R, and we compare the results of PSO-R with that of PSO-P.

To make a fair comparison, we set the same parameter values for both the PSO-R and PSO-P: $\varphi_1 = \varphi_2 = 2$, $v_{max} = 2$, the number of particles is set equal to the number of objects of the problem, and we use the ring topology as the neighborhood structure with number of neighbors set to 2. These parameter settings are regarded as optimal to the standard PSO algorithms [7].

Fig.5 describes the typical performance of PSO-P and PSO-R on a MKP instance with 50 objects and 5 resource constraints. The x-axis describes the number of executed cycles, while the y-axis describes the best fitness value that is averaged over 30 runs. From this diagram, we can see clearly that PSO-R outperforms PSO-P with quick convergence to satisfied solution, and with better solution quality.

Tab.1 shows the experimental results of PSO-R and PSO-P over 7 benchmarks named mknap1 in OR-Library. All the tests are ran with 500 executed cycles. The first column indicates the problem index, the next two columns describe the problem dimension, where n is the number of objects and m is the number of constraints. The next column is the best-known solutions from OR-Library. The final 4 columns report the best and average solutions over 30 runs of PSO-P and PSO-R respectively. For all the 7 instances of mknap1 that we tested, both PSO-R and PSO-P are able to find good solutions, but PSO-R finds better solutions than that of PSO-P as the size of the problem increases.

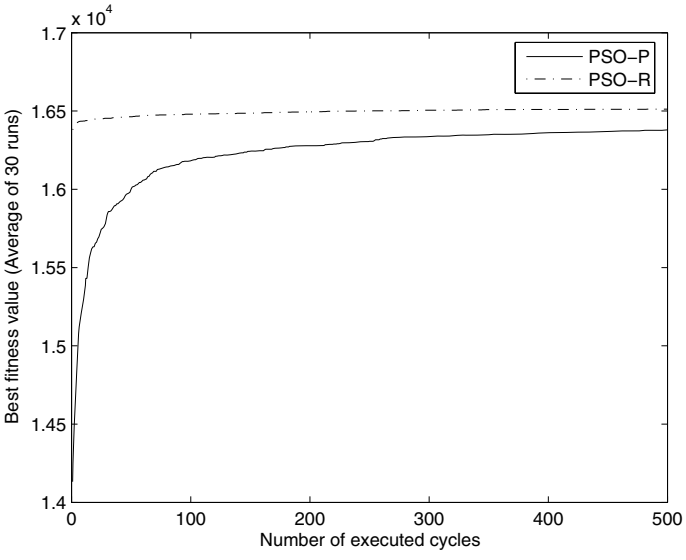


Fig. 5. Typical Performance of PSO-R and PSO-P

Table 1. Experimental Results of mknap1

N^o	n	m	Best known	PSO-P		PSO-R	
				Best	Avg.	Best	Avg.
1	6	10	3800	3800	3800	3800	3800
2	10	10	8706.1	8706.1	8570.7	8706.1	8706.1
3	15	10	4015	4015	4014.7	4015	4015
4	20	10	6120	6120	6118	6120	6119.3
5	28	10	12400	12400	12394	12400	12395
6	39	5	10618	10618	10572	10618	10592
7	50	5	16537	16491	16389	16537	16510

We also compare the PSO-R with PSO-P on some bigger size MKP instances in OR-Library, which are considered to be rather difficult for optimization approaches. The tested sets are 5.100 and 10.100, which has 5 constraints, 100 objects and 10 constraints, 100 objects respectively. We test first 5 instances of each set with maximum number of cycles of 2000, and Tab.2 reports the test results.

The first column of Tab.2 indicates the instance name, the second column is the best-known solutions from the OR-Library, and the next 4 columns record the best and average solutions over 30 runs of PSO-P and PSO-R respectively.

From Tab.2 we can see that PSO-R clearly outperforms PSO-P in all the tested instances. While PSO-P meets some difficulties in dealing with large size MKP problems, PSO-R is still able to find good solutions. Actually, PSO-R has found 5 best solutions out of 10 instances.

Table 2. Experimental Results of 5.100 and 10.100

Instance Name	Best known	PSO-P		PSO-R	
		Best	Avg.	Best	Avg.
5.100.00	24381	22525	22013	24381	24356
5.100.01	24274	22244	21719	24258	24036
5.100.02	23551	21822	21050	23551	23523
5.100.03	23534	22057	21413	23527	23481
5.100.04	23991	22167	21677	23966	23966
10.100.00	23064	20895	20458	23057	23050
10.100.01	22801	20663	20089	22781	22668
10.100.02	22131	20058	19582	22131	22029
10.100.03	22772	20908	20446	22772	22733
10.100.04	22751	20488	20025	22751	22632

5 Discussion

5.1 PSO Applied to CO

PSO has gained reputation in the field of function optimization problems, but few encouraging applications are recorded in the field of combinatorial optimization problems. The main reason is that the PSO is famous for its robustness regardless of the type of the fitness function, most of PSO rarely use the characteristic information of the problem instance, which is quite critical in tacking the combinatorial optimization problems.

Up to our knowledge, there has been no literature available concerning the application of PSO to MKP. The main purpose of this paper is to propose that the PSO technique is also effective in dealing with combinatorial optimization problems, rather than showing that PSO-R is the best algorithm overcoming MKP. The algorithm methodology, as well as the parameter setting in PSO-R, is quite normal method directly get from results of other PSO literature, however, the results presented in previous section are quite promising, indicating the potential of PSO in dealing with such kind of combinatorial optimization problems.

5.2 Role of the Repair Operator

The repair operator incorporated in PSO-R plays a critical role in quickly finding good solutions, this lies in two sides:

First, the repair operator itself improves the solution quality. Although the repair operator alone acts as a problem-specific greedy search method that can only find rather poor solutions, in cooperation with standard PSO, the repair operator acts as a local search to the solutions found by the standard PSO, which greatly improves the solution quality.

Secondly, the repair operator acts as a filter that makes all the solutions generated in the iteration being transferred to the feasible solution domain,

which makes the algorithm search around the quite promising area comparing to the normal penalty incorporated method.

The utility of a repair operator is important in applying the PSO to MKP. A good repair operator is critical in quick convergence. To find a good repair operator in other combinatorial optimization problems will be helpful for the PSO implementations.

6 Conclusions

This paper proposes a first implementation of Particle Swarm Optimization to the well-known multidimensional knapsack problem. Instead of the incorporation of the penalty function technique usually used for the constrained problems, we utilize a problem-specific repair operator to guarantee feasible solutions at each iteration cycle.

Computational results show that the modified PSO algorithm outperforms the standard PSO in MKP problems of various characteristics. Although the computational results of the modified algorithm are still not as good as the state-of-art algorithm proposed by Vasquez and Hao [11], the fact of its simplicity, quickness and that it is able to deal with large size MKP problems indicates its potential in dealing with such combinatorial optimization problems.

The repair operator technique plays a critical role in finding better solutions quickly. The procedure of our modified PSO algorithm indicates that this technique can be implemented in other combinatorial optimization problems. Further works will be on applying this technique in constrained integer programming with focus on how to apply the problem-specific information into some repair operators.

References

1. P.C. Chu and J.E.Beasley. A Genetic Algorithm for the Multidimensional Knapsack Problem. Journal of Heuristics, 4: 63-86 (1998)
2. Gavish, B. and H. Pirkuo. Efficient Algorithms for Solving Multiconstraint zero-One Knapsack Problems to Optimality, Mathematical Programming 31, 78-105. 1985
3. Gilmore, P.C. and R.E. Gomory. The Theory and Computation of Knapsack Functions, Operations Research 14, 1045-1075. 1966
4. X. Hu, R. Eberhart. Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization. Proceedings of the Sixth World Multiconference on Systemics, Cybernetics and Informatics 2002 (SCI 2002), Orlando, USA. 2002 [3]
5. J. Kennedy, R.C. Eberhart. Particle Swarm Optimization. Proceedings of the International Conference on Neural Networks, Perth, Australiz, 1995, IEEE, Piscataway, 1995, pp, 1942-1948.
6. J. Kennedy, R.C. Eberhart. A Discrete Binary Version of the Particle Swarm Algorithm. In: Proc. 1997 Conf. On systems, Man, and Cybernetics. Piscataway, NJ: IEEE Service Center, 1997, 4104-4109.

7. J. Kennedy, R.C. Eberhart and Y. Shi. Swarm Intelligence. Morgan Kaufmann Publishers. 2001.
8. S. Martello, P. Toth. Knapsack Problems, Algorithms and Computer Implementations. John Wiley & Sons Ltd. 1990.
9. K.E. Parsopoulos, M.N. Vrahatis. Particle Swarm Optimization Method for Constrained Optimization ProblemsP. Sincak, J. Vascak, V. Kvasnicka, J. Pospichal (eds.), Intelligent Technologies - Theory and Applications: New Trends in Intelligent Technologies, pp. 214-220, IOS Press (Frontiers in Artificial Intelligence and Applications series, Vol. 76), 2002, ISBN: 1-58603-256-9.
10. Shih, W. A Branch and Bound method for the Multiconstraint Zero-One Knapsack Problem, Journal of the Operational Research Society 30, 369-378 1979
11. M. Vasquez and J.K Hao, A Hybrid Approach for the 0-1 Multidimensional Knapsack Problem. In Proceedings of the 17^{th} International Joint Conference on Artificial Intelligence, 2001, (pp.328-333). San Francisco, Morgan Kaufmann.

Self-stabilizing Algorithms for Graph Coloring with Improved Performance Guarantees

Adrian Kosowski and Łukasz Kuszner

Gdańsk University of Technology, Poland
Department of Algorithms and System Modeling
{kosowski, kuszner}@sphere.pl

Abstract. In the self-stabilizing model we consider a connected system of autonomous asynchronous nodes, each of which has only local information about the system. Regardless of the initial state, the system must achieve a desirable global state by executing a set of rules assigned to each node. The paper deals with the construction of a solution to graph coloring in this model, a problem motivated by code assignment in wireless networks.

A new method based on spanning trees is applied to give the first (to our knowledge) self-stabilizing algorithms working in a polynomial number of moves, which color bipartite graphs with exactly two colors. The complexity and performance characteristics of the presented algorithms are discussed for different graph classes.

1 Introduction

Self-stabilizing algorithms are a fundamental branch of fault-tolerant computing, first introduced by Dijkstra [4] in 1974. The resilience of the system guarantees that the system will reach a desirable state even after a period of malfunction or an unexpected change of topology. These properties render self-stabilizing algorithms feasible in contexts where ordinary distributed algorithms prove insufficiently stable, especially when applied in ad-hoc wireless networks [11,15] and in state-of-the-art systems of nano-electrical sensors placed in a hostile environment [3]. The most relevant self-stabilizing algorithms deal with assigning codes or values to nodes and the links between them, thus facilitating communication and role division of the respective units. In this paper we focus our attention on the fundamental problem of coloring the system graph in such a way that neighboring nodes receive different colors and the number of colors used is as small as possible. This process of graph coloring in a distributed setting has numerous applications, including such areas as code assignment for wireless networks [1,11,13].

The model of execution. In all considerations it is assumed that the self-stabilizing system consists of nodes connected by communication channels. Each node maintains variables which determine its *local state*. The *global state* of the system is the union of all local states of its nodes. Thus we model a system

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1150–1159, 2006.
© Springer-Verlag Berlin Heidelberg 2006

by a connected graph $G = (V, E)$, where vertex set V corresponds to system nodes and the set of edges E denotes communication links between them. The algorithm for each vertex v is given as a set of rules of the form **if** $p(v)$ **then** A, where $p(v)$ is a predicate over local states of v and its neighbors, and A is an action changing a local state of v (a *move* of v). A vertex v becomes *active* when $p(v)$ is true, otherwise v is *stable*. The execution of the algorithm is controlled by a *scheduler* which allows some non-empty subset of active vertices to perform a simultaneous move defined by the rules for the respective nodes; this is referred to as a single *step*. If all vertices of the graph are stable, we say that the system is *stable* and the execution of the algorithm is complete. A *self-stabilizing system* is one which is able to achieve a legitimate global state starting from any possible global state [4].

The time complexity of such self-stabilizing algorithms is expressed in terms of the number of steps of the algorithm, or the number of moves performed by respective nodes. In this paper the total number of moves performed by all nodes is treated as the primary measure. This value is closely related to (and never less than) the number of steps of the system, and may moreover be regarded as the "energy cost" of performing the stabilization process.

Problem definition and related work. A self-stabilizing algorithm is said to perform *graph coloring* if any stable state reached by the system may be interpreted as an assignment of non-negative integer color values $c(v)$ to all system nodes $v \in V$ such that $c(v_1) \neq c(v_2)$ if $\{v_1, v_2\} \in E$. The goal of optimization is minimizing the largest color value used.

Self-stabilizing graph coloring algorithms have been intensively studied in literature. In 1993 Ghosh and Karaata [6] presented an algorithm for coloring planar graphs with at most 6 colors, assuming that all vertices have unique identifiers. This result was later improved to allow operation with bounded variable values and without identifiers in [12] and generalized for a wider class of graphs in [8]. Also in 1993, Sur and Srimani [16] gave an algorithm for exact coloring of bipartite graphs. All of the discussed papers prove the finite stabilization time of the algorithm, but leave bounds on time complexity as open problems. Later on three algorithms with improved time constraints for arbitrary graphs were presented in [9], based on a greedy assignment technique. More recently, a linear-time algorithm for coloring arbitrary graphs with $\Delta + 1$ colors (where Δ denotes the maximum vertex degree) was given in [10].

Our results. In this paper we present new graph coloring algorithms which stabilize in a polynomial number of moves and find optimal colorings for bipartite (2-colorable) graphs and cacti, thus improving the results of [16]. To achieve the results we a use recent algorithm for spanning tree construction [14].

We give different versions of the graph coloring algorithm. Depending on the configuration of the scheduler, the algorithm either operates without vertex

identifiers (for a system with a *central daemon*, i.e. when the scheduler allows the move of at most one vertex at a time, Section 3) or requires unique vertex identifiers (for a system with a *distributed daemon*, i.e. when no additional assumptions are made about the scheduler, Section 4). For these cases the number of moves performed by the algorithms is bounded by $O(mn^3 \operatorname{diam}(G))$ and $O(mn^3 \Delta \operatorname{diam}(G))$ respectively, where $\operatorname{diam}(G) < n$ is the distance between the furthest pair of nodes of the system. Finally, in Section 5 we analyze the performance of the presented algorithms color for arbitrary graphs, stating that they use not more than 2Δ colors, and that this bound is tight even for series-parallel graphs.

2 Preliminaries: Local State Definition and Assumptions

Let $G = (V, E)$ be the system graph with vertex set V and edge set E. By $n = |V|$ and $m = |E|$ we denote the number of vertices and the number of edges, respectively. In addition, let $N(v) = \{u : (u, v) \in E\}$ be the *open neighborhood* of vertex $v \in V$, and let $\deg(v) = |N(v)|$ be the *degree* of v. Each vertex has constant read-only access to the local states of all vertices in its neighborhood throughout the operation of the algorithm. When performing a move, a vertex may write to the variables of its local state. The entire move of a vertex is seen as an atomic operation by its neighbors.

In our approach we provide a *semi-uniform* algorithm, which means that exactly one of the nodes of set V, called a *root* and denoted r, needs to be distinguished and operate using a different set of rules than the other vertices. Each node has two local variables f and c, both non-negative integers, which reflect the two parts of coloring algorithm. State variable c is simply the sought color of a given vertex.

Variable f is used during the construction of a spanning tree of graph G. A *spanning tree* $T = (V, E')$ of $G = (V, E)$ is a subgraph of G consisting of the same set of nodes V, but only a subset $E' \subseteq E$ of edges such that there exists exactly one path between every pair of nodes in T. In particular, this means that each vertex $v \in V$ other than the root r has exactly one neighbor on the path connecting v and r in T, known as the *parent* of v in T. In this context, the interpretation of variable f is as follows. Consider vertex v and let us choose u such that $f(u) = \min_{w \in N(v)} f(w)$ and u is the first [1] vertex among the neighbors of v with such a property. If $f(u) < f(v)$ then we say that u is the parent of v and denote $p(v) = u$. Other case we say that a parent vertex does not exist for v in the current system configuration and denote it by $p(v) = \text{null}$. If all vertices save the root have parent vertices, then the parenthood relation defines the sought spanning tree T of G. For the distinguished vertex we set permanently $f(r) := 0$ and $c(r) := 0$.

[1] To be able to say "first" we must assume that the neighbors are somehow ordered. This is not a strong assumption as long as a node is able to distinguish between its neighbors. For example, if the neighbors of v are stored in the form of a list, the order can be given according to the list sequence.

3 A Graph Coloring Algorithm Under a Central Daemon Optimal for Bipartite Graphs

The concept of the graph coloring algorithm presented in this section is based on a simple observation. We use the *parity function* $A : \mathbb{N} \cup \{0\} \to \{0, 1\}$, defined as $A(c) \equiv c \mod 2$, to partition the space of colors available for each vertex into two subsets — the set of *even colors* $\{c \in \mathbb{N} \cup \{0\} : A(c) = 0\}$ and the set of *odd colors* $\{c \in \mathbb{N} \cup \{0\} : A(c) = 1\}$.

The constructed algorithm will attempt to assign to each vertex the smallest possible odd color if the color of its parent is even and the smallest possible even color when the color of its parent is odd. For a given vertex v such a color is denoted as $\gamma(v)$:

$$\gamma(v) = \min\{k \in \mathbb{N} \cup \{0\} \ : \ A(k) \neq A(c(p(v))) \wedge \forall_{u \in N(v)} k \neq c(u)\}.$$

Observe that $\gamma(v)$ is properly defined iff the parent of v exists; otherwise, we assume $\gamma(v) = c(v)$.

We can now write the graph coloring algorithm for the case when the scheduler selects exactly one active node at a time to make a move, which is often referred to in literature as self-stabilization under a *central daemon*. At first glance such a model can be seen as very strong, however it is equivalent to one where only local mutual exclusion of neighboring nodes is guaranteed. Moreover, there exist protocols to convert algorithms designed for the central daemon model to weaker ones [2,5].

Algorithm 1. Graph coloring under a central daemon

`F:` **if** $v \neq r \wedge f(v) \leq \min_{u \in N(v)} f(u)$
then $f(v) = \max_{u \in N(v)} f(u) + 1$

`C:` **if** $p(v) \neq \text{null} \wedge c(v) \neq \gamma(v)$
then $c(v) := \gamma(v)$

Rule `F` is the same as that used for constructing a spanning tree in Algorithm 1 [14] and is not affected by other rules, so using results from [14] at most $n \operatorname{diam}(G)$ moves using rule `F` are possible. Now we will establish a bound on the number of moves using rule `C` in between two consecutive moves using rule `F`.

Let us consider an arbitrarily chosen vertex v. We can distinguish between two types of moves according to rule `C`, namely: moves changing the parity of $c(v)$ which we call *alternating moves*, and other moves referred to as *non-alternating*.

Lemma 1. *The number of alternating moves is at most n^2 in between two consecutive moves performed according to rule `F`.*

Proof. Let $T = (V, E')$ be the forest in G induced the by parent relation p in such a way that $u, v \in E'$ iff $u = p(v)$ or $v = p(u)$. Observe that in each

component of T there is exactly one vertex u such that $p(u) = \text{null}$. Let r_v be such a vertex that $p(r_v) = \text{null}$ and both r_v and v belong to the same component of T, and let $[u, v]_T$ denote the only possible path from u to v in T, say $[u, v] = (u = u_0, u_1, \ldots, u_k = v)$. Let us consider the following function:

$$S_T(v) = \begin{cases} S_T(p(v)) + 1, & p(v) \neq \text{null} \wedge A(c(p(v)) \neq A(c(v)) \\ S_T(p(v)), & p(v) \neq \text{null} \wedge A(c(p(v)) = A(c(v)) \\ 0, & p(v) = \text{null} \end{cases}$$

Intuitively, $S_T(v)$ can be considered as the number of edges $\{w, w'\}$ in the path $[r_v, v]_T$ for which color variable c changes parity while moving from w to w'. Obviously $0 \leq S_T(v) < n$. Observe that $S_T(v)$ does not depend on $f(x)$ for $x \notin [r_v, v]_T$. Now, let us consider the influence of an alternating move `C` performed by $x \in [r_v, v]_T$ on $S_T(v)$. As r_v cannot make move `C`, thus $x \neq r_v$. There are two possibilities. First, if $v = x$, then $S_T(v)$ increases by 1. Secondly, if $v \neq x$, then $S_T(v)$ does not change or increases by 2.

Consequently, since the initial value of $S_T(v)$ is never smaller than its final value by more than n and the value $S_T(v)$ increases each time when v makes an alternating move thus v cannot perform more than n alternating moves. The total number of moves performed is thus not more than n^2. □

Theorem 2. *Algorithm 1 stabilizes in $O(mn^3\ diam(G))$ moves under a central daemon.*

Proof. We can now find a bound on the number of non-alternating moves in between two consecutive alternating moves. Again, non-alternating moves can be of two types: those increasing the value $c(v)$ (called *increasing moves*) and all other moves (*decreasing moves*). In between two alternating moves vertex v can perform at most one increasing move (as the first non-alternating move only). All following moves are decreasing moves. After the first move the value $c(v)$ belongs to the set of the smallest $\deg(v)$ color values of the given parity; the number of subsequent decreasing moves is obviously not more than $\deg(v) - 1$. Consequently, between two alternating moves each vertex performs not more than $\deg(v)$ non-alternating moves, thus the number of all non-alternating moves between alternating moves is not more than $2m$.

Finally, using Theorem 2 and Lemma 1 we have that the algorithm performs at most $2m$ non-alternating moves for each of at most n^2 alternating moves of rule `C`, for each of at most $n\,\text{diam}(G)$ moves of rule `F`. By multiplying these values we obtain an upper bound of $O(mn^3\,\text{diam}(G))$ moves, which completes the proof. □

Bearing in mind Theorem 2, we know that the system stabilizes, thus let consider the result of the process.

Theorem 3. *Algorithm 1 finds a legal coloring of graph G.*

Proof. Suppose that the system is stable, so each node is stable. Hence, according to rule `C`, the colors of any two neighboring vertices are different. □

Theorem 4. *Algorithm 1 finds a legal coloring of an arbitrary graph and uses not more than two colors when applied to bipartite graph G.*

Proof. Suppose that the system is stable and consider obtained coloring. Legality is obvious. It remains to show that if G is bipartite no more than two colors has been used.

Now, suppose to the contrary, that G is bipartite and Algorithm 1 stabilizes using more than two colors. Consider a vertex v arbitrarily chosen among of vertices with the largest value of color c. Observe that for each neighbor $u \in N(v)$ the parity of $c(v)$ and $c(u)$ is different (otherwise G would have to contain an odd cycle). Thus by property 2 of function A we have $c(v) < 2$, since otherwise v would be active due to rule `C`, a contradiction. □

3.1 Example

In Figure 1 each of the pictures illustrates the states of the vertices in successive moves. The root vertex is placed in the middle of each picture and active vertices are marked with bold circles. The state value f of each vertex is given close to the circle, while the color value is denoted inside it.

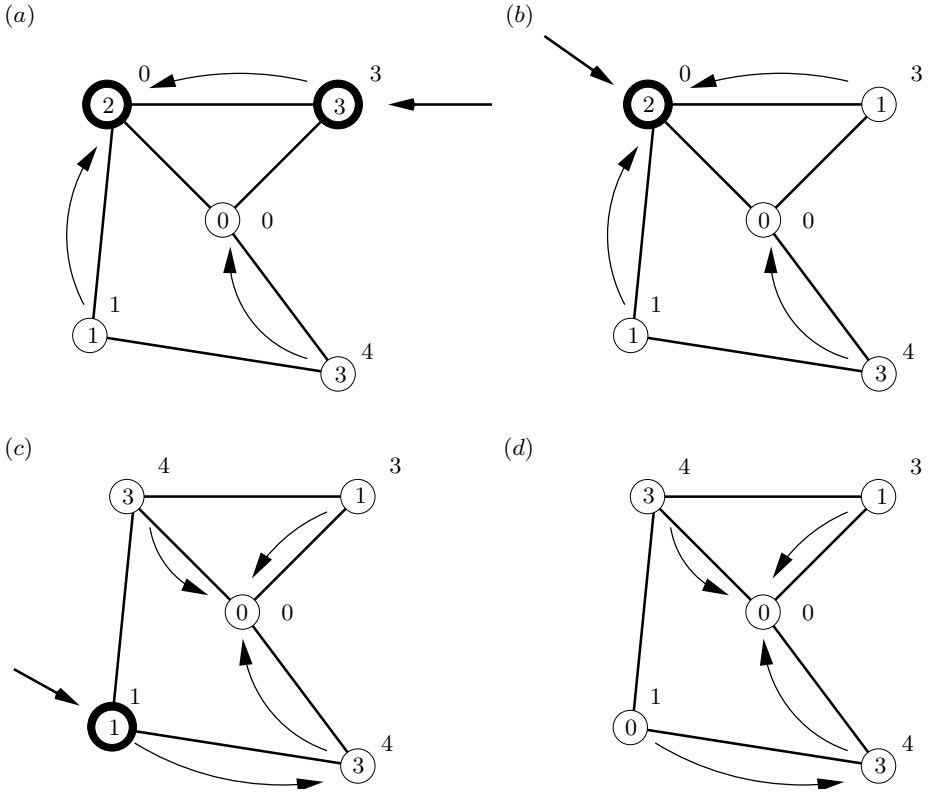


Fig. 1. Illustration of a process of Algorithm 1 (consult Subsection 3.1)

Additional arcs show the parenthood relation between vertices and a straight pointer indicates the vertex to perform the next move. The algorithm computes a coloring in three steps and the result is visible in Figure 1(d). It can be observed that in the case of non-bipartite graphs, as in the example, we cannot expect a *Grundy coloring* [7] of G, i.e. such a coloring, that no single vertex may have its color value decreased without affecting the color other vertices.

4 A Graph Coloring Algorithm Under a Distributed Daemon Optimal for Bipartite Graphs

When allowing adjacent nodes to perform moves in parallel we introduce vertex identifiers, i.e. we assume that each vertex $v \in V$ is assigned an identifier $Id(v)$ which is fixed throughout the operation of the algorithm and known to vertex v and its neighbors. Additionally, in order to allow mutual exclusion between moves of neighboring vertices, we add a new local variable $s(v) \in \{on, off\}$ to the state of each vertex, which may be regarded as a generalization of the concept of a semaphore.

Algorithm 2. Graph coloring under a distributed daemon

`F:` **if** $v \neq r \wedge f(v) \leq \min_{u \in N(v)} f(u)$
then $f(v) = \max_{u \in N(v)} f(u) + 1$

`C1:` **if** $p(v) \neq \text{null} \wedge c(v) \neq \gamma(v)$
then $s(v) := on$

`C2:` **if** $s(v) = on \wedge Id(v) < \min_{u \in N(v)} \{Id(u) \mid s(u) = on\}$
then $c(v) := \gamma(v)$; $s(v) := off$

Theorem 5. *Algorithm 2 determines a legal coloring of graph G in at most $O(mn^3 \Delta \operatorname{diam}(G))$ moves under a distributed daemon.*

Proof. By analogy to the correctness proof of Algorithm 1 we bound the number of moves using rule `F` by $n \log n$. Now, observe that rules `C1` and `C2` from Algorithm 2 are derived from rule `C` of Algorithm 1, and may be treated as two consecutive stages of its execution. First, in rule `C1` any vertex v whose color value is incorrect with respect to any legal state of the algorithm signals its intention of performing a color change by setting $s(v) := on$. The scheduler may then select a subset of vertices with a set value $s(v)$ to perform rule `C2`, which results in the desired change of color value $c(v)$ and unsets $s(v)$. Observe that the additional clause $Id(v) < \min\{Id(u) \mid u \in N(v) \wedge c(u) = c(v)\}$ in the condition of rule `C2` serves to prevent parallel moves of neighboring vertices using this rule. Thus, the proposed algorithm behaves similarly to its counterpart with a central daemon (Algorithm 1) and we can repeat the reasoning from Lemma 1 and Theorem 2. Observe that the additional Δ-factor in the stabilization time

of algorithm Algorithm 2 results from the fact that certain vertices may have a set value $s(v)$ and perform rule `C2` even when their color value $c(v)$ is legitimate (since the value $\gamma(v)$ may have changed after the prior execution of rule `C1` for vertex v due to a move made by one of its neighbors). □

5 Final Remarks

It is important to observe that Algorithms 1 and 2 can be applied to color not only bipartite graphs, and the established bounds on the number of performed moves do in fact hold for arbitrary graphs G. Furthermore, careful analysis of the presented algorithms leads to the following statement.

Corollary 6. *Any graph G is colored by Algorithm 1 in at most $O(mn^3 \operatorname{diam}(G))$ moves, and by Algorithm 2 in at most $O(mn^3\Delta \operatorname{diam}(G))$ moves, using not more than 2Δ colors.*

By comparison, general-purpose self-stabilizing algorithms for graph coloring (e.g. [10]) have been shown to use not more than $\Delta + 1$ colors for any graph; however, it is possible to construct bipartite graphs which are sometimes colored by exactly $\Delta + 1$ colors. Obviously, no similar pathological cases exist for Algorithms 1 and 2.

In general, Algorithms 1 and 2 do not color tripartite graphs optimally and it is possible to construct examples of 3-colorable series-parallel graphs for which Algorithms 1 and 2 may stabilize in a state using 2Δ colors.

Property 7. *There exists a family of series-parallel graphs $\{G_k\}$ such that for each G_k we have $\Delta = k$, and Algorithms 1 and 2 may lead to a coloring of G_k using 2Δ colors.*

Proof. Let $\{T_k\}$ be a family of trees with one selected vertex $s(T_k)$ in each, defined constructively by the following procedure:

1. Tree T_0 consists of one vertex v; $s(T_0) = v$.
2. Tree T_{k+1} is formed by adding a new vertex v to the disjoint union of trees $\bigcup_{i=0}^{k} T_i$, and inserting additional edges $\{v, s(T_0)\}, \ldots, \{v, s(T_k)\}$ to obtain a tree; $s(T_{k+1}) = v$.

Now, series-parallel graph G_k is obtained by adding the root vertex r to tree T_k, and inserting edges connecting r with all the vertices of T_k. Observe that in some executions of Algorithms 1 and 2 the spanning tree of G_k will be the star with center r, and that the algorithm may reach a legal state in which vertex $s(T_k)$ uses color $2k - 1$, thus leading to a 2Δ-coloring of G_k. □

However, it has to be noted that graphs known as *cacti* (connected graphs such that each edge belongs to at most one cycle) constitute a highly relevant class of tripartite graphs for which Algorithms 1 and 2 always determine a nearly-optimal coloring.

Corollary 8. *A cactus G is always colored by Algorithms 1 and 2 using at most 4 colors.*

Proof. It has been shown that both algorithms stabilize, so it is enough to show that the number of colors used for cacti graphs is at most 4. Let G be a cactus and T be a spanning tree computed by the algorithm. Consider an arbitrarily chosen non-root vertex $v \in G$. Let $N_A(v) = \{u \in N(v) \,|\, \mathrm{A(u)}{=}\mathrm{A(v)}\}$, clearly $c(v) < 2(|N_A(v)| + 1)$. Suppose that $u \in N_A(v)$, then $u \neq p(v)$ and $v \neq p(u)$.

Moreover u does not belong to path $[v, r]_T$, otherwise we would have a path $v = v_0, v_1, \ldots, v_k = u$, where $v_i = p(v_{i-1})$ for $i = 1, 2, \ldots, k$ and $f(v_i) < f(v_{i-1}) < \ldots < f(v_0)$, a contradiction with $p(v)$ being a neighbor of v such that $f(p(v)) = \min\{f(w) \,|\, w \in N(v)\}$. Similarly, v does not belong to path $[u, r]_T$.

Thus, we have a cycle in G involving u,v,$p(v)$, but G is a cactus, thus the edge $\{v, p(v)\}$ is involved in at most one cycle, so there is at most one vertex in $N_A(v)$ and consequently $c(v) < 4$. Remembering that $c(r)$ is equal to 0 and v was an arbitrarily chosen vertex, the proof is complete. □

It is difficult to say whether it is possible to construct a single polynomial-time self-stabilizing algorithm which optimally or near-optimally colors significantly wider graph classes than Algorithms 1 and 2. Due to the inapproximability of the 3-coloring problem in the classical model, no such algorithm may be constructed for general tripartite graphs (unless $P = NP$). However, for the relatively narrow 3-colorable classes of outerplanar graphs and series-parallel graphs the question remains open.

From the practical point of view, the algorithms presented in this paper combine the ability to take advantage of bounds on vertex degree Δ and of the 2-colorability of the system graph or its large portions. Colorings obtained for bipartite graphs, cacti and many other subclasses of planar graphs use a sufficiently small number of colors to merit application in code assignment problems in real-world conditions.

References

1. R. Battiti, A.A. Bertossi, M.A. Bonucceli, *Assigning code in wireless networks: bounds and scaling properties*, Wireless Networks, **5**, 195–209, 1999.
2. J. Beauquier, A. Kumar Datta, M. Gradinariu, F. Magniette, *Self-Stabilizing Local Mutual Exclusion and Definition Refinement*, Chicago Journal of Theoretical Computer Science, 2002.
3. G.C. Chachis, *If it walks like a duck: nanosensor threat assessment.* Proc. SPIE **5090**, 341–347, 2003.
4. E.W. Dijkstra, *Self-stabilizing systems in spite of distributed control*, Communications of the ACM **17**, 643–644, 1974.
5. S. Dolev, *Self-stabilization*, MIT Press, 2000.
6. S. Ghosh, M.H. Karaata, *A Self-Stabilizing Algorithm for Coloring Planar Graphs*, Distributed Computing **7**, 55–59, 1993.
7. Grundy, P. M.: Mathematics and games. Eureka **2** (1939), 6–8.
8. W. Goddard, S.T. Hedetniemi, D.P. Jacobs, P.K. Srimani, *Fault Tolerant Algorithms for Orderings and Colorings*, Proc. IPDPS'04, 2004.

9. M. Gradinariu, S. Tixeuil, *Self-stabilizing Vertex Coloring of Arbitrary Graphs*, Proc. OPODIS'00, 55–70, 2000.
10. S.T. Hedetniemi, D.P. Jacobs, P.K. Srimani, *Linear time self-stabilizing colorings*, Information Processing Letters **87**, 251–255, 2003.
11. L. Hu, *Distributed code assignments for CDMA Packet Radio Network*, IEEE/ACM Transactions on Networking **1**, 668–677, 1993.
12. S.T. Huang, S.S. Hung, C.H. Tzeng, *Self-stabilizing coloration in anonymous planar networks*, Information Processing Letters **95**, 307–312, 2005.
13. K.W. Hung T.S. Yum *An efficient code assignment algorithm for multihop spread spectrum packet radio networks*, Proc. GLOBECOM'90, 271–274, 1990.
14. A. Kosowski, Ł. Kuszner, *A self-stabilizing algorithm for finding a spanning tree in a polynomial number of moves*, Proc. PPAM'05, LNCS **3911** (to appear), 2006.
15. J. Li, C. Blake, D.S. De Couto, H.I. Lee, R. Morris, *Capacity of Ad Hoc wireless networks*. Proc. MobiCom'01. ACM Press, 61–69, 2001.
16. S. Sur, P.K. Srimani, *A self-stabilizing algorithm for coloring bipartite graphs*, Information Sciences **69**, 219–227, 1993.

A Novel Modeling Methodology: Generalized Nets

Maciej Krawczak

Systems Research Institute, Polish Academy of Sciences, Newelska 6,
Warsaw School of Information Technology, Newelska 6,
01-447 Warsaw, Poland
krawczak@ibspan.waw.pl

Abstract. The generalized net methodology was developed as a counterpart of Petri nets. The methodology allows to model different kinds of discrete dynamic systems. The basics of the theory of generalized nets is introduced and next the algorithm of generalized nets is described. Algebraic aspects of generalized nets as well as operator aspects of generalized nets are described. At the end, one possible application of generalized nets, namely for neural networks is shown. Here a neural network without any aggregation is considered.

Keywords: modeling, generalized nets, knowledge representation, neural networks.

1 Generalized Nets Introduction

The basic difference between generalized nets and the ordinary Petri nets is the *place – transition relation* [2], in the theory of generalized nets the transitions are objects of a very complex nature. The places are marked by $\bigcirc$, and the transitions by $\bigcirc$.

Generalized nets contain *tokens*, which are transferred from place to place. Every token bears some information, which is described by token's *characteristic*,

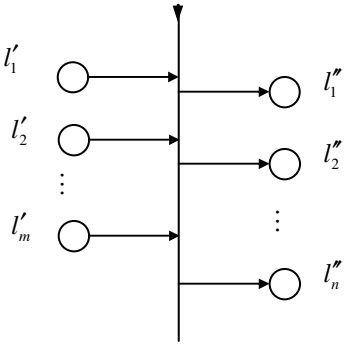


Fig. 1. A generalized net transition

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1160 – 1168, 2006.
© Springer-Verlag Berlin Heidelberg 2006

and any token enters the net with an *initial characteristic*. After passing a transition the tokens' characteristics are modified.

The transition has *input* and *output* places, as shown in Fig. 1. Formally, every transition is described by a seven-tuple

$$Z = \langle L', L'', t_1, t_2, r, M, \square \rangle . \tag{1}$$

where

- $L' = \{l'_1, l'_2, \ldots, l'_m\}$ is a finite, non empty set of the transition's input places,
- $L'' = \{l''_1, l''_2, \ldots, l''_m\}$ is a finite, non empty set of the transition's output places,
- t_1 is the current time of the transition's firing,
- t_2 is the current duration of the transition active state,
- r is the transition's *condition* determining which tokens will pass from the transition's inputs to its outputs; it has the form of an *index matrix* described in [1],
- M is an index matrix of the capacities of transition's arcs,
- $\square$ is an object of a form similar to a Boolean expression, when the value is *true*, the transition can become active, otherwise it cannot.

The following ordered four-tuple

$$G = \left\langle \langle A, \pi_A, \pi_L, c, f, \Theta_1, \Theta_2 \rangle, \langle K, \pi_k, \Theta_K \rangle, \langle T, t^0, t^* \rangle, \langle X, \Phi, b \rangle \right\rangle \tag{2}$$

is called *generalized net* if the elements are described as follows:

- A is a set of transitions,
- π_A is a function yielding the priorities of the transitions, i.e. $\pi_A : A \to N$, where $N = \{0, 1, 2, \ldots\} \cup \{\infty\}$,
- π_L is a function specifying the priorities of the places, i.e. $\pi_L : L \to N$, where $L = pr_1 A \cup pr_2 A$, and $pr_i X$ is the i-th projection of the n-dimensional set, where $n \in N$, $n \geq 1$ and $1 \leq i \leq n$ (obviously, L is the set of all generalized net places),
- c is a function providing the capacities of the places, i.e. $c : L \to N$,
- f is a function that calculates the truth values of the predicates of the transition's conditions (for the generalized net described here let the function f have the value *false* or *true*, i.e. a value from the set $\{0, 1\}$),
- Θ_1 is a function specifying the next time-moment when a given transition Z can be activated, i.e. $\Theta_1(t) = t'$, where $pr_3 Z = t$, $t' \in [T, T + t^*]$ and $t \leq t'$; the value of this function is calculated at the moment when the transition terminates its functioning,
- Θ_2 is a function yielding the duration of the active state of a given transition Z, i.e. $\Theta_2(t) = t'$, where $pr_4 Z = t \in [T, T + t^*]$ and $t' \geq 0$; the value of this function is calculated at the moment when the transition starts its functioning,

- K is the set of the generalized net's tokens,
- π_K is a function specifying the priorities of the tokens, i.e. $\pi_K : K \to N$,
- Θ_K is a function producing the time-moment when a given token can enter the net, i.e. $\Theta_K(\alpha) = t$, where $\alpha \in K$ and $t \in [T, T + t^*]$,
- T is the time-moment when the generalized net starts functioning; this moment is determined with respect to a fixed (global) time-scale,
- t^0 is an elementary time-step, related to the fixed (global) time-scale,
- t^* is the duration of the generalized net functioning,
- X is the set of all initial characteristics the tokens can receive on entering the net,
- Φ is a characteristic function that assigns new characteristics to every token when it makes the transfer from an input to an output place of a given transition,
- b is a function specifying the maximum number of characteristics a given token can receive, i.e. $b : K \to N$.

The generalized nets with lacking some components are called *reduced generalized nets*.

2 Generalized Nets Models of Neural Networks

A multilayer neural network consists of a number of simple processing units called neurons. The neurons are arranged in L layers, each layer is composed of $N(l)$ neurons, $l = 0,1,2,...,L$, where $N(0)$ denotes the number of inputs. The output of the network is equivalent to all the neurons' outputs from the last L-th layer. The network output is strictly related to the presented input, subject to the conditions resulting from the constancy of the structure (the neuron connections), the activation functions as well as the weights. In this way the neural networks realize the following simulation, that is

$$output = NN(input). \tag{3}$$

The simulation process of neural networks can be modeled by *generalized nets* methodology. In the review and bibliography on generalized nets theory and applications [4] we can find a list 353 scientific works related to the generalized nets.

The neurons, which constitute the neural network can be aggregated in many ways, e.g. any neuron is treated as a subsystem, or the neurons are aggregated within each laser, or neither separate neurons nor layers are distinguished. Here we consider only the first case treating each neuron as a separate subsystem. In this case the considered neural network consists of NL subsystems (neurons) described by the activation function as follows

$$x_{pj(l)} = f(net_{pj(l)}). \tag{4}$$

where

$$net_{pj(l)} = \sum_{i=1}^{N(l-1)} w_{i(l-1)j(l)}\ x_{pi(l-1)}\ . \tag{5}$$

while $x_{pi(l-1)}$ denotes the output of the i-th neuron with respect to the pattern p, $p = 1,2,...,P$, and the weight $w_{i(l-1)j(l)}$ connects the i-th neuron from the $(l-1)$-st layer with the j-th from the l-th layer, $j = 1,2,...,N(l)$, $l = 1,2,...,L$.

It is obvious that the different cases of aggregation of the neural network determine different streams of information passing through the system. In the subsequent sections we will describe the way of modeling the simulation process of multilayer neural networks by generalized nets for these three cases of aggregation.

The generalized net model of the considered aggregation case is shown in Fig. 2.

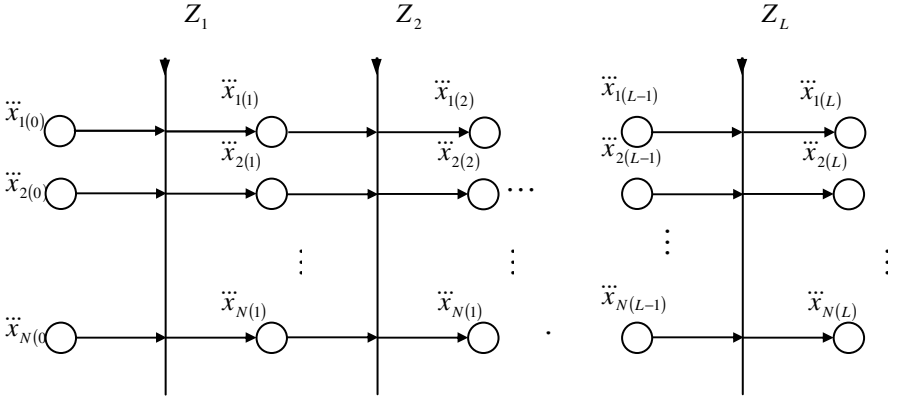


Fig. 2. The generalized net model of neural network simulation

The model consists of a set of L transitions, each transition being of the following form

$$Z_l = \left\langle \left\{ \ddot{x}_{1(l-1)}, \ddot{x}_{2(l-1)},..., \ddot{x}_{N(l-1)} \right\}, \left\{ \ddot{x}_{1(l)}, \ddot{x}_{2(l)},..., \ddot{x}_{N(l)} \right\},\ \tau_l, \tau'_l, r_l, M_l, \square_l \right\rangle . \tag{6}$$

for $l = 1,2,...,L$, where $\left\{ \ddot{x}_{1(l-1)}, \ddot{x}_{2(l-1)},..., \ddot{x}_{N(l-1)} \right\}$ - is the set of input places of the l-th transition, $\left\{ \ddot{x}_{1(l)}, \ddot{x}_{2(l)},..., \ddot{x}_{N(l)} \right\}$ - is the set of output places of the l-th transition, τ_l - is the time when the l-th transition is fired out, while it is assumed that $\tau_1 = T$ and $\tau_l = T + \sum_{k=2}^{l} \tau'_{k-1}$, τ'_l - is the duration time of firing of the l-th transition, r_l - denotes the l-th transition condition determining the transfer of tokens from the transition's inputs $\left\{ \ddot{x}_{1(l-1)}, \ddot{x}_{2(l-1)},..., \ddot{x}_{N(l-1)} \right\}$ to its outputs $\left\{ \ddot{x}_{1(l)}, \ddot{x}_{2(l)},..., \ddot{x}_{N(l)} \right\}$, and has the following index matrix form:

$$r_l = \begin{array}{c|cccc} & \vdots\!\vdots x_{1(l)} & \vdots\!\vdots x_{2(l)} & \cdots & \vdots\!\vdots x_{N(l)} \\ \hline \vdots\!\vdots x_{1(l-1)} & true & true & \cdots & true \\ \vdots\!\vdots x_{2(l-1)} & true & true & \cdots & true \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots\!\vdots x_{N(l-1)} & true & true & \cdots & true \end{array} \tag{7}$$

where the value *true* indicates that the tokens representing the neurons can be transferred from the i-th input place to the j-th output place, $i = 1, 2, ..., N(l-1)$, $j = 1, 2, ..., N(l)$, M_l - indicates an index matrix describing the capacities of transition's arcs:

$$M_l = \begin{array}{c|cccc} & \vdots\!\vdots x_{1(l)} & \vdots\!\vdots x_{2(l)} & \cdots & \vdots\!\vdots x_{N(l)} \\ \hline \vdots\!\vdots x_{1(l-1)} & 1 & 1 & \cdots & 1 \\ \vdots\!\vdots x_{2(l-1)} & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots\!\vdots x_{N(l-1)} & 1 & 1 & \cdots & 1 \end{array} \tag{8}$$

$\square_l$ - has a form of Boolean expression $\wedge\left(\vdots\!\vdots x_{1(l-1)}, \vdots\!\vdots x_{2(l-1)}, ..., \vdots\!\vdots x_{N(l-1)}\right)$ and stipulates that each input place $\vdots\!\vdots x_{i(l-1)}$, $i = 1, 2, ..., N(l-1)$, must contain a token that will be transferred to the l-th transition.

The generalized net describing the considered neural network simulation process has the following form:

$$GN = \left\langle \left\langle A, \pi_A, \pi_X, c, g, \Theta_1, \Theta_2 \right\rangle, \left\langle K, \pi_k, \Theta_K \right\rangle, \quad \left\langle T, t^0, t^* \right\rangle, \left\langle Y, \Phi, b \right\rangle \right\rangle. \tag{9}$$

where $A = \{Z_1, Z_2, ..., Z_L\}$ - is the set of transitions, π_A - is a function classifying the transitions, this classification giving the priorities of the transitions, i.e. $\pi_A : A \to N$, where $N = \{0, 1, 2, \ldots\} \cup \{\infty\}$ - in the considered neural network case this function is not valid because the transitions are arranged in a natural way (the asterisk * will be used in the subsequent text in order to denote the components of the general net structure which can be omitted), π_X - is a function describing the priorities of the places in the following way:

$$pr_1\{Z_1, ..., Z_L\} = \left\{ \vdots\!\vdots x_{1(0)}, ..., \vdots\!\vdots x_{N(0)}, \vdots\!\vdots x_{1(1)}, ..., \vdots\!\vdots x_{N(1)}, ..., \vdots\!\vdots x_{1(L-1)}, ..., \vdots\!\vdots x_{N(L-1)} \right\} \tag{10}$$

$$pr_2\{Z_1, ..., Z_L\} = \left\{ \vdots\!\vdots x_{1(2)}, ..., \vdots\!\vdots x_{N(2)}, \vdots\!\vdots x_{1(2)}, ..., \vdots\!\vdots x_{N(2)}, ..., \vdots\!\vdots x_{1(L)}, ..., \vdots\!\vdots x_{N(L)} \right\} \tag{11}$$

$$pr_1 A \cup pr_2 A = \left\{ \vdots\!\vdots x_{1(0)}, ..., \vdots\!\vdots x_{N(0)}, \vdots\!\vdots x_{1(1)}, ..., \vdots\!\vdots x_{N(1)}, \ldots, \vdots\!\vdots x_{1(2)}, ..., \vdots\!\vdots x_{N(2)}, ..., \vdots\!\vdots x_{1(L)}, ..., \vdots\!\vdots x_{N(L)} \right\} \tag{12}$$

c - is a function describing the capacities of the places; in our case it is equal 1, for $i = 1, 2,..., N(l),\ l = 0,1,2,..., L$, g - is a function that calculates the truth values of the predicates of the transition conditions, in the considered case $g(r_{l,i(l-1)j(l)}) = true$, Θ_1 - is a function yielding the next time-moment when the transitions can be again activated, Θ_2 - is a function giving the duration of activity of a given transition Z_l, K - is the set of tokens entering the generalized net, in the considered case there are $N(0)$ input places and each place contains one token; this set can be written as

$$K = \{\ \alpha_{1(0)}, \alpha_{2(0)},..., \alpha_{N(0)}\} \tag{13}$$

π_K - is a function describing the priorities of the tokens, here all tokens have the same priorities, and it will be denoted by * for $\pi_K(\alpha_{l(0)})$, $l = 1,2,..., N(0)$, Θ_K - is a function giving the time-moment when a given token can enter the net, i.e. all the tokens enter the considered generalized net at the same moment T, T - is the time when the generalized net starts functioning – here it is assumed that the net starts at the moment T, when the tokens enter the net, t^0 - is an elementary time-step, here this parameter is not used and is denoted by *, t^* - determines the duration of the generalized net functioning, that is $t^* = \sum_{l=1}^{L} \tau'_l$, Y - denotes the set of all the initial characteristics of the tokens, the characteristics of tokens describe the information which is carried by tokens and changed in transitions,

$$Y = \{\ y(\alpha_{1(0)}), y(\alpha_{2(0)}),..., y(\alpha_{N(0)})\} \tag{14}$$

where

$$y(\alpha_{i(0)}) = \langle NN1, N(0), N(1),\ imX_{i(0)},\ imW_{i(0)},\ F_{(1)},\ imout_{i(0)}\rangle \tag{15}$$

is the initial characteristic of the token $\alpha_{i(0)}$ that enters the place $\ddot{x}_{i(0)}$, $i = 1,2,..., N(0)$, where $NN1$ - the neural network identifier, $N(0)$ - the number of input places to the net as well as to the transition Z_1 (equal to the number of inputs to the neural network), $N(1)$ - the number of the output places of the transition Z_1,

$$imX_{i(0)} = [0,...,0, x_{i(0)}, 0,...,0]^T \tag{16}$$

- is the index matrix, indicating the inputs to the network, of dimension $N(0)\times 1$ in which all elements are equal 0 except for the element i whose value is equal $x_{i(0)}$ (the i-th input of the neural network),

$$imW_{i(0)} = \begin{array}{c|cccc} & \ddot{x}_{1(1)} & \ddot{x}_{2(1)} & \cdots & \ddot{x}_{N(1)} \\ \hline \ddot{x}_{i(0)} & w_{i(0)1(1)} & w_{i(0)2(1)} & \ldots & w_{i(0)N(1)} \end{array} \tag{17}$$

- has a form of an index matrix and denotes the weights connecting the i-th input with all neurons allocated to the 1-st layer

$$F_{(1)} = \left[f_{1(1)}\left(\sum_{i=1}^{N(0)} x_{i(0)}\, w_{i(0)1(1)} \right), ..., f_{N(1)}\left(\sum_{i=1}^{N(0)} x_{i(0)}\, w_{i(0)N(1)} \right) \right]^T \tag{18}$$

- denotes a vector of the activation functions of the neurons associated with the 1-st layer

$$imout_{i(0)} = \begin{array}{c|cccc} & \ddot{x}_{1(1)} & \ddot{x}_{2(1)} & \cdots & \ddot{x}_{N(1)} \\ \hline \ddot{x}_{i(0)} & x_{i(0)}\, w_{i(0)1(1)} & x_{i(0)}\, w_{i(0)2(1)} & \cdots & x_{i(0)}\, w_{i(0)N(1)} \end{array} \tag{19}$$

- is an index matrix describing the signal outgoing from the i-th input place, $i = 1, 2, ..., N(0)$, to all output places of the Z_1 transition, Φ - is a characteristic function that generates the new characteristics of the new tokens after passing the transition; for the transition Z_l, $l = 1, 2, ..., L$, there are $N(l-1)$ input places $\left\{ \ddot{x}_{1(l-1)}, \ddot{x}_{2(l-1)}, ..., \ddot{x}_{N(l-1)} \right\}$ and with each place there is associated a single token $\alpha_{i(l-1)}$, $i = 1, 2, ..., N(l-1)$, having the characteristic

$$y(\alpha_{i(l-1)}) = \left\langle NN1, N(l-1), N(l), imX_{i(l-1)}, imW_{i(l-1)}, F_{(l)}, imout_{i(l-1)} \right\rangle \tag{20}$$

where $NN1$ - the neural network identifier, $N(l-1)$ - the number of input places to the net as well as to the transition Z_l, $N(l)$ - the number of the output places of the transition,

$$imX_{i(l-1)} = [0, ..., 0, x_{i(l-1)}, 0, ..., 0]^T \tag{21}$$

- is the index matrix of dimension $N(l-1) \times 1$ in which all elements are equal 0 except the element i whose value is equal $x_{i(l-1)}$ - the i-th input value associated with the Z_l transition,

$$imW_{i(l-1)} = \begin{array}{c|cccc} & \ddot{x}_{1(l)} & \ddot{x}_{2(l)} & \cdots & \ddot{x}_{N(l)} \\ \hline \ddot{x}_{i(l-1)} & w_{i(l-1)1(l)} & w_{i(l-1)2(l)} & \cdots & w_{i(l-1)N(l)} \end{array} \tag{22}$$

- is an index matrix describing the weight connection between the i-th input places with all output places of the Z_l transition,

$$F_{(l)} = \left[f_{1(l)}\left(\sum_{i=1}^{N(l-1)} x_{i(l)}\, w_{i(l-1)1(l)} \right), ..., f_{N(l)}\left(\sum_{i=1}^{N(l-1)} x_{i(l-1)}\, w_{i(l-1)N(l)} \right) \right]^T \tag{23}$$

- is a vector of the activation functions of the neurons associated with the l-th layer of the neural network

$$imout_{i(l-1)} = \begin{array}{c|cccc} & \ddot{x}_{1(l)} & \ddot{x}_{2(l)} & \cdots & \ddot{x}_{N(l)} \\ \hline \ddot{x}_{i(l-1)} & x_{i(l-1)}\, w_{i(l-1)1(l)} & x_{i(l-1)}\, w_{i(l-1)2(l)} & \cdots & x_{i(l-1)}\, w_{i(l-1)N(l)} \end{array} \tag{24}$$

- is an index matrix describing the signals outgoing from the i-th input place, $i = 1,2,...,N(l)$, to all output places of the Z_l transition.

The tokens $\alpha_{i(l-1)}$, $i = 1,2,...,N(l-1)$, passing the transition Z_l vanish, and the new tokens $\alpha_{j(l)}$, $j = 1,2,...,N(l)$, associated with the output places $\{\ddot{x}_{1(l)}, \ddot{x}_{2(l)},..., \ddot{x}_{N(l)}\}$ of the transition Z_l are generated, their characteristics being described as follows

$$y(\alpha_{j(l)}) = \langle NN1, N(l), N(l+1), imX_{j(l)}, imW_{j(l)}, F_{(l+1)}, imout_{j(l)} \rangle \tag{25}$$

for $l = 1,2,...,L-1$, while

$$y(\alpha_{j(L)}) = \langle NN1, N(L), imX_{j(L)} \rangle \tag{26}$$

and for these new tokens the values $x_{j(l)}$, $j = 1,2,...,N(l)$, are calculated in the following way

$$imX_{j(l)} = f_{j(l)}\left(\sum_{i=1}^{N(l-1)} imout_{i(l-1)}\right),\ l = 1,2,...,L \tag{27}$$

It should be mentioned here that $imX_{j(L)}$, $j = 1,2,...,N(L)$, denotes the output of the neural network, the final state of the network after ending the simulation process, b - is a function describing the maximum number of characteristics a given token can receive; in the here considered neural network simulation process this function has a simple form

$$b(\alpha_{j(l)}) = 1, \text{ for } j = 1,2,...,N(l),\ l = 1,2,...,L, \tag{28}$$

which means that the characteristic of each token $\alpha_{j(l)}$, $j = 1,2,...,$ $N(l)$, $l = 1,2,...,L$, is constructed on the base of the characteristics of all tokens ($i = 1,2,...,$ $N(l-1)$) from the previous layer $(l-1)$, $l = 1,2,...,L$.

Due to the above considerations the transitions have the following form

$$Z_l = \langle \{\ddot{x}_{1(l-1)}, \ddot{x}_{2(l-1)},..., \ddot{x}_{N(l-1)}\}, \{\ddot{x}_{1(l)}, \ddot{x}_{2(l)},..., \ddot{x}_{N(l)}\},\ \tau_l, \tau_l', *, *, \square_l \rangle \tag{29}$$

for $l = 1,2,...,L$.

The reduced form of the generalized net describing the simulation process of the neural network has the following form:

$$GN = \langle \langle A, *, \pi_X, c, *, \Theta_1, \Theta_2 \rangle,\ \langle K, *, \Theta_K \rangle,\ \langle T, *, t^* \rangle,\ \langle Y, \Phi, b \rangle \rangle \tag{30}$$

where $A = \{Z_1, Z_2,..., Z_L\}$ - is a set of transitions, π_X - is a function describing the priorities of the places, c - is a function describing the capacities of the places, i.e. $c(x_{i(l)}) = 1$, $i = 1, 2,...,$ $N(l)$, $l = 0,1,2,...,L$, Θ_1 - is a function yielding the next time-moment when the transitions can be again activated, $\Theta_1(t_l) = t_l'$, $l = 1,2,...,L$, Θ_2 - is a function giving the duration of activity of the transition Z_l $\Theta_2(t_l) = t_l''$,

$l = 1,2,...,L$, $K = \{ \alpha_{1(0)}, \alpha_{2(0)},..., \alpha_{N(0)} \}$ - is the set of tokens entering the generalized net, $\Theta_K = T$ - for all tokens entering the net and at this moment the net starts to function, t^* - determines the duration of the generalized net's functioning and is described by (28) or (29), Y - denotes the set of all initial characteristics of the tokens described by

$$Y = \{ y(\alpha_{1(0)}), y(\alpha_{2(0)}),..., y(\alpha_{N(0)}) \} \tag{31}$$

where $y(\alpha_{i(0)}) = \langle NN1, N(0), N(1), x_{i(0)}, W_{i(0)(1)}, F_{(1)}, out_{i(0)} \rangle$, Φ - is a characteristic function that generates the new characteristics of the new tokens after passing the transition, $b(\alpha_{j(l)}) = 1$, for $j = 1,2,...,N(l), l = 1,2,...,L$, - is a function describing the number of characteristics memorized by each token.

Such generalized nets with some components missing (the components not being valid) are called *reduced generalized nets*, [2]. In the above version of the generalized nets representation of the simulation process of multilayer neural network we preserve the parallelism of computation.

3 Conclusions

We have described the concept of a generalized nets methodology for modeling discrete event systems, and then the concept of index matrix useful for aggregation as well as for separation of subsystems.

Next, as an example we considered the generalized net model representing the functioning of the multilayer neural networks - the simulation process of this class of networks, and many sophisticated tools of the generalized nets theory have been applied.

Using the applied procedure allows us to construct a subroutine-like, which describes functioning of a system – in our case a multilayer neural network. In such a subroutine the following parameters like a number of layers, numbers of neurons within each layer as well as neuron parameters can be treated as formal parameters.

References

1. Atanassov, K.: *Generalized nets*. World Scientific, Singapore New Jersey London (1991)
2. Atanassov, K.: *Generalized Nets and Systems Theory*. „Prof. M. Drinov" Academic Publishing House Sofia (1997)
3. Krawczak, M.: *Multilayer Neural Systems and Generalized Net Models*. Academic Press EXIT, Warsaw (2003)
4. Radeva, V., Krawczak, M., Choy, E.: Review and Bibliography on Generalized Nets Theory and Applications. *Advanced Studies in Contemporary Mathematics*, Vol. 4. (2002) 173-199

A Hierarchical Particle Swarm Optimization for Solving Bilevel Programming Problems

Xiangyong Li[1], Peng Tian[1], and Xiaoping Min[2]

[1] Antai College of Economics & Management, Shanghai Jiaotong University, Shanghai 200052, P.R. China
lixiangyong@163.com, ptian@sjtu.edu.cn
[2] School of Finance, Jiangxi University of Finance and Economics, Nanchang 330013, P.R. China
mxptiger@163.com

Abstract. The bilevel programming problem (BLPP) has proved to be a NP-hard problem. In this paper, we propose a hierarchial particle swarm optimization (PSO) for solving general BLPPs. Unlike most traditional algorithms designed for specific versions or based on specific assumptions, the proposed method is a hierarchical algorithm framework, which solves the general bilevel programming problems directly by simulating the decision process of bilevel programming. The solving general BLPPs is transformed to solve the upper-level and lower-level problems iteratively by two variants of PSO. The variants of PSO are built to solve upper-level and lower-level constrained optimization problems. The experimental results compared with those of other methods show that the proposed algorithm is a competitive method for solving general BLPPs.

1 Introduction

Bilevel programming problem (BLPP) is a hierarchical optimization problem, which has a second (parametric) optimization problem as part of its constraints [1]. In such a hierarchical decision framework, the decision maker at the lower-level tries to optimize his objective under the given parameters from the upper-level decision maker, who, in turn, determines the optimal parameters so as to optimize his own objective based on the complete information on the possible response of lower-level decision maker. Although each player tries to optimize his own objective without considering others' objectives, his reasonable choices are interdependent, and the decision of one player affects the objective and the feasible decision space of others. The bilevel programming model has been applied to many domains including hierarchical decision process and a wide range of real-life applications can be transformed into the bilevel or multi-level programming framework, such as economic management[1], toll-setting problem[2], and transportation network design[3]. The BLPP can be formulated as follows[2]:

$$\min_{x} \quad F(x, y) \tag{1}$$

$$s.t. \quad G(x, y) \leq 0 \tag{2}$$

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1169–1178, 2006.
© Springer-Verlag Berlin Heidelberg 2006

$$H(x, y) = 0 \tag{3}$$

where, y for each fixed x, is the optimum of the following optimization problem:

$$\min_{y} \quad f(x, y) \tag{4}$$

$$s.t. \quad g(x, y) \leq 0 \tag{5}$$

$$h(x, y) = 0 \tag{6}$$

where, $x \subseteq R^n$, and $y \subseteq R^m$ denote the upper-level and lower-level variables respectively. F and f are the upper-level and lower-level objective functions, which are a mapping $R^n \times R^m \to R$. $G(x, y) \leq 0$ ($g(x, y) \leq 0$), and $H(x, y) = 0$ ($h(x, y) = 0$) denote the upper-level (lower-level) constraints. BLPP is a mathematical optimization, where the set of all decision variables is partitioned between x and y, and y is determined as an optimal solution of the lower-level programming problem parameterized for any given x.

In the past years, many algorithms have been proposed for solving the BLPP, e.g. methods based on Kuhn-Tucker conditions[4], penalty function methods[1], fuzzy approach[5], trust region algorithm[6], and metaheuristic[7,8,9]. The most popular algorithm is that based on Kuhn-Tucker conditions, where the lower-level problem is replaced by its Kuhn-Tucker condition and a one-level problem with complementarity constraints is developed. The most proposed methods may be problem-dependent, and require that the objective of the upper or lower level problem must be differentiable, or the feasible region must be convex. Because of such a deficiency, we may not obtain satisfying result while applying them to real-life applications, especially to the BLPP with nondifferentiable objective function or nonconvex search space. On the contrary, the metaheuristic needn't differentiability of objective functions, any gradient information, or the convexity of search space. In recent years, the metaheuristic has received considerable attention to its potential as an alternative method for the BLPP, especially for complex nonlinear BLPP. Some metaheuristics for solving BLPP have been established, such as GA[7], simulated annealing[8], and tabu search[9].

As a new metaheuristic, particle swarm optimization (PSO) has proved to be a competitive algorithm for unconstrained optimization problems compared with other algorithms such as GA and SA since its introduction by Kennedy and Eberhart[10]. In this paper, we first extend the application of standard particle swarm optimization (PSO) to solving BLPP and propose a hierarchical PSO for solving general version of bilevel programming problem (HPSOBLP). The rest of this paper is organized as follows. Section 2 introduces the standard PSO. Section 3 introduces our proposed algorithm in detail. The experimental results of the proposed algorithm in comparison with other algorithms are reported in section 4. Section 5 summarizes this paper and shows some future work.

2 Standard Particle Swarm Optimization

PSO is a new adaptive and population-based stochastic optimization algorithm. In PSO, the individuals manipulate their trajectories toward the best regions of

their own previous best performance and toward the locations found by members in their neighbors dynamically and stochastically, and the particles cooperate to solve the optimization problem through the mutual interaction of individuals. The position of each particle is represented by a n-dimensional vector in problem space $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, $i = 1, 2, \ldots, N$ (N is the population size), and its performance is evaluated on the predefined fitness function related to the problem. The velocity of the *i-th* particle $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ is defined as the change of its position. The flying direction of particles is the dynamical interaction of individual and social flying experience. The position change of each particle is a function of its current velocity vector, the difference between its current position and the best position found by itself so far,the stochastically weighted difference between the individual's current position and the best position found by any member in its neighborhood. The velocity and position of *i-th* particle at time t is updated according to the following two equations:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot rand1 \cdot (p_i - x_i(t)) + c_2 \cdot rand2 \cdot (p_g - x_i(t)) \quad (7)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (8)$$

where, $p_i = (p_{i1}, \ldots, p_{in})$ is the best position encountered by *i-th* particle so far; p_g represents the best position found by any member in the neighborhood of *i-th* particle (for local version of PSO) or the whole swarm (for global version of PSO); t is the iteration counter; c_1 and c_2 are the acceleration coefficients, which are the weight of the velocity toward global and local best; $rand1$ and $rand2$ are two random numbers in $[0, 1]$; w is the inertia weight. The inertia weight controls the impact of previous histories of velocities on the current velocity, which is often used as a parameter to control the trade-off between exploration and exploitation in the search space.

3 Hierarchical Particle Swarm Optimization for Solving Bilevel Programming Problem

In this section, we introduce our proposed hierarchical PSO for solving all classes of bilevel programming problem, which doesn't require any assumption of the property of the objectives or constraints functions. As mentioned in section 2, we know that most existing algorithms for BLPP are established and implemented for specific versions of BLPP, or based on some weak or strong assumptions, such as differentiability or convexity. Here, we propose a new algorithm framework based on PSO for solving general bilevel programming problem. Unlike traditional methods, our proposed algorithm is a sequentially optimization method, which combines two variants of PSO to solve the upper-level and lower-level programming problems interactively and cooperatively.

3.1 Proposed Algorithm for Solving General BLPPs

As mentioned above, BLPP is a hierarchical and sequential optimization problem. In consideration of the characteristics of sequential decision, we can construct a hierarchical algorithm framework based on two variants of PSO to solve

BLPP(HPSOBLP). HPSOBLP solves the BLPP directly by simulating the sequential decision procedure of BLPP. The proposed algorithm consists of two variants of standard PSO (see section 3.2). The main body of our proposed approach is a variant of PSO (called PSO_L described in section 3.2), which is designed only to solve the upper-level programming problem based on the follower's optimal response. Another variant of PSO (called PSO_F described in section 3.2) is embedded in the hierarchical framework to solve the lower-level programming problem, and obtain the optimal response of the follower for each given decision variable x of the upper-level programming (x is obtained by PSO_L). The follower's optimal reaction y is then returned to upper-level programming problem as the implementation base of PSO_L.

In our proposed algorithm HPSOBLP, solving BLPP is transformed to solve the upper-level and lower-level programming problem respectively while supposing that the decision variable of upper-level or lower-level is determined respectively. But the solution information is exchanged between two variants of PSO, and the output of one algorithm is the input of another algorithm, namely y the output of PSO_F is the input of PSO_L and x the output of PSO_L is the input of PSO_F. These forms a hierarchical and sequential framework. The HPSOBLP is implemented interactively in the hierarchical structure of the two variants of

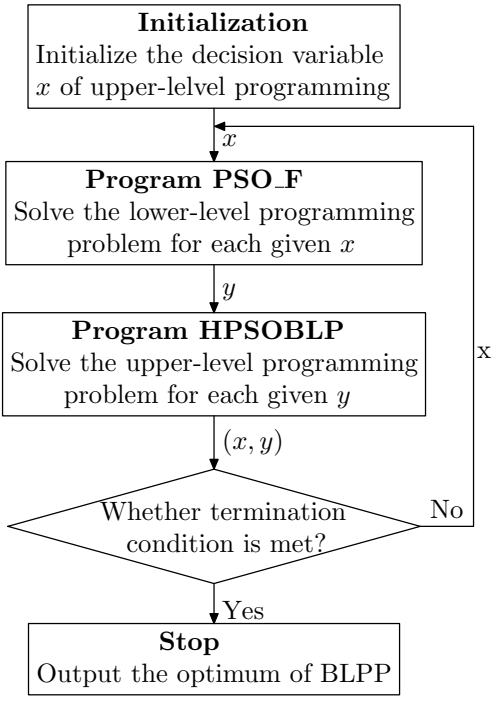


Fig. 1. The scheme procedure for HPSOBLP

PSO. Fig. 1 shows the scheme procedure of HPSOBLP to generate solutions for BLPP. Such a repeated interaction and cooperation between two variants of PSO reflects and simulates the hierarchical decision process of BLPP and can solve BLPP sequentially. The algorithm is tested in section 4. The proposed algorithm can be used to solve different classes of BLPP, even though complex BLPP with nondifferentiable or nonconvex objective functions and constraints.

3.2 A Modified PSO for Solving Constrained Optimization Problem

From equation (1)-(6), we know that the upper-level or lower-level programming problem is a standard constrained optimization problem without consideration of interaction and interdependency between the leader and the follower. Here we propose a modified PSO with a new constraint-handling mechanism (namely above PSO_L and PSO_F) to solve them(called PSOCO in the following section).

Without loss of generality, here, we choose the lower-level programming problem as a single independent constrained optimization problem to describe our proposed algorithm PSOCO when the leader's decision variable x is given. Here, we assume that the constraints of lower-level programming are composed of q inequalities constraints ($g_i(x,y) \leq 0$, for i=1,...,q), and $m-q$ equalities constraints ($h_i(x,y) = 0$, for $i = q+1$,...,m). In the following sections, the particles satisfying all the constraints are called feasible particles, otherwise infeasible particles. Inspired by the basic idea in GA for solving constrained optimization problems[11], we assume that every feasible particle is better than every infeasible one. Under this assumption, all particles violating the constraints (at least a constraint) will be associated with a bigger fitness than the ones satisfying all the constraints (for minimization problems). It is also assumed that all infeasible particles at each generation also have bigger fitness than the worst feasible particle found over the generations so far. In this paper, we introduce a variable $Worst_fit$ to keep track of the fitness value of the worst feasible particle.

In the PSOCO, all the particles (feasible or infeasible) for a given upper-level decision variable x are evaluated by the following objective function $P(x,y)$:

$$P(x,y) = \begin{cases} f(x,y) & \text{if } y \in \Omega(x) \\ f(x,y) + r\sum_{i=1}^{m} f_i(x,y) + \varphi(y,t) & \text{if } y \in S \backslash \Omega(x) \end{cases} \tag{9}$$

where S is the search space. $\Omega(x)$ is defined in section 1. r is the penalty coefficient. $f_i(x,y)$ denotes the constraint violation measure of infeasible particles for the i-th constraint. $\varphi(y,t)$ is the additional heuristic value at time t. The $f_i(x,y)$ and $\varphi(y,t)$ are defined as:

$$f_i(x,y) = \begin{cases} \max\{0, g_i(x,y)\} & \text{if } 1 \leq i \leq q \\ |h_i(x,y)| & \text{if } q+1 \leq i \leq m \end{cases} \tag{10}$$

$$\varphi(y,t) = Worst_fit(t) - \min_{y \in S \backslash \Omega(x)} \{f(x,y) + r\sum_{i=1}^{m} f_i(x,y)\} \tag{11}$$

$$Worst_fit(t) = \max\{Worst_fit(t-1), \max_{y \in \Omega(x)} \{f(x,y)\}\} \tag{12}$$

where, $\varphi(y,t)$ is an additional penalty for infeasible particles at time t. It means that any infeasible particle will be given an additional penalty to guarantee that its fitness value in current generation is worse than the value of $Worst_fit$. The $Worst_fit(t)$ keeps record of the fitness value of the worst feasible particle until the t-th generation, which is updated dynamically according to equation (12) during the implementation of PSO. By using the variable $Worst_fit(t)$, the PSOCO can make all feasible particles better than infeasible particles over the generations. While the initialization of a particle swarm, the variable $Worst_fit(t)$ is set to a random larger number and updated dynamically over the generations without requirement that there is at least a feasible particle in the initial particle swarm. The algorithms must handle the constraints of both upper-level and lower-level problems in order to keep the particles moving in the constraint region and improve the algorithm convergence.

4 Experiment and Results

To compare the performance of our proposed algorithm with other algorithms, the HPSOBLP is testing on a series of testing problems. The testing problem set is composed of different classes of bilevel programming problems, namely the problems involving linea, quadratic, and nonlinear objective functions. The complete formulations are shown in the Appendix.

In consideration of computation precision of the computer, a violation tolerance ε (a very small positive number) is set for all the constraints. For example, for a constraint $g_i(x) \leq 0$, the constraint will be assumed to be violated only if $g_i(x) > \varepsilon$. Otherwise, the constraint is satisfied. Moreover, an equality constraint can be replaced by two inequalities constraints, namely, $g_i(x) = 0$ is transformed to $g_i(x) \leq 0$ and $g_i(x) \geq 0$. The parameters for the implementation of HPSOBLP are set as follows: the swarm size N_{max} is set to 20 and 40 respectively, the number of maximum generation G_{max} of two subprograms are set to 120 and 30, both acceleration coefficients c_1 and c_2 are set to 2 as proposed in literature[10], inertia weight w is set to decrease linearly from 1.2 to 0.1 as in[12], the maximum velocity of particles V_{max} is set to the corresponding bounds of decision variables, the penalty factor r is set to 10, and the violation tolerance ε is set to 7e-5. Moreover, the control parameter $Worst_fit$ is initialized to 100000.

With the abovementioned parameters, the HPSOBLP is implemented for 30 runs for each testing problem. The results in terms of the worst("Worst_L"), the average("Avg."), the best("Best_L"), and the standard deviation("Std.") of the upper-level objective value $F(x,y)$ are summarized in Table 1. The lower-level objective value $f(x,y)$ corresponding to the best, and worst upper-level objective (namely Best_F, and Worst_F) are also reported in Table 1. For each testing problem, it can be seen that the standard deviation of the best upper-level objective values over 30 trials, and the difference between the Best_L and the Worst_L almost equal to 0, which means that the robustness of our proposed algorithm is very high, and it is a robust algorithm for considered testing instances in this paper. Note that now there is no suitable testing set of large bilevel programming

Table 1. Summary of the results of HPSOBLP over 100 runs

No.	$F(x,y)$				$f(x,y)$	
	Best_L	Worst_L	Avg.	Std.	Best_F	Worst_F
T1	0	0	0	0	100	100
T2	225	225	225	0	100	100
T3	-14.7578	-13.4538	-14.0391	0.261062	0.206732	-0.59095
T4	-36.0003	-36	-36.0001	5.36e-05	0.249985	0.25
T5	-9.27689	-8.69985	-8.95046	0.12679	-4.75611	-10.0157
T6	-7.95649	-7.66102	-7.78311	0.066825	-1.51518	-0.95865
T7	-11.99851	-11.9978	-11.9981	0.000168	-459.2248	-137.1943
T8	-3.60275	-3.59888	-3.6007	0.000703	-1.98489	-2.00553
T9	-3.92014	-3.91658	-3.91952	0.000645	-1.99292	-2.0819
T10	88.77571	88.7852	88.78349	0.001635	-0.7698	-0.7698
T11	15.44004	16.37004	16.03177	0.217296	2.72798	2.25809
T12	1.99972	2.060783	2.001623	0.006511	24.019	23.8702
T13	2.703942	2.749716	2.741455	0.005005	0.560201	0.559681

problems, and our proposed approach is only tested on some testing problems with less variables and constraints. The performance of HPSOBLP need to be tested on large bilevel programming problems and real-world applications.

Table 2. Comparison of the optimums of HPSOBLP with other algorithms

No.	Leader's objective $F(x,y)$				Follower's objective $f(x,y)$			
	HPSOBLP	GA	TRM	Original	HPSOBLP	GA	TRM	Original
T1	0	0		5	100	100		0
T2	225	225		225	100	100		100
T3	-14.7578	-12.68	-12.68	-12.68	0.206732	-1.016	-1.02	-1.016
T4	-36.0003	-29.2		-29.2	0.249985	0.3148		0.3148
T5	-9.27689	-8.92		-8.92	-4.75611	-6.14		-6.05
T6	-7.95649	-7.58		-7.56	-1.51518.	-0.574		-0.580
T7	-11.99851	-11.999		-12	-459.225	-163.42		-112.71
T8	-3.60275	-3.6		-3.6	-1.98489	-2		-2
T9	-3.92014	-3.92		-3.15	-1.99292	-2		-16.29
T10	88.77571		88.79	88.79	-0.7698		-0.77	-0.77
T11	15.44004		17	17	2.72798		2	2
T12	1.999972		2	2		24.019	24.02	24.02
T13	2.703942		2.75	2.75	0.560201		0.57	0.57

We compare the results of our algorithm with those obtained by GA [7], trust-region method[2](abbreviated to TRM) and the methods in the corresponding original references(abbreviated to Original). Here our method is only compared with the above three algorithms because there is no results of other traditional algorithms in the literatures for the considered testing problems in this paper. The best-found results of HPSOBLP and other three algorithms are reported

in Table 2. It can be observed that the best-found results of HPSOBLP are better than or equal to those other algorithms, especially for testing problem 3,4,5,6,10,11 etc.. Note that we do not compare computation time for the approaches. We believe that it is an unfair comparison for some reasons. First the authors implement their algorithms on different computers. Second, different algorithms are coded in different structures, which affect the computation time.

In a word, the results show that the proposed algorithm can better solve most testing problems than other three approaches. The most interesting is that our proposed approach has solved different classes of BLPPs without any specific assumption or any transformation of the objective or constraints functions. Note that the above results were obtained under specific violation tolerance. We can improve the computation precision by decreasing the violation tolerance.

5 Conclusions and Future Work

In this paper, we extend the application of PSO to solving BLPPs, and develop a hierarchial PSO for solving general BLPPs (HPSOBLP). By simulating the decision process of sequential decision, our proposed algorithm solve general bilevel programming problems directly without any specific assumption conditions. The proposed algorithm is a hierarchial algorithm framework combining two variants of PSO. The variants of PSO are designed to solve the upper-level and lower-level programming problems iteratively. Through such an interaction between two variants of PSO, the general BLPPs can be solved effectively. The experimental results comparing with other algorithms are reported and show that the PSO optimization can also be used to solve BLPPs, and the proposed algorithm is a competitive approach for solving general BLPPs without any specific assumptions and constraints. Future work will look into application of HPSOBLP to other testing problems, especially to complex large-scale BLPPs in the real world. It is also necessary and significant to study the effects of parameters of standard PSO such as acceleration coefficients on the performance of HPSOBLP.

References

1. Dempe, S.: Foundations of Bilevel Programming. Volume 61 of Nonconvex optimization and its application. Kluwer Academic Publisher (2002)
2. Colson, B., Marcotte, P., Savard, G.: Bilevel programming: A survey. 4OR: Quarterly Journal of the Belgian, French and Italian Operations Research Societies **3** (2005) 87–107
3. Chiou, S.W.: Bilevel programming for the continuous transport network design problem. Transportation Research Part B: Methodological **39** (2005) 383
4. Shi, C., Lu, J., Zhang, G.: An extended kuhn-tucker approach for linear bilevel programming. Applied Mathematics and Computation **162** (2005) 63
5. Shih, H., Lai, Y., Lee, E.: Fuzzy apporach for multi-level programming problems. Computers and Operations Research **23** (1983) 773–791
6. Marcotte, P., Savard, G., Zhu, D.L.: A trust region algorithm for nonlinear bilevel programming. Operations Research Letters **29** (2001) 179

7. Wang, Y., Jiao, Y., Li, H.: An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme. IEEE Transactions on Systems, Man, and Cybernetics, Part C **35** (2005) 221
8. Sahin, K.H., Ciric, A.R.: A dual temperature simulated annealing approach for solving bilevel programming problems. Computers and Chemical Engineering **23** (1998) 25
9. Genderau, M., Marcotte, P., Savard, G.: A hybrid tabu-ascent algorithm for the linear bilevel programming problem. Journal of Global Optimization **8** (1996) 217–233
10. Kennedy, J., Eberhart, R., Shi, Y.: Swarm intelligence. Morgan Kaufmann Publisher, San Francisco (2001)
11. Powell, D., Skolnick, M.: Using genetic algorithms in engineering design optimization with non-linear constraints. In: Proceedings of the Fifth International Conference on Genetic Algorithms, Los Altos, CA, Morgan Kaufmann Publisher (1993)
12. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of IEEE World Congress on Computational Intelligence 1998. (1998) 69–73
13. Aiyoshi, E., Shimizu, K.: A solution method for the static constrained stackelberg problem via penalty method. IEEE Transactions on Automatic Control **29** (1984) 1111–1114
14. Shimizu, K., Aiyoshi, E.: A new computational method for syackelberg and min-max problems by use of a penalty method. IEEE Transaction on Autommatic Control **AC-26** (1981) 460–466
15. Bard, J.: Convex two-level optimization. Mathematical Programming **40** (1988) 15–27
16. Calvete, H.I., Gale, C.: Theory and methodology: The bilevel linear/linear fractional programming problem. European Journal of Operation Research **114** (1999) 188–197
17. Outrata, J.V.: On the numerical solution of a class of stackelberg problems. Zeitschrift Fur Operation Research **34** (1990) 255–278

Appendix: Testing problems

1. Testing problem one (T1)[13]
$$\min_x \quad F(x,y) = 2x_1 + 2x_2 - 3y_1 - 3y_2 - 60$$
$$\text{s.t. } x_1 + x_2 + y_1 - 2y_2 \le 40, 0 \le x_1 \le 50, 0 \le x_2 \le 50$$
$$\min_y \quad f(x,y) = (y_1 - x_1 + 20)^2 + (y_2 - x_2 + 20)^2$$
$$\text{s.t. } 2y_1 - x_1 + 10 \le 0, 2y_2 - x_2 + 10 \le 0$$
$$-10 \le y_1 \le 20, -10 \le y_2 \le 20$$
2. Testing problem two (T2)[14]
$$\min_x \quad F(x,y) = (x_1 - 30)^2 + (x_2 - 20)^2 - 20y_1 + 20y_2$$
$$\text{s.t. } -x_1 - 2x_2 + 30 \le 0, x_1 + x_2 - 25 \le 0, x_2 \le 15$$
$$\min_y \quad f(x,y) = (x_1 - y_1)^2 + (x_2 - y_2)^2$$
$$\text{s.t. } 0 \le y_1 \le 10, 0 \le y_2 \le 10$$
3. Testing problem third (T3)[15]
$$\min_x \quad F(x,y) = -x_1^2 - 3x_2 - 4y_1 + y_2^2$$
$$\text{s.t. } (x_1)^2 + 2x_2 \le 4, x_1 \ge 0, x_2 \ge 0$$

$$\min_{y} \quad f(x,y) = 2x_1^2 + y_1^2 - 5y_2$$
$$\text{s.t. } x_1^2 - 2x_1 + x_2^2 - 2y_1 + y_2 \geq -3$$
$$x_2 + 3y_1 - 4y_2 \geq 4, y_1 \geq 0, y_2 \geq 0$$

4. Testing problem four (T4)[16]

$$\min_{x} \quad F(x,y) = -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3$$
$$\text{s.t. } x_1 \geq 0, x_2 \geq 0$$
$$\min_{y} \quad f(x,y) = \frac{1+x_1+x_2+2y_1-y_2+y_3}{6+2x_1+y_1+y_2-3y_3}$$
$$\text{s.t. } -y_1 + y_2 + y_3 + y_4 = 1, 2x_1 - y_1 + 2y_2 - \frac{1}{2}y_3 + y_5 = 1$$
$$2x_2 + 2y_1 - y_2 - \frac{1}{2}y_3 + y_6 = 1, y_i \geq 0, i = 1, \ldots, 6$$

5. Testing problem five to nine (T5-T9)[17]

$$\min_{x} \quad F(x,y) = r(x_1^2 + x_2^2) - 3y_1 - 4y_2 + 0.5(y_1^2 + y_1^2)$$
$$\min_{y} \quad f(x,y) = 0.5[y_1 \quad y2]H[y_1 \quad y_2]^T - b(x)^T[y_1 \quad y_2]^T$$
$$\text{s.t. } -0.333y_1 + y_2 - 2 \leq 0, y_1 - 0.333y_2 - 2 \leq 0, y_1 \geq 0, y_2 \geq 0$$

T5) $r = 0.1, H = \begin{bmatrix} 1 & -2 \\ -2 & 5 \end{bmatrix}, b(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

T6) $r = 1$,H,and $b(x)$ is the same as in T5

T7) $r = 0, H = \begin{bmatrix} 1 & 3 \\ 3 & 10 \end{bmatrix}, b(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

T8) $r = 0.1$,H,and $b(x)$ is the same as in T7

T9) r,H is the same as in T8. $b(x) = \begin{bmatrix} -1 & 2 \\ 3 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

6. Testing problem ten(T10)[2]

$$\min_{x} \quad F(x,y) = x^2 + (y-10)^2$$
$$\text{s.t. } x + 2y - 6 \leq 0, -x \leq 0$$
$$\min_{y} \quad f(x,y) = x^3 - 2y^3 + x - 2y - x^2$$
$$\text{s.t. } -x + 2y - 3 \leq 0, -y \leq 0$$

7. Testing problem eleven (T11)[2]

$$\min_{x} \quad F(x,y) = (x-5)^2 + (2y+1)^2$$
$$\text{s.t. } -x \leq 0$$
$$\min_{y} \quad f(x,y) = (x-1)^2 - 1.5xy + x^3$$
$$\text{s.t. } -3x + y + 3 \leq 0, x - 0.5y - 4 \leq 0$$
$$x + y - 7 \leq 0, -y \leq 0$$

8. Testing problem twelve(T12)[2]

$$\min_{x} \quad F(x,y) = (x-5)^4 + (2y+1)^4$$
$$\text{s.t. } x + y - 4 \leq 0, -x \leq 0$$
$$\min_{y} \quad f(x,y) = e^{-x+y} + x^2 + 2xy + y^2 + 2x + 6y$$
$$\text{s.t. } -x + y - 2 \leq 0, -y \leq 0$$

9. Testing problem thirteen (T13)[2]

$$\min_{x} \quad F(x,y) = (x_1 - y_2)^4 + (y_1 - 1)^2 + (y_1 - y_2)^2$$
$$\text{s.t. } -x_1 \leq 0$$
$$\min_{y} \quad f(x,y) = 2x_1 + e^{y_1} + y_1^2 + 4y_1 + 2y_2^2 - 6y_2$$
$$\text{s.t. } 6x_1 + 2y_1^2 + e^{y_2} - 15 \leq 0, -y_1 \leq 0, y_1 - 4 \leq 0$$
$$5x_1 + y_1^4 - y_2 - 25 \leq 0, -y_2 \leq 0, y_2 - 2 \leq 0$$

Methods of Artificial Intelligence in Blind People Education*

Bohdan Macukow and Wladyslaw Homenda

Faculty of Mathematics and Information Science
Warsaw University of Technology
Plac Politechniki 1, 00-660 Warszawa, Poland
{macukow, homenda}@mini.pw.edu.pl

Abstract. This paper presents the idea of recognition of music symbols to help the blind people reading music scores and operating music notation. The discussion is focused on two main topics. The first topic is the concept of the computer program, which recognizes music notation and processes music information while the second is a brief presentation of music processing methods including recognition of music notation - Optical Music Recognition technology - based on artificial neural networks. The short description and comparison of effectiveness of artificial neural networks is also given.

1 Introduction

Over the past few decades computers developed enormously. Along with the progress in hardware the researchers have been working hard to bring on computers to the activities of everyday life. Starting from turning raw interfaces to user friendly communication devices new methods have been studied and developed to make the computers not only efficient but mechanistic tools (like typewriters and counting machines) but also interacting with the human partners in an intelligent way. This required, of course, the use of methods that firmly belong to the domain of Artificial Intelligence.

In this paper we attempt to study an application of methods of Artificial Intelligence in the real life computer program that is supposed to handle musical notations. The term "Artificial Intelligence", though widely used by computer researchers, has neither a common definition nor is it uniquely understood by the academic community. However, it is not our aim to provoke a discussion on what artificial intelligence is and which methods does it embed. Instead, we rather use the term in a common sense though in an intuitive way.

Computer-based music processing methods have been developing since sixties, c.f. [13] and have found their commercial applications during last two decades, c.f. [5]. Music processing embraces several types of computer programs, including

* This work is supported under State Committee for Scientific Research Grant no 3T11C00926, years 2004-2007.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1179–1188, 2006.
© Springer-Verlag Berlin Heidelberg 2006

MIDI sequencers, AUDIO players and notation editors. However, these kinds of programs, in their basic forms, cannot be classified as employing artificial intelligence methods; they are rather simple tools like record players or music analogs of typewriters. Above and beyond implementation of simple computer programs for music processing more sophisticated methods have been developed that truly belong to the field of AI. Two most important areas of utilization of AI methods are: knowledge representation and processing focused on music information, and pattern recognition exploited in music notation recognition and recognition of music in audio form.

Music computer program for the blind is what is possible at the cutting-edge of technology. There are many commercial programs for recognition of music notation processing that have been developed during the last decade. In early nineties MIDISCAN was developed. It was then superseded by the Smart Score [17], SharpEye and PhotoScore in the forthcoming years. Several notation editors as Smart Score [17], Finale, Sibelius and many MIDI sequencers were in use for music processing. There are, however, only a few programs of music processing for the blind musicians, c.f [16,18].

In this paper we discuss application of artificial intelligence methods in music education for the blind. The discussion is focused on two aspects of computer processing of music: information representation and processing and optical music recognition. We then present a concept of computer aimed specifically at the blind people.

2 Representation and Processing of Music Information

Knowledge representation and processing is the most important part of any music processing system. Music itself is one of human communication languages. It has extremely high level of sophistication, has not been codified in its wholeness and is still evolving. Music notation, an emanation of music, is a description of music in graphical form. Music notation can also be seen as human communication tool, it is highly complicated, its codification does not describe the whole notation and it is still evolving (like other tools of human activities). Music processing is governed by rules that neither are unique, nor complete, nor certain. Music processing cast on music notation is characterized by the same features as music processing. All these features require carefulness in music notation representation. In fact, music notation is a two dimensional language in which the geometrical relations between its symbols if of similar importance that the symbols alone. Music notation representation format has to store data about symbols, symbols placement and also contextual information about relative placement of symbols. Incorrectly design music notation representation will result in difficulties in music processing and even may make such processing impossible.

There are two important aspects related to music representation: structure of music notation and music description. The first aspect is a derivative of music structure (title, composer, parts of music piece, instruments, voices, etc.) and of

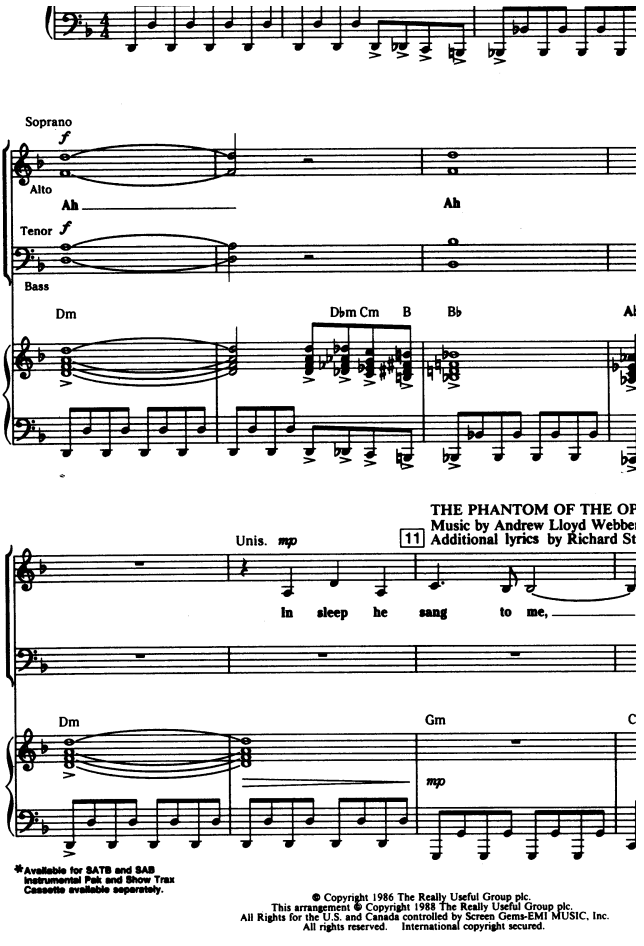


Fig. 1. An example of music notation page

geometrical limits of paper sheets (breaking music to measures, systems, pages, removing silent instruments from systems, etc.), c.f. Figure 1.

Another aspect of music representation is related to information storage about music notation symbols and their properties. Music notation has complicated structure with many implicit relations between items of music data. Music symbols vary in size, shape and are arranged in much more complex and confusing way. In fact, music notation is a two dimensional language in which the geometrical relations between its symbols if of similar importance that the symbols alone. Therefore any music notation representation has to store data about symbols, symbols placement, properties of symbols, suggestions and indications for performers, etc. It also must store contextual information about relative placement of symbols and allow for repossessing contextual information indirectly expressed by the notation.

3 Acquiring Music Information

Any music processing system must be supplied with music information. Manual inputs of music symbols are the easiest and typical source of music processing systems. Such inputs could be split in two categories. One category includes inputs form - roughly speaking - computer keyboard (or similar computer peripheral). Such input is usually linked to music notation editor, so it affects computer representation of music notation. Another category is related to electronic instruments. Such input usually produce MIDI commands which are captured by a computer program and collected as MIDI file representing live performance of music.

Besides manual inputs we can distinguish inputs automatically converted to human readable music formats. The two most important inputs of automatic conversion of captured information are automatic music notation recognition which is known as Optical Music Recognition technology and audio music recognition known as Digital Music Recognition technology. In this paper we discuss basics of automatic music notation recognition as a source of input information feeding music processing computer system.

3.1 Optical Music Recognition

Printed music notation is scanned to get image files in TIFF or similar format. Then, OMR technology converts music notation to the internal format of computer system of music processing. The structure of automated notation recognition process has two distinguishable stages: location of staves and other components of music notation and recognition of music symbols. The first stage is supplemented by detecting score structure, i.e. by detecting barlines and then systems and systems' structure and detecting other components of music notation like title, composer name, etc. The second stage is designed on finding placement and classifying symbols of music notation. The step of finding placement of music notation symbols, also called segmentation, must obviously precede the step of classification of music notation symbols. However, both steps segmentation and classification often interlace: finding and classifying satellite symbols often follows classification of main symbols.

Staff lines and systems location
Music score is a collection of staves which are printed on sheets of paper, c.f. [6]. Staves are containers to be filled in with music symbols. Stave(s) filled in with music symbols describe a part played by a music instrument. Thus, stave assigned to one instrument is often called a part. A part of one instrument is described by one stave (flute, violin, cello, etc.) or more staves (two staves for piano, three staves for organ).

Staff lines location is the first stage of music notation recognition. Staff lines are the most characteristic elements of music notation. They seem to be easily found on a page of music notation. However, in real images staff lines are distorted raising difficulties in recognition. Scanned image of a sheet of music is often skewed, staff line thickness differs for different lines and different parts

of stave, staff lines are not equidistant and are often curved, especially in both endings of the stave, staves may have different sizes, etc., c.f. [5,6] and Figure 2.

Having staves on page located, the task of system detection is performed. Let us recall that the term system (at a page of music notation) is used in the meaning of all staves performed simultaneously and joined together by beginning barline. Inside and ending barlines define system's structure. Thus, detection of systems and systems' structure relies on finding barlines.

Score structure analysis

Sometimes one stave includes parts of two instruments, e.g. simultaneous notation for flute and oboe or soprano and alto as well as tenor and bass. All staves, which include parts played simultaneously, are organized in systems. In real music scores systems are often irregular, parts which not play may be missing.

Each piece of music is split into measures which are rhythmic, (i.e. time) units defined by time signature. Measures are separated from each other by barlines.

The task of score structure analysis is to locate staves, group them in systems and then link respective parts in consecutive systems. Location of barlines depicts measures, their analysis split systems into group of parts and defines repetitions.

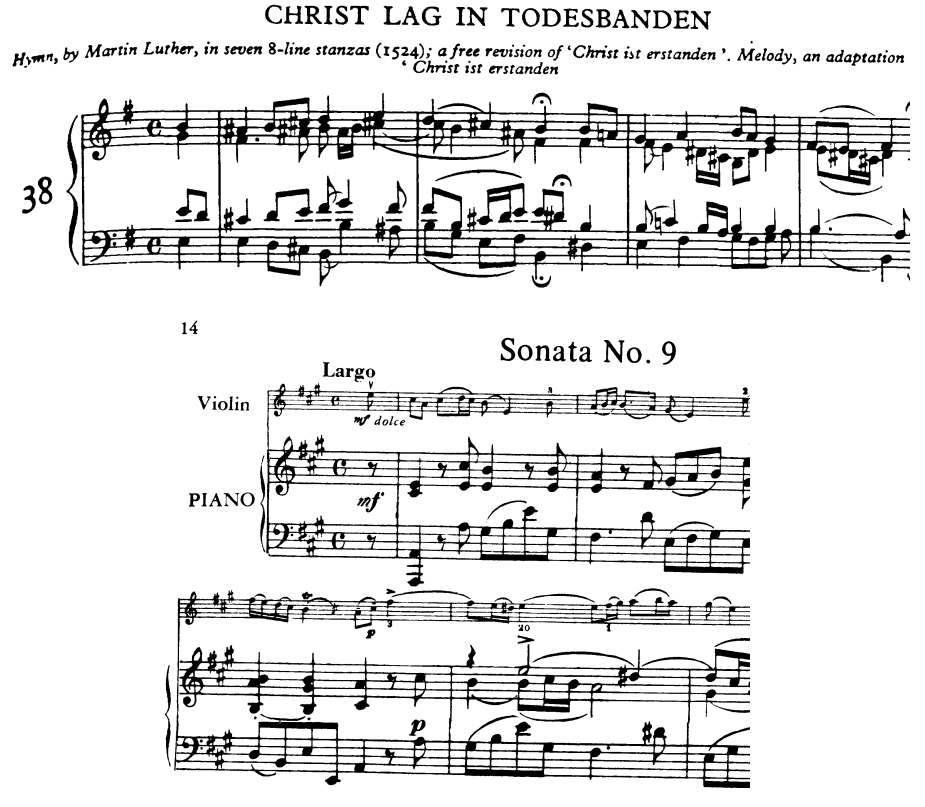



Fig. 2. Examples of real notations subjected to recognition

Music symbol recognition
Two important problems are raised by symbol recognition task: locating and classifying symbols. Due to irregular structure of music notation, the task of finding symbol placement decides about final symbol recognition result. Symbol classification could not give good results if symbol localization is not well done. Thus, both tasks are equally important in recognizing of music symbols.

Since no universal music font exits, c.f. Figures 1 and 2, symbols of one class may have different forms. Also size of individual symbols does not keep fixed proportions. Even the same symbols may have different sizes in one score. Besides usual noise (printing defects, careless scanning) extra noise is generated by staff and ledger lines, densely packed symbols, conflicting placement of other symbols, etc.

A wide range of methods are applied in music symbol recognition: neural networks, statistical pattern recognition, clustering, classification trees, etc., c.f. [1,2,4,9,11]. Classifiers are usually applied to a set of features representing processed symbols, c.f. [9]. In next section we present application of neural networks as example classifier.

3.2 Neural Networks as Symbol Classifier

Having understood the computational principles of massively parallel interconnected simple neural processors, we may put them to good use in the design of practical systems. But neurocomputing architectures are successfully applicable to many reallife problems. The single or multilayer fully connected feedforward or feedback networks can be used for character recognition, c.f. [8].

Experimental tests were targeted on classification of quarter, eight and sixteen rests, sharps, flats and naturals, c.f. Figure 3 for examples music symbols. To reduce dimensionality of the problem, the images were transformed to a space of 35 features. The method applied in feature construction was the simplest one, i.e. they were created by hand based on understanding of the problem being tackled. The list of features included the following parameters computed for bounding box of a symbol and for four quarters of bounding box spawned by symmetry axes of the bounding box:

- mean value of vertical projection,
- slope angle of a line approximating vertical projection,
- slope angle of a line approximating histogram of vertical projection;
- general horizontal moment m_{10},

Fig. 3. Printed symbols of music notation - distortions, variety of fonts

Table 1. Chromatic symbols recognition rate of selected classifiers

flats	sharps	naturals	Classifier	
99.11%	97.98%	98.81%	backpropagation	5 - 3 - 2
92.30%	86.81%	91.74%	counterpropagation	15 - 8 - 1
96.52%	93.33%	89.11%	counterpropagation	25 - 8 - 1

– general vertical moment m_{01},
– general mixed moment m_{11}.

The following classifiers were utilized: backpropagation perceptron, feedforward counterpropagation maximum input network and feedforward counterpropagation closest weights network. An architecture of neural network is denoted by a triple input - hidden - output which identifies the numbers of neurons in input, hidden and output layers, respectively, and does not include bias inputs in input and hidden layers.

Table 1 presents results for three symbols on music notation: flats, sharps and naturals, c.f. [9]. Classifier applied: backpropagation perceptron, feed-forward counterpropagation maximum input network and feedforward counterpropagation closest weights network. An architecture of neural network is denoted by a triple *input - hidden - output* which identifies the numbers of neurons in input, hidden and output layers, respectively, and does not include bias inputs in input and hidden layers.

4 Braille Score - Bringing Research to Practice

Braille Score is a computer program to process music information. Braille Score is an integrated music processing computer program directed to a broad range of people. It is the part of the project under development in Warsaw University of Technology. The program together with a man creates an integrated system. It has special features allowing its usage by blind people. It is intended as a tool supporting blind people in dealing with music. Its important application could be placed in music education of blind students. Braille Score directly applies methods of artificial intelligence in practice. Its main modules deal with music information acquisition, storage and processing, communication with both blind people and good eyesight users. Main functions of Braille Score are:

– creating scores from scratch,
– capturing existing music printings and converting them to electronic version,
– converting music between different formats of music representation,
– processing music, e.g. transposing musing to different keys, extracting parts from given score, creating a score from given parts,
– preparing teaching materials,
– creating and storing own compositions and instrumentation.

4.1 Representing and Processing of Music Information

A software for music processing is usually built around the model of an electronic format of music representation. Such a format stores music in a computer memory, processing it, exchange music data between different music equipment but also present music in a form of music notation. But from the other side the proper recognition of music notation is still under development. Information acquired at the stage of pattern recognition has to be stored in the form allowing for its further usage. This stage is based on the methods of knowledge representation and processing, c.f. [3,5,15]. The following topics could exemplify music knowledge storage and processing:

- designing a format of music representation,
- recognizing context relations: inheriting accidentals, linking articulation and ornamentation symbols to notes, linking lyrics to notes,
- structuring recognized music symbols, grouping notes into chords, grouping chords into beamed sequences, grouping accidentals into key signatures,
- identifications of rhythmic groupings,
- identifications of voices,
- converting music between different formats of music representation.

4.2 Acquiring Music Information

Braille Score is capable to acquire music information from several sources. Its main and distinguishable input source is printed music notation, which is subjected to automatic recognition of the structure and symbols. Only limited set of music notation symbols is intended to be recognized and process in Braille Score at the current version. The set of recognized symbols includes notes, chords, rests, accidentals, clefs, bar lines, key signatures, time signatures, change of key and time signature. Assuming future development of Braille Score, BSF include wider set of symbols including rhythmic, articulation and ornamentation figures and other symbols.

Braille Score can also read music information represented in MIDI, NIFF, MusicXML and Braille Music formats. Conversely, Braille Score can also output music information to the same sources. This way it is able to exchange music information with broad range of music software.

4.3 User Interface Extensions for Blind People

Braille Score is addressed to blind people. Its user interface extensions allow blind user to master the program and to perform operations on music information. The most important feature of Braille Score is its ability to read, edit and print music information in Braille format. Blind user is provided the following elements of interface: Braille Notation editor, keyboard as input tool, sound communicator.

Blind people do not use pointing devices. In consequence, all input functions usually performed with mouse must be mapped to computer keyboard. Massive communication with usage of keyboard requires careful design of interface mapping to keyboard, c.f. [12].

Blind user usually do not know printed music notation. Their perception of music notation is based on Braille music notation format, c.f. [10] presented at Braille display or punched sheet of paper. In such circumstances music information editing must be done on Braille music notation format. Since typical Braille display is only used as output device, such editing is usually done with keyboard as input device. In Braille Score Braille representation of music is online converted to internal representation and displayed in the form of music notation in usual form. This transparency will allow for controlling correctness and consistency of Braille representation, c.f. [12].

Sound information is of height importance for blind user of computer program. Wide spectrum of visual information displayed on display screen for user with good eyesight could be replaced by sound information. Braille Score provides sound information of two types. The first type of sound information collaborates with screen readers, computer programs dedicated to blind people which could read contents of display screen and communicate it to user in the form of synthesized speech. This type of communication is supported by contemporary programming environments. Braille Score uses tools provided by Microsoft .NET programming environment. The second type of sound information is based on own Braille Score tools. Braille Score has embedded mechanism of sound announcements based on own library of recorded utterances.

5 Conclusions

In this paper we describe a concept of Braille Score the specialized computer program which should help blind people to deal with music and music notation. The use of artificial intelligence tolls like neural networks can improve the program part devoted for the recognition of the music symbols. The first results with Braille Score show its to be a practical and useful tool.

References

1. Bainbridge D., Bell T., The challenge of optical music recognition, Computers and the Humanities 35 (2001) 95-121.
2. Carter N. P., Bacon R. A., Automatic Recognition of Printed Music pp. 456-465 in: Structured Document Analysis, Analysis, H.S.Baird, H.Bunke, K.Yamamoto (Eds), Springer-Verlag, 1992.
3. Dannenberg R., Music Representation Issues, Techniques, and Systems, Computer Music Journal (17:3) (1993) 20-30.
4. Fujinaga I., Adaptive optical music recognition, 16th Inter. Congress of the Inter. Musicological Society, Oxford, Oxford University Press, 2001.
5. Homenda W., Automatic recognition of printed music and its conversion into playable music data, Control and Cybernetics, vol. 25, no. 2 (1996) pp. 353-367.
6. Homenda W., Granular Computing as an Abstraction of Data Aggregation - a View on Optical Music Recognition, Archives of Control Sciences, Vol. 12 (XLVIII) (2002) No. 4, pp 433-455.
7. Homenda W., Optical Music Recognition: the Case Study of Pattern Recognition, in: Computer Recognition Systems, Kurzyski et al (Eds.), pp. 835-842, Springer Verlag, 2005.

8. Homenda W., Luckner M., Automatic Recognition of Music Notation Using Neural Networks, Proc. of the International Conference On Artificial Intelligence and Systems, Div-nomorskoye, Russia, September 3-10, 2004.
9. Homenda W., Mossakowski K., Music Symbol Recognition: Neural Networks vs. Statistical Methods, EUROFUSE Workshop On Data And Knowledge Engineering, Warsaw, Poland, September 22 - 25, 2004, pp. 265-271
10. Krolick B., How to Read Braille Music, 2nd Edition, Opus Technologies, 1998.
11. McPherson J. R., Introducing feedback into an optical music recognition system, Third Internat. Conf. on Music Information Retrieval, Paris, France, 2002,.
12. Moniuszko T., Design and implementation of music processing computer program for blind people (in Polish), Master Thesis, Warsaw University of Technology, Warsaw, 2006.
13. Pruslin D. H., Automatic Recognition of Sheet Music, PhD Thesis, Massachusetts Institute of Technology, 1966.
14. MIDI 1.0, Detailed Specification, Document version 4.1.1, February 1990.
15. http://www.musique.umontreal.ca/personnel/Belkin/NIFF.doc.html 1990.
16. http://www.dancingdots.com/goodfeel.htm
17. http://www.musitek.com/
18. http://members.optusnet.com.au/∼terryk/toccata.htm

Neural Networks and the Estimation of Hands' Strength in Contract Bridge

Krzysztof Mossakowski and Jacek Mańdziuk*

Faculty of Mathematics and Information Science, Warsaw University of Technology,
Plac Politechniki 1, 00-661 Warsaw, Poland
{mossakow, mandziuk}@mini.pw.edu.pl

Abstract. This paper is focused on a *Double Dummy Bridge Problem* (DDBP) which consists in answering the question of how many tricks are to be taken by a pair of players assuming perfect play of all four sides with all cards being revealed. Several experiments are also presented in a variant of DDBP in which the information about to whom of the two players in a given pair a particular card belongs to is hidden. In contrast to our previous works, which were devoted to no trump contracts, here we concentrate on suit contracts. Several interesting conclusions are drawn from comparison of weight patterns of networks trained on no trump contracts only vs. those trained exclusively on suit contracts. The ultimate goal of this research is to construct a computer program playing the game of contract bridge using neural networks and other CI techniques with the basic assumption of using zero-human-knowledge approach and to learn purely on examples.

1 Introduction

The game of bridge has attracted attention of many AI and CI researchers, e.g. [1,2,3,4,5]. In particular, some interest was also devoted to DDBP [5,6] which is regarded as an interesting benchmark problem at initial stage of bridge playing program's development.

In this paper we continue our research efforts devoted to DDBP having in mind the ultimate goal - construction of a bridge playing system without explicit presentation of any human expert knowledge concerning the game.

It must be emphasized that the rules of the game were not presented in any form. In all experiments the training data contained only deals (i.e. information about which player each card belongs to) with target information about the number of tricks to be taken by one pair of players.

2 Previous Work – No Trump Contracts

In this section we briefly describe previous results obtained for no trump contracts, published in [7,8,9].

* This work was supported by the Warsaw University of Technology grant no. 504G 1120 0008 000.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1189–1198, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2.1 Various Approaches to DDBP

In [7] several neural network architectures were tested in the DDBP (from NS viewpoint) for no trump contracts. The data was taken from GIB Library [10] which includes 717,102 deals with revealed all hands. Additionally the library provides a number of tricks taken by the pair NS in each contract under the assumption of a perfect play of all sides.

Several feed-forward perceptron neural networks with logistic activation function were created, trained using RProp algorithm and tested in JNNS (Java Neural Network Simulator) environment. In most cases logistic (unipolar sigmoid) activation function was used for all neurons except for the case of representation of data using negative numbers, where the hyperbolic tangent (bipolar sigmoid) activation function was applied.

Two main approaches to coding a deal suitable for neural network input representation were applied. In the first approach each card of each hand was represented by two real numbers: the value (2, 3,..., *king*, *ace*) and the suit - S (*Spades*), H (*Hearts*), D (*Diamonds*), C (*Clubs*). Both real numbers were calculated using a uniform linear transformation to the range $[0.1, 0.9]$ (see [7,8] for details).

In the other deal representation - which was superior to the above described one - the positions of cards in the input layer were fixed, i.e. from the leftmost input to the rightmost one the following cards were represented: 2 of Spades, 3 of Spades, ..., *king* of Clubs, *ace* of Clubs. An input value denoted the hand to which a given card belonged: 1.0 for N, 0.8 for S, -1.0 for W, and -0.8 for E. The simplest network $52 - 1$ accomplished the result $(94.15 \mid 76.15 \mid 31.29)$[1], and the network with one hidden layer: $52 - 25 - 1$ improved this score to $(95.81 \mid 79.95 \mid 34.02)$. A slight modification of the above way of coding a deal consisted in extending the input representation to 104 neurons. The first 52 neurons represented assignment to a pair (value 1.0 for NS pair and -1.0 for WE), and the other 52 ones represented a hand in a pair (1.0 for N or W and -1.0 for S or E). The simplest network: 104-1 accomplished the result $(94.76 \mid 77.52 \mid 32.19)$, and two-hidden layer network $104 - 30 - 4 - 1$, yielded the result $(95.64 \mid 79.63 \mid 33.74)$.

The number of iterations required to learn the training set without overfitting depended mostly on the way of coding a deal. The networks with the first type of coding needed a few tens of thousands iterations, and networks with coding by cards' assignment only several hundred ones.

A few more ways of coding a deal were also tested, but regardless of the problem representation **it was concluded that with the proposed approach exceeding the level of** $(96 \mid 80 \mid 34)$ **is a challenging task.**

As a point of reference simple estimator based on Work point count (*ace* - 4 points, *king* - 3, *queen* - 2, *jack* - 1) was proposed. The number of tricks to be

[1] Each of the three values denotes the fraction *in percent* of test deals for which the network was mistaken, respectively by no more than 2 tricks (94.15%), no more than 1 trick (76.15%), and was perfectly right (31.29%). This notation is used in the whole paper.

taken by NS was estimated as $(13/40) * points_of_NS$. This estimator achieved the result of (86.19 | 61.32 | 22.52) which was significantly inferior to the ones achieved by neural architectures. This result suggests that networks learnt some additional information besides simple estimation of Work points.

2.2 Analysis of Trained Networks

Except for numerical results it is interesting to explore what is the problem representation in connection weights of trained networks. A closer look at this data obtained for $52-25-1$ architectures revealed some interesting observations [8,9].

Firstly, weights of connections between input neurons representing *aces* and *kings* had always the biggest absolute values. This feature was simple to explain (for humans) - these cards were the most important in the game of bridge, especially in no trump contracts.

Secondly, in each trained network there were exactly 4 connections from input to hidden neurons which had absolute values noticeably bigger than all others (about 25.0 vs less than 7.0). Not surprisingly these favored connections started from 4 input neurons assigned to *aces*.

Thirdly, in all networks it was also possible to point out hidden neurons focused on one particular suit, one neuron per suit. Such neuron had much bigger absolute values of connections' weights from inputs representing the suit than weights' values from the rest of inputs. These connections are marked using long-chain lines in Fig. 1.

Another very interesting feature which appeared in all trained networks with 25 hidden neurons was the presence of four hidden neurons each specialized in five cards from one suit: *ten*, *jack*, *queen*, *king*, and *ace* (in Fig. 1 the respective connections are marked using the dotted line). In all these groups the most important were *queens* and *kings*, *jacks* were less important, but still much more relevant than *aces* and *tens*. The hypothesis is that these hidden neurons are responsible for very important aspect of the play of bridge - *the finesses*.

Finally, there existed hidden neurons with values of connections to the output close to zero. The authors were unable to find any pattern in their weights of connections from the inputs. The number of such neurons increased in case of more complicated networks.

3 Current Experiment – Suit Contracts

The new research results presented in this paper still concern the solution of DDBP, but, unlike in the previous research, the focus is now on suit contracts. For comparison with previous results in most cases the same network architectures and similar ways of coding a deal are used. These results are presented and discussed in section 3.2.

Another interesting issue is to check whether information about the exact hand location of a given card is really important in the training data, or maybe

Table 1. Results (in %) obtained for the test sets for $52-25-1$ networks. **NT** denotes *No Trump contracts*.

Description	Results
NT; input values: $N:1.0$, $S:0.8$, $W:-1.0$, $E:-0.8$	(95.81 | 79.95 | 34.02)
NT; input values: $N:1.0$, $S:0.8$, $W:-1.0$, $E:-1.0$	(95.97 | 80.46 | 34.35)
NT; input values: $NS:1.0$, $WE:-1.0$	(96.07 | 80.88 | 34.66)
Suit contracts; input values: $NS:0.5$, $WE:-0.5$	(98.68 | 87.88 | 40.11)
the above network tested on **NT**	(91.64 | 69.21 | 26.06)
NT and **suit contracts**; input values: $NS:0.5$, $WE:-0.5$	(97.72 | 84.90 | 37.56)
the above network tested on **suit contracts** only	(98.57 | 87.24 | 39.43)
the above network tested on **NT** only	(94.30 | 75.50 | 30.09)
Spades contracts; input values: $NS:1.0$, $WE:-1.0$	(98.77 | 88.00 | 40.13)
the above network tested on **Hearts contracts**	(59.18 | 39.09 | 14.12)
the above network tested on **Diamonds contracts**	(58.89 | 38.67 | 13.51)
the above network tested on **Clubs contracts**	(58.86 | 38.90 | 13.77)
Hearts contracts; input values: $NS:1.0$, $WE:-1.0$	(98.65 | 87.81 | 40.18)
Diamonds contracts; input values: $NS:1.0$, $WE:-1.0$	(98.66 | 87.68 | 39.96)
Clubs contracts; input values: $NS:1.0$, $WE:-1.0$	(98.73 | 87.90 | 40.02)

it would be enough to locate cards as belonging to either NS or WE pairs. This issue is discussed in section 3.1.

Finally, it seems worth investigating whether the results are repeatable, i.e. whether within the ensemble of neural nets the results for a given contract would be the same. This question is considered in section 3.3.

All results are summarized in Table. 1.

3.1 Hiding Opponents' Cards in NT Contracts

In previous work with NT contracts and all 4 hands revealed each of 52 input neurons pointed out the hand to which a given card was assigned, namely input equal to 1.0 denoted N hand, 0.8: S hand, -1.0: W, and -0.8: E. Let us recall that $52-25-1$ network trained using this way of coding achieved the result of (95.81 | 79.95 | 34.02).

Hiding opponent's cards was carried out by applying the following values for hands' description: $N:1.0$, $S:0.8$, W and $E:-1.0$. Results yielded by a network with the same architecture (52-25-1) were slightly improved: (95.97 | 80.46 | 34.35).

Surprisingly, hiding also information about cards' assignment in the pair NS, i.e. using input values: N and $S:1.0$, W and $E:-1.0$, yielded another slight improvement: (96.07 | 80.88 | 34.66).

These results prove that for data representation chosen in the experiments information about exact hand location of each card is not helpful and sometimes even misleading. The explanation of this phenomenon is one of our current research goals.

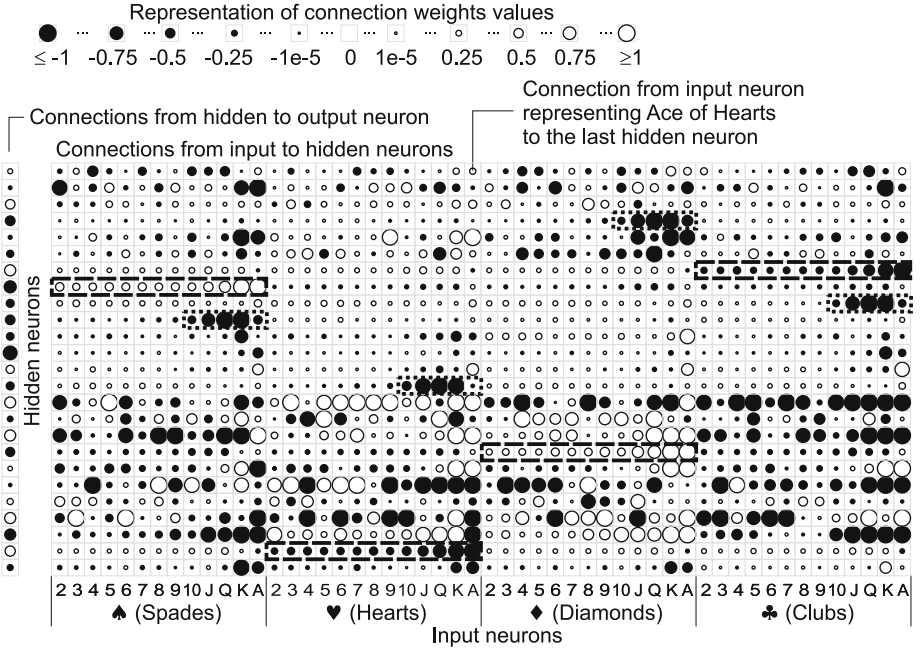


Fig. 1. NT contracts. Weights of connections of trained neural network with 25 hidden neurons ($52 - 25 - 1$). Each circle represents the weight of one connection. If the circle is placed in the leftmost column, it represents the weight of connection from hidden to output neuron, otherwise - from input to hidden neuron. The radius of the circle represents the absolute value of connection weight. Black circles denote negative and white ones positive weights.

3.2 Suit Contracts

In the experiments with suit contracts a deal was coded in the following way. Cards of the trump suit had input values equal to 1.0 for players N, S or -1.0 for players W, E. Cards of other suits: 0.5 for N, S and -0.5 for W, E. When no trump contract was presented to the network, all cards had input values equal to 0.5 or -0.5, resp. for NS and WE pairs.

Neural network with 25 hidden neurons trained on $400,000$ examples ($100,000$ deals repeated 4 times - once for each trump suit) accomplished the results (98.68 | 87.88 | 40.11). Testing this network on no trump contracts (not present in the training set) led to poorer score: (91.64 | 69.21 | 26.06).

When no trump contracts were added to the training and test sets, network with the same architecture yielded the result of (97.72 | 84.90 | 37.56). Results of testing this network separately on suit contracts were significantly better (98.57 | 87.24 | 39.43) than results of tests on no trump contracts only (94.30 | 75.50 | 30.09).

The network trained only on Spades contracts achieved the result of (98.77 | 88.00 | 40.13). Results of testing this network based exclusively on Hearts

contracts: (59.18 | 39.09 | 14.12), Diamonds contracts: (58.89 | 38.67 | 13.51), and Clubs ones: (58.86 | 38.90 | 13.77) confirmed the effect of "specialization". Furthermore, the networks trained exclusively on other suits yielded similar results, i.e. (98.65 | 87.81 | 40.18) for training on Hearts contracts only, (98.66 | 87.68 | 39.96) for Diamonds contracts, and (98.73 | 87.90 | 40.02) for Clubs contracts. It is an interesting observation that results of specialized networks presented above are on the same level as results of networks trained on all suit contracts simultaneously. Closer investigation of this counterintuitive effect is planned in the near future.

The above results imply that **no trump contracts and suit contracts should be treated separately.** Connection weights of a network trained using suit contracts only are presented in Fig. 2. The main difference between this figure and Fig. 1 representing connection weights of a network trained exclusively on no trump contracts is visibly bigger relative importance of the lowest cards in a deal. This observation is also confirmed by Table. 2, where connection weights of a network without hidden neurons (52-1) are compared. This conclusion is in line with human bridge players' experience. Moreover, it is clear from the table that no significant differences between suits exist which again is a desirable effect.

Coming back to comparisons between Fig. 1 and Fig. 2 there can also be identified some common patterns in both figures, e.g. cards from *aces* to *tens* are the most important - connections from input neurons representing them to hidden neurons have the biggest absolute values. It is also possible to point out

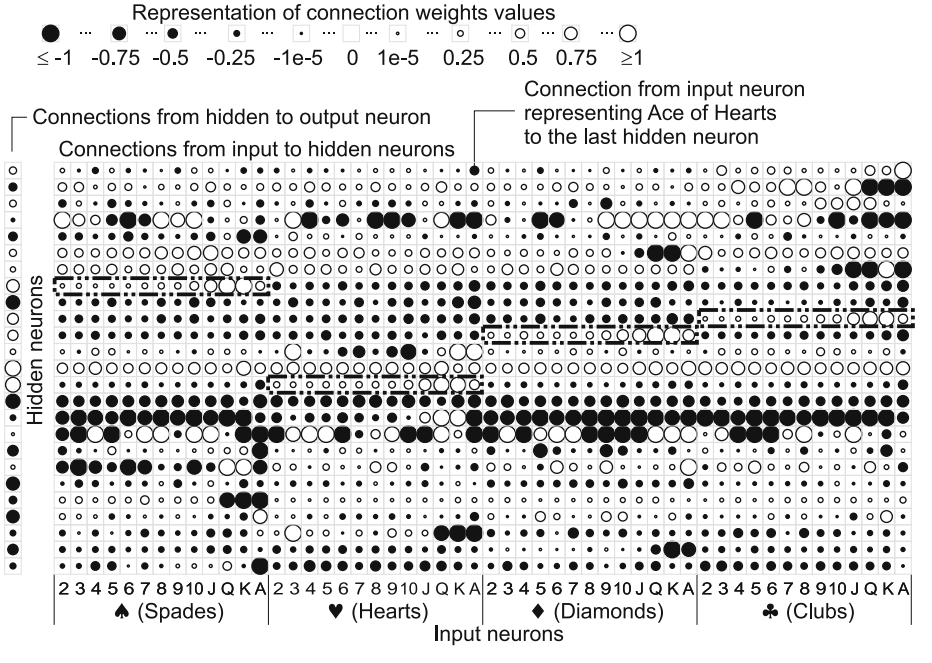


Fig. 2. Suit contracts. Weights of connections of trained network with 25 hidden neurons ($52 - 25 - 1$). See description of Fig. 1.

Table 2. Values of connections between neurons of two networks without hidden neurons $52-1$ trained respectively on no trump and suit contracts. Values were linearly scaled to interval (0,4).

	NT Contracts				Suit Contracts			
Card's value	S	H	D	C	S	H	D	C
2	0.342	0.327	0.329	0.342	1.660	1.670	1.668	1.667
3	0.340	0.334	0.328	0.353	1.664	1.667	1.663	1.660
4	0.347	0.314	0.351	0.345	1.669	1.655	1.669	1.685
5	0.341	0.332	0.341	0.344	1.660	1.673	1.676	1.663
6	0.356	0.349	0.339	0.329	1.684	1.685	1.680	1.688
7	0.380	0.331	0.354	0.356	1.680	1.684	1.687	1.697
8	0.358	0.361	0.375	0.400	1.709	1.719	1.718	1.723
9	0.496	0.469	0.461	0.473	1.782	1.791	1.780	1.783
10	0.660	0.663	0.671	0.684	1.921	1.916	1.918	1.938
J	1.047	1.032	1.056	1.030	2.174	2.167	2.177	2.172
Q	1.676	1.688	1.675	1.656	2.569	2.569	2.572	2.565
K	2.643	2.643	2.677	2.655	3.207	3.210	3.220	3.216
A	3.975	3.971	3.966	3.989	3.982	3.984	3.973	3.995

hidden neurons specialized in one suit, additionally with connections from *kings* and *queens* being the most important.

In summary, it should be emphasized that the above described weight patterns were observed also when other training sets had been used.

3.3 Reliability of Results

In order to check the reliability of results, 4 networks with one hidden layer of 25 neurons each, differing only by initial, randomly chosen connection weights, were trained based on the same set of deals. This experiment was aimed at checking the number of deals from the training set for which all 4 networks would learn the same number of tricks to be taken by NS.

For no trump contracts all 4 networks estimated the same number of tricks in 61.23% of contracts. In 37.93% of contracts estimated numbers of tricks differed by 1 trick, in 0.81% by 2 tricks and in 0.03% by 3 tricks. The same experiment for suit contracts output the following results: for 63.40% of contracts all networks were unanimous, for 36.56% of contracts there was a 1 trick difference, and for 0.04% of them the difference was equal to 2 tricks.

In 98.13% of testing deals for no trump contracts, and in 99.53% for suit contracts, real output values of all 4 trained networks differed by no more than 0.06. In these experiments target number of tricks was calculated using a uniform linear transformation to the range $[0.1, 0.9]$, so value 0.06 was the range of real output values of networks for each number of tricks.

The results prove that the confidence in the learning process is high, and the training results are repeatable.

3.4 Results by the Target Number of Tricks

Results of a network $52 - 25 - 1$ trained on Spades contracts were investigated in detail in order to test whether the efficacy of the system varies for different numbers of target tricks. The test set containing $100,000$ deals was divided into subsets which were then tested individually. Results are presented in Table. 3.

Only the results for 0, 1, 12, and 13 tricks are significantly worse than the result attained for the whole test set (the last row of the Table). Results for the other subsets are on the similar level, in spite of considerable differences in the number of deals in these subsets (e.g. $4,225$ deals with 11 tricks vs. $12,927$ ones with 6 tricks).

Table 3. Results for subsets of a testing set achieved by a network $52 - 25 - 1$ trained on Spades contracts

Target number of tricks	Number of deals	Results
0	1,138	(93.32 | 66.61 | 12.30)
1	2,725	(97.39 | 81.21 | 34.53)
2	5,156	(98.10 | 86.66 | 40.73)
3	8,043	(98.93 | 88.96 | 41.41)
4	10,447	(98.94 | 89.04 | 40.36)
5	12,201	(98.85 | 88.67 | 40.80)
6	12,927	(99.03 | 88.75 | 41.32)
7	12,709	(99.10 | 88.99 | 40.50)
8	11,467	(99.28 | 89.29 | 40.46)
9	9,618	(99.14 | 89.19 | 42.14)
10	6,866	(98.89 | 88.45 | 40.58)
11	4,225	(97.94 | 85.87 | 42.32)
12	1,935	(97.57 | 81.71 | 31.94)
13	543	(94.66 | 73.85 | 9.39)
All	100,000	(98.77 | 88.00 | 40.13)

3.5 Sample Deals

Two sample deals are presented in Fig. 3. The first deal (Fig. 3(a)) was included in the reliability test described in previous section. Each of 4 networks estimated different number of tricks to be taken by the pair NS, i.e. $5, 6, 7$, and 8. A closer analysis of this deal revealed that the number of tricks for NS in no trump contract depends on information who makes defender's lead. Defender's lead from N or S hand enables to take 8 or 7 tricks, resp. On the other hand, defender's lead from W or E limits the number of tricks for NS to 6 or 5, resp. Information about defender's lead wasn't included in the input data (desired output values were specified for defender's lead from W hand). Hence the behavior of neural networks can be "justified".

The second deal (Fig. 3(b)) is a successful example of learning suit contracts - the network predicted grand slam of spades for NS with only 26 points (Work point count). Please note that also grand slam of Hearts for NS is possible, but in

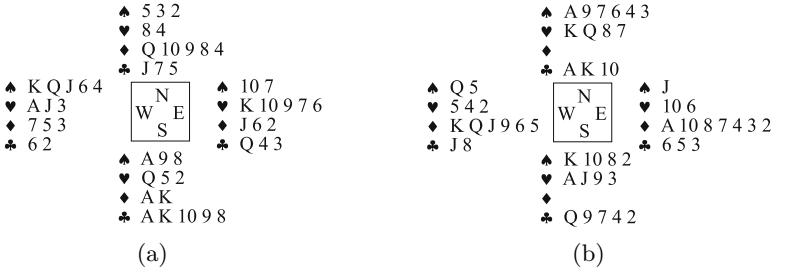


Fig. 3. Sample deals

this case the network yielded only 12 tricks. **It is an interesting observation that the network estimated higher longer suit (Spades) than stronger one (Hearts)**.

4 Conclusions and Future Research

Based on the results, it can be concluded that in the DDBP it is advisable to train neural networks separately for no trump contracts and suit ones.

Reliability tests show that the confidence in training process is high and the results are repeatable.

Interesting patterns found in figures presenting networks connection weights' values (Fig. 1 and Fig. 2), and their reasonable explanation based on human experience in the game of bridge, look very promising and suggest taking into consideration the possibility of automatic, unguided discovering of knowledge hidden in connection weights.

Currently we are focused on defining an automatic input data preprocessing system capable to find functional similarities in deals and based on that allowing to either preprocess (transform) training data or divide it into subsets suitable for specialized networks. Similarity of results achieved for subsets of deals with the same number of target tricks (Table. 3) implies that dividing training set according to the number of target tricks and then applying specialized networks to these subsets seems to be a promising direction.

The next "big" step is to advance this research into play phase.

References

1. Kohle, M., Schonbauer, F.: Experience gained with a neural network that learns to play bridge. In: 5th Austrian Artificial Intelligence Meeting. (1995) 224–229
2. Sarkar, M., Yegnanarayana, B., Khemani, D.: Application of neural network in contract bridge bidding. In: Proc. of National Conf. on Neural Networks and Fuzzy Systems, Anna University, Madras (1995) 144–151
3. Smith, S., Nau, D., Throop, T.: Computer bridge: A big win for AI planning. AI Magazine **19** (1998) 93–106
4. Frank, I., Basin, D.A.: Search in games with incomplete information: A case study using bridge card play. Artificial Intelligence **100**(1-2) (1998) 87–123

5. Ginsberg, M.: Gib: Imperfect information in a computationally challenging game. Journal of Artificial Intelligence Research **14** (2001) 303–358
6. Wheen, R.: Brute force programming for solving double dummy bridge problems. In Levy, D., Beal, D., eds.: Heuristic Programming in Artificial Intelligence: the first computer olympiad. Ellis Horwood, Chichester (1989) 88–94
7. Mossakowski, K., Mańdziuk, J.: Artificial neural networks for solving double dummy bridge problems. In Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A., eds.: Artificial Intelligence and Soft Computing - ICAISC 2004, 7th International Conference, Zakopane, Poland, June 7-11, 2004, Proceedings. Volume 3070 of Lecture Notes in Computer Science., Springer (2004) 915–921
8. Mańdziuk, J., Mossakowski, K.: Looking inside neural networks trained to solve double-dummy bridge problems. In: 5th Game-On International Conference on Computer Games: Artificial Intelligence, Design and Education (CGAIDE 2004), Reading, UK (2004) 182–186
9. Mossakowski, K., Mańdziuk, J.: Weight patterns in the networks trained to solve double dummy bridge problem. In Hryniewicz, O., et al., eds.: Issues in Intelligent Systems. Paradigms. Exit (2005) 155–165
10. Ginsberg, M. (http://www.cirl.uoregon.edu/ginsberg/gibresearch.html)

Mining Travel Resources on the Web Using L-Wrappers

Elvira Popescu[1], Amelia Bădică[2], and Costin Bădică[1]

[1] University of Craiova, Software Engineering Department
Bvd.Decebal 107, Craiova, RO-200440, Romania
{popescu_elvira, badica_costin}@software.ucv.ro
[2] University of Craiova, Business Information Systems Department
A.I.Cuza 13, Craiova, RO-200585, Romania
ameliabd@yahoo.com

Abstract. The work described here is part of an ongoing research on the application of general-purpose inductive logic programming, logic representation of wrappers (L-wrappers) and XML technologies (including the XSLT transformation language) to information extraction from the Web. The L-wrappers methodology is based on a sound theoretical approach and has already proved its efficacy on a smaller scale, in the area of collecting product information. This paper proposes the use of L-wrappers for tuple extraction from HTML in the domain of e-tourism. It also outlines a method for translating L-wrappers into XSLT and illustrates it with the example of a real-world travel agency Web site.

1 Introduction

E-tourism is a leading area in e-commerce, with an increasing number of travel agencies offering their services through online transaction brokers ([12]). They provide to human users information in areas like hotels, flights, trains or restaurants, in order to help them to plan their business or holiday trips. Travel information is heterogeneous and distributed, and there is a need to gather, search, integrate and filter it efficiently ([8]). Typically, this information is published by dynamically filling-in HTML templates with structured data taken from relational databases. Therefore, collecting information about travel resources and converting it to a form suitable for automated processing is an appropriate task for information extraction using machine learning ([10]).

This paper approaches the problem of mining travel resources using logic wrappers (L-wrappers) ([1]). As shown in [2], L-wrapper technology was successfully applied to extract tuples from Web pages written in HTML.

Note that the application of logic representations and machine learning to information extraction from the Web is not an entirely new field; several approaches and tool descriptions have already been proposed and published ([5,6,7,10,11,13,16,18]). In our opinion, the advantage of our proposal is the use of the right tool for tackling a given task, i.e. for learning extraction rules it employs inductive logic programming (ILP) systems ([15]), and for performing the extraction it employs XSLT technology ([4]).

The rest of the paper is structured as follows. First, we give a short overview of L-wrapper theory. In section 3 we discuss an approach of translating L-wrappers into XSLT stylesheets. We outline a translation algorithm that is then demonstrated with the help of an example. In section 4 we present some experiments and discuss their results. Finally, we conclude the paper, outlining some future research directions.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1199–1208, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 L-Wrappers Background

Web pages (including those publishing travel information) can be regarded as semi-structured data modeled as labeled ordered trees. A wrapper takes a labeled ordered tree and returns a subset of tuples of extracted nodes. L-wrappers use patterns defined as logic rules, which can be learned by using general-purpose ILP systems.

For our purposes, it is convenient to abstract labeled ordered trees as sets of nodes on which certain relations and functions are defined ([1]). Basically, we define two relations between tree nodes: the "parent-child" relation and the "next-sibling" linear ordering relation on the set of children of a node. Furthermore, a label is attached to each tree node, modeling a specific tag from a finite set of tag symbols Σ.

In this framework, a pattern is a labeled directed graph with arc labels specifying parent-child and next-sibling relationships and vertex labels specifying conditions such as first child ('f'), last child ('l'), a tag from Σ, or a combination thereof. A subset of the graph vertices is used for selecting the items for extraction (there is an extraction vertex for each attribute name in the relational representation of the information source).

According to the model-theoretic semantics of patterns ([1]), a labeled ordered tree is an interpretation domain for the patterns. Intuitively, patterns are matched against parts of a target labeled ordered tree. In order to have a successful matching, the labels of pattern vertices and arcs must be consistent with the corresponding relations and functions defined over tree nodes. The result of applying a pattern to a semi-structured information source is a set of extracted tuples. An extracted tuple is modeled as a function from attribute names to tree nodes, as in standard relational data modeling.

We propose a semi-automatic L-wrapper development process using ILP ([2]). During this process, two useful operations on patterns were defined ([1]): i) *pattern simplification* – the process of removing arcs in the pattern directed graph without changing the pattern semantics; more precisely we can shift one position right an arc labeled with 'c' in a pattern and we obtain an equivalent pattern. This operation is used to normalize patterns (i.e. to simplify patterns such that the out-degree of every pattern vertex is at most 1); ii) *pattern merging* – the process of combining simpler patterns that share attributes into more complex patterns, that are capable to extract tuples of higher arity [1]; this is useful to control the process of learning patterns of higher arity.

An L-wrapper can be defined as a set of patterns that share the set of attributes from the relation scheme of the information source. In this paper we restrict our attention to single-pattern L-wrappers that can be concisely defined in two steps: i) define the pattern graph together with arc labels that model parent-child and next-sibling relations and ii) extend this definition with vertex labels that model conditions on vertices, extraction vertices and assignment of extraction vertices to attributes.

Definition 1. *(Pattern graph) Let $\mathcal{W}$ be a set denoting all vertices. A* pattern graph *G is a quadruple $\langle A, V, L, \lambda_a \rangle$ such that $V \subseteq \mathcal{W}$, $A \subseteq V \times V$, $L \subseteq V$ and $\lambda_a : A \to \{'c', 'n'\}$. The set $\mathcal{G}$ of pattern graphs is defined inductively as follows:*

i) If $v \in \mathcal{W}$ then $\langle \emptyset, \{v\}, \{v\}, \emptyset \rangle \in \mathcal{G}$

ii) If $G = \langle A, V, L, \lambda_a \rangle \in \mathcal{G}$, $v \in L$, and $w, u_i \in \mathcal{W} \setminus V$, $1 \leq i \leq n$ then a) $G_1 = \langle A \cup \{(w, v)\}, V \cup \{w\}, (L \setminus \{v\}) \cup \{w\}, \lambda_a \cup \{((w, v), 'n')\} \rangle \in \mathcal{G}$; b) $G_2 = \langle A \cup \{(u_1, v), \ldots, (u_n, v))\}, V \cup \{u_1, \ldots, u_n\}, (L \setminus \{v\}) \cup \{u_1, \ldots, u_n\}, \lambda_a \cup \{((u_1, v), 'c'), \ldots,$

$((u_n, v),' c')\} \rangle \in \mathcal{G}$; c) $G_3 = \langle A \cup \{(w, v), (u_1, v), \ldots, (u_n, v))\}, V \cup \{w, u_1, \ldots, u_n\}, (L \setminus \{v\}) \cup \{w, u_1, \ldots, u_n\}, \lambda_a \cup \{((w, v),' n'), ((u_1, v),' c'), \ldots, ((u_n, v),' c')\} \rangle \in \mathcal{G}$;

Intuitively, if $\langle A, V, L, \lambda_a \rangle$ is a pattern graph then V are its vertices, A are its arcs, $L \subseteq V$ are its leaves (vertices with in-degree 0) and λ_a indicates parent-child and next-sibling arcs.

Definition 2. *(Single-pattern L-wrapper) Let $\mathcal{A}$ be the set of attribute names. A* single-pattern L-wrapper *is a tuple $W = \langle V, A, U, D, \mu, \lambda_a, \lambda_c \rangle$ such that $\langle A, V, L, \lambda_a \rangle$ is a pattern graph, $U = \{u_1, u_2, \ldots, u_k\} \subseteq V$ is the set of* pattern extraction vertices*, $D \subseteq \mathcal{A}$ is the set of attribute names, $\mu : D \rightarrow U$ is a one-to-one function that assigns a pattern extraction vertex to each attribute name, and $\lambda_c : V \rightarrow \mathcal{C}$ is the labeling function for vertices. $\mathcal{C} = \{\emptyset, \{' f'\}, \{' l'\}, \{\sigma\}, \{' f',' l'\}, \{' f', \sigma\}, \{' l', \sigma\}, \{' f',' l', \sigma\}\}$ is the set of conditions, where σ is a label in the set Σ of tag symbols.*

Examples of pattern graphs can be found in figure 2.

3 Translating L-Wrappers into XSLT Stylesheets

The process of information extraction from the Web can be structured into a sequence of stages ([1,2]): page collection, pre-processing, manual information extraction, conversion to the input format of the learning program, learning, wrapper compilation, wrapper execution. This section addresses wrapper compilation, namely the translation of L-wrappers into XSLT. Actually, we will use a subset of XSLT, called XSLT_0 ([3]).

The output of the learning stage is a set of rules. Rules are first converted into graph-like descriptions as introduced by definitions 1 and 2. These graphs are further processed using pattern operations to produce the final graph description of the wrapper.

At this point we apply an algorithm for translating L-Wrappers into XSLT that exploits the graph-like definition of L-wrappers. The idea is to generate XSLT_0 templates for all the leaf and extraction vertices of the L-wrapper, moving upwards and downwards in the graph. The extracted information is passed between templates by means of template variables and parameters. Here is an informal description of this algorithm:

Step 1. Start from the document root and generate the start template, by moving downwards to one of the vertices in $L \cup U$.
Step 2. Move from the current vertex (say w_0) to another vertex in $L \cup U$ (say w_1). The path taken depends on the type of the first vertex: if $w_0 \in L$ then we move first upwards, to the common ascendent of w_0 and w_1 and then downwards to w_1; if $w_0 \notin L$ then we follow the direct descendent path to w_1.
Step 3. Generate a template that will select the content of w_0 in case $w_0 \in U$.
Step 4. Repeat steps 2 and 3 until there are no more unvisited vertices in $L \cup U$.
Step 5. Generate the final template, which will display the extracted tuples.

Note that if the pattern graph has n vertices (i.e. $|L| = n$) then this algorithm generates $n + 1$ templates. Moreover, if there are $k \leq n$ extraction vertices (i.e. $|U| = k$) then k templates will generate a new variable.

4 Experiments

We now demonstrate the use of L-wrappers to extract travel information from the Travelocity Web site (see [17] and figure 1). That Web page displays hotel information comprising the hotel name, address and description, the check-in and check-out dates, the types of rooms offered and the corresponding average nightly rate. Adopting the relational model, we associate to this resource the following set of attribute names related to hotels: {*name*, *address*, *description*, *period*, *roomtype*, *price*}.

Fig. 1. An XHTML document fragment and its graphic view

Because of the relatively large number of attributes, we used the pattern merging approach. This overcomes the difficulty of directly learning tuples with arity greater than 2 ([2]). The following pairs of attributes were chosen: {*name*, *address*}, {*address*, *description*}, {*name*, *period*}, {*period*, *roomtype*}, and {*roomtype*, *price*}.

Then we generated extraction rules for each pair of attributes by using the FOIL program ([15]), as described in [2]. The following 5 rules were generated: (NA = *name*, AD = *address*, DE = *description*, PE = *period*, RO = *roomtype*, and PR = *price*):

$$extract(NA, AD) \leftarrow first(AD) \wedge td(AD) \wedge child(C, NA) \wedge child(D, AD) \wedge next(D, E) \wedge child(F, E) \wedge span(C) \wedge first(D) \wedge child(G, C) \wedge child(H, G) \wedge next(I, H) \wedge child(J, I) \wedge child(K, F) \wedge child(L, K) \wedge child(M, J) \wedge child(N, M) \wedge child(O, L) \wedge child(P, O) \wedge child(Q, P) \wedge child(N, Q).$$

$$extract(AD, DE) \leftarrow child(C, AD) \wedge child(D, DE) \wedge next(AD, E) \wedge next(C, F) \wedge child(G, F) \wedge child(F, D) \wedge first(G) \wedge text(DE).$$

$$extract(NA, PE) \leftarrow text(NA) \wedge child(C, NA) \wedge child(D, PE) \wedge next(E, D) \wedge child(F, E) \wedge b(D) \wedge child(G, C) \wedge child(H, G) \wedge next(I, H) \wedge child(J, I) \wedge child(K, J) \wedge next(K, L) \wedge next(L, M) \wedge child(M, N) \wedge child(O, F) \wedge child(P, O) \wedge child(N, P).$$

$$extract(PE, RO) \leftarrow child(C, PE) \wedge child(D, RO) \wedge next(D, E) \wedge next(F, C) \wedge child(G, E) \wedge child(H, F) \wedge next(I, G) \wedge child(J, I) \wedge next(K, J) \wedge first(D) \wedge child(K, L) \wedge child(L, H).$$

$$extract(RO, PR) \leftarrow child(C, RO) \wedge child(D, PR) \wedge next(C, E) \wedge next(D, F) \wedge child(G, E) \wedge next(H, G) \wedge child(I, H) \wedge next(J, I) \wedge child(G, D) \wedge first(C) \wedge last(F) \wedge text(PR).$$

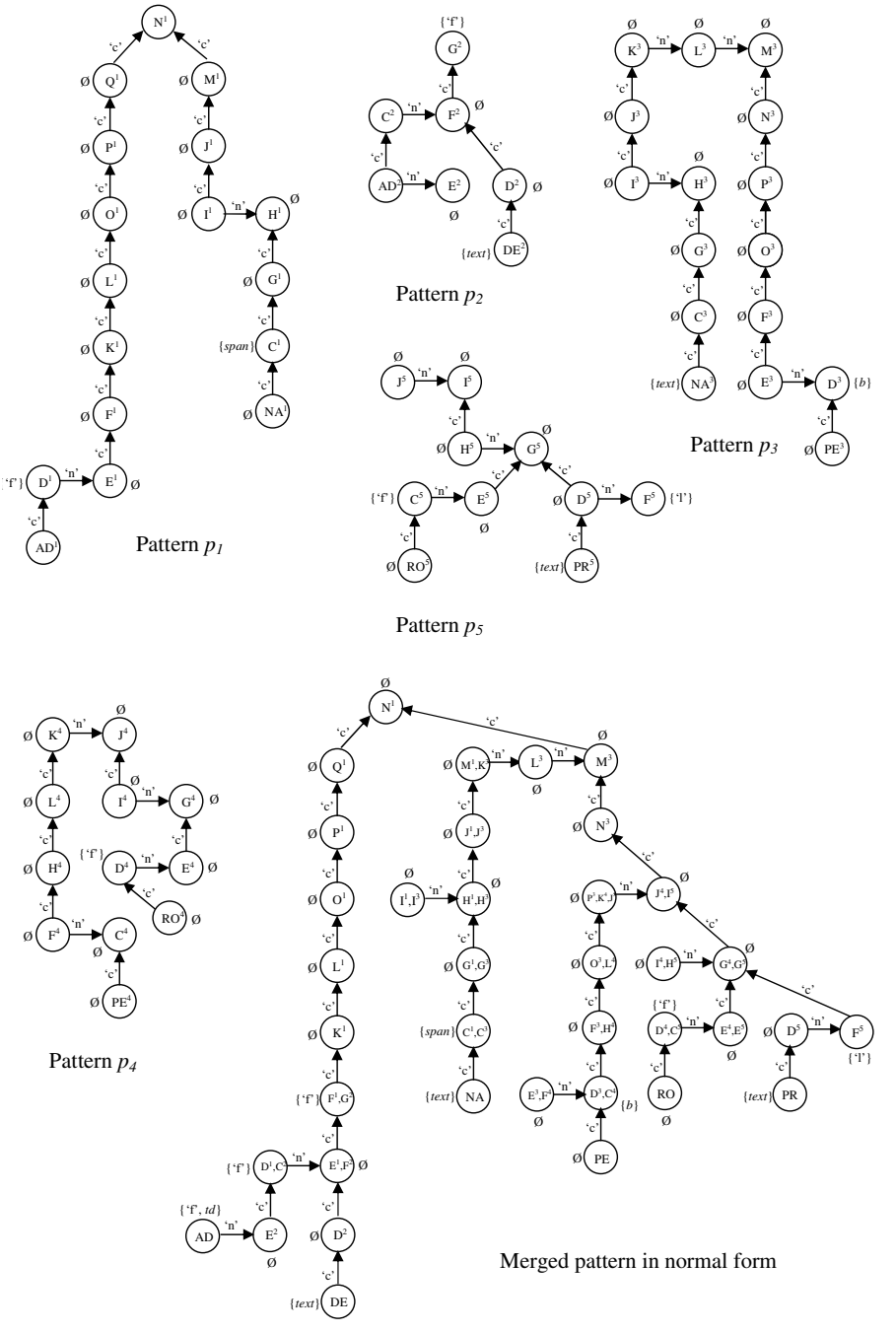


Fig. 2. Patterns and pattern merging

Table 1. Description of the sample wrapper in $XSLT_0$ pseudocode

```
template start(/)
    return
        result(sel_address(xp1))
end

template sel_address(*)
    vardef
        varAddress := content(.)
    return
        sel_description(xp2,varAddress)
end

template sel_description(*,varAddress)
    vardef
        varDescription := content(.)
    return
        sel_leaf_I1I3(xp3,varAddress,varDescription)
end

template sel_leaf_I1I3(*,varAddress,varDescription)
    return
        sel_name(xp4,varAddress,varDescription)
end

template sel_name(*,varAddress,varDescription)
    vardef
        varName := content(.)
    return
        sel_leaf_E3F4(xp5,varAddress,varDescription,
            varName)
end

template sel_leaf_E3F4(*,varAddress,varDescription,
    varName)
    return
        sel_period(xp6,varAddress,varDescription,
            varName)
end

template sel_period(*,varAddress,varDescription,
    varName)
    vardef
        varPeriod := content(.)
    return
        sel_leaf_I4H5(xp7,varAddress,varDescription,
            varName,varPeriod)
end

template sel_leaf_I4H5(*,varAddress,varDescription,
    varName,varPeriod)
    return
        sel_roomtype(xp8,varAddress,varDescription,
            varName,varPeriod)
end

template sel_roomtype(*,varAddress,varDescription,
    varName,varPeriod)
    vardef
        varRoomtype := content(.)
    return
        display(xp9,varAddress,varDescription,
            varName,varPeriod,varRoomtype)
end

template display(*,varAddress,varDescription,
    varName,varPeriod,varRoomtype)
    vardef
        varPrice := content(.)
    return
        tuple[name → varName;
              period → varPeriod;
              description → varDescription;
              address → varAddress;
              roomtype → varRoomtype;
              price → varPrice]
end
```

The patterns corresponding to the above clauses are shown in figure 2. Next we merged these patterns into a single-pattern L-wrapper and then we translated it into $XSLT_0$ using the algorithm outlined in section 3. A set of 10 $XSLT_0$ templates was obtained (see table 1).

XPath expressions xp_1, xp_2, xp_3, xp_4, xp_5, xp_6, xp_7, xp_8 and xp_9 of the wrapper from table 1 are defined as follows:

```
xp1 = //*/*/*/*/*/*/*[not(preceding-sibling::*)]/*/preceding-sibling::*[1]
      [not(preceding-sibling::*)]/*/preceding-sibling::*[1][not(preceding-sibling::*)
      and local-name()='td']
xp2 = following-sibling::*[1]/parent::*[not(preceding-sibling::*)]/following-sibling::*/
      */*[local-name()='text']
xp3 = parent::*/parent::*/parent::*[not(preceding-sibling::*)]/parent::*/parent::*/
      parent::*/parent::*/parent::*/parent::*/*/preceding-sibling::*[1]/
      preceding-sibling::*[1]/*/*/preceding-sibling::*[1]
```

xp_4 = following-sibling::*[1]/*/*[local-name()='span']/*[local-name()='text']
xp_5 = parent::*[local-name()='span']/parent::*/parent::*/parent::*/parent::*/
following-sibling::*[1]/following-sibling::*[1]/*/*/preceding-sibling::*[1]/
//*[local-name()='b']/preceding-sibling::*[1]
xp_6 = following-sibling::*[1][local-name()='b']/*
xp_7 = parent::*[local-name()='b']/parent::*/parent::*/parent::*/following-sibling::*[1]
/*/preceding-sibling::*[1]
xp_8 = following-sibling::*[1]/*/preceding-sibling::*[1][not(preceding-sibling::*)]/*
xp_9 = parent::*[not(preceding-sibling::*)]/following-sibling::*[1]/parent::*/
[not(following-sibling::)]/preceding-sibling::*[1]/*[local-name()='text']

The wrapper actually extracts the node contents rather than the nodes themselves, using the *content(.)* expression. There are two possibilities: either the current node is a leaf and has the tag *text* attached to it, or it is an internal node. (Note that we assume there is a special tag *text* $\in \Sigma$ that designates a text element, so that we can treat text and element nodes of a XHTML document in an uniform way). Thus, in case the current node is a text element, its actual text is extracted; otherwise the text content of all its descendants with tag *text* is extracted. The resulting wrapper expressed in XSLT is shown in the appendix.

For wrapper execution we can use any of the available XSLT transformation engines. In our experiments we have used Oxygen XML editor ([14]), a tool that incorporates some of these engines (see figure 3). The experimental results confirmed the efficacy of the approach: values 0.87 and 1.0 were recorded for standard measures of precision and recall.

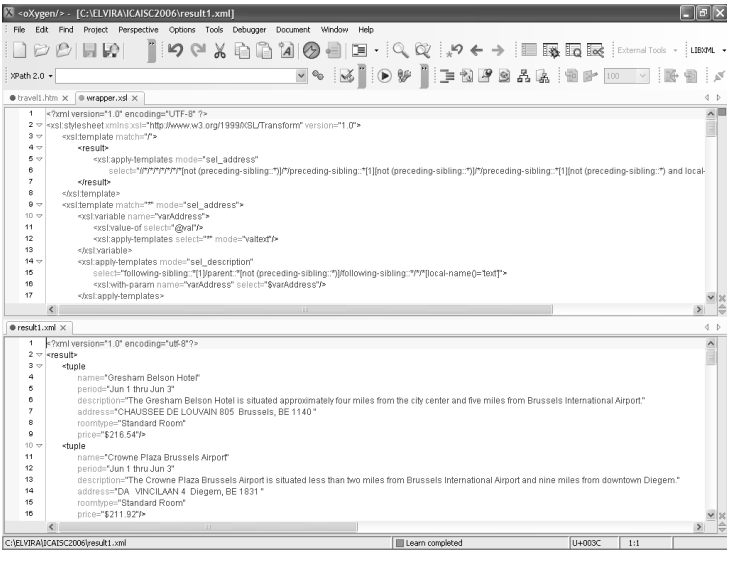


Fig. 3. Wrapper execution inside Oxygen XML editor

5 Conclusions and Future Work

This paper discusses the application of L-wrapper technology to information extraction from a travel agency Web site. The positive results of this experiment in the e-tourism

domain account as a further proof for the generality of the approach. Also, the substantially more complex hierarchical structure of the selected Web pages and the relatively large number of extracted tuples support the scalability of the proposed technique. However, the hierarchical structure of information is lost by flattening during extraction. Addressing this issue is one of our future research directions. At the same time, as future theoretical work, we are interested in giving a formal proof of the correctness of the mapping of L-wrappers to XSLT.

References

1. Bădică, C., Bădică, A.: Logic Wrappers and XSLT Transformations for Tuples Extraction from HTML. In: Bressan, S.; Ceri, S.; Hunt, E.; Ives, Z.G.; Bellahsene, Z.; Rys, M.; Unland, R. (eds): *Proc. 3rd International XML Database Symposium XSym'05*, Trondheim, Norway. LNCS 3671, Springer-Verlag (2005), 177–191
2. Bădică, C., Bădică, A., Popescu, E.: Tuples Extraction from HTML Using Logic Wrappers and Inductive Logic Programming. In: Szczepaniak, P.S., Kacprzyk, J., Niewiadomski, A. (eds.): *Proc.AWIC'05*, Lodz, Poland. LNAI 3528 Springer-Verlag (2005), 44–50
3. Bex, G.J., Maneth, S., Neven, F.: A formal model for an expressive fragment of XSLT. *Information Systems*, No.27, Elsevier Science (2002), 21–39.
4. Clark, J.: XSLT Transformation (XSLT) Version 1.0, W3C Recommendation, 16 November 1999, `http://www.w3.org/TR/xslt` (1999).
5. Chidlovskii, B.: Information Extraction from Tree Documents by Learning Subtree Delimiters. In: *Proc. IIWeb'03*, Acapulco, Mexico (2003), 3–8
6. Freitag, D.: Information extraction from HTML: application of a general machine learning approach. In: *Proc. AAAI'98*, (1998), 517–523
7. Ikeda, D., Yamada, Y., Hirokawa, S.: Expressive Power of Tree and String Based Wrappers. In: *Proc. IIWeb'03*, Acapulco, Mexoco, (2003), 16–21.
8. Knoblock, C.: Agents for Gathering, Integrating, and Monitoring Information for Travel Planning. In: Intelligent Systems for Tourism. *IEEE Intelligent Systems*. Nov./Dec. (2002), 53–66.
9. Kosala, R., Bussche, J. van den, Bruynooghe, M., Blockeel, H.: Information Extraction in Structured Documents Using Tree Automata Induction. In: *Principles of Data Mining and Knowledge Discovery, 6th European Conf.*, Helsinki, Finland, LNAI 2431, Springer-Verlag (2002), 299–310.
10. Kushmerick, N., Thomas, B.: Adaptive Information Extraction: Core Technologies for Information Agents, In: *Intelligent Information Agents R&D in Europe: An AgentLink perspective* (Klusch, Bergamaschi, Edwards & Petta, eds.). LNAI 2586, Springer-Verlag (2003), 79–103.
11. Laender, A.H.F., Ribeiro-Neto, B., Silva, A.S., Teixeira., J.S.: A Brief Survey of Web Data Extraction Tools. In: *SIGMOD Record*, Vol.31, No.2, ACM Press (2002), 84–93.
12. Laudon, K.C., Traver, C.G.: *E-commerce. business. technology. society* (2nd ed.). Pearson Addison-Wesley, (2004).
13. Li, Z., Ng, W.K.: WDEE: Web Data Extraction by Example. In: L. Zhou, B.C. Ooi, and X. Meng (Eds.): *Proc.DASFAA'2005*, Beijing, China. LNCS 3453, Springer-Verlag (2005), 347-358.
14. Oxygen XML Editor. `http://www.oxygenxml.com/`.
15. Quinlan, J. R., Cameron-Jones, R. M.: Induction of Logic Programs: FOIL and Related Systems, *New Generation Computing*, 13, (1995), 287–312.

16. Sakamoto, H., Arimura, H., Arikawa, S.: Knowledge Discovery from Semistructured Texts. In: Arikawa, S., Shinohara, A. (eds.): *Progress in Discovery Science*. LNCS 2281, Springer-Verlag (2002), 586–599.
17. Travelocity Web site. `http://www.w3.org/TR/xslt`.
18. Xiao, L., Wissmann, D., Brown, M., Jablonski, S.: Information Extraction from HTML: Combining XML and Standard Techniques fro IE from the Web. In: Monostori, L., Vancza, J., Ali, M. (eds.): *Proc. IEA/AIE 2001*. LNAI 2070, Springer-Verlag (2001), 165–174.

A XSLT Code of the Sample Wrapper

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/">
    <result>
      <xsl:apply-templates mode="sel_address" select="//*/*/*/*/*/*/*[not(preceding-
        sibling::*)]/*/preceding-sibling::*[1][not(preceding-sibling::*)]/*/preceding-
        sibling::*[1][not(preceding-sibling::*) and local-name()='td']"/>
    </result>
  </xsl:template>
  <xsl:template match="*" mode="sel_address">
    <xsl:variable name="varAddress">
        <xsl:value-of select="@val"/> <xsl:apply-templates select="*" mode="valtext"/>
    </xsl:variable>
    <xsl:apply-templates mode="sel_description" select="following-sibling::*[1]/
      parent::*[not(preceding-sibling::*)]/following-sibling::*/*/*[local-name()='text']">
      <xsl:with-param name="varAddress" select="$varAddress"/>
    </xsl:apply-templates>
  </xsl:template>
  <xsl:template match="*" mode="sel_description">
    <xsl:param name="varAddress"/>
    <xsl:variable name="varDescription">
      <xsl:value-of select="@val"/> <xsl:apply-templates select="*" mode="valtext"/>
    </xsl:variable>
    <xsl:apply-templates mode="sel_leaf_I1I3" select="parent::*/parent::*/parent::*[not
      (preceding-sibling::*)]/parent::*/parent::*/parent::*/parent::*/parent::*/parent::*/
      */preceding-sibling::*[1]/preceding-sibling::*[1]/*/*/preceding-sibling::*[1]">
      <xsl:with-param name="varAddress" select="$varAddress"/>
      <xsl:with-param name="varDescription" select="$varDescription"/>
    </xsl:apply-templates>
    </xsl:template>
  <xsl:template match="*" mode="sel_leaf_I1I3">
    <xsl:param name="varAddress"/> <xsl:param name="varDescription"/>
    <xsl:apply-templates mode="sel_name" select="following-sibling::*[1]/*/
      *[local-name()='span']/*[local-name()='text']">
      <xsl:with-param name="varAddress" select="$varAddress"/>
      <xsl:with-param name="varDescription" select="$varDescription"/>
    </xsl:apply-templates>
  </xsl:template>
  <xsl:template match="*" mode="sel_name">
    <xsl:param name="varAddress"/> <xsl:param name="varDescription"/>
    <xsl:variable name="varName">
      <xsl:value-of select="@val"/> <xsl:apply-templates select="*" mode="valtext"/>
    </xsl:variable>
    <xsl:apply-templates mode="sel_leaf_E3F4" select="parent::*[local-name()='span']/
      parent::*/parent::*/parent::*/parent::*/following-sibling::*[1]/following-sibling::*[1]/
      */*/preceding-sibling::*[1]/*/*/*[local-name()='b']/preceding-sibling::*[1]">
      <xsl:with-param name="varAddress" select="$varAddress"/>
      <xsl:with-param name="varDescription" select="$varDescription"/>
      <xsl:with-param name="varName" select="$varName"/>
    </xsl:apply-templates>
  </xsl:template>
  <xsl:template match="*" mode="sel_leaf_E3F4">
    <xsl:param name="varAddress"/><xsl:param name="varDescription"/><xsl:param name="varName"/>
    <xsl:apply-templates mode="sel_period" select="following-sibling::*[1][local-name()='b']/*">
      <xsl:with-param name="varAddress" select="$varAddress"/>
      <xsl:with-param name="varDescription" select="$varDescription"/>
      <xsl:with-param name="varName" select="$varName"/>
    </xsl:apply-templates>
```

```
  </xsl:template>
  <xsl:template match="*" mode="sel_period">
    <xsl:param name="varAddress"/><xsl:param name="varDescription"/><xsl:param name="varName"/>
    <xsl:variable name="varPeriod">
      <xsl:value-of select="@val"/> <xsl:apply-templates select="*" mode="valtext"/>
    </xsl:variable>
    <xsl:apply-templates mode="sel_leaf_I4H5" select="parent::*[local-name()='b']/
      parent::*/parent::*/parent::*/following-sibling::*[1]/*/preceding-sibling::*[1]">
      <xsl:with-param name="varAddress" select="$varAddress"/>
      <xsl:with-param name="varDescription" select="$varDescription"/>
      <xsl:with-param name="varName" select="$varName"/>
      <xsl:with-param name="varPeriod" select="$varPeriod"/>
    </xsl:apply-templates>
  </xsl:template>
  <xsl:template match="*" mode="sel_leaf_I4H5">
    <xsl:param name="varAddress"/> <xsl:param name="varDescription"/> <xsl:param name="varName"/>
    <xsl:param name="varPeriod"/>
    <xsl:apply-templates mode="sel_roomtype" select="following-sibling::*[1]/*/
      preceding-sibling::*[1][not(preceding-sibling::*)]/*">
      <xsl:with-param name="varAddress" select="$varAddress"/>
      <xsl:with-param name="varDescription" select="$varDescription"/>
      <xsl:with-param name="varName" select="$varName"/>
      <xsl:with-param name="varPeriod" select="$varPeriod"/>
    </xsl:apply-templates>
  </xsl:template>
  <xsl:template match="*" mode="sel_roomtype">
    <xsl:param name="varAddress"/><xsl:param name="varDescription"/><xsl:param name="varName"/>
    <xsl:param name="varPeriod"/>
    <xsl:variable name="varRoomtype">
      <xsl:value-of select="@val"/> <xsl:apply-templates select="*" mode="valtext"/>
    </xsl:variable>
    <xsl:apply-templates mode="display" select="parent::*[not(preceding-sibling::*)]/
      following-sibling::*[1]/parent::*/*[not(following-sibling::*)]/preceding-sibling::*[1]/
      *[local-name()='text']">
      <xsl:with-param name="varAddress" select="$varAddress"/>
      <xsl:with-param name="varDescription" select="$varDescription"/>
      <xsl:with-param name="varName" select="$varName"/>
      <xsl:with-param name="varPeriod" select="$varPeriod"/>
      <xsl:with-param name="varRoomtype" select="$varRoomtype"/>
    </xsl:apply-templates>
  </xsl:template>
  <xsl:template match="*" mode="display">
    <xsl:param name="varAddress"/><xsl:param name="varDescription"/><xsl:param name="varName"/>
    <xsl:param name="varPeriod"/><xsl:param name="varRoomtype"/>
    <xsl:variable name="varPrice">
      <xsl:value-of select="@val"/> <xsl:apply-templates select="*" mode="valtext"/>
    </xsl:variable>
    <tuple>
      <xsl:attribute name="name"> <xsl:value-of select="$varName"/> </xsl:attribute>
      <xsl:attribute name="period"> <xsl:value-of select="$varPeriod"/> </xsl:attribute>
      <xsl:attribute name="description"><xsl:value-of select="$varDescription"/></xsl:attribute>
      <xsl:attribute name="address"> <xsl:value-of select="$varAddress"/> </xsl:attribute>
      <xsl:attribute name="roomtype"> <xsl:value-of select="$varRoomtype"/> </xsl:attribute>
      <xsl:attribute name="price"> <xsl:value-of select="$varPrice"/> </xsl:attribute>
    </tuple>
  </xsl:template>
  <xsl:template match="*" mode="valtext">
    <xsl:value-of select="@val"/><xsl:value-of select="' '"/>
    <xsl:apply-templates select="*" mode="valtext"/>
  </xsl:template>
</xsl:stylesheet>
```

A New Evaluation Method for E-Learning Systems

Krzysztof Przybyszewski

Institute for Distance Education Research
Department of Computer Science, Management and Transport
AHE University, Rewolucji 1905r no 64, Lodz, Poland
kprzybyszewski@wshe.lodz.pl

Abstract. The paper describes a conception of application of the fuzzy numbers for assessing students progress in learning, knowledge assimilation and abilities. The method of judge is based on operations performed on the fuzzy numbers and fuzzy sets. It has been used for assessing one-choice test and the results have been compared with teachers marks.

1 Introduction

The evaluation units are most important parts of the e-learning systems (ELS) [1]. They allow teachers presence simulation for an asynchronous mode and enable to construct an individual learning path for each student [2]. It will be good if the decision-making processes of these units are similar teachers ones, so closely, as it is possible. The expert systems are basis of an evaluation process, even those ones for a traditional learning/teaching process. In that case, the teacher uses his own knowledge and own set of rules for assessing each students progress of learning and progress of his/her ability and for proposing for further parts of material for study [3]. As in most expert systems, the final information that is generated as a final mark, is vague (in mathematical sense). It is a verbal term (a linguistic variable) or a number that is a representative of a numbers range. It is natural, that we are able to use the fuzzy numbers and the fuzzy sets for an evaluation process. The traditional methods are mostly used for evaluation in ELS, yet. One teacher or teachers team evaluates students progress. It is independent of which level of ELS that process takes place on: it may be an evaluation of subject course or an evaluation of whole semester or even evaluation of whole study period. There are some attempts to automate these processes, but they have been made for close problems, only. In that case a decision about mark and further individual students path of learning bases on a table of rough rules. In our opinion, it is not in accordance with a view of an evaluation process that is made by teacher. The Microsoft or CISCO certification exams are examples of that kind of evaluation system.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1209–1216, 2006.
© Springer-Verlag Berlin Heidelberg 2006

2 Fuzzy Numbers as Representatives of Marking Scale

We would like to suggest a method of validation bases on fuzzy numbers and fuzzy sets.

Let $A_a \subseteq R$ is fuzzy number [4] that is determined with three parameters: m_L, a, m_P. The membership function of it is as follow:

$$\mu(x) = \begin{cases} 0 & for\ x < 0 \\ \frac{x}{a-m_L} & for\ 0 \leq x < a - m_L \quad oraz \quad (a - m_L) \neq 0 \\ 1 & for\ a - m_L \leq x \leq a + m_P \\ \frac{5-x}{5-a-m_P} & for\ a + m_P < x \leq 5 \quad oraz \quad (a - m_P) \neq 5 \\ 0 & for\ x > 5 \end{cases} \tag{1}$$

The number A_a we can call *trapezoid fuzzy number*. Parameter a is called *center ofthe number* and parameters: m_L and m_P are called appropriately: *left* and *right width of the number*. We can write a fuzzy number A_a as follow: $A_a = (m_L; a; m_P)$. A middle point of a fuzzy number A_a is defined as a middle of a numeric interval in which membership function achieves value 1.The triangular fuzzy number which has membership function as follow:

$$\mu(x) = \begin{cases} 0 & for\ x < 0 \\ \frac{x}{\widetilde{a}} & for\ 0 \leq x < \widetilde{a} \\ 1 & for\ x = \widetilde{a} \\ \frac{5-x}{5-\widetilde{a}} & for\ \widetilde{a} < x \leq 5 \\ 0 & for\ x > 5 \end{cases} \tag{2}$$

we call, that it is generated by the fuzzy number A_a and we write it in symbolic way as follow: $\widetilde{A_a}$. The following set of trapezoid fuzzy numbers:

$$SM = \{(2; 2; 0, 5), (0, 5; 3; 0, 5), (0, 5; 4; 0, 5), (0, 5; 5; 0)\}$$

is representation of the marking scale which is applied in polish high schools (without half-marks). In this case, the marking scale is linguistic variables set, as follow:

$$SM_{ling} = \{no\ pass\ mark\ (2), pass\ mark\ (3), good\ mark\ (4), very\ good\ mark\ (5)\}.$$

The graphic representation of the fuzzy numbers which are representative for that marking scale is on Fig.1 below.

There is only one condition for the fuzzy numbers which represent the marking scale: they have to be decomposition elements of the following trapezoid fuzzy number: $(2, 5; 2, 5; 2, 5)$ This condition is called *total filled scale condition*. It can be written in notation of operations on fuzzy sets as follow:

$$(2, 5; 2, 5; 2, 5) = (2; 2; 0, 5) \cup (0, 5; 3; 0, 5) \cup (0, 5; 4; 0, 5) \cup (0, 5; 5; 0)$$

For any marking scale that is represented by fuzzy numbers set: $\{(m_{Li}; a_i; m_{Pi})\}_{0 \leq i \leq N}$, total filled scale condition can be written in a broad way as follow:

$$(2, 5; 2, 5; 2, 5) = \bigcup_i (m_{Li}; a_i; m_{Pi}) \tag{3}$$

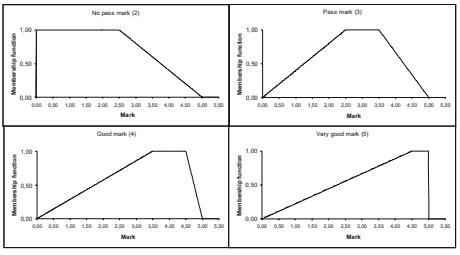


Fig. 1. The representation of the fuzzy numbers which are representative for the marking scale in polish high school

West European marking scale SM_{EU} is the most popular scale. It bases on six marks:

$$SM_{EUling} = \{F, E, D, C, B, A\}$$

The following set of fuzzy numbers is the representation of this scale:

$$SM_{EU} = \{(0;0;0,5), (0,5;1;0,5), (0,5;2;0,5), (0,5;3;0,5), (0,5;4;0,5), (0,5;5;0)\}$$

or, in symbolic way:

$$SM_{EU} = \{A_0, A_1, A_2, A_3, A_4, A_5.\}$$

It is easy to verify, that the total filled scale condition (3) is met for SM_{EU}. All fuzzy numbers that are elements of this scale (without outermost ones: A_0 and A_5) have got the same width. The center and the middle point of each one are equal (without outermost ones: A_0 and A_5, too):

$$\underset{0<i<5}{\forall} a_i = \widetilde{a}_i \ \wedge \ m_{Li} = m_{Pi}$$

Those features seem to be typical for the marking scale, which have been correctly constructed. The scale with these features is called stable one. A graphic interpretation for that scale is presented on Fig.2:

Another example of a marking scale is one that is applied in Polish Virtual University (PUW). The marks of this scale are determined by score intervals (or percentage intervals). The scores are given to students by teacher. The score intervals, marks and fuzzy numbers follow them are represented in Table 1.

That marking scale has got an interesting abnormal at *pass mark plus* : center of follow fuzzy number is outside the numeric interval in which membership function achieves value 1: $3,5 \notin [3,75; 4,0)$. It shows that the numeric intervals are selected unfortunately but it proves also that suggested method of marking scale representation by the fuzzy numbers is very flexible.

For all fuzzy numbers that are elements of this marking scale, the center of the number is not equal the middle point of it. This marking scale is not stable. The rules (algorithms) of relation between score (or percentage or numbers) intervals

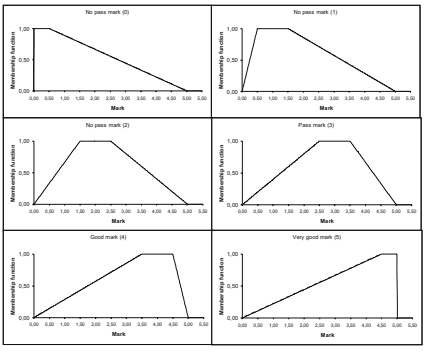


Fig. 2. The fuzzy numbers from SM_{EU} graphic presentation

Table 1. Marking scale that is applied in PUW

Score interval	Mark	Fuzzy number
[0, 60)	No pass mark	(2;2;1)
[60, 75)	Pass mark	(0,3,0,75)
[75, 80)	Pass mark plus	(-0,25;3,5;1)
[80, 90)	Good mark	(0;4;0,5)
[90, 95)	Good mark plus	(0;4,5;0,25)
[95, 100]	Very good mark	(0,25;5;0)

and trapezoid fuzzy numbers are the most important for the marking scale that is represented by the fuzzy numbers. In the case of stable scale we can describe the rules with a function.

3 Fuzzy Numbers and Final Mark Determination

There are a lot of cases then the teacher has to give to the student the final mark based on the partial marks. Those situations occur during the final or diploma exams, at the end of each learning period and even if teacher assesses the complex test.

In that case a two-part algorithm is proposed.

In the first step the fuzzy number that represents the average mark is determined. It is done by calculating the mean of all trapezoid fuzzy numbers [5] that are representative of the partial marks, according to rules of extending of mathematical operations from rough sets to fuzzy sets [4]. Let $PM = \left\{A_P^1, A_P^2, \ldots, A_P^N\right\}$ is the set of the representatives of all partial mark. Then the trapezoid fuzzy number $\overline{A_P}$ determined by the relation (4):

$$\overline{A_P} = \frac{1}{N}A_P^1 \oplus \frac{1}{N}A_P^2 \oplus \ldots \oplus \frac{1}{N}A_P^N \tag{4}$$

is the representative of the average mark. $\frac{1}{N}$ is weighting factor of each partial mark. It is possibly to assign different values to them, but sum of them must be equal 1:

$$\overline{A_P} = w_1 \cdot A_P^1 \oplus w_2 \cdot A_P^2 \oplus \ldots \oplus w_N \cdot A_P^N$$

and

$$\sum_{i=1}^{N} w_i = 1.$$

In the second step two different ways for final mark determination is proposed:

1. If the average mark represents the partial mark in further evaluation process, then it is suggested leaving it without any changes,
2. If the average mark is a base for the final mark that is represented by the trapezoid fuzzy number A_{FM}, then we choose from the set of fuzzy numbers which are epresentative of the marking scale that one for which its intersection with the triangular fuzzy number that is generated by the middle point of average mark representative, is normalized fuzzy set:

$$A_{FM} = A_{SM}^i : \quad \left(A_{SM}^i \in SM\right) \quad \wedge \quad \left(h\left(\widetilde{\overline{A_P}} \cap A_{SM}^i\right) = 1\right) \tag{5}$$

Two fuzzy numbers represent the final marks: $\overline{A_P}$ and $\overline{A_P}$. They are very useful for the expert systems of ELS. This algorithm seems to be similar to traditional method that is applied by teachers. It seems to be useful if we should compare marks that are based on the different marking scale, marks from two different schools, for example.

4 Fuzzy Numbers and Evaluation of One-Choice Test

It is suggested to apply stable marking scale SM_{EU} for assessing of one-choice tests which contains N problems. The final marks for each answer are represented only by two trapezoid fuzzy numbers: A_0 (incorrect answer) and A_5 (correct answer). The average mark we obtain according to relation (4):

$$\overline{A_{Tone}} = \frac{k}{N} \cdot A_0 \oplus \frac{N-k}{N} \cdot A_5$$

where: k is the number of incorrect answers.

The final mark for whole test is determined according to relation (5) and the final marking scale (SM_F) may be anyone, there is not necessarySM_{EU}, for example. The proposed method has been applied to assess one-choice test, which has been carried at the end of one-semester physics course. Test has contained twenty problems ($N = 20$). There have been four answers for each question and only one has been correct. First, the final marks have been determined in the traditional way with the table of rough decision rules (Table 2).

The proposed method has been applied for assessing the same test. The stable marking scale (SM_{EU}) and the marking scale bases on the decision table (SM_{DT}) have been applied. The results of both simulations are presented in Table 3.

Table 2. The table of decision rules which has been applied for traditional method of assessing of the one-choice test

Scores interval (N-k)	Mark
[0;10]	No pass mark 2
[11;15]	Pass mark - 3
[16;18]	Good mark - 4
[19;20]	Very good mark - 5

Table 3. Comparison of two methods of evaluation of one-choice test

N-k	Classical mark	Mark based on (SU_{EU})	$\widetilde{A_{Tone}}$	$\overline{A_{Tone}}$	A_{FM}	
					SM_{EU}	SM_{DT}
0	2	0 (F)	0.25	(0;0;0,5)	[2;2;0,5]	[0;0;0,5]
1			0.475	(0,025;0,25;0,475)		
2			0.7	(0,05;0,5;0,45)		
3		1 (E)	0.925	(0,075;0,75;0,425)		(0,5;1;0,5]
4			1.15	(0,1;1;0,4)		
5			1.375	(0,125;1,25;0,375)		
6		2 (D)	1.6	(0,15;1,5;0,35)		(0,5;2;0,5]
7			1.825	(0,175;1,75;0,325)		
8			2.05	(0,2;2;0,3)		
9			2.227	(0,225;2,25;0,275)		
10			2.5	(0,25;2,5;0,25)		
11	3	3 (C)	2.725	(0,275;2,75;0,225)	(0,5;3;0,75]	(0,5;3;0,5]
12			2.95	(0,3;3;0,2)		
13			3.175	(0,325;3,25;0,175)		
14			3.4	(0,35;3,5;0,15)		
15		4 (B)	3.625	(0,375;3,75;0,125)		(0,5;4;0,5]
16	4		3.85	(0,4;4;0,1)	(0,25;4;0,5]	
17			4.075	(0,425;4,25;0,075)		
18			4.3	(0,45;4,5;0,05)		
19	5	5 (A)	4.525	(0,475;4,75;0,025)	(0,5;5;0]	(0,5;5;0]
20			7.75	(0,5;5;0)		

In the next step, 168 real tests that have been assessed in the past with the table of decision rules, was validated with the fuzzy method. The results are presented in Table 4.

If we apply to assess the marking scale that bases on the decision table, then the results are the similar for both way of validation. If we compare the results based on SM_{EU} and SM_{DT} scales, we see that they are different only for 22 cases and the different is a benefit to students.

Table 4. The list of the results of validation for 168 real one-choice tests

Number of tests	N-k	Classical mark	Mark based on (SU_{EU})	$\widetilde{\overline{A_{Tone}}}$	A_{FM}	
					SM_{DT}	SM_{EU}
0	0	2	0 (F)	(0;0;0,5)	[2;2;0,5]	[0;0;0,5]
2	1			(0,025;0,25;0,475)		
6	2			(0,05;0,5;0,45)		
6	3		1 (E)	(0,075;0,75;0,425)		(0,5;1;0,5]
8	5			(0,125;1,25;0,375)		
9	6		2 (D)	(0,15;1,5;0,35)		(0,5;2;0,5]
9	7			(0,175;1,75;0,325)		
9	8			(0,2;2;0,3)		
4	10			(0,25;2,5;0,25)		
19	11	3	3 (C)	(0,275;2,75;0,225)	(0,5;3;0,75]	(0,5;3;0,5]
21	12			(0,3;3;0,2)		
19	13			(0,325;3,25;0,175)		
20	14			(0,35;3,5;0,15)		
22	15	4	4 (B)	(0,375;3,75;0,125)		(0,5;4;0,5]
10	16			(0,4;4;0,1)	(0,25;4;0,5]	
2	17			(0,425;4,25;0,075)		
1	18			(0,45;4,5;0,05)		
1	19	5	5 (A)	(0,475;4,75;0,025)	(0,5;5;0]	0,5;5;0]

5 Concluding Remarks

It is possible to apply the trapezoid fuzzy numbers as representative of any marking scale (SM). We can assess the quality of a marking scale with this method of representation. The stable scale is well constructed marking scale.

It is possible to determine average mark $(\overline{A_P})$ of some partial marks from set $\left\{A_P^i\right\}_{0\leq i\leq N}$ with using the rules of extending of mathematical operations from rough sets to fuzzy sets, even for the weighting factors w_i that are different each other.

It is possible to determine final mark A_{FM} with using triangular fuzzy number which is generated by average mark $\widetilde{\overline{A_P}}$ and the features of intersection of fuzzy sets.

Two fuzzy numbers represent the final marks: $\overline{A_P}$ and A_{FM} in that method. They are very useful for the expert systems of ELS. This algorithm seems to be similar to traditional method that is applied by teachers. It seems to be useful if we should compare marks that are based on the different marking scale.

All those algorithms are tested by using them to validation process of 168 one-choice tests for two different marking scales: one that is applied in polish high schools and other one that is used in West Europe. The results have been compared with the teacher's validation results. Sufficient compliance of those results is the best confirmation for suggested methods.

References

1. Przybyszewski K., Cader A., Filutowicz Z., Information Management in the Interactive E-learning Systems, WSHE Scientific Booklets 4 (9), (2000) 90-102 (in Polish)
2. Przybyszewski K., Tutorials and Training Modules for the E-learning, Automatyka 3 (9) (2005) 799 - 809 (in Polish)
3. Niemierko B., Between School Marks and Didactic. Closely to the Didactic., WSiP Warszawa (1997) (in Polish)
4. Rutkowski L., Methods and Techniques of Artificial Intelligence, PWN Warszawa (2005) (in Polish)
5. Debuis D., Prade H., Operations on Fuzzy Numbers, Intern. Journal System Science 9 (1978) 613-626

Parameter Estimation of Systems Described by the Relation by Maximum Likelihood Method

Jerzy Świątek

Institute of Information Science and Engineering, Wrocław University of Technology,
Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
jerzy.swiatek@pwr.wroc.pl

Abstract. In the paper the problem of parameter estimation of input - output system is discussed. It is assumed that system is described by the relation known with accuracy to some parameters. The possible observations of system are described. The estimation algorithm based on maximum likelihood method is proposed. The presented approach is illustrated by analytical examples.

1 Introduction

Investigation of computer system for decision process based on knowledge representation requires new method of system modeling. Traditional mathematical model given by system of equation was very convenient for analytical investigations. Application of computer aided method for processing observations, more generally knowledge about investigated plant allows to investigate wide class of models. Particularly, the input - output system may be described by the set of facts given by logical statements about input and output. Sometimes such a description is given by expert. The problem is to generalize expert observation and propose the system description in form of relation defined on set of input and output. In this case we can formulate the identification problem for system described by the relation, similar to the identification of systems described by the equations [1]. The problem of modeling and identification of systems described by the relation has been presented in previous works. Particularly in [3] the general problem of identification of relational system is presented. In [5] the problem of optimal model choice is discussed. Some estimation problem is presented in [6]. Now the application of maximum likelihood method is proposed.

2 System Descriptions

Let us consider the input - output static system with input x and output y. Input and output are S and L dimensional vectors, respectively. Input and output are elements of sets X and Y, which are subsets of $\mathbb{R}^S$ and $\mathbb{R}^L$ spaces, respectively, i.e.:

$$x \in X \subseteq \mathbb{R}^S,\ y \in Y \subseteq \mathbb{R}^L.$$

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1217–1222, 2006.
© Springer-Verlag Berlin Heidelberg 2006

The system is described by the set of facts concerning input and outputs. More precisely, the set of true logical statements about x and y is given. Consequently the logical function

$$F(x, y, a) \tag{1}$$

defined on the set of inputs and outputs is proposed, where F is complex logical function and a is K – dimensional vector of parameters from set of parameters A (i.e.: $a \in A \subseteq \mathbb{R}^K$). In the system description only such (x, y) from $X \times Y$ may appear, for which the statement (1) is true. In this way the description of the system is given by the relation defined on $X \times Y$ i.e.:

$$\Re_a = \{(x, y) \in X \times Y; \quad F(x, y, a)\}. \tag{2}$$

On the relation $\Re_a$ the probability density function

$$g(x, y, a) \tag{3}$$

is defined. For example, let sets of inputs and outputs are real numbers and facts concerning inputs and outputs are the following: input and output are positive numbers and sum of input and output is not grater than a. Furthermore let us assume that the probability density is monotonous. For the above system (2) and (3) have the forms:

$$\Re_a = \{(x, y) \in \mathbb{R}^2; \quad x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ x + y \leq a\}, \tag{4}$$

$$g(x, y, a) = \begin{cases} 2a^{-2} & \text{if} \quad x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ x + y \leq a \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

3 System Observations

Now it will be assumed that the description of the system is known with the accuracy to the parameters, i.e. the functions F and g in (1) – (3) are known but vector of parameters a must be estimated. To determine vector a, the following experimental data may be collected:
A – the sequence of input and output measurements of the system are collected, i.e.:

$$(x_n, y_n), \quad n = 1, 2, \ldots, N, \tag{6}$$

where (x_n, y_n) are n–th measurements of input and output, respectively, N is number of measurements.
B – the sequence of true logical statements is given, i.e.:

$$r_n = \{(x, y) \in (X \times Y); \quad f_n(x, y)\}, \quad n = 1, 2, \ldots, N, \tag{7}$$

where f_n is n–th logical statement about input and output. Such a fact may be given by expert.

For example from expert we know that input x is in the interval $x \in [\alpha_{1n}, \alpha_{2n}]$ and the output y is in the interval $y \in [\beta_{1n}, \beta_{2n}]$, what will be denoted as observation:

$$r_n = \{(x, y) \in \mathbb{R}^2; \;\; \alpha_{1n} \leq x \leq \alpha_{2n} \;\wedge\; \beta_{1n} \leq y \leq \beta_{2n}\}. \tag{8}$$

4 Parameter Estimations

The collected data of the form (6) are independent realization of random variables (x, y) with the probability density function (3). Consequently the likelihood function has the form:

$$W_A(a, X_N, Y_N) = \prod_{n=1}^{N} g(x_n, y_n, a), \tag{9}$$

where $X_N = [x_1 \; x_2 \; \ldots \; x_N]$ and $Y_N = [y_1 \; y_2 \; \ldots \; y_N]$. Estimate a_{AN} of vector a is obtained by maximization (9) with respect to a, i.e.:

$$a_{AN} = \Psi_A(X_N, Y_N) \;\rightarrow\; W_A(a_{AN}, X_N, Y) = \min_{a \in A} W_A(a, X_N, Y_N). \tag{10}$$

Let us come back to the example when sets of inputs and outputs are real numbers and facts concerning inputs and outputs are the following: input and output are positive numbers, sum of input and output is not grater than a and the probability density is monotonous. The relation $\Re_a$ and probability distribution $g(x, y, a)$ are given by (4) and (5). The respective likelihood function is:

$$W_A(a, X_N, Y_N) = \begin{cases} 2^N a^{-2N} & \text{if} \quad \forall_{n \in \{1,2,\ldots,N\}} \;\; (x_n, y_n) \in \Re_a \\ \\ 0 & \text{otherwise} \end{cases}, \tag{11}$$

and estimator a_{AN} has the form:

$$a_{AN} = \max_{1 \leq n \leq N} (x_n + y_n). \tag{12}$$

For more general case, when it is assumed that the probability density distribution (3) has the form:

$$g(x, y, a) = \begin{cases} \left(\int_{\Re_a} dxdy \right)^{-1} & \text{if} \;\; (x, y) \in \Re_a \\ \\ 0 & \text{if} \;\; (x, y) \notin \Re_a \end{cases}, \tag{13}$$

the respective likelihood function is:

$$W_A(a, X_N, Y_N) = \begin{cases} \left(\int_{\Re_a} dxdy \right)^{-N} & \text{if} \;\; \forall_{n \in \{1,2,\ldots,N\}} \;\; (x_n, y_n) \in \Re_a \\ \\ 0 & \text{if} \;\; \exists_{n \in \{1,2,\ldots,N\}} \;\; (x_n, y_n) \notin \Re_a \end{cases}, \tag{14}$$

than to obtain the estimate a_{AN} , it is enough to solve the following optimization problem:

$$a_{AN} \ \rightarrow \ \min_{a\in A} \left(\int_{\Re_a} dxdy \right) \tag{15}$$

with constraints:

$$\forall_{n\in\{1,2,\ldots,N\}} \ (x_n, y_n) \in \Re_a, \tag{16}$$

what means that relation must have minimal "volume" and all true facts must be included, i.e.:

$$\forall_{n\in\{1,2,\ldots,N\}} \ F(x_n, y_n, a).$$

For the observations of the form B the true sentence of the form (7) is given. The probability that it is possible to obtain true observation r_n is determined by the following formula:

$$P_n(r_n, a) = \begin{cases} \int_{r_n} g(x, y, a) \ dxdy & \text{if} \quad r_n \subseteq \Re_a \\ \\ 0 & \text{otherwise} \end{cases} . \tag{17}$$

For further consideration it is assumed that

$$\forall_{n,m\in\{1,2,\ldots,N\}} \quad r_n \wedge r_m = \emptyset \quad \vee \quad r_n \wedge r_m = r_n = r_m \ ,$$

consequently the likelihood function is:

$$W_B(a, R_N) = \begin{cases} \prod_{n=1}^{N} \int_{r_n} g(x, y, a) \ dxdy & \text{if} \quad \forall_{n\in\{1,2,\ldots,N\}} \ r_n \subseteq \Re_a \\ \\ 0 & \text{otherwise} \end{cases} , \tag{18}$$

where $R_N = [r_1 \ r_2 \ \ldots \ r_N]$. The estimates a_{BN} of vector a, for the measurements B – type, is obtained by maximization of likelihood function (18) with respect a, i.e.:

$$a_{BN} = \Psi_B(R_N) \ \rightarrow \ W_B(a_{BN}, R_N) = \min_{a\in A} W_B(a, R_N). \tag{19}$$

Let us come back to the system described by relation (4) with probability density distribution (5). The respective observation are of the form (8). Probability (17) to obtain observation r_n of this form is:

$$P_n(r_n, a) = \begin{cases} 2a^{-2}(\alpha_{2n} - \alpha_{1n})(\beta_{2n} - \beta_{1n}) & \text{if} \quad r_n \subseteq \Re_a \\ \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

The respective likelihood function (18) is:

$$W_B(a, R_N) = \begin{cases} 2^N a^{-2N} \prod_{n=1}^{N} (\alpha_{2n} - \alpha_{1n})(\beta_{2n} - \beta_{1n}) & \text{if } \forall_{n\in\{1,\ldots,N\}} r_n \subseteq \Re_a \\ 0 & \text{otherwise} \end{cases}, \tag{21}$$

and solution (19)

$$a_{BN} = \max_{1\le n\le N} (\alpha_{2n} + \beta_{2n}). \tag{22}$$

Let us notice that for system described by the relation with probability density function of the form (13), the estimation problem may be reduced to the following optimization task: find such a parameter a_{BN} of the system description, for which relation $\Re_a$ has the minimal "volume" and includes each observation r_n, $n = 1, 2, \ldots, N$. Particularly, in this case the likelihood function has the form

$$W_B(a, R_N) = \begin{cases} \left(\int_{\Re_a} dxdy \right)^{-N} \prod_{n=1}^{N} \int_{r_n} dxdy & \text{if } \forall_{n\in\{1,2,\ldots,N\}}\ r_n \subseteq \Re_a \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

and maximization (23) with respect to a is equivalent to the following optimization problem:

$$a_{BN} \ \rightarrow \ \min_{a\in A} \left(\int_{\Re_a} dxdy \right) \tag{24}$$

with constraints

$$\forall_{n\in\{1,2,\ldots,N\}}\ r_n \subseteq \Re_a. \tag{25}$$

5 Final Remarks

The problem of modeling of system described by the relation has been discussed. The static system is described by set of fact about input and output. Te set of true facts gives the relation defined on set of inputs and outputs. In this paper it was assumed that description is known with accuracy to parameters. To determine unknown model parameters the estimation algorithm was proposed. Two different kinds of observations were used. The first case corresponds to traditional measurements, i.e. for given input the output is measured. The other observations are true logical sentences about inputs and outputs. For both cases the estimation algorithms based on maximum likelihood method have been proposed. The presented approach is illustrated by simple analytical examples.

References

1. Bubnicki Z.: Identification of Control Plants, Elsevier, Amsterdam–Oxford–New York, (1980)
2. Bubnicki Z.: Introduction to the Expert Systems, PWN, Warsaw, (1990) (in polish)

3. Bubnicki Z.: Introduction to the relation systems theory, Proc. of 7^{th} Polish–Russian Symposium on Optimization and Control of Cybernetics Systems, Warsaw, IBS PAN, (1980) 9–15
4. Bubnicki Z.: Problems of control and identification. Expert systems. Proc. of 3^{rd} Polish-Russian Scientific–Technological Conference "Computer Automation in Industry", Technical University of Wroclaw, (1988) 27–36
5. Świątek J.: Same problems of identification for relation systems, Systems Science vol. **15**, no. **1**, 26–34
6. Świątek J.: Parameter estimation of systems described by the relation. Proc of 7^{th} International conference on Systems Engineering, Las Vegas, (1990) 274–281

A Distributed Learning Control System for Elevator Groups

Tomasz Walczak[1] and Paweł Cichosz[2]

[1] Institute of Fundamental Technological Research, Polish Academy of Sciences, Świętokrzyska 21, 00-049 Warsaw, Poland
twalczak@ippt.gov.pl
[2] Department of Electronics and Information Technology, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland
pcichosz@elka.pw.edu.pl

Abstract. Human-designed elevator control policies usually perform sufficiently well, but are costly to obtain and do not easily adapt to changing traffic patterns. This paper describes an adaptive distributed elevator control system based on reinforcement learning. Whereas inspired by prior work, the design of the system is novel, developed with the intention to avoid any unrealistic assumptions that would limit its practical usefulness. Encouraging experimental results are presented with a realistic simulator of an elevator group.

1 Introduction

Elevator group control is an important practical problem. Allocating elevators to passenger calls in real time to optimize certain performance measures (usually based on passenger waiting or service time) is a hard task in high load conditions. Typically used control policies are human-designed heuristics based on observations and performance statistics from running elevator systems and some expert knowledge about elevator traffic patterns [1,2,3]. Such heuristics give acceptable performance in most cases, but are difficult and costly to design, whereas they are still likely to leave space for performance improvement. Moreover, if the elevator usage patterns assumed by the designer no longer correctly reflect the actual traffic in the building, which could have changed due to several reasons, the performance of a hand-written policy may degrade noticeably. This justifies the effort to develop adaptive elevator control systems that would not heavily rely on human knowledge and *a priori* assumptions about traffic patterns [4].

Elevator control is a sequential decision-making process that suits well to the reinforcement learning paradigm [5], in which the learner acquires or improves a policy from a series of direct interactions with the environment. Each interaction consists in taking an action based on the current state of the environment, and observing its effects — a real reinforcement value and a next state. The learner's task is to achieve an action selection policy that maximizes reinforcement values in the long term, which means that they can represent delayed evaluation of the learner's performance. In several decision-making problems it is quite easy

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 1223–1232, 2006.
© Springer-Verlag Berlin Heidelberg 2006

to assess long-term performance *a posteriori*, whereas action utility is neither known *a priori* nor can be assessed immediately. This is clearly the case for the elevator control problem, where the performance of a running control system can be easily evaluated using any adopted performance criteria, which allows one to directly to define a reinforcement signal.

This paper presents an architecture of a distributed elevator control system based on reinforcement learning and experimental results that demonstrate its performance for a realistically simulated elevator group. The simulator was designed to closely reflect the typical daily usage pattern of a particular real state-of-the art elevator system installed in one of Warsaw office buildings. Whereas inspired by prior work by Crites and Barto [6,7], we try to make our control system better suited to practical limitations (such as elevator dynamics), which results in a considerably different design.

1.1 Elevator Control

An elevator group consists of several elevator cars servicing a common set of floors, as illustrated in Figure 1. We consider the most common setup where on each floor there is one set of up and down buttons for issuing hall calls, common for all elevator cars. Inside elevator cars there are buttons used to issue car calls, indicating the target floor.

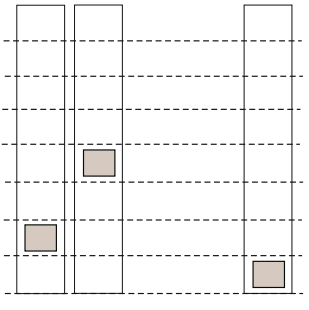


Fig. 1. Elevator group

The control system is responsible for controlling the operation of each elevator in the group, which includes making the following decisions:

for a moving elevator: the choice of the next floor in the current direction on which the elevator will stop,
for a stopped elevator: the choice of the time and direction of moving.

Two kinds of constraints must be preserved in the decision-making process:

constructional: related to elevator construction—e.g., the necessity of deciding about stopping on a floor in a sufficiently large distance from the floor,
functional: related to functional requirements—e.g., the necessity of servicing commands of passengers inside the car.

The objective of an elevator control system is to provide service to all passenger calls and commands and optimize some performance measures (usually minimize the waiting time for passenger calls and the travel time for passenger commands). The hardness of the task results from mostly from:

- large state space — there is a huge number of possible combinations of car calls, hall calls, and elevator positions or move directions,
- asynchronicity of events — hall calls or car calls can be signaled at any time moment,
- nonstationarity — the rate of incoming calls can change both in short term (during a day) and in long-term (over several days, weeks or months),
- partial osbervability — the number of passengers waiting on floors and travelling in cars is unknown.

1.2 Reinforcement Learning

The typical learning scenario of a reinforcement learning system consists in repeating a sequence of a few simple operations: observing the current state x_t, selecting and performing an action a_t, and then observing the reinforcement value r_t and the next state $x_{t+\tau}$. Here t is used to denote the time at which action selection takes place, and τ to denote the time interval after which a subsequent action will be selected. The objective of a reinforcement learning system is to maximize its reinforcement values in the long term. This is why reinforcement values are often called rewards. A standard performance measure is the total discounted reinforcement collected over time:

$$\int_0^\infty e^{-\beta t} r_t dt \tag{1}$$

where $\beta > 0$ determines the rate of discounting, i.e., giving less weight to reinforcement values more distant in time. This continuous-time formulation of the reinforcement learning task follows [8] and is adopted for this paper instead of its more common discrete-time counterpart since it is well suited to the asynchronicity of events in the elevator system.

For some tasks it may be more natural to define reinforcement in such a way that minimization would be desired instead of maximization. This will be the case for the elevator control task considered in this paper. In such tasks reinforcement values cam be called punishments or costs rather than rewards. In any case, the learner's task is to eventually reach optimal behavior, i.e., consistently selecting actions that lead to the maximization or minimization of the adopted performance measure.

Reinforcement learning algorithms usually rely on incrementally adjusting some value functions defined over the state or state and action space, which for each state or state-action pair estimate the total future reinforcement values. The most popular Q-learning algorithm [9], used by our elevator control system, maintains the Q function which, for each state-action pair $\langle x, a \rangle$, estimates the total discounted reinforcement after performing action a in state x and behaving

optimally thereafter. The single-step update rule applied at time t_2 for state-action pair from time t_1 is:

$$Q_{t_2}(x_{t_1}, a_{t_1}) := (1-\alpha)Q_{t_1}(x_{t_1}, a_{t_1}) + \alpha\Delta_{t_1,t_2} \tag{2}$$

where Δ_{t_1,t_2} is the error value for time t_1 calculated at time t_2 as:

$$\Delta_{t_1,t_2} = \int_0^{t_2-t_1} e^{-\beta\tau} r_{t_1+\tau} d\tau + e^{-\beta(t_2-t_1)} \max_a Q_{t_1}(x_{t_2}, a) \tag{3}$$

under the assumption that t_1 and t_2 are two consecutive time moments at which states are observed and action are selected. At time t_2, when the next state observation takes place, the Q-value for the state and action from time t_1 is updated, based on the reward accumulated between t_1 and t_2, and on the maximum Q-value for state x_{t_2}. The step-size parameter $\alpha \in (0, 1)$ determines the extent to which the Q-value is moved.

2 Control System Architecture

The learning elevator control system proposed in this paper is distributed, with a single separate reinforcement learning system to each elevator. This is illustrated in Figure 2. There is no direct communication between individual learners. The idea of using a team of per-elevator controllers rather than a single centralized controller for the whole group is motivated by the possibility of reducing the overall task complexity and hence operating in a smaller state space. It is also noteworthy that, since we assume that the physical operation and performance measure is the same for all elevators, all subtasks are identical, which permits using a shared control policy for individual controllers.

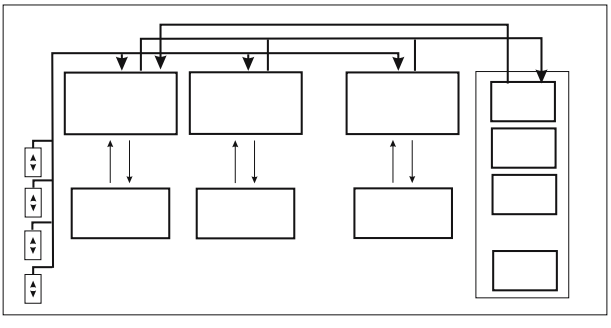


Fig. 2. Distributed reinforcement learning system

2.1 State Representation

State information provided to learning controllers has to appropriately reflect all relevant events happening in the elevator system that can be practically observed. This includes:

- awaiting hall calls and their waiting times,
- awaiting car calls and their waiting times for all elevators,
- position of all elevators,
- moving direction of all elevators,
- velocity of all elevators,
- calls to which particular elevators are allocated.

Strictly speaking, the last item in this list actually represents the internal state of the control system rather than the external environment state, but it can be considered part of state information from the perspective of individual learners. It follows from our assumptions about control actions, described below, which basically consist in assigning an elevator to an awaiting call. This assignment cannot be changed (until the elevator arrives to the stop of the assigned call) and therefore can be considered an important state variable.

Assuming the above contents of state information, for an elevator system with m elevator cars and n floors, we can calculate an estimate the size of the state space as follows:

- up to $n-1$ up hall calls — 2^{n-1} possibilities,
- up to $n-1$ down hall calls — 2^{n-1} possibilities,
- up to n car calls for each elevator (2^n possibilities),
- n possible positions for each elevator (rounded to the nearest floor),
- $2+1$ possible moving directions for each elevator (up, down, stopped)

which gives $(2^{n-1} \cdot 2^{n-1} \cdot 2^n \cdot n \cdot 3)^m$. On the other hand, for a single learner in our distributed architecture the state space size can be estimated as $2^{n-1} \cdot 2^{n-1} \cdot 2^n \cdot n \cdot 3$, which is still very large, but considerably less than before.

To further reduce the size of the state space, an aggregated state representation was used, in which a learner receives the number of awaiting hall calls in both directions and car calls rather than an exact information of calls and commands for particular floors. These counts of calls and commands are always relative to the elevator's current position and direction, respectively, i.e., the learner receives:

- the number of up and down hall calls from floors higher and lower than the current position (only not assigned to other cars),
- the number of car calls to floors in the current moving direction.

This is intended to make the aggregation represent possibly most useful state features for decision-making. It will become more clear why such aggregated state representation is appropriate when the set of actions is described below.

With aggregation, the size of the state space for a single learner in the distributed control system can be calculated as follows:

- the number of remaining up hall calls from floors higher than the current position: at most $n-1$ values,
- the number of remaining down hall calls from floors higher than the current position: at most $n-1$ values,

- the number of remaining up hall calls from floors lower than the current position: at most $n-1$ values,
- the number of remaining down hall calls from floors lower than the current position: at most $n-1$ values,
- the current position: n values,
- the number of car calls in the current direction: at most $n-1$ values,
- the current moving direction — 3 values,

which makes the total number of states equal: $(n-1)^4 \cdot n \cdot (n-1) \cdot 3$. This gives considerable reduction of the size of the state space of individual learners. For moderate values of n and m this number could permit using a tabular function representation, given that the number of actually observed states under normal operation will be usually much less than the above estimate — in an elevator system that is not extremely overloaded there will barely ever be simultaneously awaiting hall calls or car calls for more than a few floors.

2.2 Action Set

In related previous work on elevator control through reinforcement learning [6,7] the authors assumed that the control system takes actions (stop or continue moving) whenever a moving elevator is in the middle between two consecutive floors. This is hardly applicable to many practical elevator systems, where the stopping distance for an elevator moving at full speed is longer than a half of the distance between floors. Therefore we considered a more realistic approach with wider applicability, which however makes the learning task more difficult.

The learner assigned to an elevator selects its action whenever the elevator is stopped on a floor. The action consists in the selection of the target floor—the next floor on which the elevator will stop again. This selection is subject to the following constraints:

- no floor to which there is a car call can be skipped, so the target floor is selected from the range between the current floor and the floor of the nearest command (if there is one),
- the elevator cannot change direction before servicing all car calls in the current moving direction,
- the elevator can stop at a floor only if there is a call from this floor or there is a car call to this floor,
- the elevator cannot stop at a floor if the hall call from this floor has been already assigned to another elevator (unless there is a command to this floor),

Two additional heuristics incorporated in action selection are the preference for moving up and assigning incoming calls to an idle stopped elevator whenever there is one.

The number of available actions is at most equal to the number of floors. This can result in slower learning than reported in [6,7], for two actions. This is the cost of adopting more realistic assumptions about action selection. There is a positive side effect of this decision as well: since the choice of an elevator's

next stop is made at its previous stop, the arrival of an elevator to a call can be announced several seconds beforehand. This is an important merit for large elevator system, since it gives waiting passengers enough time to go towards the entry of the arriving elevator.

2.3 Reinforcement Function

The reward function used in this work is based on the definition of reinforcement for an elevator control system from [6,7]. The reinforcement value is basically calculated as the sum of squared waiting times for all calls and commands. It would be therefore more appropriate to call reinforcements defined that way *costs* rather than *rewards*, since obviously they have to be minimized instead of maximized.

Although all important events in the elevator system occur in discrete time moments, the intervals between them vary and therefore it is necessary to consider time as continuous. Therefore cost values are assigned to continuous time moments. For $\tau \in [t_1, t_2]$, where t_1 is the time of taking an action for an elevator (i.e., deciding on which floor to stop) and t_2 is the time of completing the action (i.e., actually stopping the elevator on the floor selected at time t_1), the reinforcement value is defined as follows:

$$r_\tau = \sum_c (\tau - t_1 + w_c)^2. \tag{4}$$

where w_c is the time period which waiting hall calls. It is important to underline that although the reinforcement value calculated as above is passed to the learner that selected its action and time t_1 and completed its action at time t_2, the calculation includes the waiting times of all hall calls and car calls, for other elevators as well.

3 Learning Algorithm

The reinforcement learning algorithm used in our distributed learning system is the basic Q-learning [9] algorithm, modified to meet two requirements of the elevator control task, i.e.:

- to act in continuous time, which is achieved by following the approach of [8], as summarized in Equations 2 and 3,
- to minimize costs instead of maximizing rewards, which is achieved by replacing maximization with minimization in Equation 3.

3.1 Continuous Time

In our elevator control system r_τ is non-zero in intervals between selecting and completing an action. According to Equation 4, the total discounted reinforcement for time interval $[t_1, t_2]$ is:

$$\int_{t_1}^{t_2} e^{-\beta(\tau - t_1)} \sum_c (\tau - t_1 + w_c)^2 d\tau, \tag{5}$$

which can be easily rewritten as:

$$\sum_c e^{-\beta w_c}\Big[\frac{2}{\beta^3}+\frac{2w_c}{\beta^2}+\frac{w_c^2}{\beta}\Big]-e^{-\beta(w_c+t_2-t_1)}\Big[\frac{2}{\beta^3}+\frac{2(w_c+t_2-t_1)}{\beta^2}+\frac{(w_c+t_2-t_1)^2}{\beta}\Big]. \tag{6}$$

3.2 Function Update

By reducing the size of the state-space appropriately, we were able to use tabular function representation and postpone the investigation of employing function approximators for future work. For a $[t_1, t_2]$ interval considered above, the Q-function for state x_{t_1} and action a_{t_1} is replaced at time t_2 by:

$$(1-\alpha)Q_{t_1}(x_{t_1}, a_{t_1})+\alpha\Big[\int_{t_1}^{t_2} e^{-\beta(\tau-t_1)}r_\tau d\tau + e^{-\beta(t_2-t_1)}\min_a Q(x_{t_2}, b)\Big], \tag{7}$$

where α is a step-size parameter.

4 Experimental Studies

The experiments use a realistic simulator of an elevator system, developed by the authors based on traffic statistics from a physical elevator system.[1] The elevator group consists of 6 elevators servicing 20 floors. There is an express zone between floors 1 and 7m where there are no stops and elevators travel at maximum speed. To allow using a tabular function representation rather than generalizing function approximators for this study, the number of floors in the simulation was reduced to 18, with call statistics adjusted appropriately. The Q-learning algorithm was used with roughly optimized parameters ($\alpha = 0.1$, $\beta = 0.01$, Boltzmann distribution-based action selection with temperature uniformly decreased in the course of learning).

Apart from a learning distributed control system outlined above, a simple, but good heuristic control policy was implemented and used for comparisons. This is the *best-first* heuristic which always picks up the first awaiting call in the current moving direction to be served. The results of the reinforcement learning and heuristic algorithms were compared with respect to the mean call service time and the distribution of call service times.

As one can see in Figure 3 and 4, the learning control algorithm clearly shows an improvement in mean service time in the course of learning. The improvement is particularly striking for down-calls, but also for up-calls the learning system outperforms the best-first heuristic after 10 simulated days of learning. For down-calls, the number of cases with service time between 21 and 81 is reduced by moving them to three lower intervals. For up-calls the situation is not

[1] The statistics come from an elevator system installed by the OTIS company in the Warsaw Financial Center building.

so good, since although some cases are moved from intervals $[11, 21)$ and $[21, 41)$ to two lower intervals, there is also an increase of the number of up-calls with service time between 41 and 81. There are 11 up-calls with service time above 60 for reinforcement learning and 2 for the heuristic. For down-calls there are respectively 27 and 32, so the advantage of reinforcement learning-based control system is much more evident.

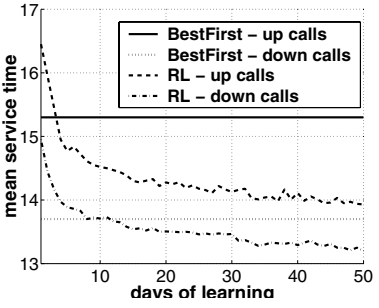


Fig. 3. Mean service time

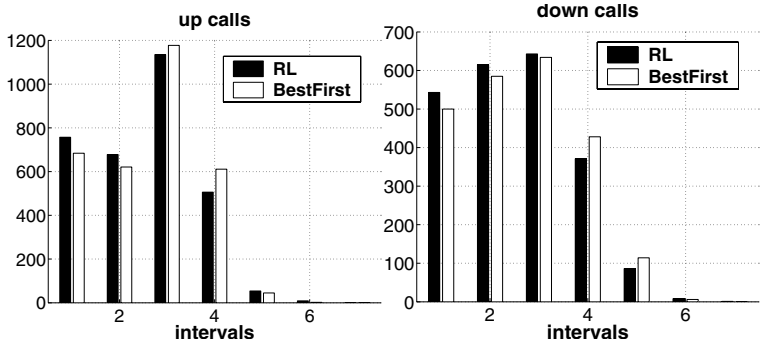


Fig. 4. Distribution of service time in intervals $[0, 6)$, $[6, 11)$,$[11, 21)$, $[21, 41)$, $[41, 81)$, $[81, 161)$, $[161, \infty)$

5 Conclusion

This work has contributed a novel approach to using reinforcement learning for elevator control. It differs in several important ways from previous approaches and makes more realistic assumptions, which brings it closer to practical applicability. To achieve this, a distributed control system architecture was designed, permitting considerable reduction of the state space size. The resulting control system is capable of online learning and therefore could be used not only to learn from scratch in simulation, but also to refine or adapt a predefined control policy in a physical elevator system.

It is particularly worthwhile to notice that reinforcement learning considerably improved the performance in comparison to a fairly good heuristic even using very limited state information, which permitted a tabular function representation. This demonstrates that our control system architecture and learning algorithm are reasonably designed and they deserve further investigations. The most promising direction for such investigations is to employ generalizing function approximator for function representation to make it possible to scale-up our solution to larger elevator systems as well as to enhance state information, which could possibly lead to even more substantial performance improvements.

References

1. Levy, D., Yardin, M., Alexandrowitz, A.: Optimal control of elevators. International Journal of Systems Science **8** (1977) 301–320
2. Pepyne, D.L., Cassandras, C.G.: Optimal dispatching control for elevator systems during uppeak traffic. IEEE Transactions on Control Systems Technology **5**(6) (1997) 629–643
3. Siikonen, M.: Planning and Control Models for Elevators in High-Rise Buildings. PhD thesis, Helsinki Unverstity of Technology, Systems Analysis Laboratory (1997)
4. Cortés, P., Larrañeta, J., Onieva, L.: Genetic algorithm for controllers in elevator groups: Analysis and simulation during lunchpeak traffic. Applied Soft Computing **4** (2004) 159–174
5. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press (1998)
6. Crites, R.H., Barto, A.G.: Improving elevator performance using reinforcement learning. In Touretzky, D.S., Mozer, M.C., Hasselmo, M.E., eds.: Advances in Neural Information Processing Systems. Volume 8., The MIT Press (1996) 1017–1023
7. Crites, R.H., Barto, A.G.: Elevator group control using multiple reinforcement learning agents. Machine Learning **33** (1998) 235–262
8. Bradtke, S.J., Duff, M.O.: Reinforcement learning methods for continuous-time Markov decision problems. In Tesauro, G., Touretzky, D., Leen, T., eds.: Advances in Neural Information Processing Systems. Volume 7., The MIT Press (1995) 393–400
9. Watkins, C.J.C.H.: Learning from Delayed Rewards. PhD thesis, King's College, Cambridge (1989)

Author Index

Abonyi, Janos 324
Abraham, Ajith 9
Albani, Cornelia 603
Amari, Shun-ichi 548
Andrysiak, Tomasz 758
Angryk, Rafal A. 170

Babczyński, Tomasz 1040
Bădică, Amelia 1199
Bădică, Costin 1050, 1199
Bailey, Trevor C. 972
Balasko, Balazs 324
Ban, Sang-Woo 814
Barreira, Noelia 712
Bartczuk, Łukasz 1060
Bartyzel, Michał 278
Bayarsaikhan, Battulga 833
Berehulyak, Olena 860
Bereta, Michal 563
Berlanga, Francisco Jos 182
Bialaszewski, Tomasz 390
Białko, Michał 470
Blachnik, Marcin 573
Blanco, Marianne 712
Błaszczyński, Jerzy 489
Bobrowski, Leon 1
Bogado, Juan M. 731
Bolanos, José M. 722
Borzemski, Leszek 192
Bożejko, Wojciech 334
Brzezinska, Izabela 1018
Burczynski, Tadeusz 563
Butkiewicz, Bohdan S. 202

Cader, Andrzej 1071
Cai, Lijun 931
Cappelletto, José 722, 731
Carreira, Maria Jose 712
Castro, Carlos 410, 1082
Chaiyaratana, Nachol 430, 1008
Chang, Jyh-Yeong 740
Chen, Yen Wei 787
Chen, Yuehui 9
Chon, Jaechoon 750
Chorąś, Michał 758
Chu, Feng 880
Cichocki, Andrzej 548, 870
Cichosz, Paweł 1223
Cierniak, Robert 890
Coats, Timothy J. 972
Cpalka, Krzysztof 212
Crawford, Broderick 1082
Cyganek, Bogusław 583
Czabański, Robert 220
Czeczot, Katarzyna 86

del Jesus, Mara Jos 182
Dembczyński, Krzysztof 489, 499
Di Barba, Paolo 344
Dominik, Andrzej 509
Duch, Włodzisław 573, 941
Dudek-Dyduch, Ewa 1091, 1101
Dyduch, Tadeusz 1091
Dzemyda, Gintautas 94
Dziwiński, Piotr 1111

Estevez, Pablo 731
Everson, Richard M. 972

Fang, Jianwen 899
Fermín, Leonardo 722, 731
Fernández-López, Gerardo 722, 731
Fieldsend, Jonathan E. 972
Fukushima, Kunihiko 814

Gabryel, Marcin 354
Gacto, Mara Jos 182
Ganzha, Maria 1050
Giordano, Daniela 768
González, Miguel A. 370
González-Rodríguez, Inés 360
Gorgoń, Marek 19
Gorzałczany, Marian B. 593
Greco, Salvatore 1018
Grieco, Juan C. 722, 731
Grześ, Marek 400
Grzymala-Busse, Jerzy W. 899

Hammer, Barbara 603
Hendzel, Zenon 778
Hermann, Wieland 603

Herrera, Francisco 182
He, Zhaohui 548
Hien, Thai Duy 787
Homenda, Wladyslaw 1179
Hong, SeokMi 823
Hori, Gen 548
Hrebień, Maciej 796

Jaroszewicz, Szymon 518
Jeong, Jong Cheol 29
Johansson, Ulf 613

Kacprzyk, Janusz 314, 1120
Kałużynski, Krzysztof 988
Karcz-Dulęba, Iwona 380
Kim, Eun-Mi 29
Kim, Hyongsuk 750, 850
Klawonn, Frank 663
Kleiber, Michal 150
Kluska, Jacek 230
Koh, Eun Jin 841
Kompanets, Leonid 1130
Kompass, Raul 548
Kong, Min 1140
König, Rikard 613
Korbicz, Józef 796
Korytkowski, Marcin 240
Korzeń, Marcin 518
Kosiński, Witold 250
Kosowski, Adrian 1150
Kotłowski, Wojciech 499
Kowalczuk, Zdzislaw 390
Krawczak, Maciej 1160
Krawiec, Krzysztof 804
Krętowska, Małgorzata 909
Krętowski, Marek 400
Krupski, Marcin 1071
Kruse, Rudolf 663
Kryzhanovsky, Boris 37
Krzanowski, Wojtek J. 972
Krzyżak, Adam 46
Kudová, Petra 56
Kulikowski, Juliusz L. 919
Kurzynski, Marek 623
Kuszner, Łukasz 1150
Kwasnicka, Halina 925

Lambert, Tony 410
Ławryńczuk, Maciej 76
Lee, Bae-Ho 29
Lee, Inwon 814
Lee, Minho 814
Lee, SeungGwan 823
Leonardi, Rosalia 768
Li, Dayou 972
Li, Xiangyong 1169
Lijewski, Patryk 804
Lin, Yaping 931
Liu, Chih-Wei 702
Liu, Qingzhong 951
Liu, Wen 66
Löfström, Tuve 613
Lopes, Heitor S. 420
Lu, Shih-Mao 740
Lu, Xinguo 931

Macukow, Bohdan 1179
Madar, Janos 324
Magomedov, Bashir 37
Magott, Jan 1040
Maiorana, Francesco 768
Mańdziuk, Jacek 1189
Markowska-Kaczmar, Urszula 86
Matykiewicz, Paweł 941
Mazurkiewicz, Jacek 479
Medina, Wilfredis 731
Medina Meléndez, Wilfredis 722
Medvedev, Viktor 94
Mehregan, Mohammad Reza 260
Mieszkowicz-Rolka, Alicja 268
Mikolajczyk, Marek 104, 123
Mikrut, Zbigniew 633
Min, Xiaoping 1169
Miyara, Kazuyoshi 787
Monfroy, Eric 410
Moritz, Guilherme L. 420
Mossakowski, Krzysztof 1189
Mukkamala, Srinivas 951

Nagata, Yasunori 787
Nakao, Zensho 787
Nam, Mi Young 833, 841
Naud, Antoine 643
Niewiadomski, Adam 278
Niklasson, Lars 613
Nowacka, Katarzyna 1120
Nowicki, Robert 240, 288

Obuchowicz, Andrzej 440
Ogiela, Lidia 962, 1027
Ogiela, Marek R. 962, 1027

Pałko, Tadeusz 988
Paprzycki, Marcin 1050
Partridge, Derek 972
Penas, Marta 712
Penedo, Manuel G. 712
Pestian, John 941
Piaskowska, Agata 633
Pieczyński, Andrzej 297
Piegat, Andrzej 104
Piroonratana, Theera 430
Pokropińska, Agata 288
Popescu, Elvira 1199
Prętki, Przemysław 440
Przybyszewski, Krzysztof 1209
Puente, Jorge 360

Qin, Zheng 450

Radzikowska, Anna Maria 528
Rafajłowicz, Ewaryst 113
Rakus-Andersson, Elisabeth 982
Raudys, Šarūnas 653
Red'ko, Vladimir 460
Rehm, Frank 663
Rejer, Izabela 104, 123
Rhee, Phill Kyu 833, 841
Robak, Silva 297
Rolka, Leszek 268
Rudziński, Filip 593
Ruta, Dymitr 672
Rutkowska, Danuta 1060, 1111
Rutkowski, Leszek 240, 354

Safari, Hossein 260
Šámalová, Terezie 56
Sanga, Gustaph 931
Sas, Jerzy 682
Saubion, Frédéric 410
Schäfer, Dominik 46
Scherer, Rafał 240, 288, 306
Schetinin, Vitaly 972
Schleif, Frank-Michael 603
Shi, Haixiang 66
Shi, Zhewen 450
Skubalska-Rafajłowicz, Ewa 133
Slot, Krzysztof 850
Słowik, Adam 470
Słowiński, Roman 489, 499, 1018
Sołtysiński, Tomasz 988
Spampinato, Concetto 768
Stasiak, Bartłomiej 142
Straszecka, Ewa 998
Sung, Andrew H. 951
Świątek, Jerzy 1217
Szczepaniak, Piotr S. 692
Szmidt, Eulalia 314

Tadeusiewicz, Ryszard 962, 1027
Tatjewski, Piotr 76
Tian, Peng 1140, 1169
Tomczyk, Arkadiusz 692
Tongsima, Sissades 1008
Tsai, Cheng-Fa 702
Tsoy, Yuri 460

Varela, Ramiro 370
Veeraghattam, Rajeev 951
Vela, Camino R. 360, 370
Viet, Nguyen Hoang 150
Villmann, Thomas 603
Vorobel, Roman 860

Wąs, Jarosław 1101
Walczak, Tomasz 1223
Walczak, Zbigniew 509
Walkowiak, Tomasz 479
Wang, Haijun 931
Wang, Hong 538
Wang, Lipo 66, 880
Wang, Yu 450
Wetcharaporn, Wannasak 1008
Wiak, Sławomir 344
Wieczorek, Tadeusz 573
Wodecki, Mieczysław 334
Wrzesiński, Mateusz 19

Yang, Xiaolin 931
Yatsymirskyy, Mykhaylo 142
Yu, Fan 450

Zadrożny, Sławomir 1120
Zak, Blazej 925
Zatwarnicki, Krzysztof 192
Zdunek, Rafal 548, 870
Zhang, Wen-Xiu 538
Zhang, Yong 9
Zhang, Youmin 160
Zliobaite, Indre 653
Zolnierek, Andrzej 623
Zurada, Jacek M. 66