

Lluís Godo (Ed.)

LNAI 3571

Symbolic and Quantitative Approaches to Reasoning with Uncertainty

8th European Conference, ECSQARU 2005
Barcelona, Spain, July 2005
Proceedings

 Springer

Lecture Notes in Artificial Intelligence 3571

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Lluís Godo (Ed.)

Symbolic and Quantitative Approaches to Reasoning with Uncertainty

8th European Conference, ECSQARU 2005
Barcelona, Spain, July 6-8, 2005
Proceedings



Springer

Series Editors

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

Volume Editor

Lluís Godo

Institut d'Investigació en Intel·ligència Artificial (IIIA)
Consejo Superior de Investigaciones Científicas (CSIC)
Campus UAB s/n, 08193 Bellaterra, Spain
E-mail: godo@iiia.csic.es

Library of Congress Control Number: 2005928377

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

ISSN	0302-9743
ISBN-10	3-540-27326-3 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-27326-4 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

springeronline.com

© Springer-Verlag Berlin Heidelberg 2005
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11518655 06/3142 5 4 3 2 1 0

Lluís Godo (Ed.)

Symbolic and Quantitative Approaches to Reasoning with Uncertainty

8th European Conference, ECSQARU 2005
Barcelona, Spain, July 6–8, 2005
Proceedings

Preface

These are the proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU 2005, held in Barcelona (Spain), July 6–8, 2005. The ECSQARU conferences are biennial and have become a major forum for advances in the theory and practice of reasoning under uncertainty. The first ECSQARU conference was held in Marseille (1991), and after in Granada (1993), Fribourg (1995), Bonn (1997), London (1999), Toulouse (2001) and Aalborg (2003).

The papers gathered in this volume were selected out of 130 submissions, after a strict review process by the members of the Program Committee, to be presented at ECSQARU 2005. In addition, the conference included invited lectures by three outstanding researchers in the area, Serafin Moral (Imprecise Probabilities), Rudolf Kruse (Graphical Models in Planning) and Jérôme Lang (Social Choice). Moreover, the application of uncertainty models to real-world problems was addressed at ECSQARU 2005 by a special session devoted to successful industrial applications, organized by Rudolf Kruse. Both invited lectures and papers of the special session contribute to this volume. On the whole, the programme of the conference provided a broad, rich and up-to-date perspective of the current high-level research in the area which is reflected in the contents of this volume.

I would like to warmly thank the members of the Program Committee and the additional referees for their valuable work, the invited speakers and the invited session organizer. I also want to express my gratitude to all of my colleagues and friends of the Executive Committee for their excellent work and unconditional support, dedicating a lot of their precious time and energy to make this conference successful. Finally, the sponsoring institutions are also gratefully acknowledged for their support.

May 2005

Lluís Godo

Organization

ECSQARU 2005 was organized by the Artificial Intelligence Research Institute (IIIA), belonging to the Spanish Scientific Research Council (CSIC).

Executive Committee

Conference Chair	Lluís Godo (IIIA, Spain)
Organizing Committee	Teresa Alsinet (University of Lleida, Spain) Carlos Chesñevar (University of Lleida, Spain) Francesc Esteva (IIIA, Spain) Josep Puyol-Gruart (IIIA, Spain) Sandra Sandri (IIIA, Spain)
Technical Support	Francisco Cruz (IIIA, Spain)

Program Committee

Teresa Alsinet (Spain)	Hector Geffner (Spain)
John Bell (UK)	Angelo Gilio (Italy)
Isabelle Bloch (France)	Michel Grabisch (France)
Salem Benferhat (France)	Petr Hájek (Czech Republic)
Philippe Besnard (France)	Andreas Herzig (France)
Gerd Brewka (Germany)	Eyke Huellermeier (Germany)
Luis M. de Campos (Spain)	Anthony Hunter (UK)
Claudette Cayrol (France)	Manfred Jaeger (Denmark)
Carlos Chesñevar (Spain)	Gabriele Kern-Isberner (Germany)
Agata Ciabattini (Austria)	Jürg Kohlas (Switzerland)
Giulianella Coletti (Italy)	Ivan Kramosil (Czech Republic)
Fabio Cozman (Brazil)	Rudolf Kruse (Germany)
Adnan Darwiche (USA)	Jérôme Lang (France)
James P. Delgrande (Canada)	Jonathan Lawry (UK)
Thierry Denœux (France)	Daniel Lehmann (Israel)
Javier Diez (Spain)	Pedro Larrañaga (Spain)
Marek Druzdzel (USA)	Churn-Jung Liao (Taiwan)
Didier Dubois (France)	Weiru Liu (UK)
Francesc Esteva (Spain)	Thomas Lukasiewicz (Italy)
Hélène Fargier (France)	Pierre Marquis (France)
Linda van der Gaag (Netherlands)	Khaled Mellouli (Tunisia)

VIII Organization

Serafín Moral (Spain)	Ken Satoh (Japan)
Thomas Nielsen (Denmark)	Torsten Schaub (Germany)
Kristian Olesen (Denmark)	Romano Scozzafava (Italy)
Ewa Orlowska (Poland)	Prakash P. Shenoy (USA)
Odile Papini (France)	Guillermo Simari (Argentina)
Simon Parsons (USA)	Philippe Smets (Belgium)
Luís Moniz Pereira (Portugal)	Claudio Sossai (Italy)
Ramon Pino-Pérez (Venezuela)	Milan Studený (Czech Republic)
David Poole (Canada)	Leon van der Torre (Netherlands)
Josep Puyol-Gruart (Spain)	Enric Trillas (Spain)
Henri Prade (France)	Emil Weydert (Luxembourg)
Maria Rifqi (France)	Mary-Anne Williams (Australia)
Alessandro Saffiotti (Sweden)	Nevin L. Zhang (Hong Kong, China)
Sandra Sandri (Spain)	

Additional Referees

David Allen	Christian Döring	Witold Pedrycz
Fabrizio Angiulli	Zied Elouedi	André Ponce de Leon
Cecilio Angulo	Enrique Herrera-Viedma	Guilin Qi
Nahla Ben Amor	Thanh Ha Dang	Jordi Recasens
Guido Boella	Jinbo Huang	Rita Rodrigues
Jesús Cerquides	Joris Hulstijn	Ikuo Tahara
Mark Chavira	Germano S. Kienbaum	Vicenç Torra
Gaetano Chemello	Beata Konikowska	Suzuki Yoshitaka
Petr Cintula	Vítor H. Nascimento	
Francisco A.F.T. da Silva	Giovanni Panti	

Sponsoring Institutions

Artificial Intelligence Research Institute (IIIA)
Spanish Scientific Research Council (CSIC)
Generalitat de Catalunya, AGAUR
Ministerio de Educación y Ciencia
MusicStrands, Inc.

Table of Contents

Invited Papers

Imprecise Probability in Graphical Models: Achievements and Challenges	1
Knowledge-Based Operations for Graphical Models in Planning	3
Some Representation and Computational Issues in Social Choice	15

Bayesian Networks

Nonlinear Deterministic Relationships in Bayesian Networks	27
Penniless Propagation with Mixtures of Truncated Exponentials	39
Approximate Factorisation of Probability Trees	51
Abductive Inference in Bayesian Networks: Finding a Partition of the Explanation Space	63
Alert Systems for Production Plants: A Methodology Based on Conflict Analysis	76
Hydrologic Models for Emergency Decision Support Using Bayesian Networks	88

Graphical Models

Probabilistic Graphical Models for the Diagnosis of Analog Electrical
Circuits 100

Qualified Probabilistic Predictions Using Graphical Models
..... 111

A Decision-Based Approach for Recommending in Hierarchical Domains
..... 123

Learning Causal Networks

Scalable, Efficient and Correct Learning of Markov Boundaries Under
the Faithfulness Assumption 136

Discriminative Learning of Bayesian Network Classifiers via the TM
Algorithm 148

Constrained Score+(Local)Search Methods for Learning Bayesian
Networks 161

On the Use of Restrictions for Learning Bayesian Networks
..... 174

Foundation for the New Algorithm Learning Pseudo-Independent
Models 186

Planning

Optimal Threshold Policies for Operation of a Dedicated-Platform with
Imperfect State Information - A POMDP Framework 198

APPSSAT: Approximate Probabilistic Planning Using Stochastic
Satisfiability 209

Causality and Independence

Racing for Conditional Independence Inference	221
Causality, Simpson's Paradox, and Context-Specific Independence	233
A Qualitative Characterisation of Causal Independence Models Using Boolean Polynomials	244

Preference Modelling and Decision

On the Notion of Dominance of Fuzzy Choice Functions and Its Application in Multicriteria Decision Making	257
An Argumentation-Based Approach to Multiple Criteria Decision	269
Algorithms for a Nonmonotonic Logic of Preferences	281
Expressing Preferences from Generic Rules and Examples – A Possibilistic Approach Without Aggregation Function	293
On the Qualitative Comparison of Sets of Positive and Negative Affects	305

Argumentation Systems

Symmetric Argumentation Frameworks	317
Evaluating Argumentation Semantics with Respect to Skepticism Adequacy	329
Logic of Dementia Guidelines in a Probabilistic Argumentation Framework	341

Argument-Based Expansion Operators in Possibilistic Defeasible Logic
Programming: Characterization and Logical Properties
..... 353

Gradual Valuation for Bipolar Argumentation Frameworks
..... 366

On the Acceptability of Arguments in Bipolar Argumentation
Frameworks
..... 378

Inconsistency Handling

A Modal Logic for Reasoning with Contradictory Beliefs Which Takes
into Account the Number and the Reliability of the Sources
..... 390

A Possibilistic Inconsistency Handling in Answer Set Programming
..... 402

Measuring the Quality of Uncertain Information Using Possibilistic Logic
..... 415

Remedying Inconsistent Sets of Premises
..... 427

Measuring Inconsistency in Requirements Specifications
..... 440

Belief Revision and Merging

Belief Revision of GIS Systems: The Results of REV!GIS
..... 452

Multiple Semi-revision in Possibilistic Logic
..... 465

A Local Fusion Method of Temporal Information
..... 477

Mediation Using ω -States	489
Combining Multiple Knowledge Bases by Negotiation: A Possibilistic Approach	501
Conciliation and Consensus in Iterated Belief Merging	514
An Argumentation Framework for Merging Conflicting Knowledge Bases: The Prioritized Case	527

Belief Functions

Probabilistic Transformations of Belief Functions	539
Contextual Discounting of Belief Functions	552

Fuzzy Models

Bilattice-Based Squares and Triangles	563
A New Algorithm to Compute Low T-Transitive Approximation of a Fuzzy Relation Preserving Symmetry. Comparisons with the T-Transitive Closure	576
Computing a Transitive Opening of a Reflexive and Symmetric Fuzzy Relation	587
Generating Fuzzy Models from Deep Knowledge: Robustness and Interpretability Issues	600
Analysis of the TaSe-II TSK-Type Fuzzy System for Function Approximation	613

Many-Valued Logical Systems

Non-deterministic Semantics for Paraconsistent C-Systems 625

Multi-valued Model Checking in Dense-Time
 Model Checking in Dense-Time Multi-valued Logics 638

Brun Normal Forms for Co-atomic Łukasiewicz Logics 650

Poset Representation for Gödel and Nilpotent Minimum Logics
 Model Checking in Nilpotent Minimum Logics 662

Uncertainty Logics

Possibilistic Inductive Logic Programming 675

Query Answering in Normal Logic Programs Under Uncertainty
 Query Answering in Normal Logic Programs Under Uncertainty 687

A Logical Treatment of Possibilistic Conditioning 701

A Zero-Layer Based Fuzzy Probabilistic Logic for Conditional Probability
 Model Checking in Fuzzy Probabilistic Logics 714

A Logic with Coherent Conditional Probabilities 726

Probabilistic Description Logic Programs 737

Probabilistic Reasoning

Coherent Restrictions of Vague Conditional Lower-Upper Probability Extensions
 Model Checking in Vague Conditional Logics 750

Type Uncertainty in Ontologically-Grounded Qualitative Probabilistic Matching	763
Some Theoretical Properties of Conditional Probability Assessments	775
Unifying Logical and Probabilistic Reasoning	788

Reasoning Models Under Uncertainty

Possibility Theory for Reasoning About Uncertain Soft Constraints	800
About the Processing of Possibilistic and Probabilistic Queries	812
Conditional Deduction Under Uncertainty	824
Heterogeneous Spatial Reasoning	836

Uncertainty Measures

A Notion of Comparative Probabilistic Entropy Based on the Possibilistic Specificity Ordering	848
Consonant Random Sets: Structure and Properties	860
Comparative Conditional Possibilities	872
Second-Level Possibilistic Measures Induced by Random Variables	884

Probabilistic Classifiers

Hybrid Bayesian Estimation Trees Based on Label Semantics	896

Selective Gaussian Naïve Bayes Model for Diffuse Large-B-Cell
Lymphoma Classification: Some Improvements in Preprocessing and
Variable Elimination
..... 908

Towards a Definition of Evaluation Criteria for Probabilistic Classifiers
..... 921

Methods to Determine the Branching Attribute in Bayesian Multinets
Classifiers
..... 932

Classification and Clustering

Qualitative Inference in Possibilistic Option Decision Trees
..... 944

Partially Supervised Learning by a Credal EM Approach
..... 956

Default Clustering from Sparse Data Sets
..... 968

New Technique for Initialization of Centres in TSK Clustering-Based
Fuzzy Systems
..... 980

Industrial Applications

Learning Methods for Air Traffic Management
..... 992

Molecular Fragment Mining for Drug Discovery
..... 1002

Automatic Selection of Data Analysis Methods
..... 1014

Author Index 1027

Imprecise Probability in Graphical Models: Achievements and Challenges (Extended Abstract)

Serafín Moral

Departamento de Ciencias de la Computación e I.A.,
Universidad de Granada, 18071 Granada, Spain
`smc@decsai.ugr.es`

This talk will review the basic notions of imprecise probability following Walley's theory [1] and its application to graphical models which usually have considered precise Bayesian probabilities [2]. First approaches to imprecision were robustness studies: analysis of the sensibility of the outputs to variations of network parameters [3, 4]. However, we will show that the role of imprecise probability in graphical models can be more important, providing alternative methodologies for learning and inference.

One key problem of current methods to learn Bayesian networks from data is the following: with short samples obtained from a very simple model it is possible to learn complex models which are far from reality [5]. The main aim of the talk will be to show that with imprecise probability we can transform lack of information into indeterminacy and thus the possibilities of obtaining unsupported outputs are much lower. The following points will be considered:

1. A review of imprecise probability concepts, showing the duality between sets of probabilities and sets of desirable gambles representations. Most of the present work in graphical models has been expressed in terms of sets of probabilities, but desirable gambles representation is simpler in many situations [6]. This will be the first challenge we propose: to develop a methodology for graphical models based on sets of desirable gambles representation.
2. We will show that independence can have different generalizations in imprecise probability, giving rise to different interpretations of graphical models [7]. We will consider the most important ones: epistemic independence and strong independence.
3. Given a network structure, the estimation of conditional probabilities in a Bayesian network poses important problems. Usually, Bayesian methods are used in this task, but we will show that the selection of concrete 'a priori' distributions in conjunction with the design of the network can have important consequences in the results of the probabilities we compute with the network. Then, we will introduce the imprecise Dirichlet model [8] and discuss how it can be applied to estimate interval probabilities in a dependence graph. Its use will allow to obtain sensible conclusions (non vacuous intervals) under weaker assumptions than precise Bayesian models.
4. In general, there are no methods based on imprecise probability to learn a dependence graph. This is another important challenge for the future. In [5] we

have introduced a new score to decide between dependence or independence taking as basis the imprecise Dirichlet model, which can be used for the design of a genuine imprecise probability learning procedure. Bayesian scores always decide between one of the options (dependence or independence) even for very short samples. The main novelty of the imprecise probability score is that in some situations will determine that there is no evidence to support any of the options. This will have important consequences on the behaviour of the learning algorithms and the strategy for searching a good model.

5. We will review algorithms for inference in graphical models with imprecise probability, showing the different optimization problems associated with the different independence concepts and estimation procedures [9]. One of the most actual challenging problems is the development of inference algorithms when probabilities are estimated under a global application of the imprecise Dirichlet model.
6. Finally we will consider the problem of supervised classification, making a survey of existing approaches [10, 11] and pointing at the necessity of developing a fair comparison procedure between the outputs of precise and imprecise models.

References

1. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London (1991)
2. Jensen, F.: Bayesian Networks and Decision Graphs. Springer-Verlag, New York (2002)
3. Fagin, R., Halpern, J.: A new approach to updating beliefs. In Bonissone, P., Henrion, M., Kanal, L., Lemmer, J., eds.: Uncertainty in Artificial Intelligence, 6. North-Holland, Amsterdam (1991) 347–374
4. Breese, J., Fertig, K.: Decision making with interval influence diagrams. In P.P. Bonissone, M. Henrion, L.K., ed.: Uncertainty in Artificial Intelligence, 6. Elsevier (1991) 467–478
5. Abellán, J., Moral, S.: A new imprecise score measure for independence. Submitted to the Fourth International Symposium on Imprecise Probability and Their Applications (ISIPTA '05) (2005)
6. Walley, P.: Towards a unified theory of imprecise probability. International Journal of Approximate Reasoning **24** (2000) 125–148
7. Couso, I., Moral, S., Walley, P.: A survey of concepts of independence for imprecise probabilities. Risk, Decision and Policy **5** (2000) 165–181
8. Walley, P.: Inferences from multinomial data: learning about a bag of marbles (with discussion). Journal of the Royal Statistical Society, Series B **58** (1996) 3–57
9. Cano, A., Moral, S.: Algorithms for imprecise probabilities. In Kohlas, J., Moral, S., eds.: Handbook of Defeasible and Uncertainty Management Systems, Vol. 5. Kluwer Academic Publishers, Dordrecht (2000) 369–420
10. Zaffalon, M.: The naive credal classifier. Journal of Statistical Planning and Inference **105** (2002) 5–21
11. Abellán, J., Moral, S.: Upper entropy of credal sets. Applications to credal classification. International Journal of Approximate Reasoning (2005). To appear.

Knowledge-Based Operations for Graphical Models in Planning

Jörg Gebhardt¹ and Rudolf Kruse²

¹ Intelligent Systems Consulting (ISC), Celle, Germany
`ISC-Gebhardt@t-online.de`

² Dept. of Knowledge Processing and Language Engineering (IWS),
Otto-von-Guericke-University of Magdeburg, Magdeburg, Germany

Abstract. In real world applications planners are frequently faced with complex variable dependencies in high dimensional domains. In addition to that, they typically have to start from a very incomplete picture that is expanded only gradually as new information becomes available. In this contribution we deal with probabilistic graphical models, which have successfully been used for handling complex dependency structures and reasoning tasks in the presence of uncertainty. The paper discusses revision and updating operations in order to extend existing approaches in this field, where in most cases a restriction to conditioning and simple propagation algorithms can be observed. Furthermore, it is shown how all these operations can be applied to item planning and the prediction of parts demand in the automotive industry. The new theoretical results, modelling aspects, and their implementation within a software library were delivered by ISC Gebhardt and then involved in an innovative software system realized by Corporate IT for the world-wide item planning and parts demand prediction of the whole Volkswagen Group.

1 Introduction

Complex products like automobiles are usually assembled from a number of prefabricated modules and parts. Many of these components are produced in specialised facilities not necessarily located at the final assembly site. An on-time delivery failure of only one of these components can severely lower production efficiency. In order to efficiently plan the logistical processes, it is essential to give acceptable parts demand estimations at an early stage of planning.

One goal of the project described in this paper was to develop a system which plans parts demand for production sites of the Volkswagen Group. The market strategy of the Volkswagen Group is strongly customer-focused — based on adaptable designs and special emphasis on variety. Consequently, when ordering an automobile, the customer is offered several options of how each feature should be realised. The consequence is a very large number of possible car variants. Since the particular parts required for building an automobile depend on the variant of the car, the overall parts demand can not be successfully estimated from total production numbers alone.

The modelling of domains with such a large number of possible states is very complex. For many practical purposes, modelling problems are simplified by introducing strong restrictions, e.g. fixing the value of some variables, assuming simple functional relations and applying heuristics to eliminate presumably less informative variables. However, as these restrictions can be in conflict with accuracy requirements or flexibility, it is rewarding to look into methods for solving the original task. Since working with complete domains seems to be infeasible, decomposition techniques are a promising approach to this kind of problem. They are applied for instance in graphical models (Lauritzen and Spiegelhalter, 1988; Pearl, 1988; Lauritzen, 1996; Borgelt and Kruse, 2002; Gebhardt, 2000), which rely on marginal and conditional independence relations between variables to achieve a decomposition of distributions. In addition to a compact representation, graphical models allow reasoning on high dimensional spaces to be implemented using operations on lower dimensional subspaces and propagating information over a connecting structure. This results in a considerable efficiency gain.

In this paper we will show how a graphical model, when combined with certain operators, can be applied to flexibly plan parts demand in the automotive industry. We will furthermore demonstrate that such a model offers additional benefits, since it can be used for item planning, and it also provides a useful tool to simulate parts demand and capacity usage in projected market development scenarios.

2 Probabilistic Graphical Models

Graphical Models have often and successfully been applied with regard to probability distributions. The term "graphical model" is derived from an analogy between stochastic independence and node separation in graphs.

Let $V = \{A_1, \dots, A_n\}$ be a set of random variables. If the underlying distribution fulfils certain criteria (see e.g. Castillo et al., 1997), then it is possible to capture some of the independence relations between the variables in V using a graph $G = (V, E)$.

2.1 Bayesian Networks

In the case of Bayesian networks, G is a directed acyclic graph (DAG). Conditional independence between variables V_i and V_j ; $i \neq j$; $V_i, V_j \in V$ given the value of other variables $S \subseteq V$ is expressed by V_i and V_j being d-separated by S in G (Pearl, 1988; Geiger et al., 1990); i.e. there is no sequence of edges (of any directionality) between V_i and V_j such that:

1. every node of that sequence with converging edges is an element of S or has a descendant in S ,
2. every other node is not in S .

Probabilistic Bayesian networks are based on the idea that the common probability distribution of several variables can be written as a product of marginal and conditional distributions. Independence relations allow for a simplification of these products. For distributions such a factorisation can be described by a

graph. Any independence map of the original distribution that is also a DAG provides a valid factorisation. If such a graph G is known, it is sufficient to store a conditional distribution for each node attribute given its direct predecessors in G (marginal distribution if there are no predecessors) to represent the complete distribution p_V , i.e.

$$p_V \left(\bigwedge_{A_i \in V} A_i = a_i \right) = \prod_{A_i \in V} p \left(A_i = a_i \mid \bigwedge_{(A_j, A_i) \in E} A_j = a_j \right).$$

2.2 Markov Networks

Markov networks are based on similar principles, but rely on undirected graphs and the u-separation criterion instead. Two nodes are considered separated by a set S if all paths connecting the nodes contain an element from S . If G is an independence map of a given distribution, then any separation of two nodes given a set of attributes S corresponds to a conditional independence of the two given values of the attributes in S . As shown by Hammersley and Clifford (1971) a strictly positive probability distribution is factorisable w.r.t. its undirected independence graph, with the factors being nonnegative functions on the maximal cliques $C = \{C_1 \dots C_m\}$ break in G .

$$p_V \left(\bigwedge_{A_i \in V} A_i = a_i \right) = \prod_{C_i \in C} \phi_{C_i} \left(\bigwedge_{A_j \in C_i} A_j = a_j \right).$$

A detailed discussion of this topic, which includes the choice of factor potentials ϕ_{C_i} is given e.g. in Borgelt and Kruse (2002). It is worthy to note that graphical models can also be used in the context of possibility distributions. The product in the probabilistic formulae will then be replaced with the minimum.

3 Analysis of the Planning Problem

The models offered by the Volkswagen Group are typically highly flexible and therefore very rich in variants. In fact many of the assembled cars are unique with respect to the variant represented by them. It should be obvious that under these circumstances a car cannot be described by general model parameters alone. For that reason, model specifications list so called *item variables* $\{F_i : i = 1 \dots n; i, n \in \mathbb{N}\}$. Their domains $dom(F_i)$ are called *item families*. The item variables refer to various attributes like for example ‘exterior colour’, ‘seat covering’, ‘door layout’ or ‘presence of vanity mirror’ and serve as placeholders for features of individual vehicles. The elements of the respective domains are called *items*. We will use capital letters to denote item variables and indexed lower case letters for items in the associated family. A variant specification is

Table 1. Vehicle specification

Class: 'Golf'

Item	short back	2.8L 150kW spark	Type alpha	5	no	...
Item family	body variant	engine	radio	door layout	vanity mirror	...

obtained when a model specification is combined with a vector providing exactly one element for each item family (Table 1.)

For the 'Golf' class there are approximately 200 item families—each consisting of at least two, but up to 50 items. The set of possible variants is the product space $\text{dom}(F_1) \times \dots \times \text{dom}(F_n)$ with a cardinality of more than 2^{200} (10^{60}) elements. Not every combination of items corresponds to a valid variant specification (see Sec. 3.1), and it is certainly not feasible to explicitly specify variant-part lists for all possible combinations.

Apart from that, there is the manufacturing point of view. It focuses on automobiles being assembled from a number or prefabricated components, which in turn may consist of smaller units. Identifying the major components—although useful for many other tasks—does not provide sufficient detail for item planning. However, the introduction of additional structuring layers i.e. 'components of components' leads to a refinement of the descriptions. This way one obtains a tree structure with each leaf representing an installation point for alternative parts.

Depending on which alternative is chosen, different vehicle characteristics can be obtained. Part selection is therefore based on the abstract vehicle specification, i.e. on the item vector. At each installation point only a subset of item variables is relevant. Using this connection, it is possible to find partial variant specifications (*item combinations*) that reliably indicate whether a component has to be used or not. At the level of whole planning intervals this allows to calculate total parts demand as the product of the relative frequency of these relevant item combinations and the projected total production for that interval. Thus the problem of estimating parts demand is reduced to estimating the frequency of certain relevant item combinations.

3.1 Ensuring Variant Validity

When combining parts, some restrictions have to be considered. For instance, a given transmission t_1 may only work with a specific type of engine e_3 . Such relations are represented in a system of technical and marketing rules. For better readability the item variables are assigned unique names, which are used as a synonym for their symbolic designation. Using the item variables T and E ('transmission' and 'engine'), the above example would be represented as:

if 'transmission' = t_1 then 'engine' = e_3

The antecedence of a rule can be composed from a combination of conditions and it is possible to present several alternatives in the consequence part.

if 'engine' = e_2 and 'auxiliary heater' = h_3 then 'generator' $\in \{g_3, g_4, g_5\}$

Many rules state engineering requirements and are known in advance. Others refer to market observations and are provided by experts (e.g. a vehicle that combines sportive gadgets with a weak motor and automatic gear will not be considered valid, even though technically possible). The rule system covers explicit dependencies between item variables and ensures that only valid variants are considered. Since it already encodes dependence relations between item variables it also provides an important data source for the model generation step.

3.2 Additional Data Sources

In addition to the rule system it is possible to access data on previously produced automobiles. This data provides a large set of examples, but in order to use it for market oriented estimations, it has to be cleared of production-driven influences first. Temporary capacity restrictions, for example, usually only affect some item combinations and lead to their underrepresentation at one time. The converse effect will be observed, when production is back to normal, so that the deferred orders can be processed. In addition to that, the effect of starting times and the production of special models may superpose the statistics. One also has to consider that the rule system, which was valid upon generation of the data, is not necessarily identical to the current one. For that reason the production history data is used only from relatively short intervals known to be free of major disturbances (like e.g. the introduction of a new model design or supply shortages). When intervals are thus carefully selected, the data is likely to be 'sufficiently representative' to quantify variable dependences and can thus provide important additional information. Considering that most of the statistical information obtained from the database would be tedious to state as explicit facts, it is especially useful for initialising planning models.

Finally we want experts to be able to integrate their own observations or predictions into the planning model. Knowledge provided by experts is considered of higher priority than that already represented by the model. In order to deal with possible conflicts it is necessary to provide revision and updating mechanisms.

4 Generation of the Markov Network Model

It was decided to employ a probabilistic Markov network to represent the distribution of item combinations. Probabilities are thus interpreted in terms of estimated relative frequencies for item combinations. But since there are very good predictions for the total production numbers, conversion of facts based on absolute frequency is well possible. In order to create the model itself one still has to find an appropriate decomposition. When generating the model there are two data sources available, namely a rule system $\mathcal{R}$, and the production history.

4.1 Transformation of the Rule System

The dependencies between item variables as expressed in the rule system are relational. While this allows to exclude some item combinations that are inconsistent with the rules, it does not distinguish between the remaining item combinations, even though there may be significant differences in terms of their frequency. Nevertheless the relational information is very helpful in the way that it rules out all item combinations that are inconsistent with the rule system.

In addition to that, each rule scheme (the set of item variables that appear in a given rule) explicitly supplies a set of interacting variables. For our application it is also reasonable to assume that item variables are at least in approximation independent from one another given all other families, if there is no common appearance of them in any rule (unless explicitly stated so, interior colour is expected to be independent of the presence of a trailer hitch). Using the above independence assumption we can compose the relation of ‘being consistent with the rule system’. The first step consists in selecting the maximal rule schemes with respect to the subset relation. For the joint domain over the variables in each maximal rule scheme the relation can directly be obtained from the rules.

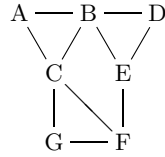
For efficient reasoning with Markov networks it is desirable that the underlying clique graph has the hypertree property. This can be ensured by graph triangulating (Figure 1c). An algorithm that performs this triangulation is given e.g. in Pearl (1988). However introducing additional edges is done at the cost of losing some more independence information. The maximal cliques in the triangulated independence graph correspond to the nodes of a hypertree (Figure 1d).

a)

$\{ABC\}$
 $\{BDE\}$
 $\{CFG\}$
 $\{EF\}$

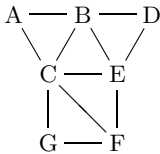
Rule schemes

b)



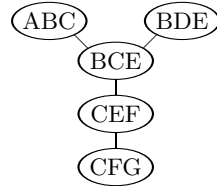
Unprocessed graph

c)



Triangulated graph

d)



Hypertree representation

Fig. 1. Transformation into hypertree structure

To complete the model we still need to assign a local distribution (i.e. relation) to each of the nodes.

For those nodes that represent the original maximal cliques in the independence graph they can be obtained from the rules that work with these item variables or a subset of them (see above). Those that use edges introduced in the triangulation process can be computed from them by combining projections, i.e. applying the conditional independence relations that have been removed from the graph when the additional edges were introduced. Since we are dealing with the relational case here this amounts to calculating a join operation.

Although such a representation is useful to distinguish valid vehicle specifications from invalid ones, the relational framework alone cannot supply us with sufficient information to estimate item rates. Therefore it is necessary to investigate a different approach.

4.2 Learning from Historical Data

A different available data source consists of variant descriptions from previously produced vehicles. However, predicting item frequencies from such data relies on the assumption that the underlying distribution does not change all too sudden. In section 3.2 considerations have been provided how to find ‘sufficiently representative’ data.

Again we can apply a Markov network to capture the distributions using the probabilistic framework this time. One can distinguish between several approaches to learn the structure of probabilistic graphical models from data. Performing an exhaustive search of possible graphs is a very direct approach. Unfortunately this method is extremely costly and infeasible for complex problems like the one given here. Many algorithms are based on dependency analysis (Sprites and Glymour, 1991; Steck, 2000; Verma and Pearl, 1992) or Bayesian statistics, e.g. K2 (Cooper and Herskovits, 1992), K2B (Khalfallah and Mellouli, 1999), CGH (Chickering et al., 1995) and the structural EM algorithm (Friedman, 1998).

Combined algorithms usually use heuristics to guide the search. Algorithms for structure learning in probabilistic graphical models typically consist of a component to generate candidate graphs for the model structure, and a component to evaluate them so that the search can be directed (Khalfallah and Mellouli, 1999; Singh and Valtorta, 1995). However even these methods are still costly and do not guarantee a result that is consistent to the rule system of our application.

Our approach is based on the fact that we do not need to rely on the production history for learning the model structure. Instead we can make use of the relational model derived from the rule system. Using the structure of the relational model as a basis and combining it with probability distributions estimated from the production history constitutes an efficient way to construct the desired probabilistic model.

Once the hypergraph is selected, it is necessary to find the factor potentials for the Markov network. For this purpose a frequentistic interpretation is assumed, i.e. estimates for the local distributions for each of the maximal cliques are ob-

tained directly from the database. In the probabilistic case there are several choices for the factor potentials because probability mass associated with the overlap of maximal cliques (separator sets) can be assigned in different ways. However for fast propagation it is often useful to store both local distributions for the maximal cliques and the local distributions for the separator sets (junction tree representation). Having copied the model structure from the relational model also provides us with additional knowledge of forbidden combinations. In the probability distributions these item combinations should be assigned a zero probability.

While the model generation based on both rule system and samples is fast, it does not completely rule out inconsistencies. One reason for that is the continuing development of the rule system. The rule system is subject to regular updates in order to allow for changes in marketing programs or composition of the item families themselves.

These problems, including the redistribution of probability mass, can be solved using belief change operations (Gebhardt and Kruse, 1998), which are described in the next section.

5 Planning Operations

A planning model that was generated using the above method, usually does not reflect the whole potential of available knowledge. For instance, experts are often aware of differences between the production history and the particular planning interval the model is meant to be used with. Thus a mechanism to modify the represented distribution is required. In addition to that we have already mentioned possible inconsistencies that arise from the use of different data sources in the learning process itself.

Planning operators have been developed to efficiently handle this kind of problem, so modification of the distribution and restoration of a consistent state can be supported.

5.1 Updating

Let us now consider the situation where previously forbidden item combinations become valid. This can result for instance from changes in the rule system. In this case neither quantitative nor qualitative information on variable interaction can be obtained from the production history. A more complex version of the same problem occurs when subsets of cliques are to be altered while the information in the remaining parts of the network is retained, for instance after the introduction of rules with previously unused schemes (Gebhardt et al., 2003). In both cases it is necessary to provide the probabilistic interaction structure—a task performed with the help of the updating operation.

The updating operation marks these combinations as valid by assigning a positive near zero probability to their respective marginals in the local distributions. Since the replacement value is very small compared to the true item frequencies obtained from the data, the quality of estimation is not affected by this alteration. Now instead of using the same initialisation for all new item

combinations, the proportion of the values is chosen in accordance to an existing combination, i.e. the probabilistic interaction structure is copied from reference item combinations. This also explains why it is not convenient to use zero itself as an initialisation. The positive values are necessary to carry qualitative dependency information. For illustration consider the introduction of a new value t_4 to item family transmission. The planners predict that the new item distributes similarly to the existing item t_3 . If they specify t_3 as a reference, the updating operation will complete the local distributions that involve T , such that the marginals for the item combinations that include t_4 are in the same ratio to each other as their respective counterparts with t_3 instead.

Since updating only provides the qualitative aspect of dependency structure, it is usually followed by the subsequent application of the revision operation, which can be used to reassess probability mass to the new item combinations.

5.2 Revision

After the model has been generated, it is further adapted to the requirements of the particular planning interval. The information used at this stage is provided by experts and includes marketing and sales stipulations. It is usually specific to the planning interval. Such additional information can be integrated into the model using the revision operator. The input data consists of predictions or restrictions for installation rates of certain items, item combinations or even sets of either. It also covers the issue of unexpected capacity restrictions, which can be expressed in this form.

Although the new information is frequently in conflict with prior knowledge, i.e. the distribution previously represented in the model, it usually has an important property—namely that it is compatible with the independence relations, which are represented in the model structure.

The revision operation, while preserving the network structure, serves to modify quantitative knowledge in such a way that the revised distribution becomes consistent with the new specialised information. There is usually no unique solution to this task. However, it is desirable to retain as much of the original distribution as possible so the principle of minimal change (Gärdenfors, 1988) should be applied. Given that, a successful revision operation holds a unique result (Gebhardt et al., 2004).

The operation itself starts by modifying a single marginal distribution. Using the iterative proportional fitting method, first the local clique and ultimately the whole network is adapted to the new information. Since revision relies on the qualitative dependency structure already present, one can construct cases where revision is not possible. In such cases an updating operation is required before revision can be applied. In addition to that the supplied information can be contradictory in itself. Such situations are sometimes difficult to recognise. Criteria that entail a successful revision and proves for the maximum preservation of previous knowledge have been provided in Gebhardt et al. (2004). Gebhardt (2001) deals with the problem of inconsistent information and how the revision operator itself can help dealing with it.

Depending on circumstances human experts may want to specify their knowledge in different ways. Sometimes it is more convenient to give an estimation of future item frequency in absolute numbers, while at a different occasion it might be preferable to specify item rates or a relative increase. With the help of some readily available data and the information which is already represented in the network before revision takes place, such inputs can be transformed to item rates. From the operator's point of view this can be very useful.

As an example for a specification using item rates experts might predict a rise of the popularity of a recently introduced navigation system and set the relative frequency of this respective item from 20% to 30%. Sometimes the stipulations are embedded in a context as in "The frequency of air conditioning for Golfs with all wheel drive in France will increase by 10%". In such cases the statements can be transformed and amount to a changing the ratio of the rates for the combination of all items in the statement (air conditioning present, all wheel drive, France) to the rates of that, which only includes the items from the context (all wheel drive, France).

5.3 Focussing

While revision and updating are essential operations for building and maintaining a distribution model, it is a much more common activity to apply the model for the exploration of the represented knowledge and its implications with respect to user decisions.

Typically users would want to concentrate on those aspects of the represented knowledge that fall into their domain of expertise. Moreover, when predicting parts demand from the model, one is only interested in estimated rates for particular item combinations (see Sec. 3). Such activities require a focussing operation. It is achieved by performing evidence-driven conditioning on a subset of variables and distributing the information through the network. The well-known variable instantiation can be seen as a special case of focussing where all probability is assigned to exactly one value per input variable.

As with revision, context dependent statements can be obtained by returning conditional probabilities. Furthermore, item combinations with compatible variable schemes can be grouped at the user interface providing access to aggregated probabilities.

Apart from predicting parts demand, focussing is often employed for market analyses and simulation. By analysing which items are frequently combined by customers, experts can tailor special offers for different customer groups. To support planning of buffer capacities, it is necessary to deal with the eventuality of temporal logistic restrictions. Such events would entail changes in short term production planning so that the consumption of the concerned parts is reduced. This in turn affects the overall usage of other parts. The model can be used to simulate scenarios defined by different sets of frame conditions, to test adapted production strategies and to assess the usage of all parts.

6 Application

The results obtained in this paper have contributed to the development of the planning system EPL (EigenschaftenPLANung, item planning). It was initiated in 2001 by Corporate IT, Sales, and Logistics of the Volkswagen Group. The aim was to establish for all trademarks a common item planning system that reflects the presented modelling approach based on Markov networks.

System design and most of the implementation work of EPL is currently done by Corporate IT. The mathematical modelling, theoretical problem solving, and the development of efficient algorithms, extended by the implementation of a new software library called MARNEJ (MARKov NETworks in Java) for the representation and the presented functionalities on Markov networks have been entirely provided by ISC Gebhardt.

Since 2004 the system EPL is being rolled out to all trademarks of the Volkswagen group and step by step replaces the previously used planning systems. In order to promote acceptance and to help operators adapt to the new software and its additional capabilities, the user interface has been changed gradually. In parallel planners have been introduced to the new functionality, so that EPL can be applied efficiently.

In the final configuration the system will have 6 to 8 Hewlett Packard Machines running Linux with 4 AMD Opteron 64-Bit CPUs and 16 GB of main memory each.

With the new software, the increasing planning quality, based on the many innovative features and the appropriateness of the chosen model of knowledge representation, as well as a considerable reduction of calculation time turned out to be essential prerequisites for advanced item planning and calculation of parts demand in the presence of structured products with an extreme number of possible variants.

References

- C. Borgelt and R. Kruse. *Graphical Models—Methods for Data Analysis and Mining*. J. Wiley & Sons, Chichester, 2002.
- E. Castillo, J.M. Gutiérrez, and A.S. Hadi. *Expert Systems and Probabilistic Network Models*. Springer-Verlag, New York, 1997.
- D.M. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks from data. *Machine Learning*, 20(3):197–243, 1995.
- G.F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.
- N. Friedman. The Bayesian structural EM algorithm. In *Proc. of the 14th Conference on Uncertainty in AI*, pages 129–138, 1998.
- P. Gärdenfors. *Knowledge in the Flux—Modeling the Dynamics of Epistemic States*. MIT press, Cambridge, MA, 1988.
- J. Gebhardt. The revision operator and the treatment of inconsistent stipulations of item rates. Project EPL: Internal Report 9. ISC Gebhardt and Volkswagen Group, GOB-11, 2001.

- J. Gebhardt. Learning from data: Possibilistic graphical models. In D. M. Gabbay and P. Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 4: Abductive Reasoning and Learning, pages 314–389. Kluwer Academic Publishers, Dordrecht, 2000.
- J. Gebhardt and R. Kruse. Parallel combination of information sources. In D. M. Gabbay and P. Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 3: Belief Change, pages 393–439. Kluwer Academic Publishers, Dordrecht, 1998.
- J. Gebhardt, H. Detmer, and A.L. Madsen. Predicting parts demand in the automotive industry – an application of probabilistic graphical models. In *Proc. Int. Joint Conf. on Uncertainty in Artificial Intelligence (UAI’03, Acapulco, Mexico), Bayesian Modelling Applications Workshop*, 2003.
- J. Gebhardt, C. Borgelt, and R. Kruse. Knowledge revision in markov networks. *Mathware and Soft Computing*, 11(2-3):93–107, 2004.
- D. Geiger, T.S. Verma, and J. Pearl. Identifying independence in Bayesian networks. *Networks*, 20:507–534, 1990.
- J.M. Hammersley and P.E. Clifford. Markov fields on finite graphs and lattices. Cited in Isham (1981), 1971.
- V. Isham. An introduction to spatial point processes and markov random fields. *Int. Statistical Review*, 49:21–43, 1981.
- F. Khalfallah and K. Mellouli. Optimized algorithm for learning Bayesian networks from data. In *Proc. 5th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQUARU’99)*, pages 221–232, 1999.
- S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 2(50):157–224, 1988.
- S.L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.
- J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, San Mateo, USA, 1988. (2nd edition 1992).
- M. Singh and M. Valtorta. Construction of Bayesian network structures from data: Brief survey and efficient algorithm. *Int. Journal of Approximate Reasoning*, 12: 111–131, 1995.
- P. Sprites and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computing Review*, 9(1):62–72, 1991.
- H. Steck. On the use of skeletons when learning Bayesian networks. In *Proc. of the 16th Conference on Uncertainty in AI*, pages 558–565, 2000.
- T. Verma and J. Pearl. An algorithm for deciding whether a set of observed independencies has a causal explanation. In *Proc. 8th Conference on Uncertainty in AI*, pages 323–330, 1992.

Some Representation and Computational Issues in Social Choice

Jérôme Lang

IRIT - Université Paul Sabatier and CNRS,
31062 Toulouse Cedex (France)
`lang@irit.fr`

Abstract. This paper briefly considers several research issues, some of which are on-going and some others are for further research. The starting point is that many AI topics, especially those related to the EC-SQARU and KR conferences, can bring a lot to the representation and the resolution of social choice problems. I surely do not claim to make an exhaustive list of problems, but I rather list *some* problems that I find important, give some relevant references and point out some potential research issues¹.

1 Introduction

For a few years, Artificial Intelligence has been taking more and more interest in collective decision making. There are two main reasons for that, leading to two different lines of research. Roughly speaking, the first one is concerned with importing concepts and procedures from social choice theory for solving questions that arise in AI application domains. This is typically the case for managing societies of autonomous agents, which calls for negotiation and voting procedures. The second line of research, which is the focus of this position paper, goes the other way round: it is concerned with importing notions and methods from AI for solving questions originally stemming from social choice.

Social choice is concerned with designing and evaluating methods of collective decision making. However, it somewhat neglects computational issues: the problem is generally considered to be solved when the existence (or the non-existence) of a procedure meeting some requirements has been shown; more precisely, knowing that the procedure *can* be computed is generally enough; now, *how hard* this computation is, and how the procedure should be implemented, have deserved less attention in the social choice community. This is where AI (and operations research, and more generally computer science) comes into play. As often when bringing together two traditions, AI probably raises more new

¹ Writing a short survey is a difficult task, especially because it always leads to leaving some relevant references aside. I'll maintain a long version of this paper, accessible at <http://www.irit.fr/recherches/RPDMP/persos/JeromeLang/papers/ecsqaru05-long.pdf>, and I'll express my gratitude to everyone who'll point to me any missing relevant reference.

questions pertaining to collective decision making than it solves old ones. One of the most relevant of these issues consists in considering group decision making problems when the set of alternative is finite and has a *combinatorial structure*.

This paper gives a brief overview of some research issues along this line. Section 2 starts with the crucial problem of eliciting and representing the individual's preferences on the possible alternatives. Section 3 focuses on preference aggregation, Section 4 on vote, and Section 5 on fair division. Section 6 evokes other directions deliberately ignored in this short paper.

2 Elicitation and Compact Representation of Preference

Throughout the paper, $N = \{1, \dots, n\}$ is the (finite) set of *agents* involved in the collective choice and X is the finite set of *alternatives* on which the decision process bears.

Any individual or collective decision making problem needs some description (at least partial) of the preferences of each of the agents involved over the possible alternatives. A *numerical* preference structure is a utility function $u : X \rightarrow \mathbb{R}$. An *ordinal* preference structure is a preorder P on X , called preference relation. $R(x, y)$ is denoted alternatively by $x \succeq y$. $\succ$ denotes strict preference ($x \succ y$ if and only if $x \succeq y$ and not $y \succeq x$) and $\sim$ denotes indifference ($x \sim y$ if and only if $x \succeq y$ and $y \succeq x$). An intermediate model between pure ordinality and pure numerical models is that of *qualitative preferences*, consisting of (qualitative) utility functions $u : X \rightarrow L$, where L is a totally ordered (yet not numerical) scale. Unlike ordinal preferences, qualitative preferences allow commensurability between uncertainty and preference scales as well as interagent comparison of preferences (see [22] for discussions on ordinality in decision making.)

The choice of a *model*, i.e. a mathematical structure, for preference, does not tell how agents' preferences are obtained from them, stored, and handled by algorithms. *Preference representation* consists in choosing a language for encoding preferences so as to spare computational resources. The choice of a language is guided by two tasks: upstream, *preference elicitation* consists in interacting with the agent so as to obtain her preferences over X , while *optimization* consists in finding nondominated alternatives from a compactly represented input.

As long as the set of alternatives has a small size, the latter problems are computationally easy. Unfortunately, in many concrete problems the set of alternatives has a *combinatorial structure*. A combinatorial domain is a Cartesian product of finite value domains for each one of a set of variables: an alternative in such a domain is a tuple of values. Clearly, the size of such domains grows exponentially with the set of variables and becomes quickly very large, which makes explicit representations and straightforward elicitation and optimization no longer reasonable. Logical or graphical *compact representation languages* allow for representing in as little space as possible a preference structure whose size would be prohibitive if it were represented explicitly. The literature on preference elicitation and representation for combinatorial domains has been growing fastly for a few years, and due to the lack of space I omit giving references here.

The criteria one can use for choosing a compact preference language include, at least, the following ones:

- *cognitive relevance*: a language should be as close as possible to the way human agents “know” their preferences and express them in natural language;
- *elicitation-friendliness*: it should be easy to design algorithms to elicit preference from an agent so as to get an output expressed in a given language;
- *expressivity*: find out the set of preference relations or utility functions that can be expressible in a given language;
- *complexity*: given an input consisting of a compactly represented preference structure in a given language, determine the computational complexity of finding a non-dominated alternative, checking whether an alternative is preferred to another one, whether an alternative is non-dominated etc.;
- *comparative succinctness*: given two languages L and L' , determine whether every preference structure that can be expressed in L can also be expressed in L' without a significant (suprapolynomial) increase of size, in which case L' is said to be at least as succinct as L .

Cognitive relevance is somewhat hard to assess, due to its non-technical nature, and has been rarely studied. Complexity has been studied in [35] for logic-based languages. Expressivity and comparative succinctness have been systematically investigated in [19] for *ordinal* preference representation.

Although these languages have been designed for single agents, they can be extended to multiple agents without much difficulty; [34] and [44] are two examples of such extensions.

3 Preference Aggregation

Preference aggregation, even on simple domains, raises challenging computational issues that have been recently investigated by AI researchers. Aggregating preferences consist in mapping a collection $\langle P_1, \dots, P_n \rangle$ of preference relations (or *profiles*) into a collective preference relation P^* (which implies circumventing Arrow’s impossibility theorem [2] by relaxing one of its applicability conditions.)

Now, even on simple domains, some aggregation functions raise computational difficulties. This is notably the case for Kemeny’s aggregation rule, consisting in aggregating the profiles into a profile (called *Kemeny consensus*) being closest to the n profiles, with respect to a distance which, roughly speaking, is the sum, for all agents, of the numbers of pairs of alternatives on which the aggregated profile disagrees with the agent’s profile. Computing a Kemeny consensus is NP-hard; [21] addresses its practical computation.

When the set of alternatives has a combinatorial structure, things get much worse. Moreover, since in that case preferences are often described in a compact representation language, aggregation should ideally operate directly on this language, without generating the individual nor the aggregated preferences explicitly. A common way of aggregating compactly represented preferences is (*logical*) *merging*. The common point of logic-based merging approaches is that

the set of alternatives corresponds to a set of propositional worlds; the logic-based representation of agent's preferences (or beliefs) then induces a cardinal function (using ranks or distances) on worlds and aggregates these cardinal preferences. These functions are not necessarily on a numerical scale but the scale has to be common to all agents. We do not have the space to give all relevant references to logic-based merging here, but we give a few of them, which explicitly mention some social choice theoretic issues: [33, 40, 13, 39]. See also [34, 6] for preference aggregation from logically expressed preferences. .

4 Vote

Voting is one of the most popular ways of reaching common decisions. Researchers in social choice theory have studied extensively the properties of various families of voting rules, but, again, have neglected computational issues. A *voting rule* maps each collection of individual preference profiles, generally consisting of linear orders over the set of candidates, to a *nonempty subset* of the set of candidates; if the latter subset is always a singleton then the voting rule is said to be *deterministic*².

For a panorama of voting rules see for instance [10]. We just give here a few of them. A *positional scoring rule* is defined from a *scoring vector*, i.e. a vector $\mathbf{s} = (s_1, \dots, s_m)$ of integers such that $s_1 \geq s_2 \geq \dots \geq s_m$ and $s_1 > s_m$. Let $\text{rank}_i(x)$ be the rank of x in $\succ_i$ (1 if it is the favorite candidate for voter i , 2 if it is the second favorite etc.), then the score of x is $S(x) = \sum_{i=1}^N s_{\text{rank}_i(x)}$. Two well-known examples of positional scoring procedures are the *Borda rule*, defined by $s_k = m - k$ for all $k = 1, \dots, m$, and the *plurality rule*, defined by $s_1 = 1$, and $s_k = 0$ for all $k > 1$. Moreover, a *Condorcet winner* is a candidate preferred to any other candidate by a strict majority of voters. (it is well-known that there are some profiles for which no Condorcet winner exists.) Obviously, when there exists a Condorcet winner then it is unique. A *Condorcet-consistent rule* is a voting rule electing the Condorcet winner whenever there is one.

The first question that comes to mind is whether determining the outcome of an election, for a given voting procedure, is computationally challenging (which is all the more relevant as electronic voting becomes more and more popular.)

4.1 Computing the Outcome of Voting Rules: Small Domains

Most voting rules among those that are practically used are computable in linear or quadratic time in the number of candidates (and almost always linear in the number of voters); therefore, when the number of candidates is small (which is typically the case for political elections where a single person has to be elected), computing the outcome of a voting rule does not need any sophisticated algorithm. However, a few voting rules are computationally complex. Here are three

² The literature of social choice theory rather makes use of the terminology “voting correspondances” and “deterministic voting rules” but for the sake of simplicity we will make use of the terminology “voting rules” in a uniform way.

of them: Dodgson’s rule and Young’s rule both consist in electing candidates that are closest to being a Condorcet winner: each candidate is given a score that is the smallest number of exchanges of *elementary changes* in the voters’ preference orders needed to make the candidate a Condorcet winner. Whatever candidate (or candidates, in the case of a tie) has the lowest score is the winner. For Dodgson’s rule, an elementary change is an exchange of adjacent candidates in a voter’s preference profile, while for Young’s rule it is the removal of a voter. Lastly, Kemeny’s voting rule elects a candidate if and only if it is the preferred candidate in some Kemeny consensus (see Section 3). Deciding whether a given candidate is a winner for any of the latter three voting rules is a $\Delta_2^P(O(\log n))$ -complete (for Dodgson’s, NP-hardness was shown in [5] and $\Delta_2^P(O(\log n))$ -completeness in [30]; $\Delta_2^P(O(\log n))$ -completeness was shown in [45] for Young’s and in [31] for Kemeny’s).

4.2 Computing the Outcome of Voting Rules: Combinatorial Domains

Now, when the set of candidates has a combinatorial structure, even simple procedures such as plurality and Borda become hard. Consider an example where agents have to agree on a common menu to be composed of a first course dish, a main course dish, a dessert and a wine, with a choice of 6 items for each. This makes 6^4 candidates. This would not be a problem if the four items to be chosen were independent from the other ones: in this case, this vote problem over a set of 6^4 candidates would come down to four independent problems over sets of 6 candidates each, and any standard voting rule could be applied without difficulty. But things get complicated if voters express dependencies between variables, such as “I prefer white wine if one of the courses is fish and none is meat, red wine if one of the courses is meat and none is fish, and in the remaining cases I would like equally red or white wine”, etc.

Obviously, the prohibitive number of candidates makes it hard, or even practically impossible, to apply voting rules in a straightforward way. The computational complexity of some voting procedures when applied to compactly represented preferences on a combinatorial set of candidates has been investigated in [35]; however this paper does not address the question of *how* the outcome can be computed in a reasonable amount of time.

When the domain is large enough, computing the outcome by first generating the whole preference relations on the combinatorial domain from their compact representation is unfeasible. A first way of coping with the problem consists in contenting oneself with an *approximation* of the outcome of the election, using incomplete and/or randomized algorithms making a possible use of heuristics. This is an open research issue.

A second way consists in decomposing the vote into *local* votes on individual variables (or small sets of variables), and gathering the results. However, as soon as variables are not preferentially independent, it is generally a bad idea: “multiple election paradoxes” [11] show that such a decomposition leads to suboptimal choices, and give real-life examples of such paradoxes, including simultaneous

referenda on related issues. We give here a very simple example of such a paradox. Suppose 100 voters have to decide whether to build a swimming pool or not (S), and whether to build a tennis court or not (T). 49 voters would prefer a swimming pool and no tennis court ($S\bar{T}$), 49 voters prefer a tennis court and no swimming pool ($\bar{S}T$) and 2 voters prefer to have both (ST). Voting separately on each of the issues gives the outcome ST , although it received only 2 votes out of 100 – and it might even be the most disliked outcome by 98 of the voters (for instance because building both raises local taxes too much). Now, the latter example did not work because there is a *preferential dependence* between S and T . A simple idea then consists in exploiting *preferential independencies* between variables; this is all the more relevant as graphical languages, evoked in Section 2, are based on such structural properties. The question now is to what extent we may use these preferential independencies to *decompose* the computation of the outcome into smaller problems. However, again this does not work so easily: several well-known voting rules (such as plurality or Borda) cannot be decomposed, *even when the preferential structure is common to all voters*. Most of them fail to be decomposable *even when all variables are mutually independent for all voters*. We give below an example of this phenomenon.

Consider 7 voters, a domain with two variables $\mathbf{x}$ and $\mathbf{y}$, whose domains are respectively $\{x, \bar{x}\}$ and $\{y, \bar{y}\}$, and the following preference relations, where each agent expresses his preference relation by a CP-net [7] corresponding to the following fixed preferential structure: preference on $\mathbf{x}$ is unconditional and preference on $\mathbf{y}$ may depend on the value given to $\mathbf{x}$.

3 voters	2 voters	2 voters
$\bar{x} \succ x$	$x \succ \bar{x}$	$x \succ \bar{x}$
$x : \bar{y} \succ y$	$x : y \succ \bar{y}$	$x : \bar{y} \succ y$
$\bar{x} : y \succ \bar{y}$	$\bar{x} : \bar{y} \succ y$	$\bar{x} : y \succ \bar{y}$

For instance, the first CP-net says that the voters prefer $\bar{x}$ to x unconditionally, prefer $\bar{y}$ to y when $\mathbf{x} = x$ and y to $\bar{y}$ when $\mathbf{x} = \bar{x}$. This corresponds to the following preference relations:

3 voters	2 voters	2 voters
$\bar{x}\bar{y}$	xy	$x\bar{y}$
$\bar{x}y$	$x\bar{y}$	xy
$x\bar{y}$	$\bar{x}\bar{y}$	$\bar{x}y$
xy	$\bar{x}y$	$\bar{x}\bar{y}$

The winner for the plurality rule is $\bar{x}y$. Now, the sequential approach gives the following outcome: first, because 4 agents out of 7 unconditionally prefer x over $\bar{x}$, applying plurality (as well as any other voting rule, since all reasonable voting rules coincide with the majority rule when there are only 2 candidates)

locally on $\mathbf{x}$ leads to choose $\mathbf{x} = x$. Now, given $\mathbf{x} = \text{true}$, 5 agents out of 7 prefer $\bar{y}$ to y , which leads to choose $\mathbf{y} = \bar{y}$. Thus, the sequential plurality winner is $(x, \bar{y})$ – whereas the direct plurality winner is $(\bar{x}, y)$.

Such counterexamples can be found for many other voting rules. This raises the question of finding voting rules which can be decomposed into local rules (possibly under some domain restrictions), following the preferential independence structure of the voters' profiles – which is an open issue.

4.3 Manipulation

Manipulating a voting rule consists, for a given voter or coalition of voters, in expressing an insincere preference profile so as to give more chance to a preferred candidate to be elected. Gibbard and Satterthwaite's theorem [29, 47] states that if the number of candidates is at least 3, then any nondictatorial voting procedure is manipulable for some profiles.

Consider again the example above with the 7 voters³, and the plurality rule, whose outcome is $\bar{x}y$. The two voters whose true preference is $xy \succ x\bar{y} \succ \bar{x}\bar{y} \succ \bar{x}y$ have an interest to report an insincere preference profile with $x\bar{y}$ on top, that is, to vote for $x\bar{y}$ – in that case, the winner is $x\bar{y}$, which these two voters prefer to the winner if they express their true preferences, namely $\bar{x}y$.

Since it is theoretically not possible to make manipulation impossible, one can try to make it less efficient or more difficult. Making manipulation less efficient can consist in making as little as possible of the others' votes known to the would-be manipulating voter – which may be difficult in some contexts. Making it more *difficult to compute* is a way followed recently by [4, 3, 15, 14, 17]. The line of argumentation is that if finding a successful manipulation is extremely hard computationally, then the voters will give up trying to manipulate and express sincere preferences. Note that, for once, the higher the complexity, the better. Randomization can play a role not only in making manipulation less efficient but also more complex to compute [17].

In a logical merging context (see Section 3), [27] investigate the manipulation of merging processes in propositional logic. The notion of a manipulation is however more complex to define (and several competing notions are discussed indeed), since the outcome of the process is a full preference relation.

4.4 Incomplete Knowledge and Communication Complexity

Given some incomplete description of the voters' preferences, is the outcome of the vote determined? If not, whose preferences are to be elicited and what is relevant so as to compute the outcome? Assume, for example, that we have 4 candidates A, B, C, D and 9 voters, 4 of which vote $C \succ D \succ A \succ B$, 2 of which vote $A \succ B \succ D \succ C$ and 2 of which vote $B \succ A \succ C \succ D$, the last vote being still unknown. If the plurality rule is chosen then the outcome is already known (the winner is C) and there is no need to elicit the last voter's profile. If the Borda rule is used then the partial scores are $A : 14, B : 10, C : 14, D : 10$,

³ I thank Patrice Perny, from whom I borrowed this example.

therefore the outcome is not determined; however, we do not need to know the totality of the last vote, but we only need to know whether the last voter prefers A to C or C to A . This *vote elicitation* problem is investigated from the point of view of computational complexity in [16].

More generally, communication complexity is concerned with the amount of information to be communicated so that the outcome of the vote procedure is determined: since the outcome of a voting rule is sometimes determined even if not all votes are known, this raises the question in designing protocols for gathering the information needed so as to communicate as little info as possible [18]. For example, plurality needs only to know top ranked candidates, while plurality with run-off needs the top-ranked candidates and then, after communicating the names of two finalists to the voters, which one they prefer between these two.

5 Fair Division

Resource allocation of *indivisible goods* aims at assigning, to each of a set of agents N , some items from a finite set R to each of a set of agents N , given their preferences over all possible combination of objects. For the sake of simplicity, we assume here that each resource must be given to one and only one agent⁴.

In *centralized* allocation problems, the assignment is determined by a central authority to which the agents have given their preferences beforehand. As it stands, a centralized fair division problem is clearly a group decision making problem on a combinatorial domain, since the number of allocations grows exponentially with the number of resources. Since the description of a fair division problem needs the specification of the agents' preferences over the set of all possible combinations of objects, elicitation and compact representation issues are highly relevant here as well. Now, is a fair division problem a vote problem, where candidates are possible allocations? Not quite, because a usual assumption is made, stating that the primary preferences expressed by agents depends only of their share, that is, agent i is indifferent between two allocations as soon as they give her the same share. Furthermore, as seen below, some specific notions for fair division problems, such as envy-freeness, have no counterpart in terms of voting.

Two classes of criteria are considered in centralized resource allocation, namely *efficiency* and *equity* (or *fairness*). At one extremity, *combinatorial auctions* consist in finding an allocation maximizing the revenue of the seller, where this revenue is the sum, over all agents, of the price that the agent is willing to pay for the combination of objects he receives in the allocation (given that these price functions are not necessarily additive.) Combinatorial auctions are a very spe-

⁴ More generally, an object could be allocated to zero, one, or more agents of N . Even if most applications require the allocation to be *preemptive* (an object cannot be allocated to more than one agent), some problems do not require it. An example of such preemption-free problems is the exploitation of shared Earth observation satellites described in [36, 8].

cific, purely utilitarianistic class of allocation problems, in which considerations such as equity and fairness are not relevant. They have received an enormous attention since a few years (see [20]). Here we rather focus on allocation problems where fairness is involved – in which case we speak of *fair division*.

The weakest efficiency requirement is that allocations should not be Pareto-dominated: an allocation $\pi : N \rightarrow 2^X$ is *Pareto-efficient* if and only if there is no allocation π' such that (a) for all i , $\pi'(i) \succeq_i \pi(i)$ and (b) there exists an i such that $\pi'(i) \succ_i \pi(i)$. Pareto-efficiency is purely ordinal, unlike the *utilitarianistic* criterion, applicable only when preference are numerical, under which an allocation π is preferred to an allocation π' if and only if $\sum_{i \in N} u_i(\pi(i)) > \sum_{i \in N} u_i(\pi'(i))$.

None of the latter criteria deals with fairness or equity. The most usual way of measuring equity is *egalitarianism*, which compares allocations with respect to the *leximin* ordering which, informally, works by comparing first the utilities of the least satisfied agents, and when these utilities coincide, compares the utilities of the next least satisfied agents and so on (see for instance Chapter 1 of [41]).

The *leximin* ordering does not need preferences to be numerical but only *interpersonally comparable*, that is, expressed on common scale. A purely ordinal fairness criterion is *envy-freeness* : an allocation π is *envy-free* if and only if $\pi(i) \succeq_i \pi(j)$ holds for all i and all $j \neq i$, or in informal terms, each agent is at least as happy with his share than with any other one's share. It is well-known that there exist allocation problems for which no there exists no allocation being both *Pareto-efficient* and *envy-free*.

In *distributed* allocation problems, agents negotiate, communicate, exchange or trade goods, in a multilateral way. Works along this line have addressed the convergence conditions towards allocations being optimal from a social point of view, depending on the acceptability criteria used by agents when deciding whether or not to agree on a propose exchange of resources, and some constraints allowed on deals – see e.g. [46, 26, 24, 23, 12]. The notion of communication complexity is revisited in [25] and reinterpreted as the minimal (with respect to some criteria) sequence of deals between agents (where minimality is with respect to a criterion that may vary, and which takes into account the number of deals and the number of objects exchanged in deals). See [38] for a survey on these issues.

Whereas social choice theory has developed an important literature on fair division, and artificial intelligence has devoted much work on the computational aspects of combinatorial auctions, computational issues in *fair division* have only started recently to be investigated. Two works addressing envy-freeness from a computational perspective are [37], who compute approximately envy-free solutions (by first making it a graded notion, suitable to optimization), and [9] who relate the search of envy-freeness and efficient allocations to some well-known problems in knowledge representation. A more general review of complexity results for centralized allocation problems in in [8]. Complexity issues for distributed allocation problems are addressed in [24].

Clearly, many models developed in the AI community should have an impact on modelling, representing compactly and solving fair division problems. Moreover, some issues addressed for voting problems and/or combinatorial auctions, such as the computational aspects of elicitation and manipulation and the role of incomplete knowledge, are still to be investigated for fair division problems.

6 Conclusion

There are many more issues for further research than those that we have briefly evoked. Models and techniques from artificial intelligence should play an important role, for (at least) the following reasons:

- the importance of *ordinal and qualitative* models in preference aggregation, vote and fair division (no need to recall that the AI research community has contributed a lot to the study of these models.) Ordinality is perhaps even more relevant in social choice than in decision under uncertainty and multicriteria decision making, due to equity criteria and the difficulty of interpersonal comparison of preference.
- the role of *incomplete knowledge*, and the need to reason about agents' beliefs, especially in utility elicitation and communication complexity issues. Research issues include various ways of applying voting and allocation procedures under incomplete knowledge, and the study of communication protocols for these issues, which may call for multiagent models of beliefs, including mutual and common belief (see e.g. [28]). Models and algorithms for group decision under uncertainty is a promising topic as well.
- the need for *compact (logical and graphical) languages* for preference elicitation and representation and measure their *spatial efficiency*. These languages need to be extended to multiple agents (such as in [44]), and aggregation should be performed directly in the language (e.g., aggregating CP-nets into a new CP-net without generating the preference relations explicitly).
- the high complexity of the tasks involved leads to interesting algorithmic problems such as finding tractable subclasses, efficient algorithms and approximation methods, using classical AI and OR techniques.
- one more relevant issue is *sequential group decision making and planning* with multiple agents. For instance, [42] address the search for an optimal path for several agents (or criteria), with respect to an egalitarianistic aggregation policy.
- measuring and localizing inconsistency among a group of agents – especially when preferences are represented under a logical form – could be investigated by extending *inconsistency measures* (see [32]) to multiple agents.

References

1. H. Andreka, M. Ryan, and P.-Y. Schobbens. Operators and laws for combining preference relations. *Journal of Logic and Computation*, 12(1):13–53, 2002.
2. K. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, 1951. revised edition 1963.

3. J.J. Bartholdi and J.B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.
4. J.J. Bartholdi, C.A. Tovey, and M.A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.
5. J.J. Bartholdi, C.A. Tovey, and M.A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(3):157–165, 1989.
6. S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Bipolar representation and fusion of preference in the possibilistic logic framework. In *Proceedings of KR2002*, pages 421–429, 2002.
7. C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole. CP-nets: a tool for representing and reasoning with conditional *ceteris paribus* statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.
8. S. Bouveret, H. Fargier, J. Lang, and M. Lemaître. Allocation of indivisible goods: a general model and some complexity results. In *Proceedings of AAMAS 05*, 2005. Long version available at <http://www.irit.fr/recherches/RPDMP/persos/JeromeLang/papers/aig.pdf>.
9. S. Bouveret and J. Lang. Efficiency and envy-freeness in fair division of indivisible goods: logical representation and complexity. In *Proceedings of IJCAI-05*, 2005.
10. S. Brams and P. Fishburn. Voting procedures. In K. Arrow, A. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, chapter 4. Elsevier, 2004.
11. S. Brams, D. M. Kilgour, and W. Zwicker. The paradox of multiple elections. *Social Choice and Welfare*, 15:211–236, 1998.
12. Y. Chevaleyre, U. Endriss, and N. Maudet. On maximal classes of utility functions for efficient one-to-one negotiation. In *Proceedings of IJCAI-2005*, 2005.
13. S. Chopra, A. Ghose, and T. Meyer. Social choice theory, belief merging, and strategy-proofness. *Int. Journal on Information Fusion*, 2005. To appear.
14. V. Conitzer, J. Lang, and T. Sandholm. How many candidates are required to make an election hard to manipulate? In *Proceedings of TARK-03*, pages 201–214, 2003.
15. V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of AAAI-02*, pages 314–319, 2002.
16. V. Conitzer and T. Sandholm. Vote elicitation: complexity and strategy-proofness. In *Proceedings of AAAI-02*, pages 392–397, 2002.
17. V. Conitzer and T. Sandholm. Universal voting protocols to make manipulation hard. In *Proceedings of IJCAI-03*, 2003.
18. V. Conitzer and T. Sandholm. Communication complexity of common voting rules. In *Proceedings of the EC-05*, 2005.
19. S. Coste-Marquis, J. Lang, P. Liberatore, and P. Marquis. Expressive power and succinctness of propositional languages for preference representation. In *Proceedings of KR-2004*, pages 203–212, 2004.
20. P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2005. To appear.
21. A. Davenport and J. Kalagnanam. A computational study of the Kemeny rule for preference aggregation. In *Proceedings of AAAI-04*, pages 697–702, 2004.
22. D. Dubois, H. Fargier, and P. Perny. On the limitations of ordinal approaches to decision-making. In *Proceedings of KR2002*, pages 133–146, 2002.
23. P. Dunne. Extremal behaviour in multiagent contract negotiation. *Journal of Artificial Intelligence Research*, 23:41–78, 2005.
24. P. Dunne, M. Wooldridge, and M. Laurence. The complexity of contract negotiation. *Artificial Intelligence*, 164(1-2):23–46, 2005.

25. U. Endriss and N. Maudet. On the communication complexity of multilateral trading: Extended report. *Journal of Autonomous Agents and Multiagent Systems*, 2005. To appear.
26. U. Endriss, N. Maudet, F. Sadri, and F. Toni. On optimal outcomes of negotiations over resources. In *Proceedings of AAMAS-03*, 2003.
27. P. Everaere, S. Konieczny, and P. Marquis. On merging strategy-proofness. In *Proceedings of KR-2004*, pages 357–368, 2004.
28. R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.
29. A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.
30. E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll’s 1876 system is complete for parallel access to NP. *JACM*, 44(6):806–825, 1997.
31. E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. Technical report, Jenaer Schriften zur Mathematik und Informatik, October 2003.
32. A. Hunter and S. Konieczny. *Approaches to measuring inconsistent information*, pages 189–234. SpringerLNC3 3300, 2004.
33. S. Konieczny and R. Pino Pérez. Propositional belief base merging or how to merge beliefs/goals coming from several sources and some links with social choice theory. *European Journal of Operational Research*, 160(3):785–802, 2005.
34. C. Lafage and J. Lang. Logical representation of preferences for group decision making. In *Proceedings of KR2000*, pages 457–468, 2000.
35. J. Lang. Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence*, 42(1):37–71, 2004.
36. M. Lemaître, G. Verfaillie, and N. Bataille. Exploiting a common property resource under a fairness constraint: a case study. In *Proceedings of IJCAI-99*, pages 206–211, 1999.
37. R. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of EC’04*, 2004.
38. Agentlink technical forum group on multiagent resource allocation. <http://www.doc.ic.ac.uk/~ue/MARA/>, 2005.
39. P. Maynard-Zhang and D. Lehmann. Representing and aggregating conflicting beliefs. *Journal of Artificial Intelligence Research*, 19:155–203, 2003.
40. T. Meyer, A. Ghose, and S. Chopra. Social choice, merging, and elections. In *Proceedings of ECSQARU-01*, pages 466–477, 2001.
41. H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.
42. P. Perny and O. Spanjaard. On preference-based search in state space graphs. In *Proceedings of AAAI-02*, pages 751–756, 2002.
43. M. S. Pini, F. Rossi, K. Venable, and T. Walsh. Aggregating partially ordered preferences: possibility and impossibility results. In *Proceedings of TARK-05*, 2005.
44. F. Rossi, K. Venable, and T. Walsh. mCP nets: representing and reasoning with preferences of multiple agents. In *Proceedings of AAAI-04*, pages 729–734, 2004.
45. J. Rothe, H. Spakowski, and J. Vogel. Exact complexity of the winner for Young elections. *Theory of Computing Systems*, 36(4):375–386, 2003.
46. T. Sandholm. Contract types for satisficing task allocation: I theoretical results. In *Proc. AAAI Spring Symposium: Satisficing Models*, 1998.
47. M. Satterthwaite. Strategyproofness and Arrow’s conditions. *Journal of Economic Theory*, 10:187–217, 1975.

Nonlinear Deterministic Relationships in Bayesian Networks

Barry R. Cobb and Prakash P. Shenoy

University of Kansas School of Business,
1300 Sunnyside Ave., Summerfield Hall,
Lawrence, KS 66045-7585, USA
{brcobb, pshenoy}@ku.edu

Abstract. In a Bayesian network with continuous variables containing a variable(s) that is a conditionally deterministic function of its continuous parents, the joint density function does not exist. Conditional linear Gaussian distributions can handle such cases when the deterministic function is linear and the continuous variables have a multi-variate normal distribution. In this paper, operations required for performing inference with nonlinear conditionally deterministic variables are developed. We perform inference in networks with nonlinear deterministic variables and non-Gaussian continuous variables by using piecewise linear approximations to nonlinear functions and modeling probability distributions with mixtures of truncated exponentials (MTE) potentials.

1 Introduction

An important class of Bayesian networks with continuous variables are those that have conditionally deterministic variables (a variable that is a deterministic function of its parents). Conditional linear Gaussian (CLG) distributions (Lauritzen and Jensen 2001) can handle such cases when the deterministic function is linear and variables are normally distributed. In models with nonlinear deterministic relationships and non-Gaussian distributions, Monte Carlo methods may be required to obtain an approximate solution. General purpose solution algorithms, e.g., the Shenoy-Shafer architecture, have not been adapted to such models, primarily because the joint density for the variables in models with deterministic variables does not exist and these methods involve propagation of probability densities.

Approximate inference in Bayesian networks with continuous variables can be performed using mixtures of truncated exponentials (MTE) potentials (Moral et al. 2001). Cobb and Shenoy (2004) define operations which allow the distributions of linear deterministic variables to be determined when the continuous variables are modeled with MTE potentials. This allows MTE potentials to be used for inference in any continuous CLG model, as well as other models that have non-Gaussian and conditionally deterministic variables. This paper extends these methods to continuous Bayesian networks with nonlinear deterministic variables.

The remainder of this paper is organized as follows. Section 2 introduces notation and definitions used throughout the paper. Section 3 describes a method for approximating a nonlinear function with a piecewise linear function. Section 4 defines operations required for inference in Bayesian networks with conditionally deterministic variables. Section 5 contains examples of determining the distributions of nonlinear conditionally deterministic variables. Section 6 summarizes and states directions for future research. This paper is based on a longer, unpublished working paper (Cobb and Shenoy 2005).

2 Notation and Definitions

This section contains notation and definitions used throughout the paper.

2.1 Notation

Random variables will be denoted by capital letters, e.g., A, B, C . Sets of variables will be denoted by boldface capital letters, e.g., $\mathbf{X}$. All variables are assumed to take values in continuous state spaces. If $\mathbf{X}$ is a set of variables, $\mathbf{x}$ is a configuration of specific states of those variables. The continuous state space of $\mathbf{X}$ is denoted by $\Omega_{\mathbf{X}}$. In graphical representations, continuous nodes are represented by double-border ovals, whereas nodes that are deterministic functions of their parents are represented by triple-border ovals.

2.2 Mixtures of Truncated Exponentials

A mixture of truncated exponentials (MTE) (Moral et al. 2001) potential has the following definition.

MTE potential. Let $\mathbf{X} = (X_1, \dots, X_n)$ be an n -dimensional random variable. A function $\phi : \Omega_{\mathbf{X}} \mapsto \mathcal{R}^+$ is an MTE potential if one of the next two conditions holds:

1. The potential ϕ can be written as

$$\phi(\mathbf{x}) = a_0 + \sum_{i=1}^m a_i \exp \left\{ \sum_{j=1}^n b_i^{(j)} x_j \right\} \quad (1)$$

for all $\mathbf{x} \in \Omega_{\mathbf{X}}$, where $a_i, i = 0, \dots, m$ and $b_i^{(j)}, i = 1, \dots, m, j = 1, \dots, n$ are real numbers.

2. The domain of the variables, $\Omega_{\mathbf{X}}$, is partitioned into hypercubes $\{\Omega_{\mathbf{X}_1}, \dots, \Omega_{\mathbf{X}_k}\}$ such that ϕ is defined as

$$\phi(\mathbf{x}) = \phi_i(\mathbf{x}) \quad \text{if } \mathbf{x} \in \Omega_{\mathbf{X}_i}, \quad i = 1, \dots, k, \quad (2)$$

where each $\phi_i, i = 1, \dots, k$ can be written in the form of equation (1) (i.e. each ϕ_i is an MTE potential on $\Omega_{\mathbf{X}_i}$).

In the definition above, k is the number of *pieces* and m is the number of exponential *terms* in each piece of the MTE potential. We will refer to ϕ_i as the i -th piece of the MTE potential ϕ and $\Omega_{\mathbf{X}_i}$ as the portion of the domain of $\mathbf{X}$ approximated by ϕ_i . In this paper, all MTE potentials are equal to zero in unspecified regions.

2.3 Conditional Mass Functions (CMF)

When relationships between continuous variables are deterministic, the joint probability density function (PDF) does not exist. If Y is a deterministic relationship of variables in $\mathbf{X}$, i.e. $y = g(\mathbf{x})$, the conditional mass function (CMF) for $\{Y \mid \mathbf{x}\}$ is defined as

$$p_{Y|\mathbf{x}} = \mathbf{1}\{y = g(\mathbf{x})\} , \quad (3)$$

where $\mathbf{1}\{A\}$ is the indicator function of the event A , i.e. $\mathbf{1}\{A\}(B) = 1$ if $B = A$ and 0 otherwise.

3 Piecewise Linear Approximations to Nonlinear Functions

3.1 Dividing the Domain

Suppose that a random variable Y is a deterministic function of a single variable X , $Y = g(X)$. The function $Y = g(X)$ can be approximated by a piecewise linear function. Define a set of ordered points $x = (x_0, \dots, x_n)$ in the domain of X , with x_0 and x_n defined as the endpoints of the domain. A corresponding set of points $y = (y_0, \dots, y_n)$ is determined by calculating the value of the function $y = g(x)$ at each point x_i , $i = 0, \dots, n$. The piecewise linear function (with n pieces) approximating $Y = g(X)$ is the function $Y^{(n)} = g^{(n)}(X)$ defined as follows:

$$g^{(n)}(x) = \begin{cases} \left(y_0 - \frac{y_1 - y_0}{x_1 - x_0} \cdot x_0 \right) + \frac{y_1 - y_0}{x_1 - x_0} \cdot x & \text{if } x_0 \leq x < x_1 \\ \left(y_1 - \frac{y_2 - y_1}{x_2 - x_1} \cdot x_1 \right) + \frac{y_2 - y_1}{x_2 - x_1} \cdot x & \text{if } x_1 \leq x < x_2 \\ \vdots & \vdots \\ \left(y_{n-2} - \frac{y_{n-1} - y_{n-2}}{x_{n-1} - x_{n-2}} \cdot x_{n-2} \right) + \frac{y_{n-1} - y_{n-2}}{x_{n-1} - x_{n-2}} \cdot x & \text{if } x_{n-2} \leq x < x_{n-1} \\ \left(y_{n-1} - \frac{y_n - y_{n-1}}{x_n - x_{n-1}} \cdot x_{n-1} \right) + \frac{y_n - y_{n-1}}{x_n - x_{n-1}} \cdot x & \text{if } x_{n-1} \leq x \leq x_n \end{cases} . \quad (4)$$

Let $g_i^{(n)}(x)$ denote the i -th piece of the piecewise linear function in (4). We refer to $g^{(n)}$ as an n -point (piecewise linear) approximation of g . In this paper, all piecewise linear functions equal zero in unspecified regions. If a variable is a deterministic function of multiple variables, the definition in (4) can be extended by dividing the domain of the parent variables into hypercubes and creating an approximation of each function in each hypercube.

3.2 Algorithm for Splitting Regions

An initial piecewise approximation is defined (minimally) by splitting the domain of X at extreme points and points of change in concavity and convexity in the function $y = g(x)$, and at endpoints of pieces of the MTE potential for X . This initial set of bounds on the pieces of the approximation is defined as $x = (x_0^S, \dots, x_\ell^S)$. The absolute value of the difference between the approximation and the function will increase, then eventually decrease within each region of the approximation. This is due to the fact that the approximation in (4) always lies “inside” the actual function.

Additional pieces may be added to improve the fit between the nonlinear function and the piecewise approximation. Define an allowable error bound, ϵ , for the distance between the function $g(x)$ and its piecewise linear approximation. Define an interval η used to select the next point at which to test the distance between $g(x)$ and the piecewise approximation. The piecewise linear approximation in (4) is completely defined by the sets of points $x = (x_0, \dots, x_n)$ and $y = (y_0, \dots, y_n)$. The following procedure in pseudo-code determines the sets of points x and y which define the piecewise linear approximation when a deterministic variable has one parent.

```

INPUT :=  $x_0^S, \dots, x_\ell^S, g(x), \epsilon, \eta$ 
OUTPUT :  $x = (x_0, \dots, x_n), y = (y_0, \dots, y_n)$ 
INITIALIZATION
 $x \leftarrow \{(x_0^S, \dots, x_\ell^S)\}$  /* Endpoints, extrema, and inflection points in  $\Omega_{\mathbf{X}}$  */
 $y \leftarrow \{(g(x_0^S), \dots, g(x_\ell^S))\}$ 
 $i = 0$  /* Index for the intervals in the domain of  $X$  */

DO WHILE   $i < |x|$  /* Continue until all intervals are refined */
 $j = 1$  /* Index for number of test points in an interval */
 $a = 0$  /* Previous distance between  $g(x)$  and approximation */
 $b = 0$  /* Current distance between  $g(x)$  and approximation */
  FOR   $j = 1 : (x_{i+1} - x_i) / \eta$ 
     $b = g(x_i + (j - 1) \cdot \eta) -$ 
       $\left( \left( y_i - \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \cdot x_i \right) + \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \cdot (x_i + (j - 1) \cdot \eta) \right)$ 
    IF   $|b| \geq a$  /* Compare current and previous distance */
       $a = |b|$  /* Distance increased; test next point */
    ELSE
      BREAK /* Distance did not increase; break loop */
    END IF
  END FOR
  IF   $a > \epsilon$  /* Test max. distance versus allowable error bound */
     $x \leftarrow \text{Rank}(x \cup \{x_i + (j - 2) \cdot \eta\})$  /* Update  $x$  and re-order */
     $y \leftarrow \text{Rank}(y \cup \{g(x_i + (j - 2) \cdot \eta)\})$  /* Update  $y$  and re-order */
  END IF
   $i = i + 1$ 
END DO

```

The algorithm refines the piecewise approximation to the function $y = g(x)$ until the maximum distance between the function and the piecewise approximation is no larger than the specified error bound. A smaller error bound, ϵ , produces more pieces in the linear approximation and a closer fit in the theoretical and approximate density functions for the deterministic variable (see, e.g., Section 5.1 of (Cobb and Shenoy 2005)). A closer approximation using more pieces, however, requires greater computational expense in the inference process.

4 Operations with Linear Deterministic Variables

Consider a random variable Y which is a monotonic function, $Y = g(X)$, of a random variable X . The joint cumulative distribution function (CDF) for $\{X, Y\}$ is given by $F_{X,Y}(x, y) = F_X(g^{-1}(y))$ if $g(X)$ is monotonically increasing and $F_{X,Y}(x, y) = F_X(x) - F_X(g^{-1}(y))$ if $g(X)$ is monotonically decreasing. The CDF of Y is determined as $F_Y(y) = \lim_{x \rightarrow \infty} F_{X,Y}(x, y)$. Thus, $F_Y(y) = F_X(g^{-1}(y))$ if $g(X)$ is monotonically increasing and $F_Y(y) = 1 - F_X(g^{-1}(y))$ if $g(X)$ is monotonically decreasing. By differentiating the CDF of Y , the PDF of Y is obtained as

$$f_Y(y) = \frac{d}{dy} F_Y(y) = f_X(g^{-1}(y)) \left| \frac{d}{dy} (g^{-1}(y)) \right|, \quad (5)$$

when $Y = g(X)$ is monotonic. If Y is a conditionally deterministic linear function of X , i.e. $Y = g(x) = ax + b$, $a \neq 0$, the following operation can be used to determine the marginal PDF for Y :

$$f_Y(y) = \frac{1}{|a|} \cdot f_X\left(\frac{y-b}{a}\right). \quad (6)$$

The following definition extends the operation defined in (6) to accommodate piecewise linear functions. Suppose Y is a conditionally deterministic piecewise linear function of X , $Y = g(X)$, where $g_i(x) = a_i x + b_i$, with each $a_i \neq 0$, $i = 1, \dots, n$. Assume the PDF for X is an MTE potential ϕ with k pieces, where the j -th piece is denoted ϕ_j for $j = 1, \dots, k$. Let n_j denote the number of linear segments of g that intersect with the domain of ϕ_j and notice that $n = n_1 + \dots + n_j + \dots + n_k$. The CMF $p_{Y|x}$ represents the conditionally deterministic relationship of Y on X . The following definition will be used to determine the marginal PDF for Y (denoted $\chi = (\phi \otimes p_{Y|x})^{\downarrow Y}$):

$$\chi(y) = (\phi \otimes p_{Y|x})^{\downarrow Y}(y) \triangleq \begin{cases} 1/a_1 \cdot \phi_1((y-b_1)/a_1) & \text{if } y_0 \leq y < y_1 \\ 1/a_2 \cdot \phi_1((y-b_2)/a_2) & \text{if } y_1 \leq y < y_2 \\ \vdots & \vdots \\ 1/a_{n_1} \cdot \phi_1((y-b_{n_1})/a_{n_1}) & \text{if } y_{n_1-1} \leq y < y_{n_1} \\ \vdots & \vdots \\ 1/a_n \cdot \phi_k((y-b_n)/a_n) & \text{if } y_{n-1} \leq y < y_n \end{cases}, \quad (7)$$

with ϕ_j being the piece of ϕ whose domain is a superset of the domain of g_i . The normalization constants for each piece of the resulting MTE potential ensures that the CDF of the resulting MTE potential matches the CDF of the theoretical MTE potential at the endpoints of the domain of the resulting PDF. From Theorem 3 in (Cobb and Shenoy 2004), it follows that the class of MTE potentials is closed under the operation in (7); thus, the operation can be used for inference in Bayesian networks with deterministic variables. Note that the class of MTE potentials is not closed under the operation in (5), which is why we approximate nonlinear functions with piecewise linear functions.

5 Examples

The following examples illustrate determination of the distributions of random variables which are nonlinear deterministic functions of their parents, as well as inference in a simple Bayesian network with a nonlinear deterministic variable.

5.1 Example One

Suppose X is normally distributed with a mean of 0 and a standard deviation of 1, i.e. $X \sim N(0, 1^2)$, and Y is a conditionally deterministic function of X , $y = g(x) = x^3$. The distribution of X is modeled with an two-piece, three-term MTE potential as defined in (Cobb et al. 2003). The MTE potential is denoted by ϕ and its two pieces are denoted ϕ_1 and ϕ_2 , with $\Omega_{X_1} = \{x : -3 \leq x < 0\}$ and $\Omega_{X_2} = \{x : 0 \leq x \leq 3\}$.

Piecewise Approximation. Over the region $[-3, 3]$, the function $y = g(x) = x^3$ has an inflection point at $x = 0$, which is also an endpoint of a piece of the MTE approximation to the PDF of X . To initialize the algorithm in Sect. 3.2, we define $x = (x_0^S, x_1^S, x_2^S) = (-3, 0, 3)$ and $y = (y_0^S, y_1^S, y_2^S) = (-27, 0, 27)$. For this example, define $\epsilon = 1$ and $\eta = 0.06$ (which divides the domain of X into 100 equal intervals).

The procedure in Sect. 3.2 terminates after finding sets of points $x = (x_0, \dots, x_8)$ and $y = (y_0, \dots, y_8)$ as follows:

$$\begin{aligned} x &= (-3.00, -2.40, -1.74, -1.02, 0.00, 1.02, 1.74, 2.40, 3.00) , \\ y &= (-27.000, -13.824, -5.268, -1.061, 0.000, 1.061, 5.268, 13.824, 27.000) . \end{aligned}$$

The function representing the eight-point linear approximation is defined as

$$g^{(8)}(x) = \begin{cases} 21.960x + 38.880 & \text{if } -3.00 \leq x < -2.40 \\ 12.964x + 17.289 & \text{if } -2.40 \leq x < -1.74 \\ 5.843x + 4.898 & \text{if } -1.74 \leq x < -1.02 \\ 1.040x & \text{if } -1.02 \leq x < 0 \\ 1.040x & \text{if } 0 \leq x < 1.02 \\ 5.843x - 4.898 & \text{if } 1.02 \leq x < 1.74 \\ 12.964x - 17.289 & \text{if } 1.74 \leq x < 2.40 \\ 21.960x - 38.880 & \text{if } 2.40 \leq x \leq 3.00 . \end{cases} \quad (8)$$

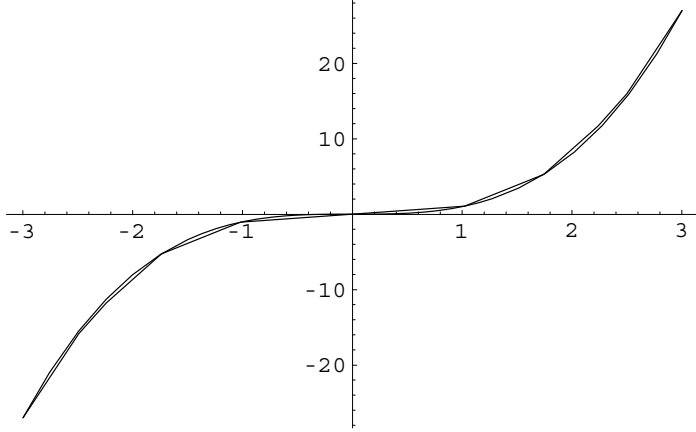


Fig. 1. The piecewise linear approximation $g^{(8)}(x)$ overlayed on the function $y = g(x)$

The piecewise linear approximation $g^{(8)}(x)$ is shown in Fig. 1, overlayed on the function $y = g(x)$.

The conditional distribution for Y is represented by a CMF as follows:

$$\psi^{(8)}(x, y) = p_{Y|x}(y) = \mathbf{1}\{y = g^{(8)}(x)\} \ .$$

Determining the Distribution of Y . The marginal distribution for Y is determined by calculating $\chi^{(8)} = (\phi \otimes \psi^{(8)})^{\downarrow Y}$. The MTE potential for Y is

$$\chi^{(8)}(y) = \begin{cases} (1/21.960) \cdot \phi^{(1)}(0.0455y - 1.7705) & \text{if } -27.000 \leq y < -13.824 \\ (1/12.964) \cdot \phi_1(0.0771y - 1.3336) & \text{if } -13.824 \leq y < -5.268 \\ (1/5.843) \cdot \phi_1(0.1712y - 0.8384) & \text{if } -5.268 \leq y < -1.061 \\ (1/1.040) \cdot \phi_1(0.9612y) & \text{if } -1.061 \leq y \leq 0.000 \\ (1/1.040) \cdot \phi_2(0.9612y) & \text{if } 0.000 \leq y < 1.061 \\ (1/5.843) \cdot \phi_2(0.1712y + 0.8384) & \text{if } 1.061 \leq y < 5.628 \\ (1/12.964) \cdot \phi_2(0.0771y + 1.3336) & \text{if } 5.628 \leq y < 13.824 \\ (1/21.960) \cdot \phi_2(0.0455y + 1.7705) & \text{if } 13.824 \leq y \leq 27.000 \ . \end{cases}$$

The CDF associated with the eight-piece MTE approximation is shown in Fig. 2, overlayed on the CDF associated with the PDF from the transformation

$$f_Y(y) = f_X(g_1^{-1}(y)) \frac{d}{dy}(g_1^{-1}(y)) \ . \quad (9)$$

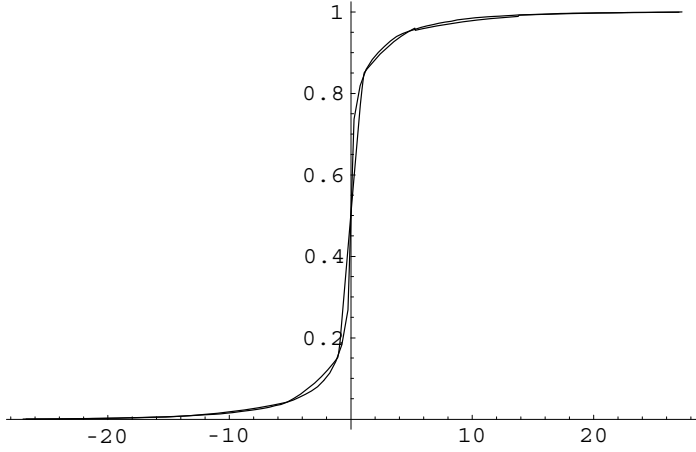


Fig. 2. CDF for the eight-piece MTE approximation to the distribution for Y overlaid on the CDF created using the transformation in (9)

5.2 Example Two

The Bayesian network in this example (see Fig. 3) contains one variable (X) with a non-Gaussian potential, one variable (Z) with a Gaussian potential, and one variable (Y) which is a deterministic linear function of its parent. The probability distribution for X is a beta distribution, i.e. $\mathcal{L}(X) \sim \text{Beta}(\alpha = 2.7, \beta = 1.3)$. The PDF for X is approximated (using the methods described in (Cobb et al. 2003))



Fig. 3. The Bayesian network for Example Two

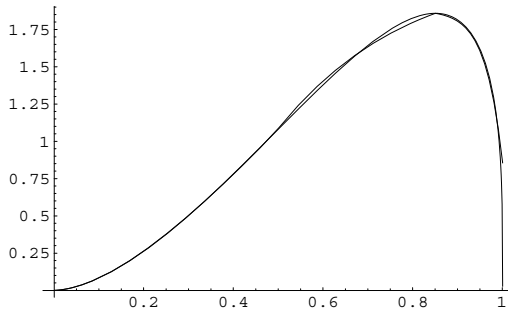


Fig. 4. The MTE potential for X overlaid on the actual $\text{Beta}(2.7, 1.3)$ distribution

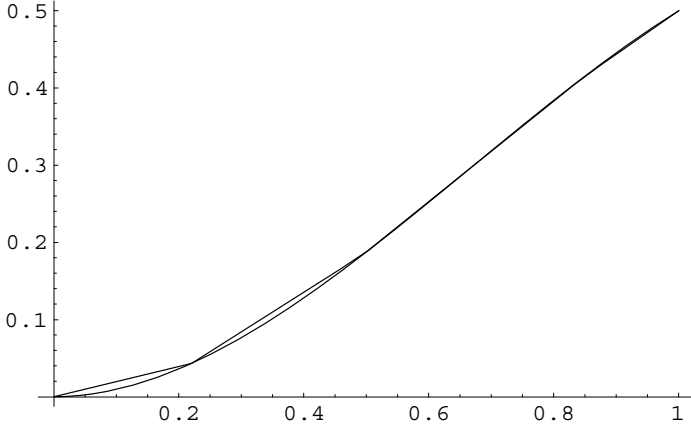


Fig. 5. The piecewise linear approximation $g^{(5)}(x)$ overlayed on the function $g(x)$ in Example Two

by a three-piece, two-term MTE potential. The MTE potential ϕ for X is shown graphically in Figure 4, overlayed on the actual $Beta(2.7, 1.3)$ distribution.

The variable Y is a conditionally deterministic function of X , $y = g(x) = -0.5x^3 + x^2$. The five-point linear approximation is characterized by points $x = (x_0, \dots, x_5) = (0, 0.220, 0.493, 0.667, 0.850, 1)$ and $y = (y_0, \dots, y_5) = (0, 0.043, 0.183, 0.296, 0.415, 0.500)$. The points x_0, x_2, x_3 , and x_5 are defined according to the endpoints of the pieces of ϕ . The point x_4 is an inflection point in the function $g(x)$ and the point $x_1 = 0.220$ is found by the algorithm in Sect. 3.2 with $\epsilon = 0.015$ and $\eta = 0.01$. The function representing the five-piece linear approximation (denoted as $g^{(5)}$) is shown graphically in Fig. 5 overlayed on $g(x)$.

The conditional distribution for Y given X is represented by a CMF as follows:

$$\psi^{(5)}(x, y) = p_{Y|X}(y) = \mathbf{1}\{y = g^{(5)}(x)\} .$$

The probability distribution for Z is defined as $\mathcal{L}(Z | y) \sim N(2y + 1, 1)$ and is approximated by χ , which is a two-piece, three-term MTE approximation to the normal distribution (Cobb et al. 2003).

5.3 Computing Messages

The join tree for the example problem is shown in Fig. 6.

The messages required to calculate posterior marginals for each variable in the network without evidence are as follows:

- 1) ϕ from $\{X\}$ to $\{X, Y\}$
- 2) $(\phi \otimes \psi^{(5)})^{\downarrow Y}$ from $\{X, Y\}$ to $\{Y\}$ and $\{Y\}$ to $\{Y, Z\}$
- 3) $((\phi \otimes \psi^{(5)})^{\downarrow Y} \otimes \chi)^{\downarrow Z}$ from $\{Y, Z\}$ to $\{Z\}$



Fig. 6. The join tree for the example problem

5.4 Posterior Marginals

The posterior marginal distribution for Y is the message sent from $\{X, Y\}$ to $\{Y\}$ and is calculated using the operation in (7). The expected value and variance of this distribution are calculated as 0.3042 and 0.0159, respectively. The posterior marginal distribution for Z is the message sent from $\{Y, Z\}$ to $\{Z\}$ and is calculated by point-wise multiplication of MTE functions, followed by marginalization (see operations defined in (Moral et al. 2001)). The expected value and variance of this distribution are calculated as 1.6084 and 1.0455, respectively.

5.5 Entering Evidence

Suppose we observe evidence that $Z = 0$ and let e_Z denote this evidence. Define $\varphi = (\phi \otimes \psi^{(5)})^{\downarrow Y}$ and $\psi^{(5)}(x, y) = \mathbf{1}\{x = g^{(5)-1}(y)\}$ as the potentials resulting from the reversal of the arc between X and Y (Cobb and Shenoy 2004). The evidence e_Z is passed from $\{Z\}$ to $\{Y, Z\}$ in the join tree, where the existing potential is restricted to $\chi(y, 0)$. This likelihood potential is passed from $\{Y, Z\}$ to $\{Y\}$ in the join tree.

Denote the unnormalized posterior marginal distribution for B as $\xi'(y) = \varphi(y) \cdot \chi(y, 0)$. The normalization constant is calculated as $K = \int_y (\varphi(y) \cdot \chi(y, 0)) dy = 0.0670$. Thus, the normalized marginal distribution for Y is found as $\xi(y) =$

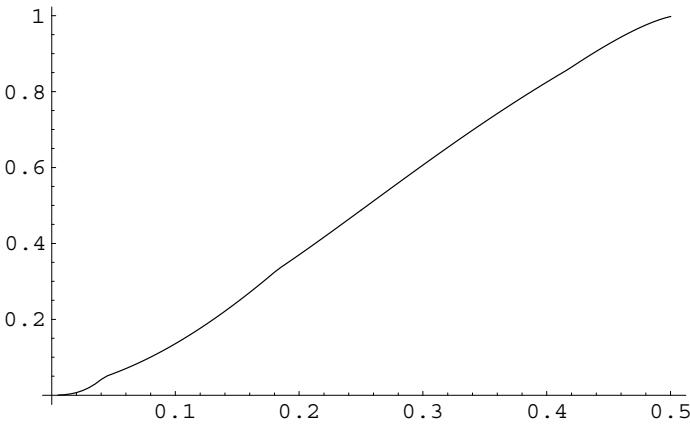


Fig. 7. The posterior marginal CDF for Y considering the evidence $Z = 0$

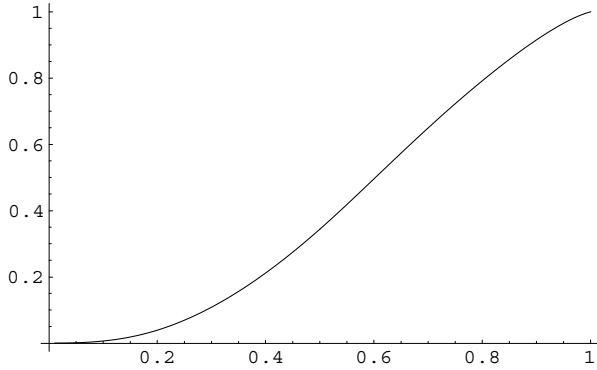


Fig. 8. The posterior marginal CDF for X considering the evidence ($Z = 0$)

$K^{-1} \cdot \xi'(y)$. The expected value and variance of this distribution (whose CDF is displayed in Fig. 7) are calculated as 0.2560 and 0.0167, respectively.

Using the operation in (7), we determine the posterior marginal distribution for X as $\vartheta = (\xi \otimes \psi^{(5)}) \downarrow^X$. The expected value and variance of this distribution are calculated as 0.5942 and 0.0480, respectively. The posterior marginal CDF for X considering the evidence is shown graphically in Figure 8.

6 Summary and Conclusions

This paper has described operations required for inference in Bayesian networks containing variables that are nonlinear deterministic functions of their continuous parents. Since the joint PDF for a network with deterministic variables does not exist, the operations required are based on the method of convolutions from probability theory. By estimating nonlinear functions with piecewise linear approximations, we ensure the class of MTE potentials are closed under these operations. Bayesian networks in this paper contain only continuous variables. In future work, we plan to design a general inference algorithm for Bayesian networks that contain a mixture of discrete and continuous variables, with some continuous variables defined as deterministic functions of their continuous parents.

References

- Cobb, B.R. and P.P. Shenoy: Inference in hybrid Bayesian networks with deterministic variables. In P. Lucas (ed.): *Proceedings of the Second European Workshop on Probabilistic Graphical Models (PGM-04)* (2004) 57–64, Leiden, Netherlands.
- Cobb, B.R. and P.P. Shenoy: Modeling nonlinear deterministic relationships in Bayesian networks. School of Business Working Paper No. 310, University of Kansas, Lawrence, Kansas (2005). Available for download at: <http://www.people.ku.edu/~brcobb/WP310.pdf>

- Cobb, B.R., Shenoy, P.P. and R. Rumí: Approximating probability density functions in hybrid Bayesian networks with mixtures of truncated exponentials. Working Paper No. 303, School of Business, University of Kansas, Lawrence, Kansas (2003). Available for download at: <http://www.people.ku.edu/~brcobb/WP303.pdf>
- Kullback, S. and R.A. Leibler: On information and sufficiency. *Annals of Mathematical Statistics* **22** (1951) 79–86.
- Larsen, R.J. and M.L. Marx: *An Introduction to Mathematical Statistics and its Applications* (2001) Prentice Hall, Upper Saddle River, N.J.
- S.L. Lauritzen and F. Jensen: Stable local computation with conditional Gaussian distributions. *Statistics and Computing* **11** (2001) 191–203.
- Moral, S., Rumí, R. and A. Salmerón: Mixtures of truncated exponentials in hybrid Bayesian networks. In P. Besnard and S. Benferhart (eds.): *Symbolic and Quantitative Approaches to Reasoning under Uncertainty*, Lecture Notes in Artificial Intelligence **2143** (2001) 156–167, Springer-Verlag, Heidelberg.

Penniless Propagation with Mixtures of Truncated Exponentials^{*}

Rafael Rumí and Antonio Salmerón

Dept. Estadística y Matemática Aplicada,
Universidad de Almería, 04120 Almería, Spain
{rrumi, Antonio.Salmeron}@ual.es

Abstract. Mixtures of truncated exponential (MTE) networks are a powerful alternative to discretisation when working with hybrid Bayesian networks. One of the features of the MTE model is that standard propagation algorithm can be used. In this paper we propose an approximate propagation algorithm for MTE networks which is based on the Penniless propagation method already known for discrete variables. The performance of the proposed method is analysed in a series of experiments with random networks.

1 Introduction

A Bayesian network is an efficient representation of a joint probability distribution over a set of variables, where the network structure encodes the independence relations among the variables. Bayesian networks are commonly used to make inferences about the probability distribution on some variables of interest, given that the values of some other variables are known. This task is usually called *probabilistic inference* or *probability propagation*.

Much attention has been paid to probability propagation in networks where the variables are discrete with a finite number of possible values. Several exact methods have been proposed in the literature for this task [8, 13, 14, 20], all of them based on *local computation*. Local computation means to calculate the marginals without actually computing the joint distribution, and is described in terms of a message passing scheme over a structure called *join tree*. Also, approximate methods have been developed with the aim of dealing with complex networks [2, 3, 4, 7, 18, 19].

In mixed Bayesian networks, where both discrete and continuous variables appear simultaneously, it is possible to apply local computation schemes similar to those for discrete variables. However, the correctness of exact inference depends on the model.

This problem was deeply studied before, but the only general solution is the discretisation of the continuous variables [5, 11] which are then treated as if they

^{*} This work has been supported by the Spanish Ministry of Science and Technology, project Elvira II (TIC2001-2973-C05-02) and by FEDER funds.

were discrete, and therefore the results obtained are approximate. Exact propagation can be carried out over mixed networks when the model is a conditional Gaussian distribution [12, 17], but in this case, discrete variables are not allowed to have continuous parents. This restriction was overcome in [10] using a mixture of exponentials to represent the distribution of discrete nodes with continuous parents, but the price to pay is that propagation cannot be carried out using exact algorithms: Monte Carlo methods are used instead.

The Mixture of Truncated Exponentials (MTE) model [15] provide the advantages of the traditional methods and the added feature that discrete variables with continuous parents are allowed. Exact standard propagation algorithms can be performed over them [6], as well as approximate methods. In this work, we introduce an approximate propagation algorithm for MTEs based on the idea of Penniless propagation [2], which is actually derived from the Shenoy-Shafer [20] method.

This paper continues with a description of the MTE model in section 2. The representation based on mixed tress can be found in section 3. Section 4 contains the application of Shenoy-Shafer algorithm to MTE networks, while in section 5 the Penniless algorithm is presented, and is illustrated with some experiments reported in section 6. The paper ends with conclusions in section 7.

2 The MTE Model

Throughout this paper, random variables will be denoted by capital letters, and their values by lowercase letters. In the multi-dimensional case, boldfaced characters will be used. The domain of the variable $\mathbf{X}$ is denoted by $\Omega_{\mathbf{X}}$. The MTE model is defined by its corresponding potential and density as follows [15]:

Definition 1. (MTE potential) *Let $\mathbf{X}$ be a mixed n -dimensional random vector. Let $\mathbf{Y} = (Y_1, \dots, Y_d)$ and $\mathbf{Z} = (Z_1, \dots, Z_c)$ be the discrete and continuous parts of $\mathbf{X}$, respectively, with $c + d = n$. We say that a function $f : \Omega_{\mathbf{X}} \mapsto \mathbb{R}_0^+$ is a Mixture of Truncated Exponentials potential (MTE potential) if one of the next conditions holds:*

- i. $\mathbf{Y} = \emptyset$ and f can be written as

$$f(\mathbf{x}) = f(\mathbf{z}) = a_0 + \sum_{i=1}^m a_i \exp \left\{ \sum_{j=1}^c b_i^{(j)} z_j \right\} \quad (1)$$

for all $\mathbf{z} \in \Omega_{\mathbf{Z}}$, where a_i , $i = 0, \dots, m$ and $b_i^{(j)}$, $i = 1, \dots, m$, $j = 1, \dots, c$ are real numbers.

- ii. $\mathbf{Y} = \emptyset$ and there is a partition $D_1, \dots, D_k$ of $\Omega_{\mathbf{Z}}$ into hypercubes such that f is defined as

$$f(\mathbf{x}) = f(\mathbf{z}) = f_i(\mathbf{z}) \quad \text{if } \mathbf{z} \in D_i,$$

where each f_i , $i = 1, \dots, k$ can be written in the form of (1).

- iii. $\mathbf{Y} \neq \emptyset$ and for each fixed value $\mathbf{y} \in \Omega_{\mathbf{Y}}$, $f_{\mathbf{y}}(\mathbf{z}) = f(\mathbf{y}, \mathbf{z})$ can be defined as in ii.

Definition 2. (MTE density) *An MTE potential f is an MTE density if*

$$\sum_{\mathbf{y} \in \Omega_{\mathbf{Y}}} \int_{\Omega_{\mathbf{Z}}} f(\mathbf{y}, \mathbf{z}) d\mathbf{z} = 1 .$$

In a Bayesian network, we find two types of densities:

1. For each variable X which is a root of the network, a density $f(x)$ is given.
2. For each variable X with parents $\mathbf{Y}$, a conditional density $f(x|\mathbf{y})$ is given.

A *conditional MTE density* $f(x|\mathbf{y})$ is an MTE potential $f(x, \mathbf{y})$ such that fixing $\mathbf{y}$ to each of its possible values, the resulting function is a density for X .

3 Mixed Trees

In [15] a data structure was proposed to represent MTE potentials: The so-called *mixed probability trees* or mixed trees for short. The formal definition is as follows:

Definition 3. (Mixed tree) *We say that a tree $\mathcal{T}$ is a mixed tree if it meets the following conditions:*

- i. *Every internal node represents a random variable (either discrete or continuous).*
- ii. *Every arc outgoing from a continuous variable Z is labeled with an interval of values of Z , so that the domain of Z is the union of the intervals corresponding to the arcs Z -outgoing.*
- iii. *Every discrete variable has a number of outgoing arcs equal to its number of states.*
- iv. *Each leaf node contains an MTE potential defined on variables in the path from the root to that leaf.*

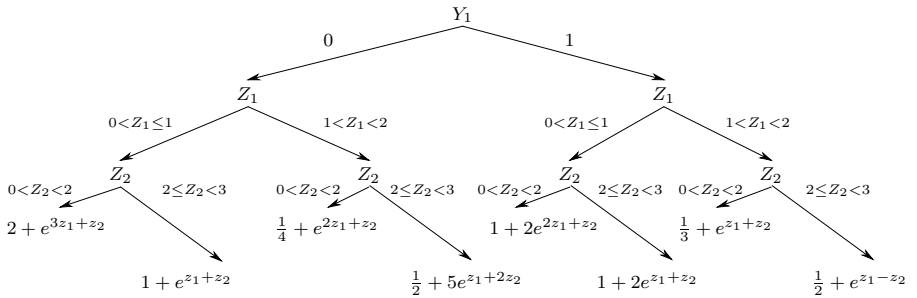


Fig. 1. A mixed probability tree representing an MTE potential

Mixed trees can represent MTE potentials defined by parts. Each entire branch in the tree determines one sub-region of the space where the potential is defined, and the function stored in the leaf of a branch is the definition of the potential in the corresponding sub-region. An example of a mixed tree is shown in Fig. 1.

The operations required for probability propagation in Bayesian networks (restriction, marginalisation and combination) can be carried out by means of algorithms very similar to those described, for instance in [11, 18].

4 Shenoy - Shafer Propagation Algorithm with MTEs

In [15] it was shown that MTE networks can be solved using Shenoy-Shafer algorithm [20]. This algorithm requires an adequate order of elimination of the variables to get the join tree, since different orders may result in join trees of distinct sizes, and the efficiency of probability propagation depends on the complexity of the join tree. This problem has been widely studied for discrete networks [1, 9], but not yet for MTE models. Here we propose a one-step lookahead strategy to determine the elimination order. We will choose the next variable to eliminate according to the size of the potential associated with the resulting clique.

Definition 4. (Size of an MTE potential) *The size of an MTE potential is defined as the number of exponentials terms, including the independent term, out of which the MTE potential is composed.*

Example 1. The potential represented in Fig. 1 has size equal to 16, because it has 8 leaves, and in each one an independent term, and one exponential term, so $8 \times (1 + 1) = 16$.

The decision on which variable to select next time, requires the knowledge about the size of the clique that would result from combining all the potentials defined for the variable. In the case of some MTE networks, it is possible to estimate it beforehand. If the MTE potentials are such that for each of them, the number of exponential terms in each leaf is the same, and the number of splits of the domain of the continuous variables also coincides, and only one variable appears in the MTE functions stored in the leaves of the mixed tree (the rest of the variables are used just to split the domain), as in [15] and [16], then there is an upper bound on the potential size:

Proposition 1. *Let $\mathcal{T}_1, \dots, \mathcal{T}_h$ be h mixed probability trees, $\mathbf{Y}_i, \mathbf{Z}_i$ the discrete and continuous variables of each of them, and n_i the number of intervals into which the domain of the continuous variables of $\mathcal{T}_i$ is split. Let Ω_{Y_i} be the set of possible values of the discrete variable Y_i . The size of the tree $\mathcal{T} = \mathcal{T}_1 \times \mathcal{T}_2 \times \dots \times \mathcal{T}_h$ is lower than*

$$\left(\prod_{Y_i \in \bigcup_{i=1}^h \mathbf{Y}_i} |\Omega_{Y_i}| \right) \times \left(\prod_{j=1}^h n_j^{k_j} \right) \times \left(\prod_{j=1}^h t_j \right) ,$$

where t_j is the number of exponential terms in each leaf of $\mathcal{T}_j$, and k_j is the number of continuous variables in $\mathcal{T}_j$.

5 Penniless Propagation with MTEs

Using the algorithm cited above, it is usual in large discrete networks that the size of the potentials involved grow so much that the propagation becomes infeasible. In the case of MTE networks, the complexity is higher, since the potentials are larger in general.

To overcome this problem in the discrete case, the Penniless propagation algorithm was proposed [2]. This propagation method is based on the Shenoy-Shafer method, but modifying it so that the results are approximations of the actual marginal distributions in exchange of lower time and space requirements.

The Shenoy-Shafer algorithm operates over the join tree built from the original network using a message passing scheme between adjacent nodes. Between every pair of adjacent nodes C_i and C_j there is a mailbox for the messages from C_i to C_j and another one for the messages from C_j to C_i . Sending a message from C_i to C_j can be considered as transferring the information contained in C_i that is relevant to C_j . Messages stored in both mailboxes are potentials defined for $C_i \cap C_j$. Initially these mailboxes are empty and once a message is stored it is full. A node C_i is allowed to send a message to its neighbor C_j if and only if every mailbox for messages arriving to C_i is full except the one from C_j to C_i .

The propagation is organised in two steps: in the first one messages are sent from leaves to a previously selected root node, and in the second one the messages are sent from the root to the leaves.

The message from C_i to C_j is recursively defined as follows:

$$\phi_{C_i \rightarrow C_j} = \left\{ \phi_{C_i} \cdot \left(\prod_{C_k \in ne(C_i) \setminus \{C_j\}} \phi_{C_k \rightarrow C_i} \right) \right\}^{\downarrow C_i \cap C_j}, \quad (2)$$

where ϕ_{C_i} is the original potential defined over C_i , $ne(C_i)$ is the set of adjacent nodes of C_i and superscript $\downarrow C_i \cap C_j$ indicates the marginal over $C_i \cap C_j$.

The main feature of the Penniless algorithm is that the messages sent are approximated, decreasing their size. This approximation [2, 4] is performed after every combination and marginalisation in (2), and also when obtaining the posterior marginals. It consists of reducing the size of the probability trees used to represent the potentials by *pruning* some of their branches (namely, those that are more similar). The same approach can be taken within the MTE framework, with the difference that this time, instead of probability trees, the potentials are represented as mixed trees. Let us consider now how the pruning operation can be carried out over mixed trees.

5.1 Pruning a Mixed Tree

The size of an MTE potential (and consequently the size of its corresponding mixed tree) is determined by the number of leaves it has and the number of

exponential terms in each leaf. Thus, a way of decreasing the size of the MTE potentials is decreasing each one of these two quantities.

But every pruning has an error associated with it. This error will be measured in terms of divergence between the mixed trees before and after the pruning.

Definition 5. (Divergence between mixed trees) *Let $\mathcal{T}$ be a mixed tree representing an MTE potential ϕ defined for $\mathbf{X} = (\mathbf{Y}, \mathbf{Z})$. Let $\mathcal{T}^*$ be a subtree of $\mathcal{T}$ with root $Z \in \mathbf{Z}$ where every child of Z is an MTE potential. Let ϕ_1 be the potential represented by $\mathcal{T}^*$. Let $\mathcal{T}_P^*$ be a tree obtained from $\mathcal{T}^*$ replacing ϕ_1 by the potential ϕ_2 for which it holds that $\int_{\Omega_{\mathbf{Z}}} \phi_1 d\mathbf{z} = \int_{\Omega_{\mathbf{Z}}} \phi_2 d\mathbf{z}$. The divergence between $\mathcal{T}^*$ and $\mathcal{T}_P^*$ is defined as*

$$D(\mathcal{T}^*, \mathcal{T}_P^*) = E_{\phi_1^*}[(\phi_1^* - \phi_2^*)^2] = \int_{\Omega_{\mathbf{Z}}} \frac{\phi_1(\mathbf{z})}{\Delta} \left(\frac{\phi_1(\mathbf{z})}{\Delta} - \frac{\phi_2(\mathbf{z})}{\Delta} \right)^2 d\mathbf{z},$$

where ϕ_i^* is the normalisation of ϕ_i and Δ is the total weight of ϕ :

$$\Delta = \sum_{\mathbf{Y}} \int_{\Omega_{\mathbf{Z}}} \phi(\mathbf{y}, \mathbf{z}) d\mathbf{z}.$$

We have considered three different kinds of pruning that are described in the next subsections.

Removing Exponential Terms. In each leaf of the mixed tree, the exponential terms that have little impact on the density function could be removed and the resulting potential would be rather similar to the original one.

Let $f(\mathbf{z}) = k + \sum_{i=1}^n a_i e^{b_i \mathbf{z}}$ be the potential stored in a leaf. The goal is to detect those exponential terms $a_i e^{b_i \mathbf{z}}$ having little influence on the entire density. We define the *weight* of each term as:

$$p_i = \int_{\Omega_{\mathbf{Z}}} a_i e^{b_i \mathbf{z}} d\mathbf{z}.$$

We think that two sensible criteria to remove terms in an MTE potential are the following:

1. A threshold α is established and the terms whose absolute weight, $|p_i|$, is lower than α are removed.
2. A maximum potential size is fixed and then the terms with lower absolute weight are removed until the size of the potential lies below the established maximum.

Once a term has been removed, the resulting potential is updated as follows :

- The maximum value of the term is computed , $m = \max_{\mathbf{z} \in \mathbf{Z}} \{a_i e^{b_i \mathbf{z}}\}$, and added to the independent term, $k^* = k + m$.
- The potential is normalised in order to make it integrate up to the total weight of the original potential.

The reason why the maximum of the potential is added to the independent term is to avoid negative points in the resulting potential.

Joining MTE Functions. Let $\mathcal{T}$ be a mixed tree whose root node, X , is continuous, and its children are MTE functions. The domain of X is divided into intervals, I_j , and for each of those intervals, a potential $f_j(\mathbf{z}) = k^j + \sum_{i=1}^n a_i^j e^{b_i^j \mathbf{z}}$ is defined. It may be that these potentials are very similar in the different intervals, I_j , and therefore some of them could be joined with little loss of information. Two intervals I_{j_1} and I_{j_2} are joined by replacing the potentials $f_{j_1}(\mathbf{z})$ and $f_{j_2}(\mathbf{z})$ by another potential $f(\mathbf{z})$, defined for over $I_{j_1} \cup I_{j_2}$.

We propose to compute $f(\mathbf{z})$ as follows. Let

$$p_{j_1} = \int_{\Omega_{\mathbf{z}}} f_{j_1}(\mathbf{z}) d\mathbf{z} \quad \text{and} \quad p_{j_2} = \int_{\Omega_{\mathbf{z}}} f_{j_2}(\mathbf{z}) d\mathbf{z}$$

be the weights of $f_{j_1}(\mathbf{z})$ and $f_{j_2}(\mathbf{z})$ respectively, the replacing function is proportional to

$$f(\mathbf{z}) = \frac{p_{j_1} f_{j_1}(\mathbf{z}) + p_{j_2} f_{j_2}(\mathbf{z})}{p_{j_1} + p_{j_2}}.$$

Since both functions must integrate up to the same quantity over $I_{j_1} \cup I_{j_2}$, a constant K must be found such that

$$\int_{\Omega_{\mathbf{z}}} K f(\mathbf{z}) d\mathbf{z} = p_1 + p_2,$$

which implies that $K = \frac{p_1 + p_2}{\int_{\Omega_{\mathbf{z}}} f(\mathbf{z}) d\mathbf{z}}$.

Let $\mathcal{T}$ be the tree corresponding to the original potential, and $\mathcal{T}_{\mathcal{P}}$ the one resulting from replacing $f_{j_1}(\mathbf{z})$ and $f_{j_2}(\mathbf{z})$ by $f(\mathbf{z})$, then the error $\mathcal{D}(\mathcal{T}, \mathcal{T}_{\mathcal{P}})$ is computed, and if it is lower than a fixed parameter, we replace $\mathcal{T}$ by $\mathcal{T}_{\mathcal{P}}$.

Discrete Pruning. In this particular MTE networks, the values of the discrete variables are used only when splitting the domain of the potential, so marginal potentials defined for discrete variables are equivalent to probability tables.

If Y is a discrete variable in a mixed tree node, and its children are MTE functions, then the tree can be pruned as defined in [18] (due to space limitations we do not provide the details here).

6 Experimental Evaluation of the Algorithm

In order to test the performance of the Penniless algorithm over MTE networks, we have carried out a simulation study, in which the algorithm is run over some MTE networks, using different levels of pruning.

Three different artificial networks have been created following these restrictions:

1. Given a variable, its number of parents is a Poisson distribution with mean 0.8 and its parents are chosen at random.

Table 1. Networks studied

Net	Number of nodes	Number of discrete nodes
Net1	42	3
Net2	77	8
Net3	86	11

Table 2. Probability distribution for the number of states of the discrete variables, the number of splits of the domain of continuous variables and the number of exponential terms of MTE functions

No. states	2	3	4	No. splits	1	2	3	No. exp. terms	0	1	2
Probability	1/3	1/3	1/3	Probability	0.2	0.4	0.4	Probability	0.05	0.75	0.20

2. Discrete variables:

- (a) Its number of states is simulated from the distribution showed in Table 2.
- (b) The probability value of each state is simulated from an Exponential distribution with mean 0.5.

3. Continuous variables:

- (a) The number of splits of the variable in a potential is simulated from the distribution showed in Table 2.
- (b) Every MTE potential has an independent term which is simulated from an Exponential distribution with mean 0.01 and a number of exponential terms determined by the distribution showed in Table 2.
- (c) In every exponential term, $a \exp\{bx\}$ the coefficient a is a real number following an Exponential distribution with mean 1, and the exponent b is a real number determined by a standard Normal distribution (mean 0 and standard deviation 1).

After simulating the parameters of the potentials, they are normalised in order to guarantee that the potentials are density functions.

For each network, the 30% of its variables are observed at random. The corresponding evidence is inserted in the network by restricting the potentials to the observed values.

The Penniless propagation is carried out over each of these networks, with different parameters of pruning. For discrete pruning and for joining intervals, some parameters are chosen, and the exponential terms in every potential are removed until there are only two terms remaining (i.e. the maximum number of terms per potential leaf in a mixed tree is set to 2).

Since the MTE framework is mainly an alternative to discretisation, the results of the propagation are compared with the results of applying Shenoy-Shafer propagation to the discretisation obtained by replacing every MTE function

$$f(\mathbf{z}) = k + \sum_{i=1}^n a_i e^{b_i \mathbf{z}} \quad \text{by a constant function } f^*(\mathbf{z}) = k^* \quad \text{so that}$$

$$\int_{\Omega_{\mathbf{z}}} f(\mathbf{z}) d\mathbf{z} = \int_{\Omega_{\mathbf{z}}} f^*(\mathbf{z}) d\mathbf{z} .$$

After each propagation, the following quantities are computed:

1. The maximum size of the potential needed to compute the marginal distribution. It is achieved after combining all the messages sent to the clique that contains the variable in the join tree.
2. The error attached to it, according to definition 5.

For each network, the mean of these quantities is computed for all the variables that do not appear in the evidence. The summary of the obtained results are shown in Figs. 2 to 4, where the notation for the pruning parameters is shown in Table 3. The "Join parameter" is the maximum error allowed for joining two intervals, while the "Discrete parameter" indicates that discrete distributions that differ less than the value of the parameter with respect to a uniform distribution, in terms of entropy, are pruned. The foundations of this discrete parameter are explained in [18].

The results of the experiments show that the use of MTEs instead of discretisations provides more accurate results. It is not surprising, since the discretisation is just a particular case of the MTE framework (a discretised density is an

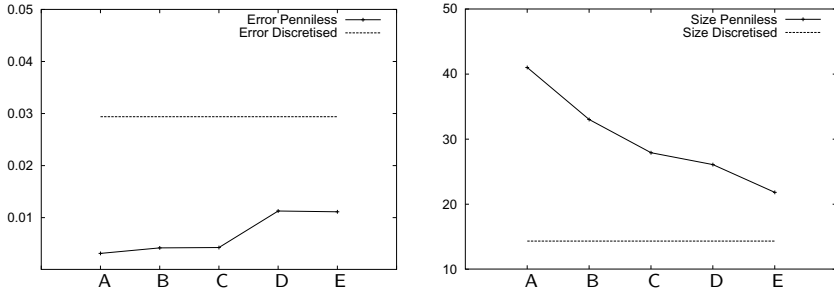


Fig. 2. Errors and sizes for Net1

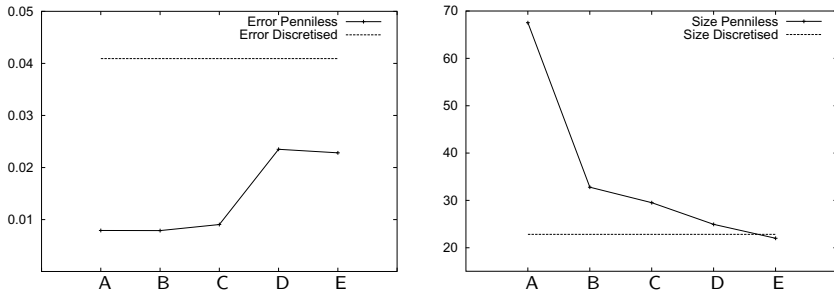
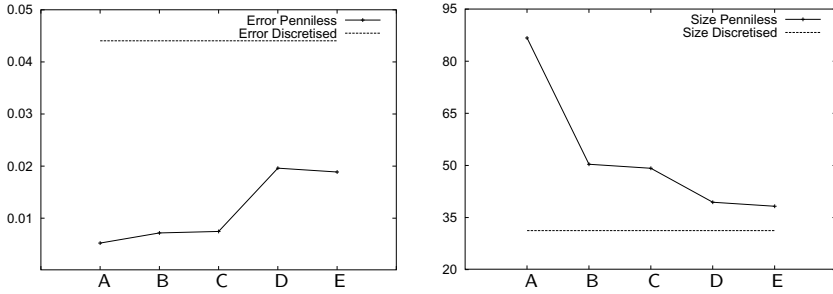


Fig. 3. Errors and sizes for Net2

**Fig. 4.** Errors and sizes for Net3**Table 3.** Different pruning parameters evaluated

Prune	Join parameter	Discrete parameter
A	0	0
B	0.005	0
C	0.005	0.01
D	0.05	0
E	0.05	0.01

MTE density with one independent term and zero exponential terms). However, it is important to point out that the increase in space required by the MTEs is significantly lower than the gain in accuracy, which means that the tradeoff space/accuracy, according to the evidence provided by the experiments reported here, is favourable to the MTE.

7 Conclusions

Some propagation methods have been successfully applied to MTE networks, as for example Shenoy-Shafer propagation [6], but so far they were not able to overcome the problem of the exponential increase of the sizes of the potentials involved in the propagation, specially when evidence is entered. In this paper we have presented a method to apply Penniless propagation to MTE networks, so that the sizes of the potentials are reduced because of the pruning operation.

The performance of the method has been tested on three artificial networks. The results of the experiments suggest that the Penniless algorithm is appropriate for MTE models, since the tradeoff between space requirements and accuracy is better than the one obtained with the discretisation.

The ideas contained in this paper can be extended to other propagation methods, specially the Lazy propagation and the class of Importance Sampling propagation algorithms, since these methods can take advantage of the reduction of the sizes of the potentials after pruning.

References

1. A. Cano and S. Moral. Heuristic algorithms for the triangulation of graphs. In B. Bouchon-Meunier, R.R. Yager, and L. Zadeh, editors, *Advances in Intelligent Computing*, pages 98–107. Springer Verlag, 1995.
2. A. Cano, S. Moral, and A. Salmerón. Penniless propagation in join trees. *International Journal of Intelligent Systems*, 15:1027–1059, 2000.
3. A. Cano, S. Moral, and A. Salmerón. Lazy evaluation in Penniless propagation over join trees. *Networks*, 39:175–185, 2002.
4. A. Cano, S. Moral, and A. Salmerón. Novel strategies to approximate probability trees in Penniless propagation. *International Journal of Intelligent Systems*, 18:193–203, 2003.
5. A. Christofides, B. Tanyi, D. Whobrey, and N. Christofides. The optimal discretization of probability density functions. *Computational Statistics and Data Analysis*, 31:475 – 486, 1999.
6. B. Cobb, P. Shenoy, and R. Rumí. Approximating probability density functions with mixtures of truncated exponentials. In *Proceedings of the Tenth International Conference IPMU'04*, Perugia (Italy), 2004.
7. F. Jensen and S.K. Andersen. Approximations in Bayesian belief universes for knowledge-based systems. In *Proceedings of the 6th Conference on Uncertainty in Artificial Intelligence*, pages 162–169, 1990.
8. F.V. Jensen, S.L. Lauritzen, and K.G. Olesen. Bayesian updating in causal probabilistic networks by local computation. *Computational Statistics Quarterly*, 4:269–282, 1990.
9. U. Kjærulff. Optimal decomposition of probabilistic networks by simulated annealing. *Statistics and Computing*, 2:1–21, 1992.
10. D. Koller, U. Lerner, and D. Anguelov. A general algorithm for approximate inference and its application to hybrid Bayes nets. In K.B. Laskey and H. Prade, editors, *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 324–333. Morgan & Kaufmann, 1999.
11. D. Kozlov and D. Koller. Nonuniform dynamic discretization in hybrid networks. In D. Geiger and P.P. Shenoy, editors, *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, pages 302–313. Morgan & Kaufmann, 1997.
12. S.L. Lauritzen. Propagation of probabilities, means and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87:1098–1108, 1992.
13. S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50:157–224, 1988.
14. A.L. Madsen and F.V. Jensen. Lazy propagation: a junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, 113:203–245, 1999.
15. S. Moral, R. Rumí, and A. Salmerón. Mixtures of truncated exponentials in hybrid Bayesian networks. In *Lecture Notes in Artificial Intelligence*, volume 2143, pages 135–143, 2001.
16. S. Moral, R. Rumí, and A. Salmerón. Estimating mixtures of truncated exponentials from data. In *Proceedings of the First European Workshop on Probabilistic Graphical Models*, pages 156–167, 2002.
17. K.G. Olesen. Causal probabilistic networks with both discrete and continuous variables. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:275–279, 1993.

18. A. Salmerón, A. Cano, and S. Moral. Importance sampling in Bayesian networks using probability trees. *Computational Statistics and Data Analysis*, 34:387–413, 2000.
19. E. Santos, S.E. Shimony, and E. Williams. Hybrid algorithms for approximate belief updating in Bayes nets. *International Journal of Approximate Reasoning*, 17:191–216, 1997.
20. P.P. Shenoy and G. Shafer. Axioms for probability and belief function propagation. In R.D. Shachter, T.S. Levitt, J.F. Lemmer, and L.N. Kanal, editors, *Uncertainty in Artificial Intelligence 4*, pages 169–198. North Holland, Amsterdam, 1990.

Approximate Factorisation of Probability Trees^{*}

Irene Martínez¹, Serafín Moral², Carmelo Rodríguez³, and Antonio Salmerón³

¹ Dept. Languages and Computation, University of Almería, Spain

`irene@ual.es`

² Dept. Computer Science and Artificial Intelligence, University of Granada, Spain

`smc@decsai.ugr.es`

³ Dept. Statistics and Applied Mathematics, University of Almería, Spain

`{crt, Antonio.Salmeron}@ual.es`

Abstract. Bayesian networks are efficient tools for probabilistic reasoning over large sets of variables, due to the fact that the joint distribution factorises according to the structure of the network, which captures conditional independence relations among the variables. Beyond conditional independence, the concept of asymmetric (or context specific) independence makes possible the definition of even more efficient reasoning schemes, based on the representation of probability functions through probability trees. In this paper we investigate how it is possible to achieve a finer factorisation by decomposing the original factors for which some conditions hold. We also introduce the concept of approximate factorisation and apply this methodology to the Lazy-Penniless propagation algorithm.

1 Introduction

Bayesian networks have been successfully used as efficient tools for knowledge representation and reasoning under uncertainty. The uncertainty is quantified in terms of a probability distribution over the domain variables, and the reasoning process conveys the computation of the posterior distribution for some variables given that the value of other variables is known. This task is called

There are several exact and approximate algorithms for probability propagation [2, 3, 6, 8, 10, 11], but the fact that it is an NP-hard problem [4, 5], justifies investing effort in the study of new algorithms with the aim of enlarging the class of affordable problems. The most recent advances in propagation have come along with methods that incorporate the ability of dealing with factorised representations of the potentials that represent the probabilistic information. These algorithms are Lazy propagation [8] and Lazy-penniless propagation [3].

A particular feature of the Lazy-penniless algorithm is that it uses probability trees [1] to represent probabilistic potentials. Probability trees are usually more

^{*} This work has been supported by the Spanish Ministry of Science and Technology, projects TIC2001-2973-C05-01,02, TIN2004-06204-C03-01 and by FEDER funds.

compact than probability tables and, what is more important, they provide a flexible way to reduce the space required to store a probabilistic potential, by pruning some of the branches of the trees. Of course, it can happen that the resulting tree be just an approximation of the original potential.

2 Bayesian Networks and Probability Trees

We will use the concept of potential to represent any probabilistic information in a Bayesian network (including ‘a priori’, conditional and ‘a posteriori’ distributions and intermediate results of operations between them). A potential ϕ for a set of variables $\mathbf{X}$ is a mapping $\phi : \Omega_{\mathbf{X}} \rightarrow \mathbb{R}_0^+$, where $\mathbb{R}_0^+$ is the set of non-negative real numbers and $\Omega_{\mathbf{X}}$ is the set of possible cases of the set of variables $\mathbf{X}$. We will consider only discrete variables with a finite number of cases.

Probability propagation is usually carried out over an auxiliary structure called join tree. A join tree is a tree where each node is a subset of the variables in the network, and such that if a variable is in two distinct nodes, then it is also in every node in the path connecting them. Every potential in the original Bayesian network (i.e. every conditional distribution) is assigned to a node containing the variables involved in the conditional distribution. A potential constantly equal to 1 (unity potential) is assigned to nodes which did not receive any conditional distribution. In this way, attached to every node V there will be a potential ϕ_V defined over the set of variables V and which is equal to the product of all the potentials assigned to it.

There are different ways to represent the potentials in the join tree (for instance, probability tables and probability trees) and it is possible to keep the potentials assigned to a node as a list instead of multiplying them initially [8, 3]. Probability propagation is carried out by a flow of messages through the edges of the join tree. A message from one node V_i to one of its neighbours, V_j , is a potential defined for the variables contained in $V_i \cap V_j$, and is obtained as the result of removing from the potentials attached to V_i all the variables not in V_j . A variable is removed multiplying the potentials containing it and then summing the variable out. This is precisely the step in which the complexity of probability propagation arises: The domain of the potential resulting from the product above mentioned may become so large that a huge amount of memory would be necessary to store it. In this paper we are concerned with the representation of probabilistic potentials by means of probability trees. We will introduce some factorisation techniques, either exact and approximate, that can help to overcome this problem.

A probability tree [1, 10] is a directed labeled tree, where each internal node represents a variable and each leaf node represents a probability value. Each internal node has one outgoing arc for each state of the variable associated with that node. Each leaf contains a non-negative real number. The size of a tree $\mathcal{T}$, denoted as $\text{size}(\mathcal{T})$, is defined as its number of leaves. A probability tree $\mathcal{T}$ on variables $\mathbf{X}_I = \{X_i | i \in I\}$ represents a potential $\phi : \Omega_{\mathbf{X}_I} \rightarrow \mathbb{R}_0^+$ if for each $\mathbf{x}_I \in \Omega_{\mathbf{X}_I}$ the value $\phi(\mathbf{x}_I)$ is the number stored in the leaf node that is reached by

starting from the root node and selecting the child corresponding to coordinate x_i for each internal node labeled with X_i .

A probability tree is usually a more compact representation of a potential than a table. Furthermore, trees allow to obtain even more compact representations in exchange of loosing accuracy. This is achieved by pruning some leaves and replacing them by the average value.

The basic operations ($\cdot$, $+$, $\wedge$, $\vee$, and $\sum_{x_i}$) over potentials required for probability propagation can be carried out directly over probability trees. The combination is done recursively and basically consists of selecting an initial node and multiplying each of its children by the other tree. A variable is marginalised out from a probability tree by replacing it by the sum of its children. We refer to [2] for the details.

3 Exact Factorisation of Probability Trees

Probability propagation basically relies on the combination and marginalisation operations, but the complexity is mainly determined by the combination. For instance, consider the situation in which we are going to delete a variable X_i in order to send a message between two nodes of the join tree. The first step is to combine the potentials (probability trees in this case) containing X_i . The result will be, in the worst case, a potential of size equal to the product of the sizes of the trees that took part in the combination. A gain in efficiency could be achieved if we managed to decompose each tree containing X_i as a product of two trees (factors) of lower size, one of them containing X_i and the other not containing it [9]. Then, the product would be actually carried out over potentials (trees) with reduced domains and therefore, the complexity of probability propagation would decrease. Clearly, it would only be true if the next two conditions hold:

1. The product of the factors into which a tree is decomposed is equal to the original tree, in order to keep the correctness of the results.
2. The propagation algorithm is able to deal with lists of potentials, instead of single potentials in each node and separator of the join tree.

We will devote the rest of the paper to investigate situations in which the probability trees can be decomposed preserving the first condition above, and also situations in which that condition holds only approximately. In this case, the results of the propagation will not be exact, but it is compensated for by the fact that the reasoning can be carried out over very large networks. With respect to the second condition, it is fulfilled by the Lazy [8] and Lazy-penniless [3] algorithms.

We have found two main situations in which probability trees can be decomposed. One is achieved when the variable to marginalise out is only in a part of the tree and the other one is met when some sub-trees of the original one are proportional.

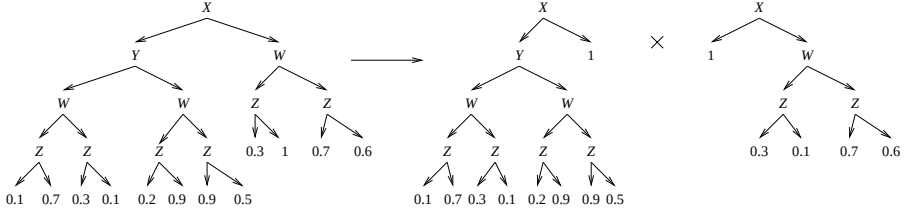


Fig. 1. A decomposition of a probability tree by splitting it with respect to Y

3.1 Tree Splitting

Assume that probability propagation is being carried out and that Y is the next variable to marginalise out, and that it is contained in a potential represented by the tree in the left side of Figure 1. Observe that Y is in the sub-tree corresponding to the first case of variable X , but not in the sub-tree corresponding to the second case. This is a very common situation in Lazy-penniless propagation, where it is possible that a variable disappears from a part of a tree after a pruning operation carried out to reduce the size of a tree.

This fact allows to decompose the original tree as the product of two factors of lower size, as displayed in Figure 1. The advantage of this decomposition is that the second factor does not take part in the product previous to the deletion of Y , because it does not contain Y , and the first factor is simpler than the original tree; Therefore, the complexity of the deletion of variable Y is reduced and thus the efficiency of Lazy propagation increased.

3.2 Proportional Sub-trees

Now assume that the next variable to marginalise out is X , and we find it in the tree shown in the upper part of Figure 2. We can see that, within context $W = 0$, all the children of X are proportional. In this case, it is possible to factorise the tree as a product of two trees, where the size of each of the factors is lower than the size of the original tree (see the lower part of Figure 2), in such a way that one of the factor keeps the information regarding X and the other contains the information irrelevant to X . More formally, trees able to be factorised in this way can be characterised by the next definition.

Definition 1. A tree $\mathcal{T}$ is said to be *proportional below X within context $(\mathbf{X}_C = \mathbf{x}_C)$* if and only if there exists a set of positive real numbers α_j such that for all $x_i \in \Omega_X$, $x_j, x_i \neq x_j \in \Omega_X$ $\exists \alpha_j > 0$ such that

$$\mathcal{T}^{R(\mathbf{X}_C=\mathbf{x}_C, X=x_i)} = \alpha_j \cdot \mathcal{T}^{R(\mathbf{X}_C=\mathbf{x}_C, X=x_j)} \quad (1)$$

where $\mathcal{T}^{R(\mathbf{X}_C=\mathbf{x}_C, X=x)}$ is the tree obtained by removing the subtree rooted at X in $\mathcal{T}$ and replacing it by a constant $\alpha = \{\alpha_j | j \neq i\}$. The α_j are called *proportionality factors*.

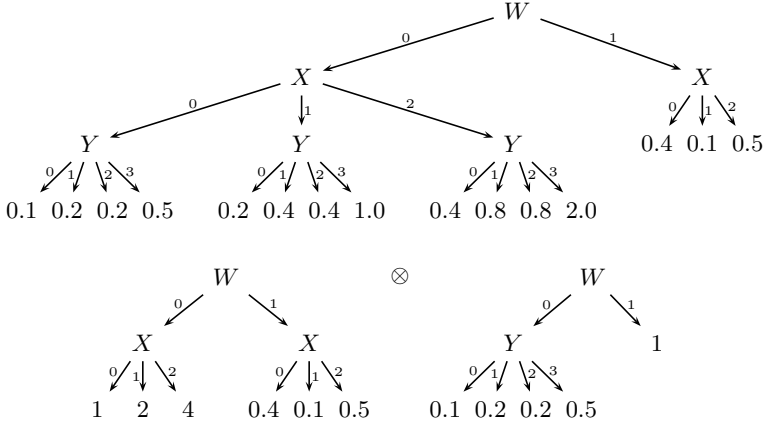


Fig. 2. A probability tree proportional below X for context ($W = 0$) and its decomposition with respect to variable X

The following definition identify each one of the factors into which a tree verifying definition 1 can be decomposed.

Definition 2. $\mathcal{T}(\mathbf{X}_C = \mathbf{x}_C, X = x_i, \alpha)$ is the **core term**, $\mathcal{T}^R(\mathbf{X}_C = \mathbf{x}_C, X = x_i)$ is the **free term**, $\mathcal{T}(\mathbf{X}_C = \mathbf{x}_C, X = x_i)$ is the **total term**, $\mathcal{T}(\mathbf{X}_D = \mathbf{x}_D)$ is the **context term**.

Observe that the core and free terms have size smaller than $\mathcal{T}$. Furthermore, the free term does not contain variable X . This, together with the result in the next proposition, show that factorisation increases the efficiency of probability propagation, in the sense that the amount of memory required is reduced.

Proposition 1. $\mathcal{T}(\mathbf{X}_C = \mathbf{x}_C, X = x_i, \alpha) = \mathcal{T}(\mathbf{X}_C = \mathbf{x}_C, X = x_i) \times \mathcal{T}(\mathbf{X}_C = \mathbf{x}_C, X = x_i, \alpha)$.

$$\mathcal{T} = \mathcal{T}(\mathbf{X}_C = \mathbf{x}_C, X = x, \alpha) \times \mathcal{T}(\mathbf{X}_C = \mathbf{x}_C, X = x). \quad (2)$$

3.3 Partially Proportional Sub-trees

Still there is another situation in which some regularities can be found in a probability tree that can be used to reduce the complexity of the operations involved in the process of marginalising out a variable. The scenario is very similar to the case of proportional sub-trees described above, but instead of all

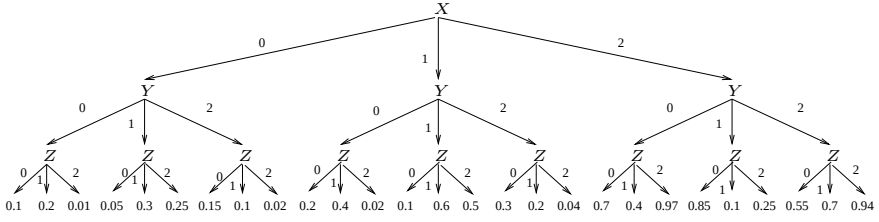


Fig. 3. A probability tree partially proportional below variable X

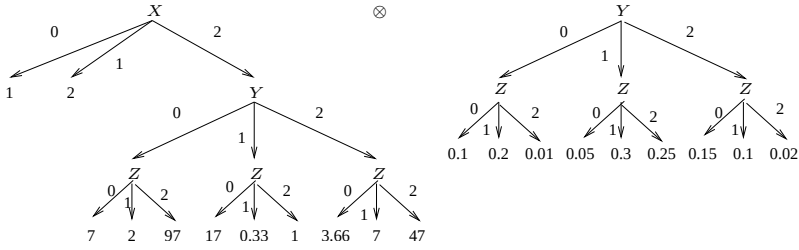


Fig. 4. Factorisation of the tree in figure 3

the children of the variable to delete, only some of them are proportional. This situation is illustrated in the next example.

Assume we have three variables X , Y and Z , each one of them taking values on the set $\{0, 1, 2\}$. Consider the conditional distribution for X given Y and Z represented by the probability tree in figure 3. Observe that the tree is not proportional below X , because the sub-trees corresponding to $X = 0$ and $X = 1$ are proportional, but the sub-tree for $X = 2$ is not. However, even though the conditions in definition 1 are not met in this case, the tree can be decomposed in the way described in figure 4. Notice that the resulting factorisation is able to represent the conditional distribution for X using just 20 numbers instead of 27. $\square$

Formally, a probability where this kind of proportionality occurs can be defined as follows.

Definition 3. Let $\mathcal{T}$ be a probability tree. We say that $\mathcal{T}$ is partially proportional below X within context $(\mathbf{X}_C = \mathbf{x}_C)$, if there exists a set $L \subset \Omega_X \setminus \{x_i\}$ such that for all $x_i \in \Omega_X$ and $x_j \in L$ there exists $\alpha_j > 0$ such that

$$\mathcal{T}^{R(\mathbf{X}_C = \mathbf{x}_C, X = x_i)} = \alpha_j \cdot \mathcal{T}^{R(\mathbf{X}_C = \mathbf{x}_C, X = x_j)} \quad (3)$$

In this setting, the concept of core term given in definition 2 must be modified in order to guarantee that the product of the core and free terms is equal to the original tree. However, the free term needs not be re-defined.

Observe that proportional and partially proportional trees for context $(\mathbf{X}_C = \mathbf{x}_C)$ are δ -factorisable, with $\delta = 0$.

Now we will consider how to factorise δ -decomposable trees, analysing different divergence measures and computing the optimum α . We will impose the next consistency restriction to all the approximate factorisation methods that we will propose: A method is said to be consistent if it introduces no error when the tree is proportional or partially proportional below the considered context (see definitions 1 and 3).

4.1 Computing the Proportionality Factor

Consider a probability tree $\mathcal{T}$. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be sub-trees of $\mathcal{T}$ below a variable X , for a given context $(\mathbf{X}_C = \mathbf{x}_C)$ with leaves $\mathcal{P} = \{p_i : i = 1, \dots, n; p_i \neq 0\}$ and $\mathcal{Q} = \{q_i : i = 1, \dots, n; \}$ respectively. As we described before, approximate factorisation is achieved by replacing $\mathcal{T}_2$ by another tree $\mathcal{T}_2^*$ such that $\mathcal{T}_2^*$ is proportional to $\mathcal{T}_1$. It means that the leaves of $\mathcal{T}_2^*$ will be $\mathcal{Q}^* = \{\alpha p_i : i = 1, \dots, n; \}$, where α is the proportionality factor between $\mathcal{T}_1$ and $\mathcal{T}_2$. Let us denote by $\{\pi_i = q_i/p_i, i = 1, \dots, n; \}$ the ratios between the leaves of $\mathcal{T}_2$ and $\mathcal{T}_1$.

We have considered several possibilities for computing the proportionality factor, α . First we will derive the value of the proportionality factor under the restriction of minimising different measures of divergence:

1. The χ^2 divergence, defined as

$$\mathcal{D}_\chi(\mathcal{T}_2, \mathcal{T}_2^*) = \sum_{i=1}^n \frac{(q_i - \alpha p_i)^2}{q_i} ,$$

is minimised for α equal to $\alpha_\chi = \frac{\sum_{i=1}^n p_i}{\sum_{i=1}^n p_i / \pi_i}$. Instead of using $\mathcal{D}_\chi$, we can consider its normalised version

$$\mathcal{D}_{\chi^*}(\mathcal{T}_2, \mathcal{T}_2^*) = \sqrt{\frac{\mathcal{D}_\chi}{\mathcal{D}_\chi + n}} ,$$

which takes values between 0 and 1 and is minimised for the same α .

2. The mean squared error

$$\mathcal{D}_{mse}(\mathcal{T}_2, \mathcal{T}_2^*) = \frac{1}{n} \sum_{i=1}^n (q_i - \alpha p_i)^2$$

is minimised for $\alpha_{mse} = \frac{\sum_{i=1}^n \pi_i p_i^2}{\sum_{i=1}^n p_i^2}$.

In case of using a weighted MSE as divergence measure, i.e.

$$\mathcal{D}_{wmse}(\mathcal{T}_2, \mathcal{T}_2^*) = \sum_{i=1}^n h_i (q_i - \alpha p_i)^2$$

with $\{h_i \geq 0, i = 1, \dots, n; \sum h_i = 1\}$, the optimum proportionality factor is

$$\alpha_{wmse} = \frac{\sum_{i=1}^n h_i \pi_i p_i^2}{\sum_{i=1}^n h_i p_i^2} .$$

A possible selection of the weights h_i is $h_i = \frac{q_i}{\sum_{i=1}^n q_i}$, in which case $\mathcal{D}_{mse}$ would be the expected MSE with respect to $\mathcal{T}_2$ (actually, with respect to a probability distribution proportional to the potential represented by $\mathcal{T}_2$).

3. The Kullback-Leibler divergence, defined as

$$\mathcal{D}_{kl}(\mathcal{T}_2, \mathcal{T}_2^*) = \sum_{i=1}^n q_i \log \left(\frac{q_i}{\alpha p_i} \right) ,$$

reaches its minimum at $\alpha_{kl} = 2^{\frac{\sum_{i=1}^n q_i \log(\pi_i)}{\sum_{i=1}^n q_i}}$. The problem of using $\mathcal{D}_{kl}$ is that it requires that the sum of the values of its arguments coincide [7]. Otherwise, $\mathcal{D}_{kl}$ can take negative values. This renders this criterion useless for our purposes.

But it is also possible to obtain the proportionality factor independently of any divergence measure. For instance, one restriction could be to ensure that the weight of the original and the approximate tree coincide, that is:

$$\text{sum}(\mathcal{T}_2^*) = \sum_{i=1}^n \alpha p_i = \sum_{i=1}^n q_i = \text{sum}(\mathcal{T}_2) .$$

We will refer to this as the *restriction of equal sum*, and the proportionality factor that corresponds to this restriction is

$$\alpha_{wp} = \frac{\sum_{i=1}^n q_i}{\sum_{i=1}^n p_i} = \frac{\sum_{i=1}^n \pi_i p_i}{\sum_{i=1}^n p_i} .$$

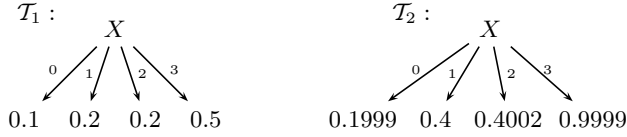
Perhaps the more straightforward way to obtain a value for α is the so-called *weighted average*, which computes it as a weighted average of the ratios between the leaves of $\mathcal{T}_1$ and $\mathcal{T}_2$. The resulting proportionality factor is

$$\alpha_{wa} = \sum_{i=1}^n h_i \pi_i ,$$

with $\{h_i \geq 0, i = 1, \dots, n; \sum h_i = 1\}$. Observe that α_{wp} and α_{mse} are particular cases of α_{wa} with $h_i = \frac{p_i}{\sum_{i=1}^n p_i}$ and $h_i = \frac{p_i^2}{\sum_{i=1}^n p_i^2}$ respectively.

Besides, there may be other divergence measures that could be applied to our problem but that cannot be minimised with respect to α . Of special interest is the divergence measure computed as the *maximum absolute difference* between the leaves of $\mathcal{T}_2$ and $\mathcal{T}_2^*$, that we will use in the experiments:

$$\mathcal{D}_{mad}(\mathcal{T}_2, \mathcal{T}_2^*) = \max_{1 \leq i \leq n} |q_i - \alpha p_i| .$$

**Fig. 5.** Almost proportional trees**Table 1.** Divergences between the tree $\mathcal{T}_2$ in Fig. 5 and the different approximations of it which are proportional to $\mathcal{T}_1$

	$\alpha_{wp} =$	$\alpha_{\chi} =$	$\alpha_{mse} =$	$\alpha_{wmse} =$	$\alpha_{kl} =$	$\alpha_{wa} =$
	2.0	1.9999998	1.9999412	1.9998733	2.0000001	2.0000002
$\mathcal{D}_{mad}$	2E-4	2.00032E-4	2.11764E-4	2.25343E-4	1.99984E-4	1.99968E-4
$\mathcal{D}_{\chi}$	0.00039997005	0.00039997003	0.000402115	0.000409859	0.00039997005	0.00039997009
$\mathcal{D}_{\chi^*}$	0.00019998502	0.00019998501	0.000201057	0.000204929	0.00019998503	0.00019998504
$\mathcal{D}_{mse}$	0.000122474	0.000122467	0.000121267	0.000122872	0.000122477	0.000122481
$\mathcal{D}_{wmse}$	5.91671E-5	5.91549E-5	5.56270E-5	5.41362E-5	5.91732E-5	5.91793E-5

The trees in figure 5 are "almost" proportional. It seems that they could be considered as proportional and the corresponding factorisation would not affect very much the results of the probability propagation algorithm. Table 1 shows the divergence between $\mathcal{T}_2$ and $\mathcal{T}_2^*$ using the different criteria for approximate factorisation described in this section. It can be seen from the results in that table how choosing α_{χ} , α_{mse} and α_{wmse} minimises the corresponding divergence measures with respect to which they were obtained. The maximum absolute divergence ($\mathcal{D}_{mad}$) is minimised, in this example, by choosing α_{wa} as proportionality factor.

If the trees in figure 5 are siblings below a given variable Y for a context ($\mathbf{X}_C = \mathbf{x}_C$) of some tree $\mathcal{T}$, it can be said that, according to definition 5 that $\mathcal{T}$ is δ -factorisable within context ($\mathbf{X}_C = \mathbf{x}_C$) for any $\delta > 0.001$, regardless the selected α and the divergence measure used.

For $\delta \leq 0.001$, $\mathcal{T}$ would not always be considered δ -factorisable. For example, if we selected a tolerance $\delta = 0.0002$ and the divergence measure $\mathcal{D}_{mad}$, $\mathcal{T}$ is δ -factorisable within context ($\mathbf{X}_C = \mathbf{x}_C$) only for proportionality factors α_{wp} , α_{wa} and α_{kl} .

5 Experiments

In order to illustrate how the techniques above described can be used to tradeoff accuracy for space requirements, we have tested the Lazy-penniless algorithm [3] with the added feature of factorising the potentials before deleting a variable, using different real networks. In order to analyse the impact of the factorisation, we have used the simplest version of Lazy-penniless (no heuristic is used to select the order of combination of the potentials), and the trees are not pruned. Due to space limitations, we only report the results for two well known networks

Table 2. Experimental results for network Munin1

δ	D_χ divergence			Weight Preserving		
	Mean	MSE	nAp	Mean	MSE	nAp
0.025	27556.85	3.06E-6	3070	23704.26	1.49E-6	2788
0.050	27286.13	1.52E-4	3387	23609.30	2.68E-6	2982
0.075	26885.23	2.40E-4	3699	23300.01	1.33E-5	3443
0.1	26238.68	7.04E-4	4645	23499.51	1.42E-5	3655
0	31947.58	0	132	31947.58	0	132

Table 3. Experimental results for network Water

δ	D_χ divergence			Weight Preserving		
	Mean	MSE	nAp	Mean	MSE	nAp
0.025	1884.80	1.93E-5	368	1884.54	1.93E-5	367
0.050	1735.47	2.23E-5	435	1737.91	2.22E-5	419
0.075	1692.30	9.88E-6	530	1693.14	1.02E-5	512
0.1	1581.15	3.28E-5	570	1581.74	3.35E-5	558
0	1733.23	0	2	1733.23	0	2

(Munin1 and Water) borrowed from the Decision Support Systems Group at Aalborg University. The results are displayed in table 2 and table 3 respectively, where the first column δ indicates the error allowed when factorising (tolerance in terms of distance D_{χ^*}). The reason to use D_{χ^*} is that it is easier to control, since it is between 0 and 1. We have computed the mean of the sizes of the potentials used during the propagation, the average mean squared error (MSE) for all the unobserved variables after the propagation and the number of factorisations actually carried out. In the experiments, we have only searched for proportional subtrees which root is not located beyond half of the depth of the tree, in order to avoid useless factorisations (for instance, factorising only the leaves). With respect to the computing times, they are about a 20% higher compared with the Lazy propagation (or exact Lazy-penniless), but the space requirements are lower. The mean clique sizes for Lazy propagation are 31905.37 for Munin1 and 1733.2 for Water.

Even though the analysis is still rather preliminary, the results seem to indicate that approximate factorisation is a valid method for controlling the space requirements during propagation.

6 Conclusions

In this paper we have extended the factorisation technique presented in [9] by introducing the possibility of decomposing the trees that are approximately proportional. The results suggest that this method provides a valid tradeoff between space requirements and approximation error, and that it can be controlled by means of the δ parameter. A deeper experimental analysis is necessary to know how far this technique can go, and which of the proposed distance measure achieves the best results. Besides, we have not yet checked the joint behaviour of factorising and splitting, but we believe that the results must significantly improve. We are also implementing the use of factorisation in compilation time,

in order to obtain smaller initial probability distributions for the propagation phase.

References

1. C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. In E. Horvitz and F.V. Jensen, editors, *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, pages 115–123. Morgan & Kaufmann, 1996.
2. A. Cano, S. Moral, and A. Salmerón. Penniless propagation in join trees. *International Journal of Intelligent Systems*, 15:1027–1059, 2000.
3. A. Cano, S. Moral, and A. Salmerón. Lazy evaluation in Penniless propagation over join trees. *Networks*, 39:175–185, 2002.
4. G.F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.
5. P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60:141–153, 1993.
6. F.V. Jensen, S.L. Lauritzen, and K.G. Olesen. Bayesian updating in causal probabilistic networks by local computation. *Computational Statistics Quarterly*, 4:269–282, 1990.
7. S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:76–86, 1951.
8. A.L. Madsen and F.V. Jensen. Lazy propagation: a junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, 113:203–245, 1999.
9. I. Martínez, S. Moral, C. Rodríguez, and A. Salmerón. Factorisation of probability trees and its application to inference in Bayesian networks. In J.A. Gámez and A. Salmerón, editors, *Proceedings of the First European Workshop on Probabilistic Graphical Models*, pages 127–134, 2002.
10. A. Salmerón, A. Cano, and S. Moral. Importance sampling in Bayesian networks using probability trees. *Computational Statistics and Data Analysis*, 34:387–413, 2000.
11. P.P. Shenoy. Binary join trees for computing marginals in the Shenoy-Shafer architecture. *International Journal of Approximate Reasoning*, 17:239–263, 1997.

Abductive Inference in Bayesian Networks: Finding a Partition of the Explanation Space

M. Julia Flores¹, José A. Gámez¹, and Serafín Moral²

¹ Departamento de Informática, Universidad de Castilla-La Mancha,
02071 Albacete, Spain

² Departamento de Ciencias de la Computación e I.A.,
Universidad de Granada, 18071 Granada, Spain

Abstract. This paper proposes a new approach to the problem of obtaining the most probable explanations given a set of observations in a Bayesian network. The method provides a set of possibilities ordered by their probabilities. The main novelties are that the level of detail of each one of the explanations is not uniform (with the idea of being as simple as possible in each case), the explanations are mutually exclusive, and the number of required explanations is not fixed (it depends on the particular case we are solving). Our goals are achieved by means of the construction of the so called *explanation tree* which can have asymmetric branching and that will determine the different possibilities. This paper describes the procedure for its computation based on information theoretic criteria and shows its behaviour in some simple examples.

1 Introduction

Although the most common probabilistic inference in Bayesian networks (BNs) is probability or evidence propagation [18, 1, 11], that is, computation of posterior probability for all non-observed variables given a set of observations ($X_O = x_O$) (the *evidence*), there are other interesting inference tasks. In this paper we are concerned with the inference task that attempts to generate explanations for a given evidence. Generating explanations in Bayesian networks can be understood in two (main) different ways:

1. *Explaining the reasoning process* (see [12] for a review). That is, trying to justify *how* a conclusion was obtained, *why* new information was asked, etc.
2. *Diagnostic explanations* or *abductive inference* (see [9] for a review). In this case the explanation reduces to *factual* information about the state of the world, and the best explanation for a given evidence is the state of the world (configuration) that is the most probable given the evidence [18].

In this paper we focus on the second approach. Therefore, given a set of observations or evidence ($X_O = x_O$ or x_O in short) known as the *explanandum*, we aim to obtain the best configuration of values for the explanatory variables (the *explanation*) which is consistent with the *explanandum* and which needs

to be assumed to predict it. Depending on what variables are considered as *explanatory*, two main abductive tasks in BNs are identified:

- *Most Probable Explanation* (MPE) or *total abduction*. In this case all the unobserved variables (X_U) are included in the explanation [18]. The *best* explanation is the assignment $X_U = x_U^*$ which has maximum a posteriori probability given the *explanandum*, i.e.,

$$x_U^* = \arg \max_{x_U \in \Omega_{X_U}} P(x_U | x_O). \quad (1)$$

Searching for the best explanation has the same complexity (NP-hard [23]) as probability propagation, in fact the best MPE can be obtained by using probability propagation algorithms but replacing summation by maximum in the marginalisation operator [3]. However, as it is expected to have several competing hypothesis accounting for the *explanandum*, our goal usually is to get the K best MPEs. Nilsson [15] showed that using algorithm in [3] only the first three MPEs can be correctly identified, and proposed a clever method to identify the remaining $(4, \dots, K)$ explanations.

One of the main drawbacks of the MPE definition is that as it produces complete assignments, the explanations obtained can exhibit the *overspecification* problem [21] because some non-relevant variables have been used as explanatory.

- *Maximum a Posteriori Assignment* (MAP) or *partial abduction* [14, 21]. The goal of this task is to alleviate the overspecification problem by considering as target variables only a subset of the unobserved variables called the *explanation set* (X_E). Then, we look for the maximum a posteriori assignment of these variables given the explanandum, i.e.,

$$x_E^* = \arg \max_{x_E} P(x_E | x_O) = \arg \max_{x_E} \sum_{x_R} P(x_E, x_R | x_O), \quad (2)$$

where $X_R = X_U \setminus X_E$. This problem is more complex than the MPE problem, because it can be NP-hard even for cases in which MPE is polynomial (e.g., polytrees) [17, 5], although recently Park and Darwiche [16, 17] have proposed exact and approximate algorithms to enlarge the class of *efficiently* solved cases. With respect to looking for the K best explanations, exact and approximate algorithms which combine Nilsson algorithm [15] with probability trees [19] has been proposed in [6].

The question now is which variables should be included in the explanation set. Many algorithms avoid this problem by assuming that the explanation set is provided as an input, e.g., given by the experts or users. Many others interpret the BN as a causal one and only ancestors of the *explanandum* are allowed to be included in the explanation set (sometimes only root nodes are considered) [13]. However, including all the ancestors in the explanation set does not seem to avoid the overspecification problem and even so, what happens if the network does not have a causal interpretation?, e.g., it has been learnt from a data base

or it represents an agent's beliefs [2]. Shimony [21, 22] goes one step further and describes a method which tries to identify the relevant variables (among the *explanandum* ancestors) by using independence and relevance based criteria. However, as pointed out in [2] the explanation set identified by Shimony's method is not as concise as expected, because for each variable in the *explanandum* all the variables in at least one path from it to a root variable are included in the explanation set. Henrion and Druzdzel [10] proposed a model called *scenario-based explanation*. In this model a tree of propositions is assumed, where a path from the root to a leaf represents a scenario, and they look for the scenario with highest probability. In this model, partial explanations are allowed, but they are restricted to come from a set of predefined explanations.

As stated in [2] *conciseness* is a desirable feature in an explanation, that is, the user usually wants to know only the most influential elements of the complete explanation, and does not want to be burdened with unnecessary detail. Because of this, a different approach is taken in [4]. The idea is that even when only the relevant variables to the *explanandum* are included in the explanation set, the explanations can be simplified due to context-specific irrelevance. This idea is even more interesting when we look for the K MPEs, because it allows us to obtain explanations with different number of literals. In [4] the process is divided into two stages: (1) the K MPEs are obtained for a given prespecified explanation set, and (2) then they are simplified by using different independence and relevance based criteria.

In this paper we try to obtain simplified explanations directly. The reason is that the second stage in [4] requires to carry out several probabilistic propagations and so its computational cost is high (and notice that this process is carried out after -a complex- MAP computation). Another drawback of the procedure in [4] is that it is possible, that after simplification, the explanations are not mutually exclusive, we can have even the case of two explanations such that one is a subset of the other. Here, our basic idea is to start with a predefined explanation set X_E , and then we build a tree in which variables (from X_E) are added in function of their explanatory power with respect to the *explanandum* but taken into account the current context, that is, the partial assignment represented by the path obtained from the root to the node currently analysed. Variables are selected based on the idea of *stability*, that is, we can suppose that our system is (more or less) stable, and that it becomes unstable when some (*unexpected*) observations are entered into the system. The instability of a variable will be measured by its entropy or by means of its (im)purity (GINI index). Therefore, we first select those variables that reduce most the uncertainty of the non-observed variables of the explanation set, i.e., the variables better determining the value of the explanation variables. Of course, the tree does not have to be symmetric and we can decide to stop the growing of a branch even if not all the variables in X_E have been included. In any case, our set of explanations will be mutually exclusive, and will have the additional property of being exhaustive, i.e., we will construct a true partition of the set of possible configurations or scenarios of the values of the variables in the explanation set.

The subsequent sections describe our method in detail and illustrate it by using some (toy) study cases. Finally in Section 4 we present our conclusions and outline future works.

2 How to Obtain an *Explanation Tree*

Our method aims to find the best explanation(s) for the observed variables that do not necessarily have a fixed number of literals. The provided explanations will adapt to the current circumstances. Sometimes that a variable X takes a particular value it is an explanation by itself (Occam’s razor) and including other variables to this explanation will not add any new information. We have then decided to represent our solutions by a tree, the *Explanation Tree* (ET).

In the ET, every node will denote a variable of the explanation set and every branch from this variable will indicate the instantiation of this variable to one of its possible states. Each node of the tree will determine an assignment for the variables in the path from the root to it: each variable equal to the value on the edge followed by the path. This assignment will be called the configuration of values associated to the node. In the explanation tree, we will store for each leaf the probability of its associated configuration given the evidence. The set of explanations will be the set of configurations associated to the leaves of the explanation tree ordered by their posterior probability given the evidence. For example, in Fig. 5.a we can see three variables $A1$, $A2$ and $N2$ that belong to the explanation set, since they are nodes in the ET. In this particular example there are four leaves nodes, i.e., four possible explanations. What this ET indicates is that, given the observed evidence, $A1 = f$ is a valid explanation for such situation (with its probability). But if it is not the case then we should look into other factors, in this case $N2$. For example, we can see that adding $N2 = f$ to the current path ($A1 = ok$) will be enough to provide an explanation. Otherwise, when $N2 = ok$ the node needs to be *expanded* and we will look for other involved factors in order to find a valid explanation (in this example, by using $A2$).

Although the underlying idea is simple, how to obtain this tree is not so evident. There are two major points that have to be answered:

- As the ET is created in a top-down way, given a branch of the tree, how to select the next variable?
- Given our goals, i.e. allow asymmetry and get concise explanations, how to decide when to stop branching?

To solve the two previous questions we have used information measures. For the first one, we look for the variable that once instantiated the uncertainty of the rest explanation variables is reduced at maximum. In other words, given the context provided by the current branch, we identify the most explicative as the one that helps to determine the values of the other variables as much as possible.

Algorithm 1 (CREATE-NEW-NODE) recursively creates our ET. In this algorithm we assume the existence of an inference engine that provides us with the probabilities needed during tree growing. We comment on such engine in Section 2.1. The algorithm is called with the following parameters:

1. The evidence/observations to be explained x_O .
2. The *path* corresponding to the branch we are growing. In the first call to this algorithm, i.e. when deciding the root node, this parameter will be null.
3. The current explanation set (X_E). That is, the set of explanatory variables already available given the context (*path*). In the first call X_E is the original explanation set. Notice also that if $X_E = X_U$ in the first call, i.e., all non-observed variables belong to the explanation set, then the method has to select those variables relevant to the explanation without prior information.
4. Two real numbers α and β used as thresholds (on information and probability respectively) to stop growing.
5. The final explanation tree that will be recursively and incrementally constructed as an accumulation of branches (*paths*). Empty in the initial call.

Algorithm 1. Creates a new node for the explanation tree

```

1: procedure CREATE_NEW_NODE( $x_O, path, X_E, \alpha, \beta, ET$ )
2:   for all  $X_j, X_k \in X_E$  do
3:      $\mathbf{Info}[X_j, X_k] = \mathbf{Inf}(X_j, X_k | x_O, path)$ 
4:   end for
5:    $X_j^* = \arg \max_{X_j \in X_E} \sum_{X_k} \mathbf{Info}[X_j, X_k]$ 
6:   if CONTINUE( $\mathbf{Info}[], X_j^*, \alpha$ ) and  $P(path | x_O) > \beta$  then
7:     for all state  $x_j$  of  $X_j^*$  do
8:        $new\_path \leftarrow path + X_j^* = x_j$ 
9:       CREATE_NEW_NODE( $x_O, new\_path, X_E \setminus X_j^*, \alpha, \beta, ET$ )
10:    end for
11:   else
12:      $ET \leftarrow ET \cup \langle path, P(path | x_O) \rangle$   $\triangleright$  update the ET adding path
13:   end if
14: end procedure

```

In algorithm 1, for each variable in the explanation set, X_j , we compute the sum of the amount of information that this variable provides about all the current explanation variables conditioned to the current observations x_O^* . We are interested in the variable that maximises this value. In our study we have considered two classical measures: *mutual information* ($\mathbf{Inf}(X_j, X_k | x_O^*) = \sum_{x_j, x_k} P(x_j, x_k | x_O^*) \log \left(\frac{P(x_j, x_k | x_O^*)}{P(x_j | x_O^*) \cdot P(x_k | x_O^*)} \right)$) and *GINI index* ($\mathbf{Inf}(X_j, X_k | x_O^*) = 1 - \sum_{x_j, x_k} P(x_j, x_k | x_O^*)^2$). Thus, there are different instances of the algorithm depending on the criterion used as $\mathbf{Inf}$.

Once we have selected the next variable to be placed in a branch, we have to decide whether or not to expand this node. Again, we will use the measure $\mathbf{Inf}$. The procedure CONTINUE is the responsible to take this decision by considering the vector $\mathbf{Info}[]$. This procedure considers the list of values $\mathbf{Info}[X_j^*, X_k]$ for $X_k \neq X_j^*$, then it computes the **maximum**, **minimum**, or **average** of them, depending on the particular criterion we are using. If this value is greater than α it decides to continue. Of course the three criteria give rise to different behaviour, being **minimum** the most restrictive, **maximum** the most permissive and having **average** and intermediate behaviour.

Notice that when only two variables remain in the explanation set, the one selected in line 5 is in fact that having greater entropy ($I(X, X) = H(X)$) if mutual information is used. Also, when only one variable is left, it is of course the selected one, but it is necessary to decide whether or not it should be expanded. For that purpose, we use the same information measure, that is, $I(X, X)$ or $\text{GINI}(X, X)$, and only expand this variable if it is at least as uncertain (unstable) as the distribution $[1/3, 2/3]$ (Normalising with more than two states). That is, we only add a variable if it has got more uncertainty than a given threshold.

2.1 Computation

Our inference engine is (mainly) based on Shenoy Shafer running over a binary join tree [20]. Furthermore, we have forced the existence of a single clique (being a leaf) for each variable in X_E , i.e. a clique which contains only a variable. We use these cliques to enter *as evidence* the value to which an explanatory variable is instantiated, as well as to compute its posterior probability.

Here we comment on the computation of the probabilities needed to carry out the construction of the explanation tree. Let us assume that we are considering to expand a new node in the tree which is identified by the configuration (path) $C = c$. Let x_O^* be the configuration obtained by joining the observations $X_O = x_O$ and $C = c$. Then, we need to calculate the following probabilities:

- $P(X_i, X_j | x_O^*)$ for $X_i, X_j \in X_E \setminus C$. To do this we use a two stage procedure:
 1. Run a full propagation over the join tree with x_O^* entered as evidence. In fact, many times only the second stage (i.e., *DistributeEvidence*) of Shenoy-Shafer propagation is needed. This is due to the single cliques included in the join tree, because if only one evidence item (say X) has changed¹ from the last propagation, we locate the clique containing X , modify the evidence entered over it and run *DistributeEvidence* by using it as root.
 2. For each pair (X_i, X_j) whose joint probability is required, locate the two closest cliques (C_i and C_j) containing X_i and X_j . Pick all the potentials in the path between C_i and C_j and obtain the joint probability by using variable elimination [7]. In this process, we can take as basis the deletion sequence implicit in the join tree (but without deleting the required variables) and then the complexity is not greater than the complexity of sending a series of messages along the path connecting C_i with C_j for each possible value of X_i . But, the implicit triangulation has been optimized to compute marginal distributions for single variables, and it is possible to improve it to compute the marginal of two variables as in our case. The complexity of this phase is also decreased by using caching/hashing techniques, because some sub-paths can be shared between different pairs, or even a required potential can be directly obtained by marginalisation over one previously cached.

¹ Which happens frequently because we build the tree in depth, and (obviously) the create-node algorithm and the probabilistic inference engine are synchronised.

- $P(C = c|x_O) = \frac{P(C=c,x_O)}{P(x_O)}$. This probability can be easily obtained from previously described computations. We just use $P(x_O)$ that is computed in the first propagation (when selecting the variable to be placed in the root of our explanation tree) and $P(x_O^*) = P(C = c, x_O)$ which is computed in the current step (full propagation with x_O^* as evidence).

Though this method requires multiple propagations, all of them are carried out over a join tree obtained without constraining the triangulation sequence, and so it (generally) has a size considerably smaller than the join tree used for partial abductive inference over the same explanation set [17, 5]. Besides, the join tree can be pruned before starting the propagations [5].

3 Cases of Study: Explanation and Diagnosis

Because we are in an initial stage of research about the ET method, in order to show how it works and the features of the provided explanations, we found interesting to use some (toy) networks having a familiar meaning for us, to test whether the outputs are reasonable. We used the following two cases:

1. **academe** network: it represents the evaluation for a subject in an academic environment, let us say, university, for example. This simple network has got seven variables, as Fig. 3 shows. Some of them are intermediate or auxiliary variables. What this network tries to model is the final mark for a student, depending on her practical assignments, her mark in a theoretical exam, on some possible extra tasks carried out by this student, and on other factors such as behaviour, participation, attendance... We have chosen this particular topic because the explanations are easily understandable from an intuitive point of view.

In this network we consider as evidence that a student has failed the subject, i.e., $x_O \equiv \{finalMark=failed\}$, and we look for the best explanations that could lead to this fact. We use $\{Theory, Practice, Extra, OtherFactors\}$ as the explanation set. In this first approach we run our ET-based algorithm with $\beta = 0.0$, $\alpha=0.05|0.07$ and $criterion = \max|min|avg$. Figure 3 summarises the obtained results (variables are represented by using their initials).

2. **gates** network: this second net represents a logical circuit (Fig. 2.a). The network (Fig. 2.b) is obtained from the circuit by applying the method described in [8]. The network has a node for every input, output, gate and intermediate output. Again, we use an example easy to follow, since the original circuit only has got seven gates (two NOT-gates, two OR-gates and three AND-gates) and the resulting network has 19 nodes.

In this case, we consider as evidence one possible input for the circuit (ABCDE=01010) plus an erroneous output (given such input), KL=10. Notice that the correct output for this case is KL=00, and also notice that from the transformation carried out to build the network, even when some gates are wrong the output could be correct (see [8]). So our evidence is ABCDEKL =

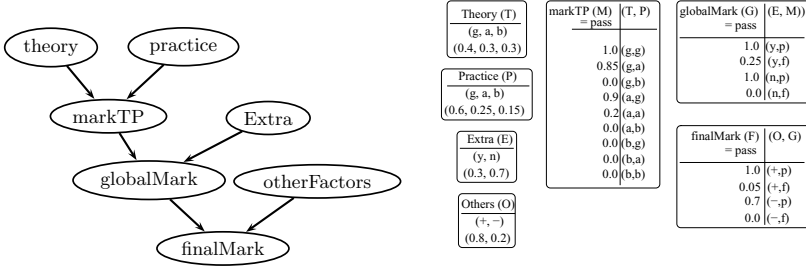


Fig. 1. Case of study 1: academe network

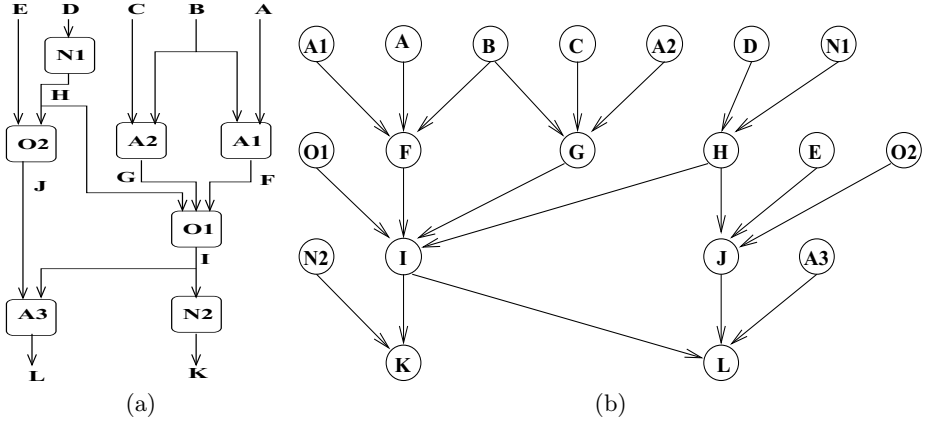


Fig. 2. (a) Original logic circuit. (b) Network gates obtained from (a) by using the transformation described in [8]

0101010 and we consider $X_E = \{A1, A2, A3, O1, O2, N1, N2\}$ as the explanation set with the purpose of detecting which gate(s) is(are) faulty. Figures 4 and 5 show the trees obtained for MI and GINI respectively. The same parameters as in the previous study case are used but $\beta = 0.05$.

3.1 Analysis of the Obtained Trees

The first thing we can appreciate from the obtained trees is that they are *reasonable*, i.e., the produced explanations are those that could be expected.

Regarding the *academe* network, when a student is failed, it seems reasonable that the most explicative variable is *theory* because of the probability tables introduced in the network. Thus, in all the cases *Theory* is the root node, and also in all the cases $\{theory=bad\}$ constitutes an explanation by itself, being in fact the most probable explanation (0.56).

The other common point for the obtained ETs is that the branch with *theory* as good is always expanded. It is clear that being *theory* ok another reason must

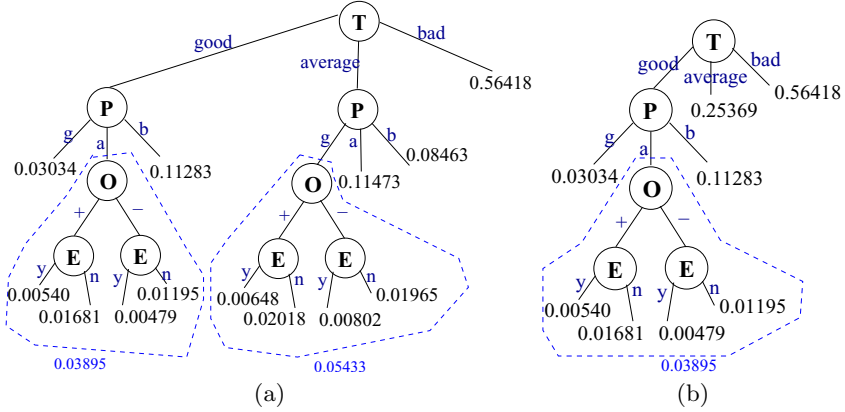


Fig. 3. Results for academe: (a) is the obtained tree for all MI cases except (MI, $\alpha=0.05$, min) which produces tree (b) together with all (gini, $\alpha=0.05$) cases and (gini, $\alpha=0.07$, max). Finally it is necessary to remark that (gini, $\alpha=0.07$, min|avg) leads to an empty tree, $\emptyset$, that is no node is expanded. β is 0.0

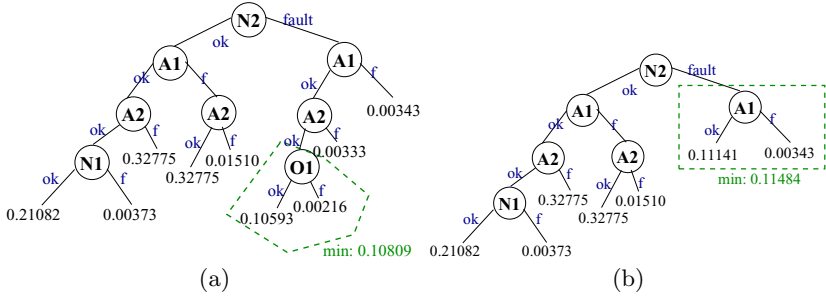


Fig. 4. Results for gates and MI: (a) is the obtained ET for (MI, $\alpha=0.05$, max|avg) and also (MI, $\alpha=0.07$, max); (b) is for (MI, $\alpha=0.07$, avg). In both cases min prunes more the tree than avg, so the dotted area would not be expanded. β is 0.05

explain the failure. On the other hand, the main difference between the two ETs is that 3.(a) expands the branch $\{theory=average\}$ and (b) does not. It is obvious that a bigger α makes the tree more restrictive. If this tree is expanded, as $\alpha=0.05$ does, is because when *theory* is average it can be interesting to explore what happens with the *practical* part of the subject.

It is possible that variables that are not part of an explanation and that change their 'a priori' usual value or that have an important change in its 'a priori' probability distribution could be added to the explanation as this could be useful to the final user to fully understand some situations. An example can be the case of academe network with $\{theory = good, practice = good\}$. This branch is not expanded. The reason is that in this situation, the other variables have small entropy: *Extra* should be 'no' and *OtherFactors* '-', with high proba-

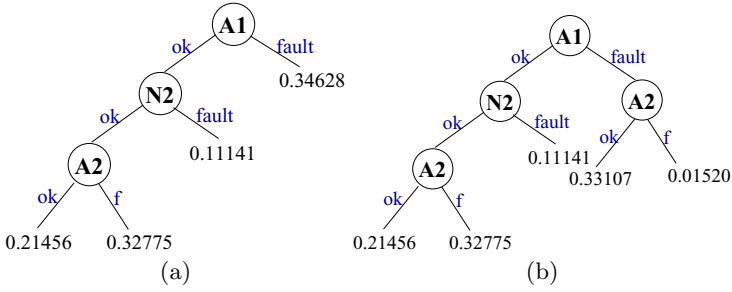


Fig. 5. Results for **gates** and GINI: (a) represents the tree for all gini cases, except (gini, $\alpha=0.05,\max$) which produces tree in part (b). β is 0.05

bility. This implies an important change with respect to 'a priori' probabilities for these values, and then these variables with their respective values could be added to the explanation $\{theory = good, practice = good\}$, making its meaning more evident.

We also used this case to show the influence of β . As $\beta = 0.0$ was used, we can see that some branches represent explanations with a very low posterior probability (those in the dashed area in Fig. 3), and so they will not be useful. The dashed areas in Fig. 3 represent the parts of the tree that are not constructed if we use $\beta \simeq 0.05$, which apart of producing a simpler and more understandable tree is also of advantage to reduce the computational effort (probabilistic propagations) required to construct the tree.

With respect to the resulting trees for the **gates** case, we can appreciate two clear differences: (1) GINI produces simpler trees than MI, and (2) the most explicative variable is different depending on the used measure. Regarding this last situation, we can observe in the circuit that there are many independent causes² (faults) that can account for the erroneous output. Choosing the AND gate A1 as GINI does is reasonable (as well as choosing A2) because AND gates have (in our network) greater a priori fault probability. On the other hand, choosing N2 as MI does is also reasonable (and perhaps closer to human behaviour) because its physical proximity to the wrong output. If we were a technician this would probably be the first gate to test. In this way, it seems that MI manages in some way the fact that the impact a node has in the value of the remaining nodes is attenuated with the distance in the graph.

Once the first variable has been decided, the algorithm tries to grow the branches until they constitute a good explanation. In some cases, it seems that some branches could be stopped early (i.e. once we know that $N2 = \text{fault}$), but these situations depend on the thresholds used and it is clear that studying how to fix them is one of the major research lines for this work.

² However, it is interesting to observe that applying probability propagation, the posterior probability of each gate given the evidence, e.g. $P(A1|x_O)$, indicates that that for all the gates it is more probable to be ok.

Perhaps an interesting point is to think about why O1 is not selected by MI when N2=ok as could be expected given the distance-based preference previously noticed. But, if we look carefully the circuit, we can see that output L (which is correct) also receives as input the output of gate O1, so it is quite probable that O1 is working properly.

Of course, we get different explanations depending on the used measure, the value of α or the criterion, but in general we can say that all the generated explanations are quite reasonable. Finally, in all the trees there is a branch, and so an explanation which indicates that a set of gates are ok. Perhaps this cannot be understood as an explanation to a fault, but we leave it in the tree in order to provide a full partitioning. Some advice about these explanations can be given to the user by indicating for example if such explanations raise or not the probability of the fault with respect to its prior probability.

4 Conclusions and Further Work

This paper has proposed a procedure providing explanations at different level of complexity for the same evidence. The method gives a partition of the different possible scenarios for the explanation variables. The partition can have different levels of granularity depending on the values of the some variables.

We have shown that the results are reasonable in some simple examples and that computations are feasible: though they involve several probabilistic propagations, they are carried out in any junction tree associated to the original Bayesian network, without any restriction. The complexity can be controlled with two parameters (α and β) which at the same time will determine the level of detail of the provided explanations. In fact, the number of explanations (number of leaves in the explanation tree) is bounded by $O(1/\beta)$. Also the expansion of each node of the explanation tree can involve a quadratic number (with respect to the size of the explanation set) of probabilistic propagations, but these are partial propagations and usually we need far less computations than in a complete propagation.

We are conscious that this is an initial step and that additional work is necessary. In the future, we plan to test different criteria to select the variable to branch and to stop branching, specially in the last point where we aim to integrate the two parameters into a single one. Also, we want to make experiments with large Bayesian networks and refine the algorithms to improve its performance. We are studying different ways in which the results can be presented to the user: for example it is possible that variables that are not part of an explanation and that change their usual value (without evidence) could be added to the explanation, as this can be useful to the final user. Finally, for the evaluation of the different procedures it would be necessary a set of experiments in which final users rank the solutions according to their degree of satisfaction with them.

Acknowledgements. This work has been supported by FEDER and Spanish MCYT and MEC: TIC2001-2973-CO5-{01,05} and TIN2004-06204-C03-{02,03}.

References

1. E. Castillo, J.M. Gutiérrez, and A.S. Hadi. *Expert Systems and Probabilistic Network Models*. Springer-Verlag, 1997.
2. U. Chajewska and J. Y. Halpern. Defining explanation in probabilistic systems. In *Proc. of 13th Conf. on Uncertainty in Artificial Intelligence*, pages 62–71. 1997.
3. A.P. Dawid. Applications of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, 2:25–36, 1992.
4. L.M. de Campos, J.A. Gámez, and S. Moral. Simplifying explanations in Bayesian belief networks. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, 9:461–489, 2001.
5. L.M. de Campos, J.A. Gámez, and S. Moral. On the problem of performing exact partial abductive inference in Bayesian belief networks using junction trees. In B. Bouchon, J. Gutierrez, L. Magdalena, and R.R. Yager, editors, *Technologies for Constructing Intelligent Systems 2: Tools*, pages 289–302. Springer Verlag, 2002.
6. L.M. de Campos, J.A. Gámez, and S. Moral. Partial abductive inference in Bayesian networks by using probability trees. In O. Camp, J. Filipe, S. Hammoudi, and M. Piattini, editors, *Enterprise Information Systems V*, pages 146–154. Kluwer Academic Publishers, 2004.
7. R. Dechter. Bucket elimination: A unifying framework for probabilistic inference. In *Proc. of the 12th Conf. on Uncertainty in Artificial Intelligence*, pages 211–219, 1996.
8. J. deKleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987.
9. J.A. Gámez. Abductive inference in Bayesian networks: A review. In J.A. Gámez, S. Moral, and A. Salmerón, editors, *Advances in Bayesian Networks*, pages 101–120. Springer Verlag, 2004.
10. M. Henrion and M.J. Druzdzel. Qualitative propagation and scenario-based schemes for explaining probabilistic reasoning. In P.P. Bonissone, M. Henrion, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 6*, pages 17–32. Elsevier Science, 1991.
11. F.V. Jensen. *Bayesian Networks and Decision Graphs*. Springer Verlag, 2001.
12. C. Lacave and F.J. Díez. A review of explanation methods for Bayesian networks. *The Knowledge Engineering Review*, 17:107–127, 2002.
13. Z. Li and D'Ambrosio B. An efficient approach for finding the MPE in belief networks. In *Proc. of the 9th Conf. on Uncertainty in Artificial Intelligence*, pages 342–349. 1993.
14. R. E. Neapolitan. *Probabilistic Reasoning in Expert Systems. Theory and Algorithms*. Wiley Interscience, 1990.
15. D. Nilsson. An efficient algorithm for finding the M most probable configurations in Bayesian networks. *Statistics and Computing*, 9:159–173, 1998.
16. J.D. Park and A. Darwiche. Solving map exactly using systematic search. In *Proc. of the 19th Conf. in Uncertainty in Artificial Intelligence*, pages 459–468, 2003.
17. J.D. Park and A. Darwiche. Complexity results and approximation strategies for map explanations. *Journal of Artificial Intelligence Research*, 21:101–133, 2004.
18. J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.
19. A. Salmerón, A. Cano and S. Moral. Importance sampling in Bayesian networks using probability trees. *Computational Statistics and Data Analysis*, 34:387–413, 2000.

20. P.P. Shenoy. Binary join trees for computing marginals in the shenoy-shafer architecture. *International Journal of Approximate Reasoning*, 17(2-3):239–263, 1997.
21. S.E. Shimony. Explanation, irrelevance and statistical independence. In *Proc. of the National Conf. in Artificial Intelligence*, pages 482–487, 1991.
22. S.E. Shimony. The role of relevance in explanation I: Irrelevance as statistical independence. *International Journal of Approximate Reasoning*, 8:281–324, 1993.
23. S.E. Shimony. Finding maps for belief networks is NP-hard. *Artificial Intelligence*, 68:399–410, 1994.

Alert Systems for Production Plants: A Methodology Based on Conflict Analysis

Thomas D. Nielsen and Finn V. Jensen

Department of Computer Science, Aalborg University,
Fredrik Bajers vej 7E, 9220 Aalborg Ø, Denmark
{tdn, fvj}@cs.aau.dk

Abstract. We present a new methodology for detecting faults and abnormal behavior in production plants. The methodology stems from a joint project with a Danish energy consortium. During the course of the project we encountered several problems that we believe are common for projects of this type. Most notably, there was a lack of both knowledge and data concerning possible faults, and it therefore turned out to be infeasible to learn/construct a standard classification model for doing fault detection. As an alternative we propose a method for doing on-line fault detection using only a model of normal system operation, i.e., it does not rely on information about the possible faults. We illustrate the proposed method using real-world data from a coal driven power plant as well as simulated data from an oil production facility.

1 Introduction

Most production plants are equipped with sensors providing information to a control room where operators monitor the production process. Based on skill and experience the operators are alerted if something unusual happens, and through inspection of sensor readings, or derivatives thereof (so-called soft sensors), a diagnostic process may be initiated.

In connection to a joint project with an energy consortium, we have been working on establishing an alert system for a coal driven power plant. By an alert system we mean a system that, based on sensor readings, raises a flag in case of an abnormal situation. We intended to base the system on a Bayesian network representation [15, 10] of the power plant, and to help establish the model we had access to process engineers and an extensive database of logged sensor data. However, during the course of the project we encountered several problems, which we believe are common for projects of this type:

1. The engineers' knowledge of the plant is not sufficient for providing a causal structure.
2. The production process is so complex that it is difficult for the engineers to specify the possible faults (abnormal situations) and, in particular, how these faults would manifest themselves in the sensor readings.

3. The time constants, describing the delay from event to effect, are difficult to determine.
4. Faults are so rare that statistics cannot be used to learn neither the structure nor the parameters of a model of the faults.
5. As there is a difference between a true value and its sensor reading, true values should appear as hidden variables.

Faced with these problems, one approach would be to get as much causal structure from the engineers as possible and to combine this information with a data driven learning method. Unfortunately, state of the art of structural learning algorithms cannot cope with domains with a massive set of hidden variables. Furthermore, due to the lack of knowledge about the possible faults it is not obvious how such a model should subsequently be used for classifying abnormal behavior.

In this paper we propose an alternative methodology for on-line detection of abnormal behavior in production systems. The method focuses on systems which are prone to the problems described above, and it has the desirable property that it does *not* require information about the possible faults nor a model of abnormal behavior. We illustrate the proposed method using real-world data from the above mentioned power plant as well as simulated data from an oil production facility.

2 The Proposed Methodology

As implied above, it is not obvious how to construct a classifier (encoding the possible faults) for detecting abnormal behavior; neither in the form of a causal model nor in the form of e.g. a Naïve Bayes model [7] or a tree augmented Naïve Bayes model [8].

Instead, we propose to learn a Bayesian network representing normal operation only. At each time step the model is then used to calculate the probability of the set of sensor readings for that time step. This probability is in turn used to evaluate whether the sensor readings are jointly outside the scope of normal operation. That is, the methodology we propose basically consists of two steps: (i) learning a model of the sensors for normal operation, and (ii) using the learned model to monitor the system, initiate alerts and perform on-line diagnostics. Note that the use of models for describing normal operation has also been explored in the model-based diagnosis community [6]: Based on a pre-specified model of normality (formulated in first-order logic), each component in the system is assigned a state (either normal or abnormal) which is consistent with both the model and any observations made of the system.

2.1 Learning a Model

The available database consists of sensor readings that have been logged during normal system operation; each instance in the database can be seen as a “snapshot” of the overall production process. In what follows we shall assume that

this production process is composed of an ordered collection $(C_1, C_2, \dots, C_n)$ of components (or sub-processes). The output of component C_i serves as input to component C_{i+1} , and (for ease of exposition) each component, C_i , is assumed to be equipped with a single sensor, S_i . For instance, when tracking the coal in a power plant we can, at an abstract level, describe the overall production process as being composed of three components: the silo, the coal mill, and the furnace. Since the production process is a physical non-instantaneous process we also have a delay (or time constant) associated with each of the components C , i.e., the time it takes for a particular unit (e.g. a piece of coal) to pass through that component.

Based on this perspective, we initially considered learning a model of the flow of one unit (e.g. coal) through the production plant. The variables in the learned model would then represent the sensors in the system. One approach for learning such a model would be to first transform the original database s.t. a case in the transformed database would correspond to the sensor readings related to one particular unit (this transformation is illustrated in Table 1). However, making such a transformation requires information about the time constants, and this information was unfortunately not available.

An alternative approach would be to learn a dynamic Bayesian network model directly from the database by treating the cases as representing a trajectory through the system [9, 1]. Unfortunately, learning such a model also requires information about the time constants.

Instead, we simply focused on learning a Bayesian network model over the sensor variables directly from the database. This approach, however, has a potential computational drawback in the sense that we must expect the learned model to be very dense (this was also confirmed in the empirical experiments). To see this, consider Fig. 1 which illustrates a simplified temporal causal model of the data generation process for a production plant. Learning a model for the sensor variables can now conceptually be seen as learning a model that describes

Table 1. The original database is transformed s.t. each case in the resulting database contains the sensor readings related to one particular unit in the system. Note that in the tables below we have assumed that the time delay between sensor S_1 and S_2 corresponds to the sampling delay between case/snapshot c_1 and c_j in the original database

	S_1	S_2	$\dots$	S_n
c_1	x_1^1	x_2^1	$\dots$	x_n^1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_j	x_1^j	x_2^j	$\dots$	x_n^j
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_k	x_1^k	x_2^k	$\dots$	x_n^k
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_N	x_1^N	x_2^N	$\dots$	x_n^N

 $\Rightarrow$

	S_1	S_2	$\dots$	S_n
c_1	x_1^1	x_2^j	$\dots$	x_n^k
c_2	x_1^j	x_2^l	$\dots$	$x_n^{l'}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_k	x_1^k	x_2^m	$\dots$	$x_n^{m'}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$c_{N'}$	$x_1^{N'}$	$x_2^{N'}$	$\dots$	x_n^N

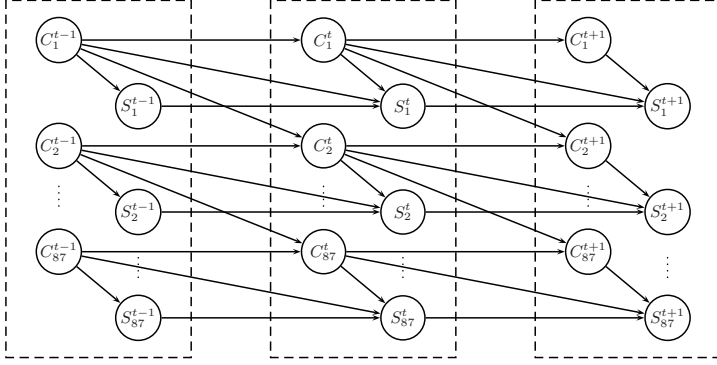


Fig. 1. The figure illustrates a dynamic Bayesian network representation of the data generation process for a production plant. The variable S_i represents the sensor associated with component C_i , and the arcs going into a sensor variable from a previous time slice models that the state of a sensor (correct, faulty or drifting) has an impact on the next sensor reading

the marginal distribution over the sensor variables S_i in a time slice. However, according to Fig. 1 we see that after very few time steps, each pair of variables in a time slice are dependent no matter how we condition on the other variables in the time slice. This is not only due to the hidden variables (modeling the components in the system), but also because standard learning methods treat the cases as being independent [4]; the latter corresponds to the past being unobserved.

2.2 Initiation of Alerts

The sensor readings are received in a constant flow, which is chopped up into time steps of, say, 1 second. This means that for every second we have evidence consisting of a value for each variable in the model.

Let the evidence be $\bar{e} = \{e_1, \dots, e_n\}$, where e_i is a sensor reading. We can now calculate the conflict measure for the evidence as [11]:

$$\text{conf}(\bar{e}) = \log \left[\frac{P(e_1) \cdot \dots \cdot P(e_n)}{P(\bar{e})} \right].$$

The probabilities $P(e_i)$ can be read directly from the Bayesian network in its initial state, and it does not require any propagation. As all variables in the model are instantiated, $P(\bar{e})$ is also very easy to calculate: It is simply the product of the appropriate entries in the conditional probability tables of the Bayesian network, and no propagation is required, i.e., the complexity is linear in the number of variables in the model.

Since the learned model represents normal system operation we would in general expect that sensor readings recorded during normal operation are positively correlated (i.e., $\text{conf}(\bar{e}) \leq 0$) relative to the model. Thus, when $\text{conf}(\bar{e}) > 0$ then this is an indication of an abnormal situation, and an alert may be triggered,

see also [13, 12]. The conflict measure can also be interpreted as a soft measure of inconsistency: If a case is inconsistent with the model, then it has probability 0, and if it is close to being inconsistent then it has an unusual low probability; “unusual” is for this measure calculated relative to the model for complete independence. For the conflict measure above, we expect a rather constant level for $\text{conf}(\cdot)$ under stable normal operation. When the process is changed, and it transforms from one mode of normal operation to another, we should expect oscillations in the conflict values until the changes have propagated and resulted in a new stable mode of normal operation.

As noted above, a positive conflict value is an indication of an abnormal situation. On the other hand, a negative conflict value does not necessarily imply that we have a normal situation as it may hide a serious conflict: If the sensors are strongly correlated during normal operation, the conflict level will be very negative, and a few conflicting sensor readings may therefore not cause the entire conflict to be positive. This can also be seen from the following proposition.

Proposition 1. *Let $\bar{e}^x = \{e_1^x, \dots, e_n^x\}$, $\bar{e}^y = \{e_1^y, \dots, e_m^y\}$, and $\bar{e} = \bar{e}^x \cup \bar{e}^y$. Then*

$$\text{conf}(\bar{e}) = \text{conf}(\bar{e}^x, \bar{e}^y) + \text{conf}(\bar{e}^x) + \text{conf}(\bar{e}^y),$$

where $\text{conf}(\bar{e}^x, \bar{e}^y) = \log \left[\frac{P(\bar{e}^x)P(\bar{e}^y)}{P(\bar{e})} \right]$.

So, it may happen that $\bar{e}^x$ and $\bar{e}^y$ are internally so strongly correlated that they dominate a conflict between the two sets. Thus, even when the conflict is negative, we shall watch out for jumps in the conflict level that may indicate a potential abnormal situation.

When an alert has been triggered, the system can start tracing the source of the alert. Various ways of tracing the conflict may be used. In our case we perform a greedy conflict resolution: recursively remove the sensor reading that reduces the conflict the most, and continue until the conflict is below a pre-defined threshold. This procedure can be performed very fast by exploiting lazy propagation [14] or fast retraction [5], as can be seen from the following proposition.

Proposition 2. *Let $\bar{e}$ be evidence, X a variable with evidence e_x , and $\bar{e}^{-x}$ the remaining evidence. Then*

$$\text{conf}(\bar{e}) = \log \left[\frac{P(e_x)}{P(e_x|\bar{e}^{-x})} \right] + \text{conf}(\bar{e}^{-x}).$$

That is, the reading with lowest normalized likelihood given the other readings contributes the most to the conflict. Note that as the Markov blanket of X is instantiated, the calculation of $P(e_x|\bar{e}^{-x})$ can be performed locally.

3 Empirical Results

The proposed methodology has been tested on real-world data from a coal based power plant as well as simulated data from an oil production facility; in the latter

case the data was generated based on a model that includes the dynamics of the facility as well as control loops.

3.1 Power Plant Data

We received data about the power plant under normal system operation with load average 90 – 100%, i.e., the power plant operated between 90% and 100% of its full capacity. The data set contains 9600 cases, and each case consists of 87 simultaneous observations with no missing values.¹ The cases does not only contain actual sensor values, but they also include soft sensors, i.e., artificial “sensors” that have been computed based on the values of other sensors, as well as set-points and other indirect signals. As a preprocessing step, all data sets were naively discretized using equal width binning, where the number of bins were chosen (based on several tests) to be 3. Based on the preprocessed data, we learned a Bayesian network model as described in Section 2.1; the actual learning was performed using the software tool PowerConstructor with a 0.1-threshold for the conditional independence tests [2, 3].² Since the database is complete, the parameters of the model could simply be estimated using frequency counts.

In addition to the data sets for normal system operation, we received three data sets that each contained 1441 cases. Two of the data sets covered actual errors/abnormal situations whereas the last represented an “unusual behavior” that it would be interesting to detect:

- The fall-pipe leading coal into the power plant becomes clogged.
- A temperature sensor becomes faulty.
- A load change (from 60 – 75% to 90 – 100%) occurs while the water concentration is high.

We have tested the proposed methodology by simulating on-line performance using the “clogged fall-pipe” data set as well as the “faulty-sensor” data set. Both tests were performed “blind-folded”, i.e., we first analyzed the data and then, *after* the analysis, we discussed our findings with the domain experts.

A plot of the conflict measures for the “clogged fall-pipe” data set is depicted in Fig. 2. From the plot we see that we have positive conflict measures from observation 1136 and forward, i.e., the conflict measures indicate that the system makes a transition from a normal to an abnormal system state at 1136. This is also consistent with the information provided to us, namely that the system entered an abnormal state (the fall-pipe became clogged) between 1100 and 1144. Another interesting aspect of the plot is the fluctuations in the conflict measure that appears around observation 700 and lasts until approximately 780. We were later told that in this interval the system actually made a short change in load average from 99% to 84% and then back again.

¹ Since each case contains sensors readings for a particular point in time, the database can also be interpreted as a sequence of “snapshots” of the plant.

² The structure of the learned model is not included in this paper, since it is only used as a factorization of the joint probability distribution and should not be subject to interpretation from e.g. a causal point of view.

When performing conflict resolution, the algorithm indicates that the sensor measuring the water-percentage in the coal can explain all the conflicts. Ideally, we would have liked the system to pinpoint that the fall-pipe is clogged, however, this would require a sensor placed at that location. Since the system does not include such a sensor, we interpret the result as indicating that there is an inconsistency in the energy balance of the system and that this inconsistency can best be explained by the water percentage in the coal; this was also consistent with the analysis by the engineers.

A similar test was made on the “faulty sensor” data set, where the conflict measures can be seen in Fig. 3. As suggested by the plot, the conflict measure indicates that the system entered the abnormal state prior to the first observation; this was later confirmed by the engineers. We were also informed that in the beginning of the data set and around observation 600, there were two quick changes

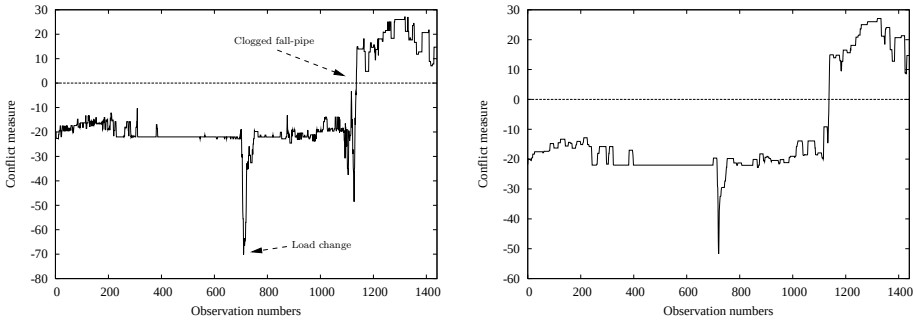


Fig. 2. The left hand figure shows a plot of the conflict measure for each case in the “clogged fall-pipe” data set; a value above 0 indicates a conflict. Note how the conflict measure is affected by the load-change and the fall-pipe becoming clogged. To reduce the noise in the data, the right hand figure shows the 0.9 percentile of the last 30 cases

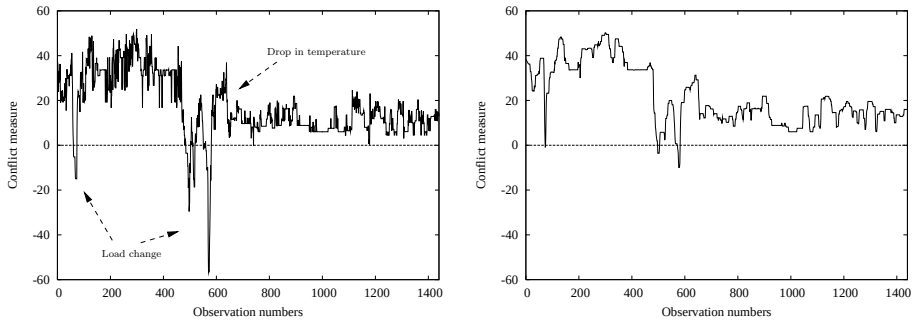


Fig. 3. A plot of the conflict measure for each case in the “faulty-sensor” data set; a value above 0 indicates a conflict. Note how the conflict measure is affected by the load-changes and the drop in temperature. The right hand figure shows the 0.9 percentile of the last 30 cases

in the load averages (from 90 – 100% to 80% and back again); these changes are reflected as quick changes in the calculated conflict measures. Finally, we were told that around observation 600 the temperature drops from 100°C to 90°C (at which level it stays for the remaining observations). Observe, that around this observation we also see a permanent drop in the conflict measure.

When performing conflict resolution we found that after observation 600, there were six significant sensors that could explain the conflict. We were informed that four of the sensors were actually significant for this scenario, but that the other two “sensors” should not have been picked out since they were set-points rather than sensors. However, the identification of these sensors actually makes sense as there is a conflict between the system sensors and the set-points. A simple approach for solving this problem could be to take such prior knowledge into account during conflict resolution.

Finally, we have made a tentative analysis of the “load-change” data set. A difficulty with this data set is that the learned model only covers normal operation during load average 90 – 100%. Hence, we have only considered the observations made after the load change has been completed, and where the distinguishing characteristics of the data set is that the coal has a high water concentration. That is, the data set has *not* been produced from a system state which should be classified as being abnormal, but rather an unusual system state that it would be interesting to detect (in case it would eventually result in an abnormal state). Fig. 4 shows a plot of the conflicts after observation 550 where the load change has been completed.

As can be seen from the figure, the conflict values are all below 0 (except for a few single cases). This is consistent with the system not being in an abnormal state. However, from the measurements we can also see that the average conflict value is higher than for normal operation: For the “load-change” data set, the average conflict value is -7.44 , but during normal operation in the “clogged-fall-pipe” data set the average conflict value is between -10.34 and -22.8 with an average of -19.96 . That is, you may be able to discriminate be-

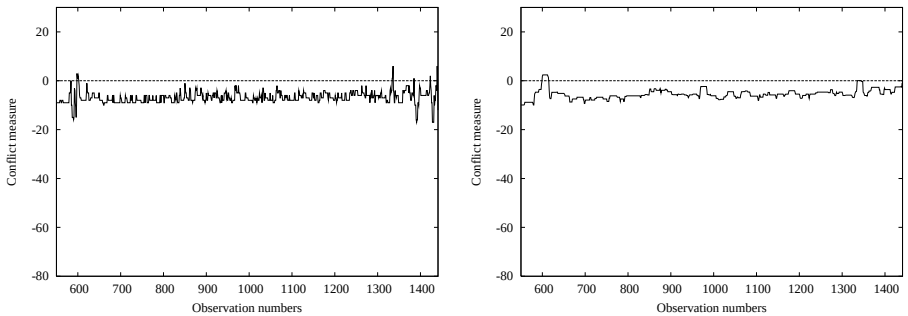


Fig. 4. A plot of the conflict measure for the “load-change” data set after the change has taken effect. The system is correctly classified as *not* being in an abnormal state. The right hand figure shows the 0.9 percentile of the last 30 cases

tween different types of normal system operation by also considering the value of the conflict measure and not only whether it is positive or negative.

3.2 Oil Production Data

We have received a database with 10000 simulated cases for normal system operation for an oil production facility; each case in the database covers 140 sensors with white noise added to the sensor values.³ The database was generated from a temporal causal model, which also simulated standard process variations. Hence, the database shares the same characteristics w.r.t learning as the power plant database (see Section 2.1). All of the sensor values appeared as real-valued output, so as a preprocessing step all variables/sensors were discretized. The actual discretization was performed using cross-validation to find the number of bins (with a maximum of 5) that maximizes the estimated likelihood of the data; the actual discretization was performed using Weka [16].

In order to test the proposed methodology in this setting, we used two other data sets both containing 10000 cases. The first data set had been generated by simulating faults in the pumping system whereas the second data set had been generated by simulating faults in the cooling system (see also Table 2).

Table 2. The table summarizes the changes in the production process for the “Pump” data set and the “Cooling” data set, respectively. Note that the changes in the two scenarios are initiated at the same points in time

Time:	“Pump” data set	“Cooling” data set
30	Small leak in the pump	Small external leak in the cooling system
1500	Large leak in the pump	Large external leak in the cooling system
3000	Normal operation	Normal operation
3500	Small degradation of motor efficiency	Small internal leak in the cooling system
5000	Large degradation of motor efficiency	Large internal leak in the cooling system
6500	Normal operation	Normal operation
7000	Small degradation of pump efficiency	Moderate fouling
8500	Large degradation of pump efficiency	Significant fouling

A plot of the conflict measure for the “Pump” data set is depicted in Fig. 5(a); as in the previous section, Fig. 5(b) shows the 0.9-percentile over the last 30 cases. The vertical lines in the two plots correspond to the points in time where changes are initiated (see Table 2). As can be seen from Fig. 5, there are significant changes in the conflict measure at time 1500, 3000, 5000, 6500 and 8500, which either correspond to large errors in system operation or changes back to normal system operation. From Table 2 we see that the changes appearing at 30,

³ Similar to the power plant database, the database can be interpreted as a sequence of “snapshots” of the facility.

3500 and 7000 correspond to small errors in the system operation and, accordingly, they are also less apparent in the plots. In particular, the change which appears at 3500 occurs before the system has settled into stationary normal system operation.

A similar plot of the conflict measure for the “Cooling” data set is depicted in Fig. 6(a). Analogously to the previous data set, there is a significant change in the conflict measure for all errors except at time 30, 3500 and 7000.

Observe that the conflict measures for both databases are all negative, which is a consequence of the decomposition property (Proposition 1) as discussed in Section 2.2. Thus, in order to detect changes in system operation we need to track jumps in the conflict measure. However, a method for performing this analysis is a subject for future research.

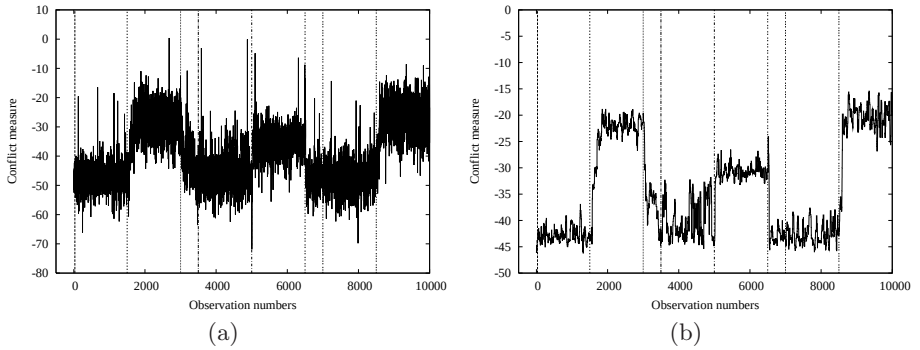


Fig. 5. The left hand figure shows a plot of the conflict measure for each case in the “Pump” data set. The vertical lines indicates when a change in the production process is initiated as specified in Table 2. The figure to the right shows the 0.9 percentile of the last 30 cases

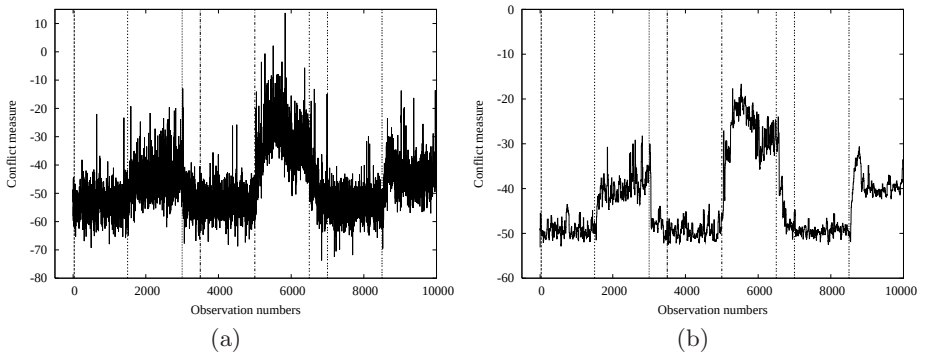


Fig. 6. The figure to the left shows a plot of the conflict measure for each case in the “Cooling” data set. The vertical lines indicates when a change in the production process is initiated as specified in Table 2. The right hand figure shows the 0.9 percentile of the last 30 cases

4 Conclusion and Future Work

We have proposed an alert system methodology based on conflict analysis. A distinguishing characteristic of the proposed methodology is that it only relies on a model for normal system operation, i.e., knowledge about the possible faults is not required. Moreover, the computational complexity of the algorithm ensures that on-line analysis is feasible. The methodology has been successfully tested on both real-world data from a power plant and simulated data from an oil production facility.

As part of ongoing research and future work, we are working on establishing alternative straw models in order to perform a more refined conflict analysis; see also the discussion in [13, 12] concerning the independence straw model [11]. Having an alternative straw model might also reduce the effect of the decomposition property. I.e., when faulty sensors' impact on the conflict measure is dominated by strongly correlated sensors. Furthermore, we are considering procedures for tracking changes in the actual value of the conflict measure in order to perform early fault detection by identifying trends in the behavior of the system being monitored, e.g. if the system "drifts" towards an abnormal state.

Acknowledgments

We would like to thank Rasmus Madsen and Babak Mataji from ELSAM engineering for providing us with data from the power plant. We would also like to thank John-Morten Godhavn from Statoil ASA for supplying us with data from the oil production facility, and Erling Lunde from Dynamica AS for helpful comments regarding the technical layout of the facility. Finally, we would like to thank Helge Langseth for valuable discussions and comments, and **Hugin Expert** (www.hugin.com) for giving us access to the *Hugin Decision Engine* that forms the basis of our implementation.

References

1. Xavier Boyen, Nir Friedman, and Daphne Koller. Discovering the hidden structure of complex dynamic systems. In Kathryn B. Laskey and Henri Prade, editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 91–100. Morgan Kaufmann Publishers, 1999.
2. Jie Cheng, David A. Bell, and Weiru Liu. Learning belief networks from data: An information theory based approach. In *Proceedings of the Sixth ACM International Conference on Information and Knowledge Management*, pages 325–331, 1997.
3. Jie Cheng, Russell Greiner, Jonathan Kelly, David Bell, and Weiru Liu. Learning Bayesian networks from data: an information-theory based approach. *Artificial Intelligence*, 137(1-2):43–90, 2002.
4. Gregory F. Cooper and Edward Herskovits. A Bayesian Method for Constructing Bayesian Belief Networks from Databases. In Bruce D. D'Ambrosio, Philippe Smets, and Piero P. Bonissone, editors, *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 86–94, 1991.

5. A. Philip Dawid. Applications of a general propagation algorithm for a probabilistic expert system. *Statistics and Computing*, 2:25–36, 1992.
6. Johan de Kleer and James Kurien. Fundamentals of model-based diagnosis. In *Proceedings of the fifth IFAC symposium on Fault Detection, Supervision, and Safety of technical Processes (Safeprocess)*, pages 25–36, 2003.
7. Richar O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.
8. Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2–3):131–163, 1997.
9. Nir Friedman, Kevin Murphy, and Stuart Russell. Learning the Structure of Dynamic Probabilistic Networks. In Gregory F. Cooper and Serafin Moral, editors, *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers, 1998.
10. Finn V. Jensen. *Bayesian networks and decision graphs*. Springer-Verlag New York, 2001. ISBN: 0-387-95259-4.
11. Finn V. Jensen, Bo Chamberlain, Torsten Nordahl, and Frank Jensen. Analysis in hugin of data conflict. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 1990. Also published in *Uncertainty in AI 6*, 519–528, North-Holland, Amsterdam, 1991.
12. Young-Gyun Kim and Marco Valtorta. On the detection of conflicts in diagnostic Bayesian networks using abstraction. In Philippe Besnard and Steve Hanks, editors, *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 362–367. Morgan Kaufmann Publishers, 1995.
13. Kathryn Blackmond Laskey. Conflict and surprise: Heuristics for model revision. In Bruce D. D'Ambrosio, Philippe Smets, and Piero P. Bonissone, editors, *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 197–204. Morgan Kaufmann Publishers, 1991.
14. Anders L. Madsen and Finn V. Jensen. Lazy evaluation of symmetric Bayesian decision problems. In Kathryn B. Laskey and Henri Prade, editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 382–390. Morgan Kaufmann Publishers, 1999.
15. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Representation and Reasoning. Morgan Kaufmann Publishers, San Mateo California, 1988. ISBN 0-934613-73-7.
16. Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco, 2000. Version 3.4.3.

Hydrologic Models for Emergency Decision Support Using Bayesian Networks

Martin Molina¹, Raquel Fuentetaja², and Luis Garrote³

¹ Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Spain
mmolina@fi.upm.es

² Departamento de Informática, Universidad de Carlos III, Madrid, Spain
raquel.fuentetaja@uc3m.es

³ Departamento de Ingeniería Civil: Hidráulica y Energética,
Universidad Politécnica de Madrid, Spain
garrote@caminos.upm.es

Abstract. In the presence of a river flood, operators in charge of control must take decisions based on imperfect and incomplete sources of information (e.g., data provided by a limited number sensors) and partial knowledge about the structure and behavior of the river basin. This is a case of reasoning about a complex dynamic system with uncertainty and real-time constraints where bayesian networks can be used to provide an effective support. In this paper we describe a solution with spatio-temporal bayesian networks to be used in a context of emergencies produced by river floods. In the paper we describe first a set of types of causal relations for hydrologic processes with spatial and temporal references to represent the dynamics of the river basin. Then we describe how this was included in a computer system called SAIDA to provide assistance to operators in charge of control in a river basin. Finally the paper shows experimental results about the performance of the model.

1 Introduction

The SAIH National Programme (Spanish acronym for Automatic System Information in Hydrology) has been developed in Spain with the goal of installing sensor devices and telecommunications networks in the main river basins to get on real time in a control center the information on rainfall, water levels and flows in river channels. One of the main tasks of this type of control centers is to help to react in the presence emergency situations as a consequence of river floods.

During a river flood, operators in charge of control use knowledge about the physical system and hydrologic processes of the river basin to estimate future states and make decisions about defensive actions. The exact details about the physical system and behavior are normally difficult to know and therefore certain simplifications are made in order to provide quick and efficient decisions in the presence of problems. Operators use their experience trying to identify similar situations, either measured in past events or simulated with models, in order to forecast similar outcomes.

This is a case of reasoning about the behavior of a complex dynamic system (the whole river basin) with uncertainty and real-time constraints using data recorded by a limited number of imperfect sensors. To help operators in this task with automatic tools, a solution based on traditional mathematical models with deterministic simulation (e.g. [1] [2]) cannot be directly applied. The probabilistic nature of the rainfall forecast, the uncertainty on model parameters, the noise of sensor measures and the discrepancy between model results and observations are difficult to incorporate into a decision-support system that uses deterministic simulation models, especially if the problem area is composed of many small basins that need to be monitored simultaneously for flash flood warning. In addition to that, numerical forecasts obtained via deterministic simulation models do not include an assessment of their accuracy, so it is a task for decision makers to assign degrees of credibility to the values based on their experience in the operation with the models.

As an alternative approach, we describe in this paper a solution where hydrologic models are formulated as bayesian networks. Bayesian networks can be appropriate to model the intuitive understanding of physical hydrologic processes with an explicit representation of this uncertainty together with a natural representation corresponding to the causal relations typically present in river basins. Based on this approach, we have developed a computer system called SAIDA to provide assistance in making decisions about hydraulic actions during floods.

In this paper we describe first the type of bayesian networks that we have considered with spatial and temporal references to model different hydrological processes. Then we describe how they are integrated and used in the SAIDA tool to help operators in decision-making during floods. Finally we show experimental results corresponding to the evaluation of the performance of the bayesian model.

2 Modeling Hydrologic Processes as Spatio-Temporal Bayesian Networks

In order to provide an acceptable level of decision support in a real-time context we have designed a model considering the different meaningful hydrologic variables associated to the physical processes of a river basin. For each process, one or several types of causal relations have been identified that constitute the basic pieces for the complete bayesian model. Each variable X_t^i corresponds to a state (rain, flow, volume, potential damage, etc.) at location i in time t . In the model, time is divided into intervals of fixed duration Δt (for example, $\Delta t=1h$ according to the time interval of the data collection network). The current time interval is identified as time t and past intervals are referred to as $t-1$, $t-2$, etc.

As a result of an experimental analysis of physical influences, the general format of the causal relations that we have considered to estimate the value of a physical variable X_t^i with the upstream variable X_t^j is $P(X_t^i | X_{t-1}^i, X_t^j, X_{t-1}^j, \dots, X_{t-k}^j)$ (more than one upstream variable can be considered). This type of relation can be used together with a conditional probability that relates X_t^i with the observation E_t^i corresponding to gauge stations in the river basin: $P(E_t^i | X_t^i)$. This relation is especially useful when the hydrologic variable cannot directly measured by a gauge station as it happens for example with raingages.

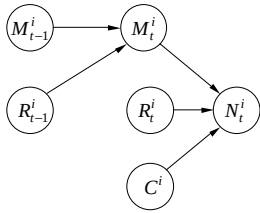
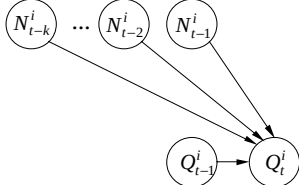
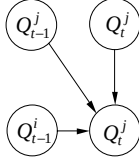
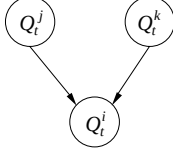
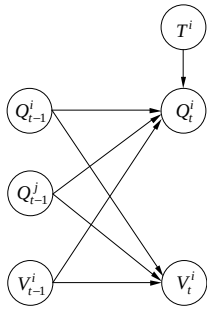
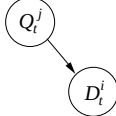
Process	Causal relations	Partial network
<i>Runoff generation</i>	$P(N_t^i R_t^i, M_t^i, C^i),$ $P(M_t^i R_{t-1}^i, M_{t-1}^i)$	
<i>Runoff concentration</i>	$P(Q_t^i Q_{t-1}^i, N_{t-1}^i, \dots, N_{t-k}^i)$	
<i>Discharge propagation</i>	$P(Q_t^j Q_{t-1}^j, Q_b^j, Q_{t-1}^i)$	
<i>River junction</i>	$P(Q_t^i Q_t^j, Q_t^k)$	
<i>Reservoir operation</i>	$P(V_t^i V_{t-1}^i, Q_{t-1}^i, Q_{t-1}^j),$ $P(Q_t^i Q_{t-1}^i, V_{t-1}^i, T^i, Q_{t-1}^j)$	
<i>Potential damages</i>	$P(D_t^i Q_t^j)$	

Fig. 1. Examples of basic types of causal relations for hydrological processes

Six basic processes have been considered: (1) runoff generation, (2) runoff concentration, (3) discharge propagation, (4) river junction, (5) reservoir operation and (6) potential damages. The first three processes resemble the equations applied in conventional lumped rainfall-runoff modeling (Hortonian infiltration, linear response to rainfall excess and hydrologic flood routing). The *runoff generation* process represents the causal influence between rainfall and net rainfall and includes two basic relations to estimate basin moisture content and infiltration. Three variables are included in the infiltration model: basin average rainfall R_t^i , cumulative basin moisture content M_t^i and average net rainfall N_t^i , all of them corresponding to the current temporal interval t and the spatial location i . Each variable is formulated in a qualitative domain composed of a finite set of discrete values relevant for decision support purposes. For instance, rainfall during a time step may have the following discrete set of significant values $\{0, 3, 6, 9, 12, 20, 30, 50, >50\}$, all of them expressed in mm. Two additional variables are required for the basin moisture model: basin moisture content M_{t-1}^i and basin average rainfall R_{t-1}^i in the previous time interval $t-1$. According to this model, runoff generation is assumed to be Hortonian, and net rainfall N_t^i is directly explained by rainfall intensity R_t^i and basin moisture content M_t^i . In its turn, cumulative moisture content M_t^i is directly explained by moisture in the previous time interval, M_{t-1}^i , and rainfall in the previous time interval R_{t-1}^i . Initially, these causal relations were formulated as $P(N_t^i | R_t^i, M_t^i)$ and $P(M_t^i | R_{t-1}^i, M_{t-1}^i)$.

The results of model calibration of this model showed a lot of variability that was attributed to the basin initial condition. If the bayesian network is built using the full range of inter-storm curve number variability, the dispersion in the result is so large that many forecasts show flat probability distributions. In this case, instead the first relation of the the previous bayesian model, an alternative causal relation was used for N_t^i in the form of $P(N_t^i | R_t^i, M_t^i, C^i)$ which includes as additional cause the variable C^i (SCS curve number). In real time, the bayesian model uses explicitly an estimate of this parameter as input, which can be provided by the operator using knowledge of initial conditions with the help of a simulation model. In this model, conditional independence between R_{t-1}^i and R_t^i is assumed considering that the time interval Δt is large enough (e.g., one hour). This assumption is based on empirical studies of the behavior of torrential rain that presents low persistency and consequently low level of correlation between consecutives values of rain.

The *runoff concentration* represents the response to rainfall excess. In this case, the variables are $N_{t-1}^i, \dots$ and N_{t-k}^i which correspond to net rainfall for k previous time intervals and Q_t^i the average discharge (in m^3/s) in the current time interval. The number of temporal intervals of net rainfall (k) is chosen balancing the need to represent the length of the unit hydrograph (an hydrological parameter associated to each river basin) and the need to limit the number of explaining variables to a manageable size. In practice, it should be reduced to three or four intervals. The causal relationship is expressed as $P(Q_t^i | Q_{t-1}^i, N_{t-1}^i, \dots, N_{t-k}^i)$.

Another type of relation corresponds to the process of *discharge propagation*. This model represents the flow transportation from a certain location (spatial location j) to a downstream location (spatial location i), assuming hydrologic routing. The variables are Q_t^i , that corresponds to the flow at location i , and Q_t^j that corresponds to the flow at location j . The causal relationship is expressed as $P(Q_t^i | Q_{t-1}^i, Q_t^j, Q_{t-1}^j)$. For a *river*

junction another dependency can be established as $P(Q_t^i | Q_t^j, Q_t^k)$ where Q^j and Q^k are upstream flows of the flow Q^i .

Reservoir operation is included in the model as an additional set of causal relations. A model of reservoir behavior was formulated with the following variables: Q_t^j inflow discharge, V_t^i stored volume, T^i target volume and Q_t^i outflow discharge. The bayesian network includes two types of conditional probabilities for causal relations $P(V_t^i | V_{t-1}^i, Q_{t-1}^j, Q_{t-1}^i)$ and $P(Q_t^i | Q_{t-1}^i, V_{t-1}^i, T^i, Q_{t-1}^j)$. Note that this model uses the decision variable T^i (target volume) that describes the management strategy expressed as the desired volume in the reservoir.

Another application in this type of models is the interpretation of the prediction in terms of *potential damages*. For this purpose, additional relations were included to interpret the hydrologic values. There are two variables for each location with potential flood problems in the river basin: Q_t^j flow at location j and D_t^i damage level at location i . The values of D_t^i represent levels of problems with qualitative values such as normal, material damages, severe material damages, personal damages, severe personal damages, etc. The interpretation of the flow values in terms of problem levels is expressed by the conditional probability $P(D_t^i | Q_t^j)$.

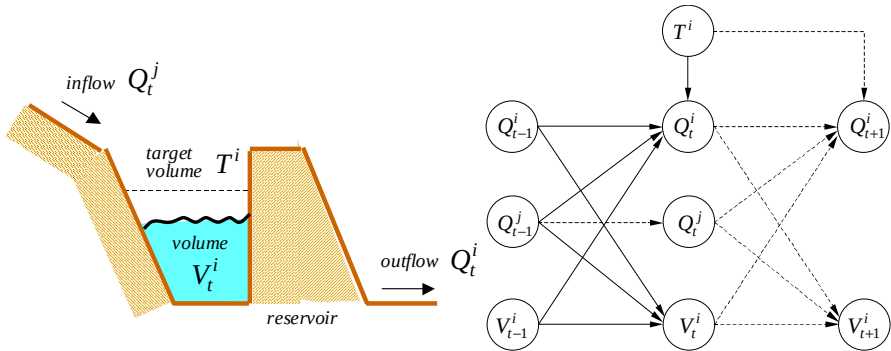


Fig. 2. Example of temporal extension for the bayesian network of the reservoir operation

In the context of prediction for decision support, it is normally required to make a forecast for several consecutive time steps. In order to perform this process, besides the spatial references of nodes corresponding to the specific locations of physical variables, a temporal extension is required as it is done in dynamic bayesian networks [3] [4]. For this purpose, the elementary bayesian network for each physical process is considered with additional nodes and causal relations corresponding to consecutive timeslices. Figure 2 shows this idea for the case of the reservoir operation. In dynamic bayesian networks, the first order Markov property indicates that the parents of a variable in timeslice t must occur in either slice t or $t-1$. This is a property that is not always satisfied by the hydrologic processes presented here¹. For example, in the runoff concentration we have identified the causal relation $P(Q_t^i | Q_{t-1}^i, N_{t-1}^i, \dots, N_{t-k}^i)$ (in the particular model for Guadalhorce river $k=3$).

¹ Nevertheless, variables can be transformed to satisfy this property as it is described by [5].

3 Operation with the SAIDA Application

SAIDA is a computer system that was developed in a three-year project during 1998-2000 and promoted by the Spanish Ministry of the Environment with the purpose of operating in connection with the information hydrologic systems in several Spanish basins (details about the operation and the complete software architecture of SAIDA can be found at [6] [7] [8]). SAIDA receives as input the available data provided by sensors about discharge, water level and rainfall at different locations in the river basin. SAIDA provides answers that evaluate the current situation, predict a short term evolution and recommend control actions. The answers are produced with time constraints and the conclusions are justified at a reasonable level of abstraction given that the operator must take the final responsibility of decisions.

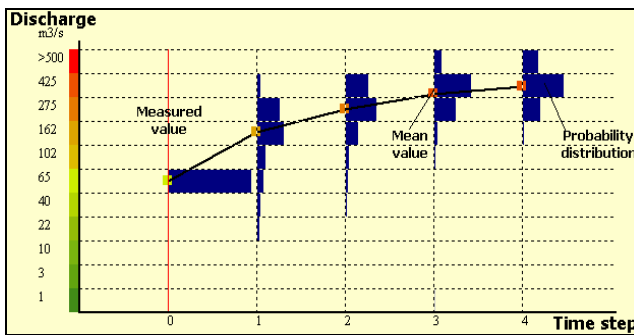


Fig. 3. Example of presentation of the predicted values for a variable in consecutive time steps

The bayesian approach was applied for the development of models for prediction as part of the SAIDA system. In this context, SAIDA receives as input: (1) *values recorded by sensors* about past and recent rainfall in different areas, current discharge at significant locations and water level in reservoirs, and (2) *hypotheses of future behavior*, i.e. the operator makes hypotheses of values for significant cause variables like future rain based on global meteorological information, future discharge policy of reservoirs (target volume), basin condition expressed in terms of model parameter values (e.g., curve number), etc.

SAIDA uses the model of bayesian networks to determine values about future volumes stored in reservoirs and flows at certain locations. The model also provides information about potential damages in areas at risk. All these values are expressed as probability distributions showing a range of potential behaviors according to model uncertainty. Figure 3 shows an example of how SAIDA shows the future evolution of a variable at certain location. The graphical representation shows the probability distribution associated to each time step. The mean values are explicitly connected to show the temporal trend of the variable. This graphical representation is a synthetic image that covers a wide range of potential behaviors for a particular variable taking into account the uncertainty of different processes.

In order to perform total predictions for the whole river basin the local bayesian models are connected and linked to the real-time hydrologic information network. Individual bayesian models are combined in a larger network, that connects the set of variables according to river basin topology. Inference is carried out with an adaptation of a general inference algorithm for multiply connected networks [9]. SAIDA shows a complete view about the causal relations in a global image (Figure 4). This global view corresponds to a summarized view of the instantiation of the type of bayesian networks described in the previous section for a particular river (e.g. the Guadalhorce River in Málaga). The model for a particular river basin is built by linking together several instances of the bayesian networks according to the topology of the river basin. Each specific bayesian network for a particular physical process at certain location presents differences (e.g., discrete values and conditional probabilities) compared to another network for the same process at a different location.

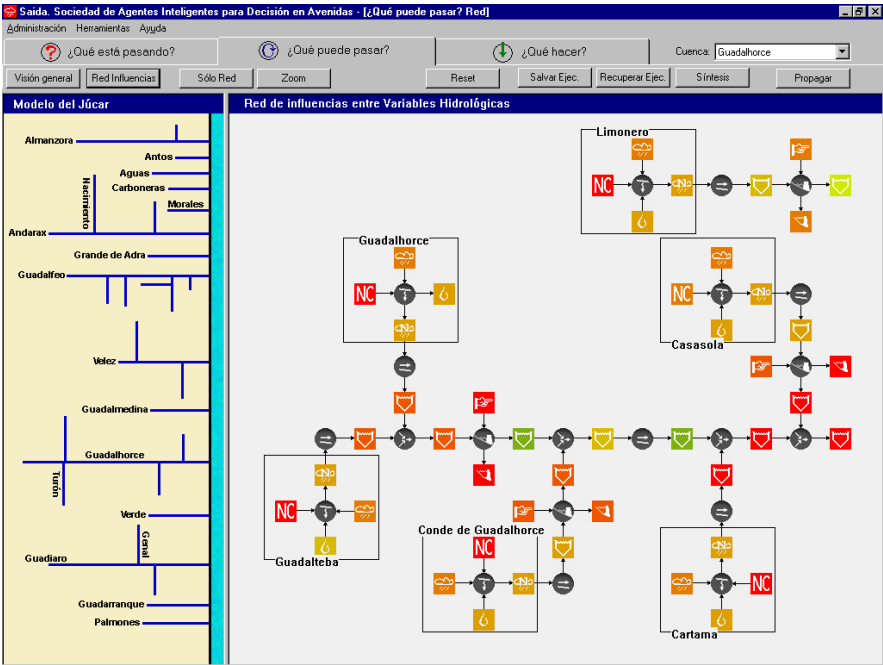


Fig. 4. Interactive analysis tool for hydrologic prediction provided by the SAIDA user interface with bayesian networks

The window of figure 4 shows a visualization using a color code for each variable that goes from the lowest value (green) to the highest value (red). Each node of the diagram corresponds to a physical process (runoff generation, reservoir operation, etc.). This provides a global image of the causal explanation of flows at different locations. The operator can consult individually the temporal evolution of input, output and intermediate variables displaying additional windows where the probability distributions for different time steps are presented. This user interface is actually an interactive analysis

tool where the user can also change the values of some of these variables to produce a new prediction. This feature is very useful to analyze different hydrologic scenarios in an appropriate level of abstraction in the presence of problematic situations.

4 Experimental Results

Following the previous approach several models were developed for the control centers located in Valencia and Málaga (Spain). This section describes details of the case of Málaga to show results about an experimental evaluation. Málaga is located in a flash-flood prone area, at the outlet of two rivers, Guadalhorce and Guadalmedina. Contributing areas to the Guadalhorce and Guadalmedina basins are of 3,158 km² and 147 km² respectively. The climate is semiarid, with steep slopes covered by brush at the headwaters and irrigated land at the floodplain. Several reservoirs have been built to regulate the Guadalhorce basin and to protect Malaga from flooding. The *Confederación Hidrográfica del Sur* is the management authority responsible for the operation of the reservoirs during floods.

Physical Process	Spatial location	Causal relations	CE	A
Runoff generation	Guadalhorce	$P(N_t^i R_t^i, M_t^i, C_t^i)$	0.16	87
		$P(M_t^i R_{t-1}^i, M_{t-1}^i)$	0.04	
	Guadalteba	$P(N_t^i R_t^i, M_t^i, C_t^i)$	0.19	87
		$P(M_t^i R_{t-1}^i, M_{t-1}^i)$	0.04	
	Conde de Guadalhorce	$P(N_t^i R_t^i, M_t^i, C_t^i)$	0.15	86
		$P(M_t^i R_{t-1}^i, M_{t-1}^i)$	0.05	
	Casasola	$P(N_t^i R_t^i, M_t^i, C_t^i)$	0.19	87
		$P(M_t^i R_{t-1}^i, M_{t-1}^i)$	0.04	
	Cártama	$P(N_t^i R_t^i, M_t^i, C_t^i)$	0.18	87
		$P(M_t^i R_{t-1}^i, M_{t-1}^i)$	0.04	
	Limonero	$P(N_t^i R_t^i, M_t^i, C_t^i)$	0.17	87
		$P(M_t^i R_{t-1}^i, M_{t-1}^i)$	0.04	
Runoff concentration	Guadalhorce	$P(Q_d^i Q_{t-1}^i, N_{t-1}^i, N_{t-2}^i, N_{t-3}^i)$	0.2	89
	Guadalteba	$P(Q_d^i Q_{t-1}^i, N_{t-1}^i, N_{t-2}^i, N_{t-3}^i)$	0.2	89
	C.Guadalhorce	$P(Q_d^i Q_{t-1}^i, N_{t-1}^i, N_{t-2}^i, N_{t-3}^i)$	0.16	90
	Casasola	$P(Q_d^i Q_{t-1}^i, N_{t-1}^i, N_{t-2}^i, N_{t-3}^i)$	0.22	89
	Cártama	$P(Q_d^i Q_{t-1}^i, N_{t-1}^i, N_{t-2}^i, N_{t-3}^i)$	0.19	90
	Limonero	$P(Q_d^i Q_{t-1}^i, N_{t-1}^i, N_{t-2}^i, N_{t-3}^i)$	0.16	90
Reservoir operation	C.Guadalhorce	$P(Q_t^i Q_{t-1}^i, V_{t-1}^i, T_t^i, Q_{t-1}^i)$	0.54	78
		$P(V_t^i V_{t-1}^i, Q_{t-1}^i, Q_{t-1}^i)$	0.01	99
	Casasola	$P(Q_t^i Q_{t-1}^i, V_{t-1}^i, T_t^i, Q_{t-1}^i)$	0.05	97
		$P(V_t^i V_{t-1}^i, Q_{t-1}^i, Q_{t-1}^i)$	0.04	98
	Limonero	$P(Q_t^i Q_{t-1}^i, V_{t-1}^i, T_t^i, Q_{t-1}^i)$	0.18	91
		$P(V_t^i V_{t-1}^i, Q_{t-1}^i, Q_{t-1}^i)$	0.07	96
River junction	Cartama Guadalhorce	$P(Q_t^i Q_b^i, Q_b^i)$	0.12	93
	Cuadalhorce Campanillas	$P(Q_t^i Q_b^i, Q_b^i)$	0.12	92
	Guadalhorce G., Conde G.	$P(Q_t^i Q_b^i, Q_b^i)$	0.13	92
Discharge prop.	Guadalhorce	$P(Q_d^i Q_{t-1}^i, Q_b^i, Q_{t-1}^i)$	0.45	82

Fig. 5. Experimental results for the model of the South of Spain (CE: conditional entropy, A: accuracy)

Data gathered from an automatic data collection network are analyzed at a control center to provide assistance to decision makers in selecting the best management strategies for reservoir operation and to issue warnings to Civil Defense authorities and to the population. Hydrologic information is received at one hour time intervals from 29 raingages, 5 reservoirs (Guadalhorce, Guadalteba, Conde de Guadalhorce, Casasola and Limonero) and from a gaging station in the Guadalhorce river located near Málaga, in Cártama.

A deterministic simulation model was taken as the basic framework to build the probabilistic decision model. Hydrological knowledge about a river basin is typically encoded in deterministic simulation models. A great deal of expert knowledge and effort is applied in model formulation, discretization and calibration, using information about the basin, field surveying and data from observed events. After the calibration process, the values of model parameters are only partially known, and they are best described by a confidence interval or a probability distribution.

The deterministic model was run with random parameters and forced with a stochastic rainfall simulator, creating a large database of synthetic storms. During the simulations, parameter values were sampled randomly from their estimated probability distributions to obtain an ensemble of basin behaviors consistent with the results of the calibration process. The database of simulated events contains a variety of basin behaviors expressed in numerical values that were converted to the discrete domains of the bayesian network variables. The qualitative time series generated were processed to collect cases as combinations of values for cause variables and the corresponding value for the effect variable.

The resulting models were validated to determine their ability to produce probability distributions that describe accurately the behavior of the deterministic model and are useful for decision making. Two different types of model evaluation were performed: (1) evaluation of bayesian network structure and (2) evaluation of prediction quality. The first type of evaluation was useful to compare different versions of structures of bayesian networks and discrete domains. This evaluation was applied to different versions of bayesian networks that were accordingly refined until a satisfactory version was obtained. In order to evaluate the structure of each bayesian network, the *conditional entropy* was used. *Conditional entropy* H is computed with the following equation [10]:

$$H(X / Y_1, \dots, Y_n) = - \sum_{Y_1=y_1, \dots, Y_n=y_n} P(y_1, \dots, y_n) \sum_{X=x} P(x | y_1, \dots, y_n) \ln P(x | y_1, \dots, y_n)$$

where X represents a node of the network, $Y_1, \dots, Y_n$ is the set of parent nodes of X and $Y_j = y_j$ expresses that the variable Y_j gets the qualitative value y_j . This parameter estimates the disorder of information, so lower values are considered better results.

The evaluation of the network prediction quality was computed with the *accuracy* parameter. The *accuracy* parameter A evaluates the quality of the answers of the bayesian network. This parameter is computed with the formula:

$$A = \frac{\sum_{\forall i} P_i}{N} 100$$

where, i designates a case, P_i is the probability assigned by the bayesian network to the corresponding value of the effect variable of case i , and N is the total number of cases.

Bayesian models were calibrated with a set S1 of about 300,000 cases produced by simulation. Another different set S2 with the same number of cases was generated for evaluation of model performance. The number of cases in these sets was adjusted verifying that all combinations of discrete values for each set of nodes cause in S2 should be also be present in S1. This guarantees that the bayesian network learned from S1 includes all the physically possible situations (this requirement was efficiently verified with the help of a particular data structure for the bayesian network that included the combinations derived from S1).

The evaluation of the bayesian network structure was applied to different versions of structures for bayesian networks with different discrete values that were refined until a satisfactory version was obtained. The resulting final values of evaluation parameters for the case of the Guadalhorce and Guadalmedina basins are shown in figure 5. As shown in the table, all local bayesian networks have a good behavior for the degree of accuracy in accordance to each model level of uncertainty. The resulting values for these parameters after the evaluation process prove that the bayesian networks provide a satisfactory behavior.

5 Conclusions

The approach for hydrologic prediction presented in this paper is a practical solution to be used in a context of real-time decision support. The proposed model is a case of spatio-temporal bayesian network, i.e. a bayesian network where nodes have both spatial and temporal references. This is a solution that facilitates rational decisions in probabilistic terms as it required in the field of hydrology about future states of a river basin [11].

A number of solutions have been proposed to generate probabilistic forecasts using deterministic models [12] [13] [14]. However, these solutions show a high degree of artificial mathematical sophistication that makes that, from a practical point of view, a computer system with this approach operates like a black box. In a decision context, on the contrary, it is very important to use a natural representation model closer to the background of decision makers, in order to build confidence in the results produced by the system. Bayesian networks as presented in this paper provide a natural and intuitive description of hydrologic processes based on a symbolic representation with qualitative variables and causal relations. This is very useful to formulate decision models with high levels of abstraction and explicit meaning. The bayesian representation shows explicitly the uncertainties of the information, which is a novelty, compared to classical deterministic models (e.g. [1] [2]). This feature is useful to show explicitly the degree confidence that the system gives to its own answers. This task is normally performed by operators who give partial credibility to the answers of deterministic simulation models according to their experience with those tools.

Bayesian models can be automatically created using information currently available in flood control centers. For example, these types of models can take advantage

of the knowledge about the river basin encoded in a classical deterministic simulation model but also they can easily take advantage of historical information recorded in control centers (e.g., in Valencia the SAIH infrastructure has been recorded hydrological data of near 20 years). This feature favors the transfer of the technology to the operational stage.

The experimental evaluation of the bayesian networks associated to hydrologic processes with data obtained from the river basins in the South of Spain (Guadalhorce and Guadalmedina) showed a satisfactory performance for prediction. This approach was applied to develop part of a software environment called SAIDA which besides the capability of prediction using bayesian networks includes additional features (identification of problem scenarios, recommendation of hydraulic actions, etc.).

Bayesian networks have also been applied in the field of meteorology [15] [16] but, to our knowledge, our approach to model physical processes in the field of hydrology is an original contribution. The success of this development suggests to continue with this work in the following lines: (1) a more extensive use for new river basins in different parts of Spain with additional physical processes (for this purpose the Spanish Ministry of Environment is currently opening a new project), (2) according to the particular type of dynamic bayesian network, alternative inference methods can be applied to gain efficiency, and (3) automatic tools can be designed to facilitate the construction of models (with a suite of software tools for model edition, simulation, and machine learning).

Acknowledgements. The development of the SAIDA system was mainly supported by the Ministry of Environment of Spain (*Dirección General de Obras Hidráulicas y Calidad de las Aguas*) and local public organizations from river basins (*Confederación Hidrográfica del Júcar* and *Confederación Hidrográfica del Sur de España*.) with the collaboration of the private companies SYNCONSULT and PAGESEI. It was also partially supported by the Ministry of Science and Technology of Spain within the RIADA Project.

References

1. Brath, A., Rosso, R.: "Adaptive calibration of a conceptual model for flash flood forecasting". *Water Resources Research*, 29(8) 2561-2572, 1993.
2. Madsen, H.: "Automatic calibration of a conceptual rainfall--runoff model using multiple objectives" *Journal of Hydrology*, (235)3-4, 276-288, 2000.
3. Dean T., Kanazawa K.: "A model for reasoning about persistence and causation". *Computational Intelligence*, 5(3): 142-150. 1989.
4. Z. Ghahramani. "Learning dynamic Bayesian network". In C.L. Giles and M. Gori, editors, *Adaptive Processing of Sequences and Data Structures*, volume 1387 of *Lecture Notes in Computer Science*, pages 168-197. Springer, 1998.
5. Murphy, K. P., "Dynamic Bayesian Networks: Representation". *Inference and Learning*, Ph.D. thesis, UC Berkeley, Computer Science Division, July, 2002.
6. Cuenca, J. Molina, M.: "A Multi-agent System for Emergency Management in Floods", in "Multiple Approaches to Intelligent Systems", Iman I., Kodratoff Y. (eds.). *Lecture Notes in Artificial Intelligence*, Springer, 1999.

7. Molina M., Blasco G.: "A Multi-agent System for Emergency Decision Support". Proceedings of the Fourth International Conference Intelligent Data Engineering and Automated Learning IDEAL 03. LNCS, Springer, 2003.
8. Garrote L., Molina M.: "A Framework for making probabilistic forecast using deterministic rainfall-runoff models". Proceedings of the ESF LESC Exploratory Workshop held at Bologna, Italy, October 24-25, 2003.
9. Lauritzen S. L., Spiegelhalter D. J.: "Local computations with probabilities on graphical structures and their application to expert systems". Journal of the Royal Statistical Society B 50(2): 157-224, 1988.
10. Herskovitz E.H., Cooper G.F.: "Kutató: an entropy-driven system for the construction of probabilistic expert systems from data". Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence, pp 54-62, 1990.
11. Krzysztofowicz R.: "The case for probabilistic forecasting in hydrology". Journal of Hydrology, 249, 2-9, 2001.
12. Georgakakos K. P., Bras R. L.: "A Hydrologically Useful Station Precipitation Model 1. Formulation". Water Resources Research, 20, 1585-1596, 1984.
13. Lardet P., Obled C.: "Real-time flood forecasting using a stochastic rainfall generator". Journal of Hydrology, Volume 162, Issues 3-4, Pages 391-408, November, 1994.
14. Krzysztofowicz R.: "Bayesian theory of probabilistic forecasting via deterministic hydrologic model". Water Resources Research, 35(9) 2739-2750, 1999.
15. Kennett R., Korb K., Nicholson A.: "Seabreeze Prediction Using Bayesian Networks: A Case Study". Proceedings of the 5th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining PAKDD, Springer Verlag, 2001.
16. Cano R., Sordo C, Gutiérrez J.M.: "Applications of Bayesian Networks in Meteorology" In J.A. Gámez, S. Moral and A. Salmerón, eds., Advances in Bayesian Networks, 309-327, 2004 Springer Verlag, 2004.

Probabilistic Graphical Models for the Diagnosis of Analog Electrical Circuits

Christian Borgelt and Rudolf Kruse

School of Computer Science, University of Magdeburg,
Universitätsplatz 2, D-39106 Magdeburg, Germany
{borgelt, kruse}@iws.cs.uni-magdeburg.de

Abstract. We describe an algorithm to build a graphical model—more precisely: a join tree representation of a Markov network—for a steady state analog electrical circuit. This model can be used to do probabilistic diagnosis based on manufacturer supplied information about nominal values of electrical components and their tolerances as well as measurements made on the circuit. Faulty components can be identified by looking for high probabilities for values of characteristic magnitudes that deviate from the nominal values.

1 Introduction

In electrical engineering several approaches to the diagnosis of electrical circuits have been developed [10, 11]. Examples are: the fault dictionary approach, which collects a set of common or relevant faults and associates them with (sets of) measurements by which they can be identified [2], the model-based diagnosis of digital circuits based on constraint propagation and an assumption-based truth maintenance system (ATMS) [8], and the simulation of a circuit for different predefined faults to generate training data for a classifier, for example, an artificial neural network [1, 13]. In particular the diagnosis of digital electrical circuits is well-developed. However, this theory is difficult to transfer to analog circuits due to problems like *soft faults* (i.e. significant deviations from nominal values) and the non-directional behavior of analog circuits.

The existing methods for the diagnosis of analog circuits suffer from several drawbacks, like difficulties to take tolerances of components and measurements into account. In addition, there is often the need for a predefined set of faults, which are common or relevant for the circuit. In this paper we develop a method that is based on a probabilistic description of the state of the circuit with the help of a graphical model, an approach that is able to handle these problems.

This paper is organized as follows: first we review very briefly in Section 2 the ideas underlying graphical models and in Section 3 the basics of iterative proportional fitting, which we need for initialization purposes. Section 4 discusses some core problems of modeling analog electrical networks in order to justify our approach. In Section 5 we describe our algorithm, which is based on the direct construction of a join tree, and illustrate it with a simple example in Section 6. Finally, in Section 7, we draw conclusions and point out future work.

2 Graphical Models

In the last decade graphical models have become one of the most popular tools to structure uncertain knowledge about complex domains [14, 9, 3] in order to make reasoning in such domains feasible [12, 6]. Their most prominent representatives are Bayes networks, which are based on directed graphs and conditional probability distributions, and Markov networks, which are based on undirected graphs and marginal probability distributions or so-called factor potentials.

More formally: let $V = \{A_1, \dots, A_m\}$ be a set of (discrete) random variables. A *Bayes network* is a directed graph $G = (V, E)$ of these random variables together with a set of conditional probability distributions, one for each variable given its parents in the graph. A *Markov network*, on the other hand, is an undirected graph $G = (V, E)$ of the random variables together with a set of functions on the spaces spanned by the variables underlying the maximal cliques¹ of the graph. In both cases the structure of the graph encodes conditional independence statements between (sets of) random variables that hold in the joint probability distribution represented by the graphical model. This encoding is achieved by *node separation* criteria, with Bayes networks relying on *d*-separation [12] and Markov networks employing *u*-separation [4].

Conditional independence of X and Y given Z , written $X \perp\!\!\!\perp Y \mid Z$, means

$$p_{XY|Z}(x, y \mid z) \equiv p_{X|Z}(x \mid z) \cdot p_{Y|Z}(y \mid z),$$

where x , y and z are value vectors from the spaces spanned by the random variables in X , Y , and Z , respectively. For both Bayes networks and Markov networks it can be shown [9] that if the graph encodes only correct conditional independences by *d*- or *u*-separation, respectively, then the joint probability distribution p_V factorizes, namely according to

$$p_V(v) \equiv \prod_{i=1}^m p_{A_i|\text{parents}(A_i)}(v[\{A_i\}] \mid v[\text{parents}(A_i)]).$$

for Bayes networks and according to

$$p_V(v) \equiv \prod_{C \in \mathcal{C}} \phi_C(v[C])$$

for Markov networks. Here v is a value vector over the variables in V and $v[X]$ denotes the projection of v to the variables in the set X . The $p_{A_i|\text{parents}(A_i)}$ are conditional probability distributions of the different variables A_i given their parents in the directed graph G . The set $\mathcal{C}$ is the set of all sets C of variables underlying the maximal cliques of the undirected graph G and the ϕ_C are functions on the spaces spanned by the variables in the sets $C \in \mathcal{C}$. They are called

¹ A clique is a complete (fully connected) subgraph and it is called maximal if it is not contained in another complete subgraph.

factor potentials [6] and can be defined in different ways from the corresponding marginal probability distributions.

For reasoning purposes a Bayes or Markov network is often preprocessed into a singly connected structure to avoid update anomalies and incorrect results, which we discuss in somewhat more detail below. The preprocessing consists in forming the moral graph (for Bayes networks only) by “marrying” all parents of a variable, triangulating the graph² and turning the resulting *hypertree-structured* graph into a join tree [6]. In a join tree there is one node for each maximal clique of the graph it is constructed from. In addition, if a variable (node) of the original graph is contained in two nodes of the join tree, it is also contained in all nodes on the path between these nodes in the join tree. A join tree is usually enhanced by so-called *node separators* on each edge, which contain the intersection of the variables assigned to the connected join tree nodes.

For join trees there exist efficient evidence propagation methods [6] that are based on a message passing scheme, in which the node separators transmit the information between the nodes. In the approach we present below we work directly with join trees and neglect that our model is actually a Markov network.

3 Iterative Proportional Fitting

Iterative proportional fitting (IPF) is a well-known method for adapting the marginal distributions of a given joint probability distribution to desired values [14]. It consists in computing the following sequence of probability distributions:

$$p_V^{(0)}(v) \equiv p_V(v)$$

$$\forall i = 1, 2, \dots : p_V^{(i)}(v) \equiv p_V^{(i-1)}(v) \frac{p_{A_j}^*(a)}{p_{A_j}^{(i-1)}(a)}$$

where j is the $((i - 1) \bmod |J| + 1)$ -th element of J , the index set that indicates the variables for which marginal distributions are given. $p_{A_j}^*$ is the desired marginal probability distribution on the domain of the variable A_j and $p_{A_j}^{(i-1)}$ is the corresponding distribution as it can be computed from $p_V^{(i-1)}$ by summing over the values of all variables in V except A_j .

In each step the probability distribution is modified in such a way that it satisfies one given marginal distribution (namely the distribution $p_{A_j}^*$). However, this will, in general, disturb the marginal for a variable A_k , which has been processed in a preceding step. Therefore the adaptation has to be iterated, traversing the set of variables several times.

It can be shown that if there is a solution, iterative proportional fitting converges to a (uniquely determined) probability distribution that has the desired marginals as well as some other convenient properties [5, 7]. Convergence may

² An undirected graph is called *triangulated* or *chordal* if all cycles of length greater than three have a chord, i.e., an edge between two nodes that are nonadjacent in the cycle.

be checked in practice, for instance, by determining the maximal change of a marginal probability: if this maximal change falls below a user-defined threshold, the iteration is terminated.

Iterative proportional fitting can easily be extended to probability distributions represented by Markov networks or the corresponding join trees [7]. The idea of this extension is to assign each variable, the marginal distribution of which is to be set, to a maximal clique of the Markov network (or to a node of the join tree it has been turned into), to use steps of iterative proportional fitting to adapt the marginal distributions on the maximal cliques, and to distribute the information added by such an adaptation to the other maximal cliques by join tree propagation.

4 Modeling Electrical Networks

In this section we discuss some problems of modeling analog electrical circuits with probabilistic graphical models. Straightforward intuitive approaches fail due to two reasons: (1) cycles in underlying graph structure and (2) difficulties to specify the probability distributions in a plausible and consistent way.

We illustrate these problems with the very simple resistive direct current circuit shown in Figure 1 (left). A very natural approach to construct a graphical model for this circuit would be to set up a clique graph like the one shown in Figure 1 (right), in which there is one node for each electrical law needed to describe the circuit. The nodes at the four corners encode Kirchhoff's junction law for the four corners of the circuit and the diamond-shaped node in the middle represents Kirchhoff's mesh law. The remaining three nodes describe the three resistors with Ohm's law. (The two nodes on the left may be removed, since it is $I_0 = I_1 = I_2 = I_3$ and thus the two corner nodes on the right suffice.)

The obvious problem with this clique graph is that it is cyclic and thus evidence propagation can lead to inconsistent results. The crucial point is that all four currents must be equal and thus, depending on the resistors, only certain combinations of values for the voltages U_1 , U_2 and U_3 are possible. However, these relations are not enforced by the network, so that actually impossible states of the circuit are not ruled out.

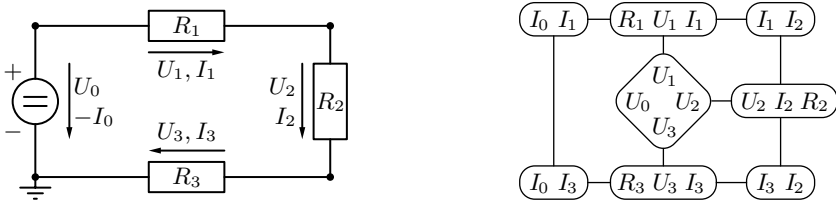


Fig. 1. A simple resistive circuit and an intuitive graph structure for this circuit

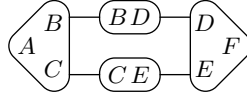


Fig. 2. An illustration of the propagation problem

Table 1. Probability distributions for the graph structure shown in Figure 2

A/F		0		1	
B/D		0	1	0	1
C/E	0	0	0.25	0.25	0
	1	0.25	0	0	0.25

B/C		0	1
D/E	0	0.5	0
	1	0	0.5

This problem is best understood if we consider a minimum example with binary variables, as shown in Figure 2. It is $\text{dom}(A) = \dots = \text{dom}(F) = \{0, 1\}$. The marginal probability distributions for the four nodes, with $p_{ABC} \equiv p_{FDE}$ and $p_{BD} \equiv p_{CE}$, are shown in Table 1. Suppose now that $A = 1$ is observed. Since this enforces $B = C$ and thus $D = E$, we should get the result $P(F = 0) = 0$ and $P(F = 1) = 1$. However, the marginal distributions on the *individual* variables B and C do not change due to this observation (it is still $P(B = 0) = P(B = 1) = P(C = 0) = P(C = 1) = 0.5$). Thus no information is transmitted to the right half of the network, leading to $P(F = 0) = P(F = 1) = 0.5$.

Basically the same problem we encounter for the electrical circuit in the graph structure shown in Figure 1. For instance, if we set the variables for the three resistors to the same value, all voltages U_1 , U_2 and U_3 must be equal. However, this information does not or not completely reach the center node. To cope with this problem we would have to merge nodes in order to obtain an acyclic structure, which, if done inappropriately, can lead to fairly large cliques (and doing it optimally is a non-trivial issue—actually it is NP-hard, since finding an optimal triangulation of an undirected graph is NP-hard [4])

The second problem we encounter when we try to construct a graphical model results from the fact that the electrical laws and the prior information about marginal probability distributions over, for example, the resistor values do not allow for a direct initialization of the quantitative part of the network. In a way, we have too little information. To see this, consider how one may try to build a Bayes network for the circuit shown in Figure 1. We would like to have parentless nodes only for those variables, for which we can specify marginal distributions, that is, for the resistor values and maybe the supply voltage. Every other variable should be the child of one or more variables, with the conditional probability distribution encoding the electrical law that governs the dependence, because we cannot easily specify a marginal distribution for it.

However, this is not possible as Figure 3 demonstrates (note that the current I_0 is left out, because it must be identical to all other currents anyway). The Bayes network shown in this figure is constructed as follows: First we choose one of the voltages U_1 , U_2 and U_3 as the child for Kirchhoff's mesh law. For

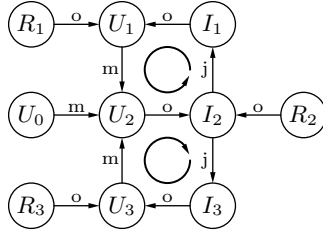


Fig. 3. Attempt at building a Bayes network for the circuit shown in Figure 1: Two cycles result

reasons of symmetry we choose U_2 , which leads to the edges marked with an m , but other choices lead to the same problem in the end. As U_1 and U_3 cannot be left parentless, because we cannot easily specify marginal distributions for them, we use Ohm's law to make them dependent on the corresponding currents and resistor values. This leads to the edges marked with an o in the top and bottom row of the network. For the second resistor, however, we make I_2 the child, because U_2 already has all the parents it needs and R_2 should be parentless. This leads to the remaining two edges marked with an o . Finally we make I_1 and I_3 children of some other variable, because we cannot specify marginal distributions for them easily. The only law left for this is Kirchhoff's junction law, which leads to the edges marked with a j . However, the final graph has two cycles and thus cannot be used as a Bayes network.

5 Constructing the Graphical Model

In this section we do not use voltages over electrical components anymore, as in Figure 1, but turn to node voltages (potentials). This has advantages not only w.r.t. the measurement process (since node voltages against ground are simpler to measure), but also has some advantages w.r.t. the construction of the graphical model. Note, however, that the problems pointed out in the preceding section are not solved by this transition. In the preceding section we used voltages over components, because this made it easier to demonstrate the core problems.

In the following we describe an algorithm to construct a join tree representation of a Markov network for an analog electrical circuit. Let a time-invariant $n + 1$ node, b branch steady state circuit with known topology be given, the nodes of which are accessible terminals for measurements. One of them is taken as a reference (ground) and the node voltages are used to study the circuit. We assume that for each component the electrical law that governs its behavior (for example, Ohm's law for a resistor), its nominal value(s) and a tolerance provided by the manufacturer are known. We use the following notation:

$$\begin{aligned} U_i, \quad i = 0, \dots, n - 1 : & \text{node voltages,} \\ I_j, \quad j = 0, \dots, b - 1 : & \text{branch currents,} \\ R_k, \quad k = 0, \dots, b - 1 : & \text{branch resistances.} \end{aligned}$$

(Note that all magnitudes may be complex numbers, making it possible to handle steady state alternating current circuits. For reasons of simplicity, however, we confine ourselves to direct current circuits here.)

To build a join tree of a Markov network for this circuit, we have to find partitions of the set of variables $V = \{U_0, \dots, U_{n-1}, I_0, \dots, I_{b-1}, R_0, \dots, R_{b-1}\}$ into three disjoint subsets X_1 , X_2 and X_3 , such that the variables in X_1 and X_2 are conditionally independent given the variables in X_3 . That is, if the values of the variables in X_3 are fixed, a change of the value of a variable in X_1 has no effect on the values of the variables in X_2 and vice versa.

To find such partitions, we consider virtual cross-sections through the circuit (only through wires, not through components). Each of these cross-sections defines a set of variables, namely the voltages of the wires that are cut and the currents flowing through them. Since this set of variables obviously has the property of making the variables on one side of the cross-section independent of those on the other side (and thus satisfies the conditional independence property), we call it a *separator set*. We select a set of cross-sections so that each component is enclosed by two or more cuts or is cut off from the rest of the circuit by a single cut (terminal cross-section). Then the electrical law governing a component describes how the variables of its enclosing cross-sections relate to each other. Note that there are usually several ways of selecting the cross-sections and that an appropriate selection is crucial to the complexity of the network. However, selecting appropriate cross-sections is easier than finding good node mergers in the approach discussed above.

Given a set of cross-sections we construct the join tree as follows: the separator sets form the node separators. For each circuit part (containing one component) we create a node containing the union of the separator sets of the bounding cross-sections. In addition, we create a node for each component, comprising the variables needed to describe its behavior, and connect it to the node corresponding to the circuit part the component is in. If the component node contains currents not yet present in the circuit part node, we add these currents to it. The connection is made through an appropriate node separator, containing the intersection of the sets of variables assigned to the connected nodes.

Next this initial graphical model is simplified in two steps. In the first step, the number of variables is reduced by exploiting trivial Kirchhoff junction equations (like the identity of two currents). In the second step, we merge adjacent nodes where the variables in one of them is a subset of the variables in the other. The result is the qualitative part of the graphical model, i.e. the graph structure of the join tree, enhanced with node separators.

To find the quantitative part (the probability distributions), we initialize all node distributions to uniform. Next we enforce the component laws as well as Kirchhoff's laws (wherever applicable) by zeroing the entries of the probability distributions that correspond to impossible value combinations. Finally we incorporate the manufacturer supplied information about nominal values and tolerances by iterative proportional fitting (see Section 3), thus setting the marginal

component distributions. The resulting graphical model can then be used to diagnose the modeled circuit by propagating node voltage measurements.

From the theory of evidence propagation in graphical models and in particular in join trees it is well known that the computational complexity of operations (iterative proportional fitting and evidence propagation) is governed by the size of the node distributions, which depends on the number of variables in a join tree node and the sizes of their domains. If the distributions can be kept small by a proper selection of cross-sections, the computation is very efficient.

6 A Simple Example

To illustrate our approach we consider the simple resistive circuit shown in Figure 4, where $n = 5$, $b = 7$. It is fed by a voltage supply U_0 , whose internal resistance R_0 we assume to be zero. The set of (real valued) variables is $V = \{U_0, \dots, U_4, I_0, \dots, I_6, R_0, \dots, R_6\}$. We select the set of six cross-sections S_1 to S_6 that are shown in Figure 5. As an example of the conditional independences consider the cross-section S_3 : once we know the voltage of the cut wires (U_1 and U_2) and the currents through them (I_1 and I_3 , $I_3 = I_1$), all the magnitudes to the left of S_3 become independent of those to the right of S_3 .

The initial graphical model, as it is constructed from the separator sets, is shown in Figure 6. The node separators (rectangles) are labeled by the cross-sections S_1 to S_6 they correspond to. The nodes are drawn with rounded corners and thicker lines. To simplify the network, we exploit $I_0 = I_1 = I_3$ and $I_4 = I_5 = I_6$. Furthermore, we merge (1) the four leftmost nodes (two from the top row and two from the bottom row), (2) the third and the fourth node on the

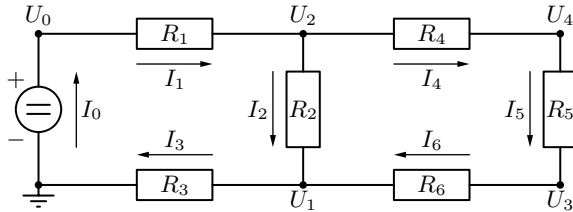


Fig. 4. Another very simple resistive circuit

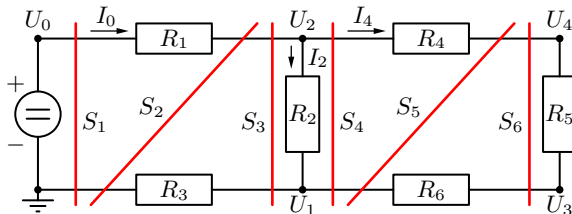
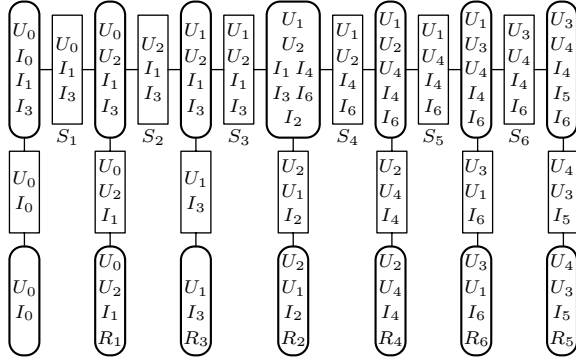
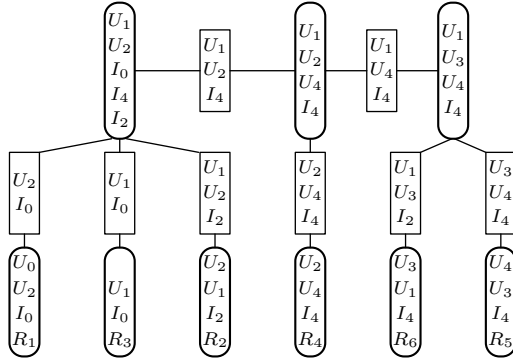


Fig. 5. The resistive circuit with cross-sections

**Fig. 6.** Initial graphical model for the example**Fig. 7.** Simplified graphical model for the example

top row and (3) the two rightmost nodes (the last nodes from the top and the bottom row). The result is shown in Figure 7.

For our experiments we implemented the described method for this example and a discrete Markov network in C.³ We discretized the continuous ranges of values as follows⁴: resistors: 1 to 5Ω with 1Ω steps, voltages: 0 to 20V with 1V steps, currents: 0 to 4A with 1A steps. For the six resistors we set an initial probability distribution that is roughly normal and centered at 3Ω, that is, for $i = 1, \dots, 6$: $p_{R_i}(r) = (0.1, 0.2, 0.4, 0.2, 0.1)$.

³ We plan to make the C sources available at the URL <http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html>

⁴ An alternative to handle metric attributes, which comes to mind immediately, is a Gaussian network. Unfortunately, in its standard form (that is, with a covariance matrix) a Gaussian network is restricted to linear dependencies. Ohm's law, however, specifies a nonlinear dependences as it involves the product of two quantities.

Table 2. Resistor marginals after propagating the supply voltage $U_0 = 20$ and the measurement $U_4 = 5$

	$U_0 = 20$					$U_0 = 20 \wedge U_4 = 5$				
R_1	.11	.22	.39	.19	.09	.00	.04	.33	.32	.31
R_2	.09	.18	.41	.21	.11	.17	.23	.38	.16	.07
R_3	.12	.22	.40	.18	.08	.53	.29	.15	.03	.00
R_4	.11	.21	.40	.19	.09	.05	.15	.39	.27	.15
R_5	.11	.21	.40	.19	.09	.16	.25	.37	.16	.07
R_6	.11	.21	.40	.19	.09	.16	.25	.37	.16	.07

The initial probability distributions are determined as described in Section 5, that is, by enforcing the electrical laws and incorporating the resistor marginals by iterative proportional fitting. To mitigate the effects of the discretization of the value ranges, we set a zero probability only if there is no combination of values from the represented intervals that is valid, i.e., satisfies the electrical law. With a threshold of 10^{-6} the iterative proportional fitting procedure converges after 5 iterations. This yields the diagnostic network.

Next we set the voltage supply to 20V and propagate this information using join tree propagation. This changes the marginals of the resistors only slightly as can be seen on the left in Table 2. Suppose now that we measure the node voltage U_4 and find it to be 5V. Propagating this evidence yields the resistor marginals shown on the right in Table 2. It can be seen that due to the measurement the distributions for R_1 and R_3 change considerably, indicating that at least resistor R_3 is highly likely to deviate from its nominal value.

7 Conclusions and Future Work

We presented a method for modeling and diagnosing analog electrical circuits that exploits probabilistic information about production tolerances of electrical components. It consists of: the construction of a join tree representation of a Markov network from a set of cross-sections of an analog electrical circuit; the iterative proportional fitting procedure for the initialization of the probability distributions; and the join tree propagation algorithm for the incorporation of measurements. For our experiments we used a simple example to keep things comprehensible, but the approach is fully general and can be applied to any steady state, alternating or direct current electrical circuit. Faults like shortcuts or open connections can easily be included by adding them as possible states to the variable(s) describing a circuit component.

In the future we plan to make our method more efficient by exploiting the sparsity of the (discrete) probability distributions (the electrical laws rule out a large number of value combinations) and by using parameterized continuous distributions. Furthermore, we plan to develop a theory of how to select measurements in a diagnosis process. The basic idea is to propagate possible outcomes of measurements through the network, to compute (and to aggregate) the result-

ing reductions in entropy of the distributions on component values, and finally to select the measurement that leads to the highest expected entropy reduction (similar to the approach suggested in [8]).

References

1. F. Aminian, M. Aminian, and H.W. Collins. Analog Fault Diagnosis of Actual Circuits Using Neural Networks. *IEEE Trans. Instrumentation and Measurement* 51(3):544–550. IEEE Press, Piscataway, NJ, USA 2002
2. J.W. Bandler and A.E. Salama. Fault Diagnosis of Analog Circuits. *Proc. IEEE* 73:1279–1325. IEEE Press, Piscataway, NJ, USA 1985
3. C. Borgelt and R. Kruse. *Graphical Models — Methods for Data Analysis and Mining*. J. Wiley & Sons, Chichester, UK 2002
4. E. Castillo, J.M. Gutierrez, and A.S. Hadi. *Expert Systems and Probabilistic Network Models*. Springer-Verlag, New York, NY, USA 1997
5. I. Csiszar. I-Divergence Geometry of Probability Distributions and Indirect Observations. *Studia Scientiarum Mathematicarum Hungarica* 2:299–318. Hungarian Academy of Sciences, Budapest, Hungary 1975
6. F.V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, London, UK 1996
7. R. Jiroušek and S. Pœuël. On the Effective Implementation of the Iterative Proportional Fitting Procedure. *Computational Statistics and Data Analysis* 19:177–189. Int. Statistical Institute, Voorburg, Netherlands 1995
8. J. de Kleer and B.C. Williams. Diagnosing Multiple Faults. *Artificial Intelligence* 32(1):97–130. Elsevier Science, New York, NY, USA 1987
9. S.L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, UK 1996
10. R.-W. Liu, ed. *Selected Papers on Analog Fault Diagnosis*. IEEE Press, New York, NY, USA 1987
11. R.-W. Liu. *Testing and Diagnosis of Analog Circuits and Systems*. Van Nostrand Reinhold, New York, NY, USA 1991
12. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (2nd edition)*. Morgan Kaufman, San Mateo, CA, USA 1992
13. R. Spina and S. Upadhyaya. Linear Circuit Fault Diagnosis Using Neuromorphic Analyzers. *IEEE Trans. Circuits and Systems II* 44(3):188–196. IEEE Press, Piscataway, NJ, USA 1997
14. J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. J. Wiley & Sons, Chichester, UK 1990

Qualified Probabilistic Predictions Using Graphical Models

Zhiyuan Luo and Alex Gammerman

Computer Learning Research Centre,
Royal Holloway, University of London,
Egham, Surrey TW20 0EX, UK
{zhiyuan, alex}@cs.rhul.ac.uk

Abstract. We consider probabilistic predictions using graphical models and describe a newly developed method, fully conditional Venn predictor (FCVP). FCVP can provide upper and lower bounds for the conditional probability associated with each predicted label. Empirical results confirm that FCVP can give well-calibrated predictions in online learning mode. Experimental results also show the prediction performance of FCVP is good in both the online and the offline learning setting without making any additional assumptions, apart from i.i.d.

1 Introduction

We are interested in making probabilistic predictions about a sequence of examples $z_1, z_2, \dots, z_n$. Each example z_i consists of an object x_i and its label y_i . The objects are elements of an object space X and the labels are elements of a finite label space Y . The example space Z can be defined as $Z = X \times Y$. It is assumed that the example sequence is generated according to an i.i.d. (independently and identically distributed) probability distribution P in Z^n .

Suppose that the label space Y is enumerated for all possible classification labels $1, 2, \dots, |Y|$. The learner Γ is a function on a finite sample of n training examples $(z_1, z_2, \dots, z_n) \in Z^n$ that makes a prediction for a new object $x_{n+1} \in X$

$$\Gamma : Z^n \times X \rightarrow [0, 1]^{|Y|}. \quad (1)$$

Probability forecasting estimates the conditional probability of a possible label given an observed object. For each new object x_{n+1} (with true label y_{n+1} withheld from the learner), a set of predicted conditional probabilities for each possible labels are produced. In the online learning setting, examples are presented one by one. The learner Γ takes object x_i , predicts $\hat{y}_i$, and then gets a feedback y_i . The new example $z_i = (x_i, y_i)$ is then included in the training set for the next trial. In the offline setting, the learner Γ is given a training set $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ to predict on a test set $x_{n+1}, x_{n+2}, \dots, x_{n+k}$.

This paper considers probabilistic predictions by using graphical models where examples are structured and more importantly, the data generating probability distribution P can be decomposed [4]. Firstly, we briefly discuss the

Bayesian belief network approach to probabilistic predictions and a technique called sequential learning to represent and update imprecision of conditional probabilities in the light of new cases [7]. Then we present a newly developed approach, called Venn probability machine, to probabilistic predictions [8]. In particular, we discuss fully conditional Venn predictor (FCVP) designed for the graphical models and its implementation. Finally, experiments are carried out on the simulated datasets to evaluate the FCVP. The empirical results confirm that predictions of FCVP are well-calibrated, in the sense that the error probability intervals produced by the FCVP bound the number of prediction errors. The experimental results demonstrate the performance of FCVP is good.

2 Bayesian Belief Networks

Bayesian belief networks are graphical knowledge representations. A Bayesian belief network can be represented as a pair (G, P) . Qualitative knowledge G is a directed acyclic graph where the nodes V of the graph G are random variables. In this paper, we only consider the nodes V that take a finite set of values. The graph G is a representation of quantitative knowledge P that factorises in the form

$$P(V) = \prod_{v_i \in V} P(v_i \mid pa(v_i)), \quad (2)$$

where $pa(v_i)$ is a set of parent nodes of v_i in G .

Various algorithms exist to exploit and take full advantage of the independence relationships embodied in the network and efficient evidence propagation algorithms have been developed [1]. One of these approaches is called junction tree algorithm [3]. The junction trees are tree-like data structures whose vertices are labelled by cliques and whose edges labelled by separator sets form by intersection of two cliques on either side. Given a Bayesian belief network, a junction tree can be obtained [1]. This is done by (1) constructing an undirected graph called the moral graph from the Bayesian belief network; (2) selectively adding arcs to the moral graph to form a triangulated graph; (3) identifying the maximal cliques from the triangulated graph; (4) building the junction tree, starting with cliques as the nodes, where each link between two cliques is labelled by a separator. It has been shown that the joint probability distribution $P(V)$ in a junction tree can be represented as

$$P(V) = \frac{\prod_{c_i \in C} \Psi(c_i)}{\prod_{s_i \in S} \Psi(s_i)}, \quad (3)$$

where Ψ indicates the potential function on the cliques (C) and separators (S) which takes non-negative values. Note that $\Psi(c_i) \propto P(c_i)$, $\Psi(s_i) \propto P(s_i)$ for $c_i \in C$ and $s_i \in S$.

The junction tree can be used for efficient inference. When evidence arrives in a network, it is first absorbed into the junction tree. Then message passing

protocol is used to propagate the evidence. The marginal distribution of a variable, conditional on some evidence can be found by local computation on the junction tree [1].

3 Parameter Learning

So far, we have assumed that the conditional probability tables in (2) for a given Bayesian belief network can be specified precisely. However, this assumption may not be realistic. The conditional probabilities derived from subjective assessments or specific dataset are subject to inevitable imprecision. The goal of parameter learning is to revise conditional probabilities for a given network topology as new cases arrive [5]. One such parameter learning method was proposed by Spiegelhalter and Lauritzen [7], namely sequential learning, and we follow here their approach.

The basic idea of sequential learning is to represent the imprecision of these conditional probabilities explicitly as parameters θ . In a Bayesian belief network setting, it is reasonable to partition the space θ into a set of small spaces θ_i concerning each node v_i and assume θ_i is independent a priori to each other. That is, $P(\theta) = \prod_{i=1}^n P(\theta_i)$, where n is the number of nodes in V . Each conditional probability table attached to node v_i is determined uniquely by the parameter θ_i . $P(V | \theta)$ can be written as follows due to the conditional independence reflected in the model $P(V | \theta) = \prod_{v_i \in V} P(v_i | pa(v_i), \theta_i)$. The joint probability distribution on V and θ is then calculated as $P(V, \theta) = \prod_{v_i \in V} P(v_i | pa(v_i), \theta_i) P(\theta_i)$. It is clear that the parameter θ_i may be considered as another parent node of v_i in the network. These θ_i parameters represent summary of past cases.

Given the network structure, $P(v_i | pa(v_i), \theta_i)$ and $P(\theta_i)$ specified for each node v_i , the task now is to calculate the posterior distribution $P(\theta | e)$ when an instantiation of variables e is obtained. Three basic operations are involved: dissemination of experience, propagation of evidence and retrieval of new information. The procedure can be repeated in the same manner as more instantiation of variables arrive. Different assumptions are made for different operations to simplify the computation. Firstly, independence of each parameter θ_i over node v_i is assumed. This allows the dissemination operation to be carried out locally. For each variable v_i , we apply

$$P(v_i | pa(v_i)) = \int P(v_i | pa(v_i), \theta_i) P(\theta_i) d\theta_i \quad (4)$$

to get the means of the conditional probabilities $P(v_i | pa(v_i), \theta_i)$ for each node v_i . Secondly, the current ‘marginal probabilities’ are used to initialise the standard evidence propagation methods, such as the junction tree algorithm described before. Finally, in the retrieval operation, the following calculation is performed:

$$P(\theta_i | e) = \sum_{v_i, pa(v_i)} P(\theta_i | v_i, pa(v_i), e) P(v_i, pa(v_i) | e). \quad (5)$$

Since θ_i is conditional independent of e given v_i and $pa(v_i)$, thus

$$P(\theta_i | e) = \sum_{v_i, pa(v_i)} P(\theta_i | v_i, pa(v_i)) P(v_i, pa(v_i) | e). \quad (6)$$

It is clear that there is a mixture distribution for the parameter θ_i if v_i and $pa(v_i)$ are not observed in the new case e . To simplify the retrieval operation, it is assumed that the individual parameter θ_i for node v_i can be further partitioned and is conditional on each possible configuration of its parent set $pa(v_i)$. Therefore, each conditional probability distribution under a configuration of the parent nodes can be individually updated in the light of e .

In this paper, we model θ_i as Dirichlet distributions and update these parameter θ_i with complete new cases. In particular, we use a Dirichlet prior distribution as a conjugate form and the mean of the Dirichlet distribution is used as the estimation of $P(v_i | pa(v_i))$. The Dirichlet has a simple interpretation in terms of pseudo counts. Both dissemination and retrieval operations are straightforward with completed data. Note that we use BDeu prior (likelihood equivalent uniform Bayesian Dirichlet) in our experiments [2].

4 Venn Probability Machines

The Venn Probability Machine (VPM) is a simple yet powerful framework for probability forecasts [8]. Unlike many conventional probabilistic prediction approaches, VPM gives several probability distributions for the predicted label. These probability distributions are close to each other so that probabilistic prediction made by the VPM will be practically useful. Therefore, VPM is a type of multiprobability predictor.

The basic idea behind the VPM is as follows. Given the training example sequence $(x_1, y_1), \dots, (x_{n-1}, y_{n-1})$ and a new test example x_n , we consider each possible completion for x_n . For each possible completion $y \in |Y|$, we have n examples $(x_1, y_1), \dots, (x_n, y)$ and then divide all the examples into a number of categories. It is required that such division of examples is independent of the order of examples. Many existing supervised machine learning algorithms can be used to perform the division. For example, a simple way to divide the examples into different categories is based on the 1-nearest neighbour algorithm. Two examples are assigned to the same category if their nearest neighbours have the same label. Taking the category T containing the example (x_n, y) , we can estimate the relative frequency of examples labelled j in T as

$$A_{y,j} = \frac{|\{(x', y') \in T : y' = j\}|}{|T|}. \quad (7)$$

Those relative frequencies obtained in (7) are interpreted as empirical probability distributions for the predicted labels.

Having considered all possible completion for x_n , we have a $|Y| \times |Y|$ Venn probability matrix A . The rows of the matrix A represent the frequency count

of each class label in the training examples set which have the same type as the new test example. The minimum and maximum frequency counts within each row give us the lower and upper bounds for conditional probabilities of possible labels given x_n . VPM predicts the label for the new test example using the respective column which contains the largest of minimum entries.

VPM is different from Bayesian learning theory and PAC learning [8]. Unlike Bayesian learning theory, VPM requires no empirical justification for probabilities. In contrast with PAC learning which aims to estimate the unconditional probability of error, VPM tries to estimate the conditional distribution of the label given the new object. A useful property of VPM is its self-calibration nature in the online learning setting. It has been proved that the probability intervals generated by the VPM is well-calibrated in the sense that the VPM can bound the true conditional probability for each new test object in an online test [8]. Using the VPM's upper and lower intervals for conditional probabilities, we can estimate the bounds for the number of errors made.

4.1 Online Compression Models

The Venn probability machine can be generalised to online compression models which can summarise statistical information efficiently and perform lossy compression [9]. Formally, an online compression model (OCM) is defined as $M = (\Sigma, \square, Z, (F_n), (B_n))$ where

- Σ is a measurable space called summary space containing summaries σ .
- $\square \in \Sigma$ is special summary called the empty summary and we set $\sigma_0 = \square$.
- Z is a measurable space containing the examples z_i
- $F_n, n = 1, 2, \dots$ are measurable functions of the type $\Sigma \times Z \rightarrow \Sigma$. F_n are called forward functions that allow us to update summary σ_{n-1} to σ_n given the example z_n in an online fashion. Therefore, we have $F_n(\sigma_{n-1}, z_n) = \sigma_n$.
- $B_n, n = 1, 2, \dots$ are backward kernels of the type $\Sigma \hookrightarrow \Sigma \times Z$. It is required that B_n are inverse to F_n in the sense that $B_n(F_n^{-1}(\sigma) \mid \sigma) = 1$ for each $\sigma \in F_n(\Sigma \times Z)$. B_n map $\sigma \in \Sigma$ to probability distributions in Z .

Intuitively, the summaries σ can be considered as sufficient statistics for the observed example sequence. For example, the summaries σ can be the number of ones in a binary sequence generated by Bernoulli models. We start with the empty summary $\square$ which indicates that we do not have information about the data, i.e. $\sigma_0 = \square$. When the first example z_1 arrives, we update our summary to σ_1 using $F_1(\sigma_0, z_1)$. We update our summary to $\sigma_2 = F_2(\sigma_1, z_2)$ given the second example z_2 , so on and so forth. Basically, forward functions F_n extract all useful information from the observed example sequence and perform lossy compression. It is important that the summaries are calculated in an online fashion, i.e. F_n updates σ_{n-1} to σ_n given z_n . On the other hand, backward kernels B_n perform decompression and allow us to find the conditional distribution of a particular example sequence $(z_1, z_2, \dots, z_n)$ given the summary σ_n . This is done iteratively. Given σ_n , we generate (σ_{n-1}, z_n) from the distribution $B_n(\sigma_n)$. Then we generate (σ_{n-2}, z_{n-1}) from $B_{n-1}(\sigma_{n-1})$, so on and so forth.

VPM can be generalised to an OCM. When we have seen $n - 1$ examples, the OCM summarises these examples and has σ_{n-1} . Given a test example x_n , we can try all possible completion $y \in |Y|$ and have $\sigma_n = F_n(\sigma_{n-1}, (x_n, y))$. We specify a partition A_n and use it to divide the set $F_n^{-1} \subseteq \Sigma_{n-1} \times Z$ into a number of categories. This is done by assigning (σ', z') and (σ'', z'') to the same category if and only if $A_n(\sigma', z') = A_n(\sigma'', z'')$ where $A_n(\sigma, z)$ represents the element of the partition A_n containing (σ, z) . Consider the category $T = A_n(\sigma_{n-1}, (x_n, y))$, we estimate the probability distribution of the label y as

$$p_y = \frac{B_n(\{(\sigma^*, (x^*, y^*)) \in T : y^* = y\} | \sigma_n)}{B_n(T | \sigma_n)}. \quad (8)$$

4.2 Fully Conditional Venn Predictor

When examples z_i are generated from Bayesian belief networks, an explicit OCM can be defined and an efficient Venn predictor called fully conditional Venn predictor (FCVP) can be constructed. The junction tree constructed from the Bayesian belief network can serve as a basis for efficient summaries of observed data sequence. As discussed earlier, a junction tree is a graphical data structure consisting of cliques and separators. For convenience, we refer to both the cliques and the separators of a junction tree as clusters. We can associate a table with each cluster where the index of the table is determined by the configurations on the cluster and each entry of the table is a non-negative integer. Obviously, the number of entries in the table on a cluster depends on the number of possible configurations of the variables in the cluster. The table size is defined as the sum of all entries. All the tables on the clusters form a table set for a junction tree. We are only interested in table sets all of whose tables have the same size.

We define an example z is consistent with a configuration of a cluster E if the configuration coincides with the restriction $z|_E$ of z to E . If we assign the number of past examples which are consistent with each configuration of the clusters to appropriate entries of the tables on the clusters, we have a table set σ generated by the example sequence. The length of each example sequence generating σ will be equal to the table size of σ . The number of example sequences generating a table set σ is specified as $\#\sigma$. One of possible operations on the table set σ is to query the number assigned to a configuration of a cluster u , which is defined as the σ -count of the configuration. For example, $\sigma_u((x_i, y_i))$ will return the count assigned by the table set σ to a configuration of a cluster u which is consistent with the example (x_i, y_i) .

An OCM $M = (\Sigma, \square, Z, (F_n), (B_n))$ can be defined for the junction tree model as follows:

- Summary space Σ consists of summaries defined by the consistent table sets σ .
- Empty summary $\square$ is a table set with size 0.
- Z consists of the set of all examples. An example z_i is simply a particular configuration on V .

- Given an example z_n , the forward function F_n will update the table set by adding 1 to the entries of the table set which are consistent with z_n .
- An example z is consistent with a summary σ if the σ -count of each configuration that is consistent with z is positive. For the size of $\sigma = n$, backward kernels B_n can be defined as

$$B_n(\{(\sigma \downarrow z, z) \mid \sigma\}) = \frac{\#(\sigma \downarrow z)}{\#\sigma} \quad (9)$$

where $\sigma \downarrow z$ means subtracting 1 from the σ -count of any configuration that is consistent with z .

The junction tree has the property that for each pair U, V of cliques with intersection S , all cliques on the path between U and V contain S . For a table set σ defined on a junction tree, it is consistent if and only if: (1) each table in σ has the same size, and (2) if clusters E_1 and E_2 intersect, the marginalisations of their tables to $E_1 \cap E_2$ coincide. Given a summary σ of size n in the junction tree model, the number of example sequences of length n that are consistent with the table set σ is

$$\#\sigma = \frac{n! \prod_{s \in S} \text{fp}_\sigma(s)}{\prod_{c \in C} \text{fp}_\sigma(c)} \quad (10)$$

where $\text{fp}_\sigma(E)$ is the factorial-product of a cluster E in a summary σ and $\text{fp}_\sigma(E) = \prod_{a \in \text{configurations of } E} \sigma_E(a)!$. It has been proved in [9] that given the summary σ_n of the first n examples, the conditional probability that $z_n = (x_n, y)$ based on maximum likelihood estimation is

$$\frac{\prod_{c \in C} \sigma_c((x_n, y))}{n \prod_{s \in S} \sigma_s((x_n, y))}. \quad (11)$$

Note that the ratio defined in (11) is set to 0 if any of the factors in the numerator or denominator is 0; in this case $z_n = (x_n, y)$ is not consistent with the summary σ .

Having specified the OCM for junction tree model, we are now ready to describe the Venn predictor. When a junction tree OCM has one or more variables as labels, a Venn predictor called fully conditional Venn predictor can be defined by determining partition A_n in which $A_n(\sigma, z)$ consists of all (σ, z') for which z and z' match on all non-label variables. Once the partition A_n is established, the VPM can make predictions and provide upper and lower bounds for the conditional probability associated with each predicted label. The FCVP algorithm in the online learning mode is presented below.

5 Experiments

5.1 Dataset

The well-known ‘Visit to Asia’ example is used for our experiments [4]. There are 8 binary variables in this example, see Figure 1. For the online learning experiments, three datasets with 1000, 2000 and 5000 examples were randomly

Algorithm 1. Fully Conditional Venn Predictor

Require: a list of variables and the values each variable can take
Require: junction tree with its cliques C and separators S
Require: object space X , label space Y and target label space $Y^t \subseteq Y$
Require: N examples $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$

$\sigma_0 = \square$
for $n = 1$ **to** N **do**
 get $x_n \in X$ of example (x_n, y_n)
 for $y = 1$ **to** $|Y|$ **do**
 $\sigma = F_n(\sigma_{n-1}, (x_n, y))$
 for $y' = 1$ **to** $|Y|$ **do**
 $A_{y,y'} = \frac{\prod_{c \in C} \sigma_c((x_n, y'))}{\prod_{s \in S} \sigma_s((x_n, y'))}$ $\{A_{y,y'}$ is set to 0 if any of the factors in the numerator
 or denominator is 0 $\}$
 end for
 end for
 $A_{y,y'} = \frac{A_{y,y'}}{\sum_{y'} A_{y,y'}}$
 for $y^t = 1$ **to** $|Y^t|$ **do**
 $A_{y,y^t} = \sum_{y' \setminus y^t} A_{y,y'}$
 end for
 predict $\hat{y}^t = \arg \max_{y^t \in Y^t} (\min_{y \in Y} A_{y,y^t})$
 output predicted probability interval for $\hat{y}^t$ as $[\min_y A_{y,\hat{y}^t}, \max_y A_{y,\hat{y}^t}]$
 get $y_n \in Y$ of example (x_n, y_n)
 $\sigma_n = F_n(\sigma_{n-1}, (x_n, y_n))$
end for

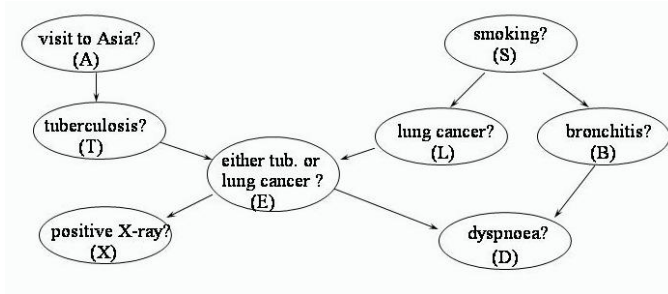


Fig. 1. ‘Visit to Asia’ example

generated using the network structure and the associated conditional probabilities. For the offline learning experiments, another three datasets were randomly generated: (training size=3000, test size=1000), (training size=2000, test size=2000) and (training size=1000, test size=3000).

5.2 Methods

For experiment purpose, we assume that we have evidence on the variables A, S, X and D (patient history and diagnostic tests) and would like to predict the

conditional probabilities of the variables B, T, E and L (medical diagnosis), respectively, given these observations.

Fully conditional Venn predictor (FCVP) was implemented using Bayes net toolbox (BNT) for Matlab [6]. In order to evaluate prediction performance of FCVP, we also implemented the junction tree algorithm and the sequential learning algorithm using BNT. The junction tree algorithm is specified with precise conditional probabilities, i.e. it has the same conditional probabilities as those used to generate the datasets in the previous section. On the other hand, both the FCVP and sequential learning algorithm will have to learn these conditional probabilities from the past examples. The implemented systems, namely FCVP, junction tree algorithm (JT) and sequential learning (SL) are evaluated on the datasets generated in the previous section. The conditional probabilities were calculated on each of the label variables $\{B, T, L, E\}$ and predictions made.

The junction tree algorithm and sequential learning produce a single probability distribution on a label and predict the class label with the largest associated conditional probability $\hat{y}_i = \arg \max_{y \in Y} \hat{p}_{i,y}$ given the test example x_i . On the other hand, FCVP outputs an interval for the probability that the predicted label is correct. If the interval is $[a_i, b_i]$ at the trial i , the complementary interval $[1 - b_i, 1 - a_i]$ is the error probability interval.

If more than one label has the largest associated conditional probability, we have multiple predictions. A prediction is correct if the true label for the example matches the predicted label. Otherwise it is an error.

5.3 Results

The predictions made by FCVP on the variables B, T, E and L were obtained in the experiments. Figure 2 shows the performance of FCVP on the variable B in the online learning setting on the datasets of 1000 examples. In this figure, the cumulative lower and upper probability error bounds, the prediction errors and multiple predictions are presented. These plots confirm that the error probability intervals generated by FCVP are well-calibrated. FCVP can produce a multiple prediction in the sense that the predicted probability interval for each label is $[0, 1]$. Note that the total number of multiple predictions is small and the multiple predictions occur at the beginning of the trials when some combination is observed for the first time. For example, 7 multiple predictions were observed on a dataset of 1000 examples. Similar prediction behaviour was observed for the variables T, L and E.

In our experiments, the three algorithms were tested and evaluated on the same datasets. Figure 3 displays the comparative performance results on B on 1000 examples in terms of the number of prediction errors. It is clear that the prediction performance of FCVP is very similar to the one produced by the junction tree algorithm with precise conditional probabilities and much superior to that of the sequential learning. Table 1 presents the summaries of results on different datasets in terms of the cumulative number of prediction errors and the number of multiple predictions. For example, the junction tree algorithm made 31 prediction errors on the variable B over 1000 examples, see Table 1. On the

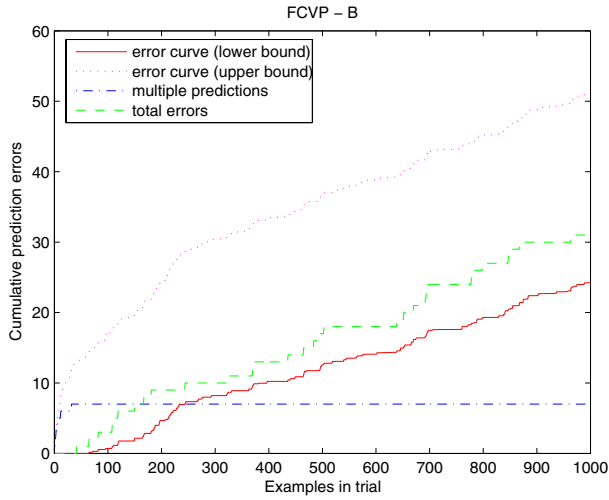


Fig. 2. FCVP results (1000 examples) - online learning mode

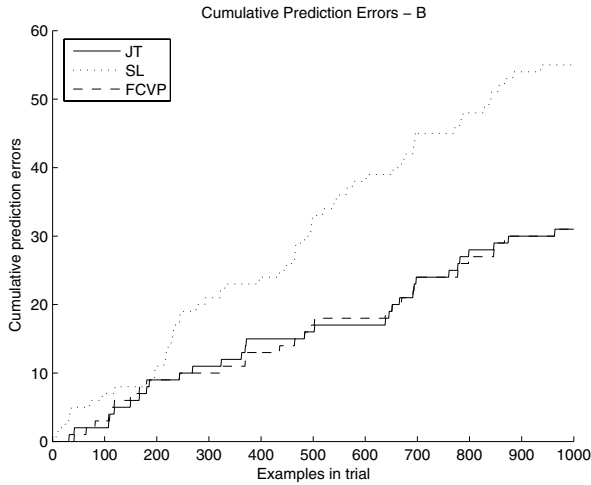


Fig. 3. Comparative performance (1000 examples) - online learning mode

other hand, the prediction errors made by the sequential learning method and FCVP were 55 and 31, respectively.

Three experiments were carried out to compare the performance of FCVP with the junction tree algorithm with precise conditional probabilities and the sequential learning in offline learning setting. The results are shown in Table 2. These results demonstrate that FCVP achieves similar performance with the junction tree algorithm and outperforms the sequential learning method in almost all the experiments.

Table 1. Comparative performance - online learning mode

No. of Examples		1000		2000		5000	
Method	Label	#errs	#multi. preds	#errs	#multi. preds	#errs	#multi. preds
JT	B	31	0	73	0	144	0
	T	6	0	25	0	47	0
	L	222	0	427	0	1068	0
	E	30	0	76	0	155	0
SL	B	55	2	103	3	270	2
	T	7	1	25	1	45	1
	L	368	1	705	2	1794	1
	E	39	1	84	1	162	1
FCVP	B	31	7	69	7	143	6
	T	8	7	25	7	45	6
	L	221	7	434	7	1071	6
	E	33	7	77	7	163	6

Table 2. Comparative performance – offline learning mode

Dataset		training set=3000, test set=1000		training set=2000, test set=2000		training set=1000, test set=3000	
Method	Label	#errs	#multi. preds	#errs	#multi. preds	#errs	#multi. preds
JT	B	43	0	68	0	99	0
	T	12	0	19	0	37	0
	L	217	0	437	0	637	0
	E	40	0	77	0	104	0
SL	B	50	0	102	0	182	0
	T	12	0	19	0	37	0
	L	348	0	676	0	1069	0
	E	43	0	71	0	115	0
FCVP	B	38	0	65	0	95	0
	T	12	0	19	0	37	0
	L	221	0	437	0	648	0
	E	40	0	76	0	104	0

6 Conclusions

We present a newly developed probabilistic prediction method using graphical models, fully conditional Venn predictor (FCVP). FCVP can provide well-calibrated probabilistic predictions in the online learning setting. Unlike the sequential learning method, FCVP makes no additional independence assumptions about probability distributions associated with the graphical structure. Empirical results have shown FCVP can achieve good prediction performance over the sequential learning method in both the online and offline learning setting.

Acknowledgements. We thank Volodya Vovk and Tony Bellotti for their discussions and comments. Financial support has been received from the following bodies: MRC through grant S505/65 and Royal Society through grant “Efficient randomness testing of random and pseudorandom number generators”.

References

- [1] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter: Probabilistic Networks and Expert Systems. Statistics for Engineering and Information Science. Springer-Verlag (1999).
- [2] D. Heckerman and D. Geiger: Likelihoods and parameter priors for bayesian networks. Technical Report MSR-TR-95-54, Microsoft Research (1995).
- [3] Finn V. Jensen: An introduction to Bayesian Networks. Taylor and Francis, London, UK (1996).
- [4] S. L. Lauritzen and D. J. Spiegelhalter: Local computations with probabilities on graphical structures and their application to expert systems (with discussion). J. Royal Statist. Soc. **series B**, (50):157–224 (1988).
- [5] Z. Luo and A. Gammerman: Parameter learning in Bayesian belief networks. Proceeding of IPMU’92, 25–28 (1992).
- [6] K. Murphy: The Bayes Net Toolbox for Matlab. Computing Science and Statistics. **33** (2001).
- [7] D. J. Spiegelhalter and S. L. Lauritzen: Sequential updating of conditional probabilities on directed graphical structures. Networks, **20**(5):579–605 (1990).
- [8] V. Vovk, G. Shafer, and I. Nouretdinov: Self-calibrating probability forecasting. In S. Thrun, L. Saul, and B. Schölkopf (ed.), Advances in Neural Information Processing Systems 16. MIT Press, Cambridge, MA (2004).
- [9] V. Vovk, A. Gammerman, and G. Shafer: Algorithmic learning in a random world. Springer-Verlag, (*To appear*) (2005).

A Decision-Based Approach for Recommending in Hierarchical Domains

L.M. de Campos, J.M. Fernández-Luna, M. Gómez, and J.F. Huete

Departamento de Ciencias de la Computación e Inteligencia Artificial,
E.T.S.I. Informática, Universidad de Granada,
18071 – Granada, Spain
{lci, jmfluna, mgomez, jhg}@decsai.ugr.es

Abstract. Recommendation Systems are tools designed to help users to find items within a given domain, according to their own preferences expressed by means of a user profile. A general model for recommendation systems based on probabilistic graphical models is proposed in this paper. It is designed to deal with hierarchical domains, where the items can be grouped in a hierarchy, each item being only contained in another, more general item. The model makes decisions about which items in the hierarchy are more useful for the user, and carries out the necessary computations in a very efficient way.

1 Introduction

In this paper we present an approach to recommending in hierarchical domains that poses this problem as a decision-based task. Broadly speaking, a Recommendation System (RS) provides specific suggestions about items or actions, within a given domain, that may be considered interesting to the user [11].

The input of a RS is normally expressed by means of information given by the user about his/her tastes or preferences, provided either explicitly (by means of a form or a questionnaire) or implicitly (using purchase records, viewing or rating items, visiting links, taking into account the membership to a certain group,...). All the information about the user that the RS stores is known as the *user profile*. The main characteristic of RSs is that they do not only return the requested information, but also try to anticipate user's needs.

There are two main types of RSs: *Content-based and Collaborative filtering RSs*. The former tries to recommend items based exclusively on the user preferences, whereas the latter tries to identify groups of people with tastes similar to that of the user and recommends items that they have liked [1]. A much more exhaustive classification of RSs is found in [8].

In order to place the problem as a decision task we shall use the probabilistic graphical models formalism. Different approaches to the RS are found in the literature: One of these are Bayesian networks (BN) that have been used in this field basically in two areas: as the tool on which the user profile is built [14, 10, 15, 3] and collaborative filtering, employed in classification tasks [2, 9, 12].

A new content-based RS is presented in this paper. In this case, decisions about what to recommend will not only depend on the probability of relevance (as BN-based approaches) of the items but also in terms of the usefulness of these items for the user. The RS will be modeled using a methodology supported by Influence Diagrams (ID) [7]. Particularly, the system has been specifically designed to deal with domains that may be represented as a hierarchy of items.

The application domain is composed of a set of items, which could be divided into two groups: those items used to express the user's preferences (*evidence items*), and those which could be recommended (*advisable items*). The elements of the first group are related to certain items in the second. The advisable items in the domain constitute a hierarchy, in which one item is only contained in/related to another item. As it could be noticed, the structure of compositions of advisable items gives rise to a hierarchical structure in the form of an inverted tree (a forest, more precisely). Another important feature of the proposed model is the way in which inference is performed, facilitating that the model scales well with the number of variables.

The paper is organized in the following way: in Section 2 we describe the general type of application domain that our model is able to deal with, as well as several examples. Then, in Section 3, we shall formalize the ID. Section 4 shows how inference is performed in order to give recommendations to the user on the application domain. Section 5 includes the conclusions and some remarks about further research.

2 Hierarchical Domains: A Description of the Problem

Example 1. Imagine that we are moving to London and we need to rent or buy a house in this city. In this case, probably the first task is to find which are the best areas (the ones which fit our preferences) of the city to move on. Suppose that we would like to use a RS to advise about the different alternatives. Then, when we log on, we need to select a group of services we are interested in, for example, the presence of shops, schools, medical health services, entertainment attractions, etc. Then the system must decide which areas are the best to be recommended to the user. In this case, the items to be recommended are geographical units (streets, postcodes, boroughs, for instance), which are organized hierarchically: London area is divided in boroughs; each borough contains postcodes; and so on. Finally, the smallest units (streets) contain the list of generic services. In this example, the recommended items should be considered as good entry points that satisfies the user preferences, i.e., locations where the user's might look for a house to rent.

Therefore, and considering the example above, services are evidence items; streets, postcodes and boroughs are advisable items. Boroughs are not included in any other item. The basic philosophy of the recommendation operation in a hierarchical structure must consider both:

- Specificity: The system is committed to the greatest possible specificity. If, on the one hand, a particular postcode matches our needs, but mainly because

there is a street having most of the required features, then the RS must show the street and not the postcode. If, on the other hand, the fact is that many streets of the postcode satisfy better the user's request then it is convenient to recommend the postcode as a whole and not to show each particular street. Thus, when a general unit is recommended none of the units included in it will be also recommended by the system.

- **Multiplicity:** The system can provide for each request as many structural units as it deems necessary. In the case of multiple recommendations it is convenient to give a ranking that allows us to select those that fit better our preferences.

Many different domains adapt to these conditions. For instance, Structured Information Retrieval [4]. A document, a book, for instance, is composed of a well-defined structure: the book contains chapters, which are divided into sections. These include subsections, and so on until the last unit that could be considered, for example, paragraphs. In the paragraphs there are words, some of them used to index the document (index terms). When a user formulates a query (a list of terms), he is interested in retrieving not only complete documents dealing with the query matter, but units of them that better match the information need. For example, a paragraph, a section or even a complete chapter may be possible answers of the system. A different example can be found if we consider a tourism recommendation system that advises a user about the different regions or countries that he could like to visit, according to the type of tourist attractions in which he is interested. The items to be recommended are geographical units (countries, regions, provinces and cities, for instance), which are organized hierarchically: a country is divided in regions; each region contains provinces; and so on. Finally, the smallest units, i.e. cities, contain the list of generic tourist attractions (for example, science museums, castles, cathedrals,...). Another example can be stated if we consider hierarchical categorization (for instance, www.yahoo.com). In this case, the hierarchy of categories represents the advisable items (for example, sports contains football which contains "Champions League") and the evidence items are the set of features used to represent a specific category. Now, the problem is: given a new document to try to assign the set of categories that better describes its contents.

3 Model's Specification

To construct the RS we use an approach posing the problem as a decision problem that will be modeled using ID's. First of all, we shall describe the different kinds of nodes in the ID and how they are related to each other.

- **Chance Nodes:** Two types of chance nodes can be found:
 - The set of items by which the user can express his preferences named evidence items or features (the set of services in Example 1), represented by the set $\mathcal{F} = \{F_1, F_2, \dots, F_l\}$. In this paper we consider that each

node F_k , has associated a random binary variable, which can take its values from the set $\{f_k^-, f_k^+\}$, representing that the feature do not match or match, respectively, the user's preferences¹. These nodes have been represented by ellipses in Fig. 1.

- The set of items that may be shown (recommended) to the user, i.e., advisable items (geographical units in Example 1). Since the problem is modeled as a hierarchical structure, these nodes will be referred as *structural units*. There are two types of these units: *basic structural units*, those which only are related to evidence items (streets in Example 1), and *complex structural units*, that are composed of other basic or complex units² (boroughs and post codes in Example 1). The notation for these nodes is $\mathcal{U}_b = \{B_1, B_2, \dots, B_m\}$ and $\mathcal{U}_c = \{S_1, S_2, \dots, S_n\}$, respectively. Therefore, the set of all structural units is $\mathcal{U} = \mathcal{U}_b \cup \mathcal{U}_c$. In this text, B or B_i represents a basic structural unit, and S or S_i represents a complex structural unit. Generic structural units (either basic or complex) will be denoted as U_i or U . Each node B_i or S_j (generically U_i) has associated a random binary variable, which can take its values from the sets $\{b_i^-, b_i^+\}$ or $\{s_j^-, s_j^+\}$ (generically $\{u_i^-, u_i^+\}$) representing that the unit is not relevant or is relevant, respectively, to satisfy the user preferences. These nodes have been represented by circles in Fig. 1.
- **Decision Nodes:** These nodes model the decision variables, representing the possible alternatives available to the RS. In our case, we consider one decision node, R_i , for each structural unit $U_i \in \mathcal{U}$. R_i represents the decision variable related to whether or not to return the advisable item U_i to the user. The two different values for R_i are r_i^+ and r_i^- , meaning 'recommend U_i ' and 'do not recommend U_i ', respectively. These nodes are represented by boxes in Fig. 1.
- **Utility Nodes:** These nodes are used to measure the value of utility of the corresponding decisions. Since one of our objectives is to achieve specificity, we need to express the utility values considering a variable and its context. Thus, we shall use a utility node $V_{i,j}$ for each pair of variables (U_i, U_j) being U_j a unit directly included in U_i . These nodes are diamonds in Fig. 1.

We shall describe the topology of the ID, starting with the relationships between chance nodes. In this case, there is an arc from any given node (either feature or structural unit) to the particular structural unit it belongs to. With these arcs we are expressing the fact that the relevance of a given structural unit to the user will depend on the relevance values of the different elements (units or features) that comprise it. It should be noted that with this criteria we obtain a hierarchical topology, where feature nodes (evidence items) have no parent, that

¹ Although in this paper we consider only bivaluated evidence items, the system can handle evidence items with a finer granularity scale in order to get finer information when the user's preferences are elicited.

² Notice that, if it is necessary, evidence items can be associated to a complex advisable item through a fictitious basic advisable item.

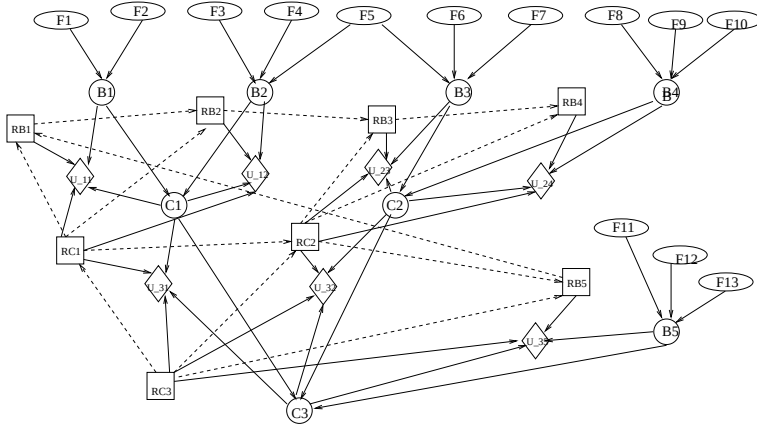


Fig. 1. Topology of the Influence Diagram

represents properly the hierarchical structure of the domain (see Example 1). Thus, when it is convenient we will use graph terminology, for example, given a node U_i we can talk about the child of U_i , $C(U_i)$, being the unique unit which directly contains U_i and parents of U_i , $Pa(U_i)$, being the set of units that directly comprise it.

The second step will be to describe those arcs pointing to an utility node $V_{i,j}$. These arcs are employed to indicate which variables have a direct influence on the desirability of a given decision, i.e., the profit obtained will depend on the value of these variables. Note that our objective is to give recommendations taking into account the context. So that, we shall consider that the utility function $V_{i,j}$ will depend on the relevance value of the structural unit U_i and also on the relevance value of the structural unit included in it, U_j . Obviously, the utility values will also depend on the decisions of showing or not these structural units, R_i and R_j . We shall also consider a utility node, denoted by Σ , that represents the joint utility of the whole model. It contains all the utility nodes as its parents. This node has not been presented in Fig. 1. These arcs represent that the joint utility of the model will depend (additively) on the values of the individual utilities.

Finally, we shall also consider arcs pointing to decision nodes R_i , $\forall i = 1, \dots, |\mathcal{U}|$. They would indicate that the value of the source node is available when the decision is made. In this case, and taking into account the hierarchical structure of the model, it will be convenient not to recommend a unit U_i if we have previously recommended a unit U_k that contains it, i.e., $U_i \subset U_k$. This restriction imposes a partial ordering between decision nodes: the first decision will be the one represented by the most general structural unit. Then, for each decision R_i related to the structural unit U_i , we include the arc that connects $R_{C(U_i)}$ with R_i . Finally, and in order to complete the ordering between decision nodes, we include arcs connecting decision nodes from left to right if they are in the same level of the hierarchy and an arc that connect the last node in one level (rightmost decision node) with the first node (leftmost decision node) in

the immediate upper level. All arcs between decision nodes are represented with dashed lines in Fig. 1. Note that no arc points from decision nodes to chance nodes. This implies that the relevance of a structural unit will not depend on the decision of showing (recommend) or not showing any structural unit.

The presented topology implies the following independence relationships: a complex structural unit S is conditionally independent on any other element which does not contain S , given the structural units that compose S ; a basic structural unit B is conditionally independent on any other element which does not contain B , given the features contained in B ; a feature F is marginally independent of any other feature. This last assumption (restrictive in some domains) could be relaxed to include relationships between evidence items [6].

To complete the specification of the model, the numerical values for the conditional probabilities and utilities have to be assessed. The required values are, on the one hand, $p(f_k^+)$, $p(b_i^+ | pa(B_i))$, $p(s_j^+ | pa(S_j))$, for every node in $\mathcal{F}$, $\mathcal{U}_b$ and $\mathcal{U}_c$, respectively, and every configuration of the corresponding parent sets ($pa(X)$ denotes a configuration or instantiation of the parent set of X , $Pa(X)$); on the other hand, for each node $V_{i,j}$ we need to assess 2^4 numerical values representing the utilities for the corresponding combination of its parents. All these values should be estimated when constructing the RS.

4 Inference

In order to use the proposed model, and therefore to recommend structural units, first we have to recall that a recommendation operation is defined as the process of showing to the user the units which best describe her/his preferences. The user's requests are expressed by means of a query, $\mathcal{Q}$, representing, for instance, that he/she is interested in a location having *nursery and primary schools, hospitals and sport centers* in its surroundings. The RS could recommend the best locations as the street "Abbey Road" or the postcode "E1". Formally, let $\mathcal{Q} \subseteq \mathcal{F}$ be the set of features whose relevance values are known (each feature $F_i \in \mathcal{Q}$ is instantiated to either f_i^+ or f_i^-) and let q be the corresponding configuration (i.e., the user profile).

Therefore, solving the ID implies the computation of the expected utility of each of the possible decision strategies, considering both specificity and multiplicity, and selecting the strategy with the highest expected utility. In this case we should take into account that the problem is highly asymmetric in the sense that whenever we decide to show a structural unit we do not need to make any decision about all the structural units included in it. Therefore, the number of strategies being considered will be reduced considerably. Nevertheless, even considering such restriction we will need to study a huge number of valid strategies. For example, consider a simple model with a general unit that includes three other units and each one including also three basic structural units. In this case, the number of valid strategies to be considered is 730. In general, we can say that the number of valid strategies is doubly-exponential in the number of basic advisable items.

Note that our purpose is not only to make decisions about what to recommend but also to give a ranking of those units. In the case of an optimal strategy having multiple recommendations the simplest way to do it is to show them in decreasing order of the utility of recommending U_i , $EU(r_i^+|q)$ ³.

In this case, because hierarchical models might contain a large number of structural units (it is possible to have thousand of units) and that a unit might have hundreds of units as its parents, it is not possible to use classical algorithms to solve ID's[13], mainly due to the computation cost of the decision tables. Therefore, and in order to ensure an efficient recommendation system being able to scale well in the size of the hierarchical domain considered, we propose to use a two steps approach:

- Probability Inference: This first step computes the posterior probabilities of relevance for all the structural units $U \in \mathcal{U}$, $p(u^+|q)$. In order to compute these values it is enough to consider the BN that is subsumed in the ID. Left hand side of Fig. 2 represents the BN for the model in Fig. 1. In subsection 4.1 we will give some guidelines to perform this process efficiently.
- Decision Making: Then, taking into account these probability values, we compute the final strategy by solving a set of simplified ID's, one for each complex structural unit (see right hand side of Fig. 2). With this simplification we can reduce considerably the computation cost of the optimal strategy. Subsection 4.2 presents the proposed approach.

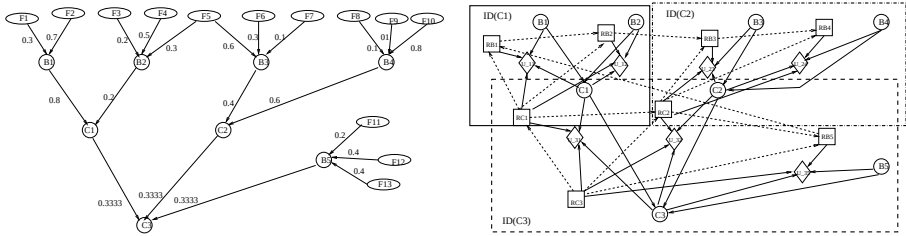


Fig. 2. Two step inference process

4.1 Probability Inference

As we have seen in the previous section, in order to provide the user with an ordered list of recommendations, we have to be able to compute the posterior probabilities of relevance of all the structural units $U \in \mathcal{U}$, $p(u^+|q)$. In the context of RSs, the number of features and structural units considered may be quite large (thousands or even hundred thousands). Moreover, the topology of the BN

³ Other options would also be possible, for example to rank the units using the difference between both expected utilities, $EU(r_i^+|q) - EU(r_i^-|q)$.

contains multiple pathways connecting nodes (because features may be associated to different basic structural units) and possibly nodes with a great number of parents (so that it can be quite difficult to assess and store the required conditional probability tables). For these reasons we propose the use of a canonical model to represent the conditional probabilities [5], which will allow us to design a very efficient inference procedure. We have to consider the conditional probabilities for the basic structural units, having a subset of features as their parents, and for the complex structural units, having other structural units as their parents. We define these probabilities as follows:

$$\forall B \in \mathcal{U}_b, p(b^+ | pa(B)) = \sum_{F \in R(pa(B))} w(F, B), \quad (1)$$

$$\forall S \in \mathcal{U}_c, p(s^+ | pa(S)) = \sum_{U \in R(pa(S))} w(U, S), \quad (2)$$

where $w(F, B)$ is a weight associated to each feature F belonging to the basic unit B , $w(U, S)$ is a weight measuring the importance of the unit U within S , with $w(F, B) \geq 0$, $w(U, S) \geq 0$, $\sum_{F \in Pa(B)} w(F, B) \leq 1$, and $\sum_{U \in Pa(S)} w(U, S) \leq 1$. In either case $R(pa(U))$ is the subset of parents of U (features for B , and either basic or complex units for S) that are relevant in the configuration $pa(U)$, i.e., $R(pa(B)) = \{F \in Pa(B) \mid f^+ \in pa(B)\}$ and $R(pa(S)) = \{U \in Pa(S) \mid u^+ \in pa(S)\}$. So, the more parents of U relevant the greater the probability of relevance of U .

As we can see [5], the posterior probabilities can be computed efficiently using the following formula, where the posterior probabilities of the basic units are obtained directly and the posterior probabilities of the complex units can be calculated in a top-down manner, starting from the basic units.

$$\begin{aligned} \forall B \in \mathcal{U}_b, p(b^+ | q) &= \sum_{F \in Pa(B) \setminus \mathcal{Q}} w(F, B) p(f^+) + \sum_{F \in Pa(B) \cap R(q)} w(F, B), \\ \forall S \in \mathcal{U}_c, p(s^+ | q) &= \sum_{U \in Pa(S)} w(U, S) p(u^+ | q). \end{aligned} \quad (3)$$

4.2 Making Decisions

In this section, we are going to make decisions about what advisable items will be recommended to the user. To obtain the optimal strategy, a compatible strategy with the maximal expected utility, we have to compute an exponential number of valid strategies. In this case, we have to consider two different situations that will help us to prune the search: on the one hand, it seems natural that whenever the evidence (the query) has no effect on a particular unit we shall decide “not to recommend” the unit and none of the units included in it; on the other hand, and considering the specificity requirement, if we decide “to recommend” a unit, none of the units included in it will be also recommended.

Nevertheless, considering the high dimensionality of the problem and that we will need to compute an exponential number of compatible strategies, it is not feasible (considering both the size and the time needed to perform the computations) to study all the possible alternatives, even for small problems. Therefore, we propose to split the above model into a set of local decision problems, one for each complex structural unit, that will be solved independently. Each local influence diagram, ID_{U_i} , will consider all the relationships relating a variable U_i with the set of parents of U_i , $Pa(U_i)$, (see right hand side of Fig. 2).

To obtain the final strategy, we propose to start from the most general complex units of the ID and using a bottom-up approach make decisions at each one of the levels of the hierarchy with the information that can be computed locally. But, in a general case, by solving the local IDs, we have two decisions for each complex structural unit U_i (except the most general one); one when considering ID_{U_i} that includes the relationships with the units contained by U_i and the other when considering $ID_{C(U_i)}$ that includes the relationships with the unique structural unit containing U_i and all the units contained by $C(U_i)$. Now, we are going to consider how they are related:

- Decision at $ID_{C(U_i)}$ is “to recommend” and decision at ID_{U_i} is “to recommend”: In this case, there is no doubt and it can be considered convenient to recommend the unit U_i .
- Decision at $ID_{C(U_i)}$ is “to recommend” and decision at ID_{U_i} is “not recommend”: In this case, on the one hand, we have that the decision of recommending is done when considering the information given by the set of siblings of U_i (probably because it is more relevant than the rest). But, on the other hand, when we are considering how U_i is related with its parents, decision is not to recommend (probably because it is preferable to recommend some of its parents). Therefore, in this case, the final decision might be “not to recommend” node U_i .
- Decision at $ID_{C(U_i)}$ is “not to recommend” and decision at ID_{U_i} is “to recommend”: This is the opposite of the previous one, and using similar argument we shall decide “to recommend” unit U_i .
- Decision at $ID_{C(U_i)}$ is “not to recommend” and decision at ID_{U_i} is “not to recommend”: In this case, it obvious that we will make the decision of “not to recommend” unit U_i .

These facts will be essential since we can say that the decision about unit U_i will only depend on the strategy of maximum expected utility computed when considering the influence diagram ID_{U_i} , i.e., the one considering the relationships with the units contained by the node U_i . Thus, if decision for unit U_i is “to recommend” we will stop the process, otherwise we will recursively study the decision for each structural unit in $Pa(U_i)$.

Solving the Simplified Influence Diagrams: Now, we will focus on ID_{U_i} and the problem of finding the decision of maximum expected utility for node U_i . Considering how variables are related to each other in the model, to compute this strategy using classical algorithms [13] we will need to work with final

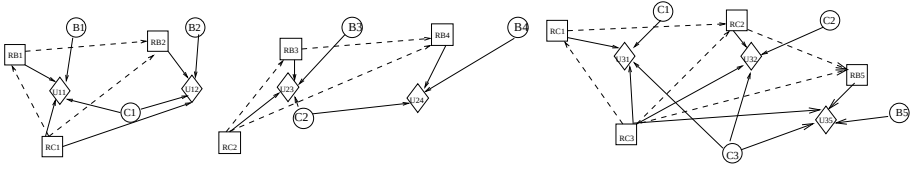


Fig. 3. Local Influence Diagrams

potentials including all chance and decision nodes and therefore with size equals to $2^{(|Pa(U_i)|+1)}$, being $|Pa(U_i)|$ the number of units in $Pa(U_i)$. Even considering small problems, with units having tens of parents, the process becomes prohibitive. The situation becomes worse if we expect a fast answer of the RS.

To solve this problem we propose to approximate the solution by using a simpler ID where we have removed all the edges connecting chance nodes (see Fig. 3). Thus, all the structural units $U \in \mathcal{U}$ become roots nodes and will store the computed probability of relevance given the query (obtained using eq. 3), i.e., they will use the values $p(u^+|q)$ and $p(u^-|q)$ as their marginal probability. Note that with this approach the dependence relationships between chance variables have been previously considered when computing the a posteriori probability of relevance. For each chance variable, U_i , we include a decision node R_i and for each pair of variables, U_i and U_j (with U_j in $Pa(U_i)$), a utility node $V_{i,j}$ is also included. Finally we add the same set of arcs pointing to decision and utility nodes than in the original model.

Now, taking into account the topology of these local ID's, we can compute the decision of maximum expected utility for a unit U_i efficiently, with a cost (in size and time) linear with the number of parents of U_i , as indicate the following expressions:

$$EU(r_i^+) = \sum_{U_j \in Pa(U_i)} \max \left\{ \begin{array}{l} \sum_{\substack{u_j \in \{u_j^-, u_j^+\}, \\ u_i \in \{u_i^-, u_i^+\}}} V_{i,j}(u_i, u_j, r_i^+, r_j^+) p(u_j|q) p(u_i|q), \\ \sum_{\substack{u_j \in \{u_j^-, u_j^+\}, \\ u_i \in \{u_i^-, u_i^+\}}} V_{i,j}(u_i, u_j, r_i^+, r_j^-) p(u_j|q) p(u_i|q) \end{array} \right\} \quad (4)$$

and similarly for $EU(r_i^-)$ (replacing r_i^+ by r_i^- in the previous equation).

Finally, all the recommended structural units will be presented to the user after sorting them in a decreasing order of their expected utility.

Example 2. To illustrate the behavior of the proposed model, let us consider the example in Fig. 1. To set quantitative values we use the scheme proposed in subsection 4.1, where the used weights, $W(\cdot, \cdot)$, are displayed in the BN at left hand side of Fig. 2. The prior probabilities of all the evidence items have been set to 0.5. Finally, all the utility nodes have the same set of values. In this example the values for each configuration of $V_{i,j} = \{U_i, U_j, R_i, R_j\}$, where $U_i = C(U_j)$ and a given configuration $v(u_i^+, u_j^+, r_i^-, r_j^-)$ is represented by means of $v(++--)$, are:

$$\begin{aligned}
v(++++) &= 0 & v(+++-) &= 5 & v(++-+) &= 0 & v(++--) &= -5 \\
v(+ -++) &= 0 & v(+ -+-) &= 0 & v(+ - -+) &= -15 & v(+ - --) &= -15 \\
v(- +++) &= -15 & v(- + -+) &= -15 & v(- + - -) &= 15 & v(- + --) &= 0 \\
v(- -++) &= -15 & v(- -+-) &= -15 & v(- - -+) &= -15 & v(- - --) &= 15
\end{aligned}$$

In order to illustrate the behavior of the final approach that considers local computations, we will compare its results with the ones obtained when considering the complete ID. First of all, it must be noticed that both models propose not to recommend any unit when there is no evidence, as it could be expected. In the next table, the results obtained when considering the complete ID (see Fig. 1) for the queries $Q_1 = \{f_2^+, f_5^+, f_{10}^+\}$, $Q_2 = \{f_2^+, f_6^+, f_{10}^+\}$ and $Q_3 = \{f_2^+, f_5^-, f_{10}^+\}$ are displayed; second column presents those structural units to be recommended in the optimal strategy, sorted by their respective expected utilities (in brackets) and third column presents the a posteriori probability values for structural nodes.

Q	Optimal Strategy	C3	C1	C2	B1	B2	B3	B4	B5
Q_1	$r_{c2}^+(1.12) \ r_{c1}^+(-1.35)$	0.703	0.750	0.86	0.85	0.65	0.80	0.90	0.50
Q_2	$r_{b1}^+(0.94) \ r_{c2}^+(0.25)$	0.658	0.675	0.80	0.85	0.50	0.65	0.90	0.50
Q_3	$r_{b4}^+(2.82) \ r_{b1}^+(1.89)$	0.593	0.600	0.62	0.85	0.35	0.20	0.90	0.50

It is interesting to see how the system decides to show a complex structural unit even considering that it is not the more relevant node to the query. This is the case of node C2 for queries Q_1 and Q_2 . These queries also illustrate some cases where the system decides to recommend some more specific structural units, for example it does not recommend C3 in any query and also it is the case of B1 and B4 in query Q_3 .

Next table shows the results obtained when using local ID's. Second, third and fourth columns present the computed optimal strategies for each ID and fifth column shows the structural units finally recommended by the system sorted by their expected utility. In these cases, the final performance of the system is similar than before. Note that for all the queries ID_{C3} shall propose to recommend C1 and C2, but in some cases these decisions will be revoked when considering the strategies proposed by ID_{C1} and ID_{C2} and therefore recommending more basic structural units.

Q	ID_{C1}	ID_{C2}	ID_{C3}	System Output
Q_1	$r_{c1}^+, r_{b1}^-, r_{b2}^-$	$r_{c2}^+, r_{b3}^-, r_{b4}^-$	$r_{c3}^-, r_{c1}^+, r_{c2}^+, r_{b5}^-$	$r_{c2}^+(3.11) \ r_{c1}^+(-1.87)$
Q_2	$r_{c1}^-, r_{b1}^+, r_{b2}^-$	$r_{c2}^+, r_{b3}^-, r_{b4}^-$	$r_{c3}^-, r_{c1}^+, r_{c2}^+, r_{b5}^-$	$r_{b1}^+(1.89) \ r_{c2}^+(0.2)$
Q_3	$r_{c1}^-, r_{b1}^+, r_{b2}^-$	$r_{c2}^-, r_{b3}^-, r_{b4}^+$	$r_{c3}^-, r_{c1}^+, r_{c2}^+, r_{b5}^-$	$r_{b4}^+(3.63) \ r_{b1}^+(2.85)$

5 Concluding Remarks

A general, ID-based model for recommendation systems in hierarchical domains has been proposed in this paper. Taking into account efficiency considerations and that the evaluation of a whole influence diagram in this context, by means

of classic algorithms, can not be afforded, we propose a two stage inference mechanism to cope efficiently with this problem. In the first step, the posterior probabilities of chance nodes from the underlying BN are computed using a very efficient method based on canonical models. A second step removes the arcs joining these nodes, incorporates these posterior probabilities, and considers the existing influence diagram, which is viewed as several smaller influence diagrams that could be solved locally with the aim of giving the user the corresponding recommendations. Moreover, not all of them have to be solved, because it depends on the decisions taken in previous evaluations.

Taking into account the huge dimension of the problem, we think that using approximations is the only way to cope with it. As future works, we are planning to evaluate the model with real problems, involving real users to determine the quality of the recommendations provided. We are also studying mechanisms to incorporate in it user profiles and collaborative filtering.

Acknowledgments. This work has been supported by the Spanish Fondo de Investigación Sanitaria, under Project PI021147.

References

1. M. Balabanovic and Y. Shoham. 1997. Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72.
2. J.S. Breese, D. Heckerman, and C. Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52.
3. C.J. Butz. 2002. Exploiting contextual independencies in web search and user profiling. In *Proc. of World Congress on Computational Intelligence*, pages 1051–1056.
4. F. Crestani, L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. 2003. A multi-layered Bayesian network model for structured document retrieval. *Lecture Notes in Artificial Intelligence*, 2711:74–86.
5. L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. 2003. The BNR model: Foundations and performance of a Bayesian network retrieval model. *International Journal of Approximate Reasoning*, 34:265–285.
6. L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. 2004. Clustering terms in the Bayesian network retrieval model: a new approach with two term-layers. *Applied Soft Computing*, 4:149–158.
7. F.V. Jensen. 2001. *Bayesian Networks and Decision Graphs*. Springer Verlag.
8. S. Kangas. 2002. Collaborative filtering and recommendation systems. VTT Information Technology, Research report TTE4-2001-35.
9. K. Miyahara and J. Pazzani. 2000. Collaborative filtering with the simple Bayesian classifier. In *Proc. of the Pacific Rim International Conference on Artificial Intelligence*, pages 679–689.
10. P. Nokelainen, H. Tirri, M. Miettinen, and T. Silander. 2002. Optimizing and profiling users online with Bayesian probabilistic modelling. In *Proceedings of the NL Conference*.

11. P. Resnick and H.R. Varian. 1997. Recommender systems. *Communications of the ACM*, 40(3):56–58.
12. V. Robles, P. Larrañaga, J.M. Peña, O. Marbán, J. Crespo, and M.S. Pérez. 2003. Collaborative filtering using interval estimation naive Bayes. *Lecture Notes in Artificial Intelligence*, 2663:46–53.
13. P.P. Shenoy, 1993. A new method for representing and solving Bayesian decision problems, *Artificial Intelligence Frontiers in Statistics: AI and Statistics* 119-138, Chapman and Hall, London.
14. S.N. Schiaffino and A. Amandi. 2000. User profiling with case-based reasoning and Bayesian network. *Proc. of the Iberoamerican Conf. of Artificial Intelligence*, 12–21.
15. S. Wong and C. Butz. 2000. A Bayesian approach to user profiling in information retrieval. *Technology Letters*, 4(1):50–56.

Scalable, Efficient and Correct Learning of Markov Boundaries Under the Faithfulness Assumption

Jose M. Peña¹, Johan Björkegren², and Jesper Tegnér^{1,2}

¹ Computational Biology, Department of Physics and Measurement Technology, Linköping University, Sweden

² Center for Genomics and Bioinformatics, Karolinska Institutet, Sweden

Abstract. We propose an algorithm for learning the Markov boundary of a random variable from data without having to learn a complete Bayesian network. The algorithm is correct under the faithfulness assumption, scalable and data efficient. The last two properties are important because we aim to apply the algorithm to identify the minimal set of random variables that is relevant for probabilistic classification in databases with many random variables but few instances. We report experiments with synthetic and real databases with 37, 441 and 139352 random variables showing that the algorithm performs satisfactorily.

1 Introduction

Probabilistic classification is the process of mapping an assignment of values to some random variables $\mathbf{F}$, the features, into a probability distribution for a distinguished random variable C , the class. Feature subset selection (FSS) aims to identify the minimal subset of $\mathbf{F}$ that is relevant for probabilistic classification. The FSS problem is worth of study for two main reasons. First, knowing which features are relevant and, thus, which are irrelevant is important in its own because it provides insight into the domain at hand. Second, if the probabilistic classifier is to be learnt from data, then knowing the relevant features reduces the dimension of the search space.

In this paper, we are interested in solving the FSS problem following the approach proposed in [9, 10, 11]: Since the Markov boundary of C , $MB(C)$, is defined as any minimal subset of $\mathbf{F}$ such that C is conditionally independent of the rest of $\mathbf{F}$ given $MB(C)$, then $MB(C)$ is a solution to the FSS problem. Under the faithfulness assumption, $MB(C)$ can be obtained by first learning a Bayesian network (BN) for $\{\mathbf{F}, C\}$: In such a BN, $MB(C)$ is the union of the parents and children of C and the parents of the children of C [6]. Unfortunately, the existing algorithms for learning BNs from data do not scale to databases with thousands of features [3, 10, 11] and, in this paper, we are interested in solving the FSS problem for databases with thousands of features but with many less instances. Such databases are common in bioinformatics and medicine.

In this paper, we propose an algorithm for learning MBs from data and prove its correctness under the faithfulness assumption. Our algorithm scales to databases with thousands of features because it does not require learning a complete BN. Furthermore, our algorithm is data efficient because the tests of conditional independence that it performs are not conditioned on unnecessarily large sets of features. In Section 3, we review other existing scalable algorithms for learning MBs from data and show that they are either data inefficient or incorrect. We describe and evaluate our algorithm in Sections 4 and 5, respectively. We close with some discussion in Section 6. We start by reviewing BNs in Section 2.

2 Preliminaries on BNs

The following definitions and theorems can be found in most books on BNs, e.g. [6, 8]. We assume that the reader is familiar with graph and probability theories. We abbreviate if and only if by iff, such that by st, and with respect to by wrt.

Let $\mathbf{U}$ denote a nonempty finite set of discrete random variables. A Bayesian network (BN) for $\mathbf{U}$ is a pair (G, θ) , where G is an acyclic directed graph (DAG) whose nodes correspond to the random variables in $\mathbf{U}$, and θ are parameters specifying a conditional probability distribution for each node X given its parents in G , $p(X|Pa_G(X))$. A BN (G, θ) represents a probability distribution for $\mathbf{U}$, $p(\mathbf{U})$, through the factorization $p(\mathbf{U}) = \prod_{X \in \mathbf{U}} p(X|Pa_G(X))$. In addition to $Pa_G(X)$, two abbreviations that we use are $PC_G(X)$ for the parents and children of X in G , and $ND_G(X)$ for the non-descendants of X in G .

Any probability distribution p that can be represented by a BN with DAG G , i.e. by a parameterization θ of G , satisfies certain conditional independencies between the random variables in $\mathbf{U}$ that can be read from G via the d-separation criterion, i.e. if $d\text{-sep}_G(\mathbf{X}, \mathbf{Y}|\mathbf{Z})$, then $\mathbf{X} \perp_p \mathbf{Y}|\mathbf{Z}$ with $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ three mutually disjoint subsets of $\mathbf{U}$. We say that $d\text{-sep}_G(\mathbf{X}, \mathbf{Y}|\mathbf{Z})$ holds when for every undirected path in G between a node in $\mathbf{X}$ and a node in $\mathbf{Y}$ there exists a node Z in the path st either (i) Z does not have two incoming edges in the path and $Z \in \mathbf{Z}$, or (ii) Z has two incoming edges in the path and neither Z nor any of its descendants in G is in $\mathbf{Z}$. The d-separation criterion in G enforces the local Markov property for any probability distribution p that can be represented by a BN with DAG G , i.e. $X \perp_p (ND_G(X) \setminus Pa_G(X))|Pa_G(X)$. A probability distribution p is said to be faithful to a DAG G when $\mathbf{X} \perp_p \mathbf{Y}|\mathbf{Z}$ iff $d\text{-sep}_G(\mathbf{X}, \mathbf{Y}|\mathbf{Z})$.

Theorem 1. *If a probability distribution p is faithful to a DAG G , then (i) for each pair of nodes X and Y in G , X and Y are adjacent in G i $X \not\perp_p Y|\mathbf{Z}$ for all $\mathbf{Z}$ st $X, Y \notin \mathbf{Z}$, and (ii) for each triplet of nodes X , Y and Z in G st X and Y are adjacent to Z but X and Y are non-adjacent, $X \rightarrow Z \leftarrow Y$ is a subgraph of G i $X \not\perp_p Y|\mathbf{Z}$ for all $\mathbf{Z}$ st $X, Y \notin \mathbf{Z}$ and $Z \in \mathbf{Z}$.*

Let p denote a probability distribution for $\mathbf{U}$. The Markov boundary of a random variable $X \in \mathbf{U}$, $MB_p(X)$, is defined as any minimal subset of $\mathbf{U}$ st $X \perp_p (\mathbf{U} \setminus (MB_p(X) \cup \{X\}))|MB_p(X)$.

Table 1. *IAMB*

IAMB(T, D)

```

/* add true positives to MB */
1  $MB = \emptyset$ 
2 repeat
3    $Y = \arg \max_{X \in \mathcal{U} \setminus (MB \cup \{T\})} dep_D(X, T|MB)$ 
4   if  $Y \not\perp_D T|MB$  then
5      $MB = MB \cup \{Y\}$ 
6 until  $MB$  does not change
/* remove false positives from MB */
7 for each  $X \in MB$  do
8   if  $X \perp_D T|(MB \setminus \{X\})$  then
9      $MB = MB \setminus \{X\}$ 
10 return  $MB$ 

```

Theorem 2. *If a probability distribution p is faithful to a DAG G , then $MB_p(X)$ for each node X is unique and is the union of $PC_G(X)$ and the parents of the children of X in G .*

We denote $MB_p(X)$ by $MB_G(X)$ when p is faithful to a DAG G .

3 Previous Work on Scalable Learning of MBs

In this section, we review two algorithms for learning MBs from data that Tsamardinos et al. introduce in [9, 10, 11, 12], namely the incremental association Markov blanket (*IAMB*) algorithm and the max-min Markov blanket (*MMMB*) algorithm. To our knowledge, these are the only algorithms that have been experimentally shown to scale to databases with thousands of features. However, we show that *IAMB* is data inefficient and *MMMB* incorrect. In the algorithms, $X \not\perp_D Y|\mathbf{Z}$ ($X \perp_D Y|\mathbf{Z}$) denotes conditional (in)dependence wrt a learning database D , and $dep_D(X, Y|\mathbf{Z})$ is a measure of the strength of the conditional dependence wrt D . In particular, the algorithms run a test with the G^2 statistic in order to decide on $X \not\perp_D Y|\mathbf{Z}$ or $X \perp_D Y|\mathbf{Z}$ [8], and use the negative p-value of the test as $dep_D(X, Y|\mathbf{Z})$. Both algorithms are based on the assumption that D is faithful to a DAG G , i.e. D is a sample from a probability distribution p faithful to G .

3.1 IAMB

Table 1 outlines *IAMB*. The algorithm receives the target node T and the learning database D as input and returns $MB_G(T)$ in MB as output. The algorithm works in two steps. First, the nodes in $MB_G(T)$ are added to MB (lines 2-6). Since this step is based on the heuristic at line 3, some nodes not in $MB_G(T)$ may be added to MB as well. These nodes are removed from MB in the second step (lines 7-9). Tsamardinos et al. prove the correctness of *IAMB* under some assumptions.

Theorem 3. *Under the assumptions that the learning database D is an independent and identically distributed sample from a probability distribution p faithful to a DAG G and that the tests of conditional independence and the measure of conditional dependence are correct, the output of $IAMB(T, D)$ is $MB_G(T)$.*

The assumption that the tests of conditional independence and the measure of conditional dependence are correct should be read as follows: $X \perp\!\!\!\perp_D Y | \mathbf{Z}$ and $dep_D(X, Y | \mathbf{Z}) = -1$ if $X \perp\!\!\!\perp_p Y | \mathbf{Z}$, and $X \not\perp\!\!\!\perp_D Y | \mathbf{Z}$ and $dep_D(X, Y | \mathbf{Z}) = 0$ otherwise. In order to maximize accuracy in practice, $IAMB$ performs a test if it is reliable and skips it otherwise. Following the approach in [8], $IAMB$ considers a test to be reliable when the number of instances in D is at least five times the number of degrees of freedom in the test. This means that the number of instances required by $IAMB$ to identify $MB_G(T)$ is at least exponential in the size of $MB_G(T)$, because the number of degrees of freedom in a test is exponential in the size of the conditioning set and some tests will be conditioned on at least $MB_G(T)$. However, depending on the topology of G , it can be the case that $MB_G(T)$ can be identified by conditioning on sets much smaller than $MB_G(T)$, e.g. if G is a tree (see Sections 3.2 and 4). Therefore, $IAMB$ is data inefficient because its data requirements can be unnecessarily high. Note that this reasoning applies not only to the G^2 statistic but to any other statistic as well. Tsamardinos et al. are aware of this drawback and describe some variants of $IAMB$ that alleviate it, though they do not solve it, while still being scalable and correct: The first and second steps can be interleaved (*interIAMB*), and the second step can be replaced by the PC algorithm [8] (*interIAMBnPC*). Finally, as Tsamardinos et al. note, $IAMB$ is similar to the grow-shrink (GS) algorithm [5]. In fact, the only difference is that GS uses a simpler heuristic at line 3: $Y = \arg \max_{X \in \mathbf{U} \setminus (MB \cup \{T\})} dep_D(X, T | \emptyset)$. GS is correct under the assumptions in Theorem 3, but it is data inefficient for the same reason as $IAMB$.

3.2 MMBB

$MMMB$ aims to reduce the data requirements of $IAMB$ while still being scalable and correct. $MMMB$ identifies $MB_G(T)$ in two steps: First, it identifies $PC_G(T)$ and, second, it identifies the rest of the parents of the children of T in G . $MMMB$ uses the max-min parents and children ($MMPC$) algorithm to solve the first step. Table 2 outlines $MMPC$. The algorithm receives the target node T and the learning database D as input and returns $PC_G(T)$ in PC as output. $MMPC$ is similar to $IAMB$, with the exception that $MMPC$ considers any subset of the output as the conditioning set for the tests that it performs and $IAMB$ only considers the output. Tsamardinos et al. prove that, under the assumptions in Theorem 3, the output of $MMPC$ is $PC_G(T)$. We show that this is not always true. The flaw in the proof is the assumption that if $X \notin PC_G(T)$, then $X \perp\!\!\!\perp_p T | \mathbf{Z}$ for some $\mathbf{Z} \subseteq PC_G(T)$ and, thus, any node not in $PC_G(T)$ that enters PC at line 7 is removed from it at line 11. This is not always true for the descendants of T . This is illustrated by running $MMPC(T, D)$ with D faithful to the DAG (a) in Table 2. Neither P nor R enters PC at line 7 because $P \perp\!\!\!\perp_p T | \emptyset$ and $R \perp\!\!\!\perp_p T | \emptyset$. Q enters PC because $Q \not\perp\!\!\!\perp_p T | \mathbf{Z}$ for all $\mathbf{Z}$ st $Q, T \notin \mathbf{Z}$. S enters PC

Table 2. *MMPC* and *MMMB*

<u><i>MMPC</i>(<i>T</i>, <i>D</i>)</u>	(a)
<pre> /* add true positives to PC */ 1 PC = 2 repeat 3 for each X ∈ U (PC ∪ T) do 4 Sep[X] = arg min_{Z ⊆ PC} dep_D(X, T Z) 5 Y = arg max_{X ∈ U \ (PC ∪ {T})} dep_D(X, T Sep[X]) 6 if Y ⊥_D T Sep[Y] then 7 PC = PC ∪ Y 8 until PC does not change /* remove false positives from PC */ 9 for each X ∈ PC do 10 if X ⊥_D T Z for some Z ⊆ PC \ X then 11 PC = PC \ X 12 return PC </pre>	<pre> graph TD T((T)) --> Q((Q)) P((P)) --> Q P --> R((R)) Q --> S((S)) R --> S </pre>
<u><i>MMMB</i>(<i>T</i>, <i>D</i>)</u>	(b)
<pre> /* add true positives to MB */ 1 PC = <i>MMPC</i>(<i>T</i>, <i>D</i>) 2 MB = PC 3 CanMB = PC ∪ {X ∈ PC <i>MMPC</i>(X, <i>D</i>)} /* add more true positives to MB */ 4 for each X ∈ CanMB \ PC do 5 find any Z st X ⊥_D T Z and X, T ⊄ Z 6 for each Y ∈ PC do 7 if X ⊥_D T Z ∪ Y then 8 MB = MB ∪ X 9 return MB </pre>	<pre> graph TD P((P)) --> Q((Q)) P --> R((R)) Q --> T((T)) T --> S((S)) R --> S </pre>

because $S \not\perp_p T | \emptyset$ and $S \not\perp_p T | Q$. Then, $PC = \{Q, S\}$ at line 9. Neither Q nor S leaves PC at line 11. Consequently, the output of *MMPC* includes S which is not in $PC_G(T)$ and, thus, *MMPC* is incorrect.

Table 2 outlines *MMMB*. The algorithm receives the target node T and the learning database D as input and returns $MB_G(T)$ in MB as output. The algorithm works in two steps. First, PC and MB are initialized with $PC_G(T)$ and $CanMB$ with $PC_G(T) \cup_{X \in PC_G(T)} PC_G(X)$ by calling *MMPC* (lines 1-3). $CanMB$ contains the candidates to enter MB . Second, the parents of the children of T in G that are not yet in MB are added to it (lines 4-8). This step is based on the following observation. The parents of the children of T in G that are missing from MB at line 4 are those that are non-adjacent to T in G . These parents are in $CanMB \setminus PC$. Therefore, if $X \in CanMB \setminus PC$ and $Y \in PC$, then X and T are non-adjacent parents of Y in G iff $X \not\perp_p T | Z \cup \{Y\}$ for any Z st $X \perp_p T | Z$ and $X, T \notin Z$. Note that Z can be efficiently obtained at line 5: *MMPC* must have found such a Z and could have cached it for later retrieval. Tsamardinos et al. prove that, under the assumptions in Theorem 3, the output of *MMMB* is $MB_G(T)$. We show that this is not always true even if *MMPC* were correct. The flaw in the proof is the observation that motivates the second step of *MMMB*, which is not true. This is illustrated by running *MMMB*(T, D) with D faithful to the DAG (b) in Table 2. Let us assume that *MMPC* is correct. Then, $MB = PC = \{Q, S\}$ and $CanMB = \{P, Q, R, S, T\}$

Table 3. *AlgorithmPCD*, *AlgorithmPC* and *AlgorithmMB*

<u><i>AlgorithmPCD</i>(T, D)</u>	<u><i>AlgorithmPC</i>(T, D)</u>
<pre> 1 $PCD = \emptyset$ 2 $CanPCD = \mathbf{U} \setminus \{T\}$ 3 repeat /* remove false positives from $CanPCD$ */ 4 for each $X \in CanPCD$ do 5 $Sep[X] = \arg \min_{\mathbf{Z} \subseteq PCD} dep_D(X, T \mathbf{Z})$ 6 for each $X \in CanPCD$ do 7 if $X \perp_D T Sep[X]$ then 8 $CanPCD = CanPCD \setminus \{X\}$ /* add the best candidate to PCD */ 9 $Y = \arg \max_{X \in CanPCD} dep_D(X, T Sep[X])$ 10 $PCD = PCD \cup \{Y\}$ 11 $CanPCD = CanPCD \setminus \{Y\}$ /* remove false positives from PCD */ 12 for each $X \in PCD$ do 13 $Sep[X] = \arg \min_{\mathbf{Z} \subseteq PCD \setminus \{X\}} dep_D(X, T \mathbf{Z})$ 14 for each $X \in PCD$ do 15 if $X \perp_D T Sep[X]$ then 16 $PCD = PCD \setminus \{X\}$ 17 until PCD does not change 18 return PCD </pre>	<pre> 1 $PC = \emptyset$ 2 for each $X \in AlgorithmPCD(T, D)$ do 3 if $T \in AlgorithmPCD(X, D)$ then 4 $PC = PC \cup \{X\}$ 5 return PC AlgorithmMB(T, D) /* add true positives to MB */ 1 $PC = AlgorithmPC(T, D)$ 2 $MB = PC$ /* add more true positives to MB */ 3 for each $Y \in PC$ do 4 for each $X \in AlgorithmPC(Y, D)$ do 5 if $X \notin PC$ then 6 find $\mathbf{Z}$ st $X \perp_D T \mathbf{Z}$ and $X, T \notin \mathbf{Z}$ 7 if $X \not\perp_D T \mathbf{Z} \cup \{Y\}$ then 8 $MB = MB \cup \{X\}$ 9 return MB </pre>

at line 4. P enters MB at line 8 if $\mathbf{Z} = \{Q\}$ at line 5, because $P \in CanMB \setminus PC$, $S \in PC$, $P \perp_p T | Q$ and $P \not\perp_p T | \{Q, S\}$. Consequently, the output of $MMMB$ can include P which is not in $MB_G(T)$ and, thus, $MMMB$ is incorrect even if $MMPC$ were correct.

In practice, $MMMB$ performs a test if it is reliable and skips it otherwise. $MMMB$ follows the same criterion as $IAMB$ to decide whether a test is reliable or not. If $MMMB$ were correct, then it would be data efficient because the number of instances required to identify $MB_G(T)$ would not depend on the size of $MB_G(T)$ but on the topology of G .

4 Scalable, Efficient and Correct Learning of MBs

In this section, we present a new algorithm for learning MBs from data that scales to databases with thousands of features. Like $IAMB$ and $MMMB$, our algorithm is based on the assumption that the learning database D is a sample from a probability distribution p faithful to a DAG G . Unlike $IAMB$, our algorithm is data efficient. Unlike $MMMB$, our algorithm is correct under the assumptions in Theorem 3. Our algorithm identifies $MB_G(T)$ in two steps: First, it identifies $PC_G(T)$ and, second, it identifies the rest of the parents of the children of T in G . Our algorithm, named *AlgorithmMB*, uses *AlgorithmPCD* and *AlgorithmPC* to solve the first step. $X \not\perp_D Y | \mathbf{Z}$, $X \perp_D Y | \mathbf{Z}$ and $dep_D(X, Y | \mathbf{Z})$ are the same as in Section 3. Table 3 outlines *AlgorithmPCD*. The algorithm receives the target node T and the learning database D as input and returns a superset of $PC_G(T)$ in PCD as output. The algorithm tries to minimize the number of nodes not in $PC_G(T)$ that are returned in PCD . The algorithm repeats three steps until PCD does not change. First, some nodes not in $PC_G(T)$

are removed from *CanPCD*, which contains the candidates to enter *PCD* (lines 4-8). This step is based on the observation that $X \in PC_G(T)$ iff $X \not\perp_p T | \mathbf{Z}$ for all $\mathbf{Z}$ st $X, T \notin \mathbf{Z}$. Second, the candidate most likely to be in $PC_G(T)$ is added to *PCD* and removed from *CanPCD* (lines 9-11). Since this step is based on the heuristic at line 9, some nodes not in $PC_G(T)$ may be added to *PCD* as well. Some of these nodes are removed from *PCD* in the third step (lines 12-16). This step is based on the same observation as the first step.

Theorem 4. *Under the assumptions that the learning database D is an independent and identically distributed sample from a probability distribution p faithful to a DAG G and that the tests of conditional independence and the measure of conditional dependence are correct, the output of $\text{AlgorithmPCD}(T, D)$ includes $PC_G(T)$ and does not include any node in $ND_G(T) \setminus Pa_G(T)$.*

Proof. First, we prove that the nodes in $PC_G(T)$ are included in the output *PCD*. If $X \in PC_G(T)$, then $X \not\perp_p T | \mathbf{Z}$ for all $\mathbf{Z}$ st $X, T \notin \mathbf{Z}$ (Theorem 1). Consequently, X enters *PCD* at line 10 and does not leave it thereafter.

Second, we prove that the nodes in $ND_G(T) \setminus Pa_G(T)$ are not included in the output *PCD*. It suffices to study the last time that lines 12-16 are executed. At line 12, $Pa_G(T) \subseteq PCD$ (see paragraph above). Therefore, if *PCD* still contains some $X \in ND_G(T) \setminus Pa_G(T)$, then $X \perp_p T | \mathbf{Z}$ for some $\mathbf{Z} \subseteq PCD \setminus \{X\}$ (local Markov property). Consequently, X is removed from *PCD* at line 16. $\square$

The output of *AlgorithmPCD* must be further processed in order to obtain $PC_G(T)$, because it may contain some descendants of T in G other than its children. These nodes can be easily identified: If X is in the output of $\text{AlgorithmPCD}(T, D)$, then X is a descendant of T in G other than one of its children iff T is not in the output of $\text{AlgorithmPCD}(X, D)$. *AlgorithmPC*, which is outlined in Table 3, implements this observation. The algorithm receives the target node T and the learning database D as input and returns $PC_G(T)$ in *PC* as output. We prove that *AlgorithmPC* is correct under some assumptions.

Theorem 5. *Under the assumptions that the learning database D is an independent and identically distributed sample from a probability distribution p faithful to a DAG G and that the tests of conditional independence and the measure of conditional dependence are correct, the output of $\text{AlgorithmPC}(T, D)$ is $PC_G(T)$.*

Proof. First, we prove that the nodes in $PC_G(T)$ are included in the output *PC*. If $X \in PC_G(T)$, then $T \in PC_G(X)$. Therefore, X and T satisfy the conditions at lines 2 and 3, respectively (Theorem 4). Consequently, X enters *PC* at line 4.

Second, we prove that the nodes not in $PC_G(T)$ are not included in the output *PC*. Let $X \notin PC_G(T)$. If X does not satisfy the condition at line 2, then X does not enter *PC* at line 4. On the other hand, if X satisfies the condition at line 2, then X must be a descendant of T in G other than one of its children and, thus, T does not satisfy the condition at line 3 (Theorem 4). Consequently, X does not enter *PC* at line 4. $\square$

Finally, Table 3 outlines *AlgorithmMB*. The algorithm receives the target node T and the learning database D as input and returns $MB_G(T)$ in *MB* as

output. The algorithm works in two steps. First, MB is initialized with $PC_G(T)$ by calling *AlgorithmPC* (line 2). Second, the parents of the children of T in G that are not yet in MB are added to it (lines 3-8). This step is based on the following observation. The parents of the children of T in G that are missing from MB at line 3 are those that are non-adjacent to T in G . Therefore, if $Y \in PC_G(T)$, $X \in PC_G(Y)$ and $X \notin PC_G(T)$, then X and T are non-adjacent parents of Y in G iff $X \not\perp_p T | \mathbf{Z} \cup \{Y\}$ for any $\mathbf{Z}$ st $X \perp_p T | \mathbf{Z}$ and $X, T \notin \mathbf{Z}$. Note that $\mathbf{Z}$ can be efficiently obtained at line 6: *AlgorithmPCD* must have found such a $\mathbf{Z}$ and could have cached it for later retrieval. We prove that *AlgorithmMB* is correct under some assumptions.

Theorem 6. *Under the assumptions that the learning database D is an independent and identically distributed sample from a probability distribution p faithful to a DAG G and that the tests of conditional independence and the measure of conditional dependence are correct, the output of *AlgorithmMB*(T, D) is $MB_G(T)$.*

Proof. First, we prove that the nodes in $MB_G(T)$ are included in the output MB . Let $X \in MB_G(T)$. Then, either $X \in PC_G(T)$ or $X \notin PC_G(T)$ but X and T have a common child Y in G (Theorem 2). If $X \in PC_G(T)$, then X enters MB at line 2 (Theorem 5). On the other hand, if $X \notin PC_G(T)$ but X and T have a common child Y in G , then X satisfies the conditions at lines 3-5 (Theorem 5) and at lines 6-7 (Theorem 1). Consequently, X enters MB at line 8.

Second, we prove that the nodes not in $MB_G(T)$ are not included in the output MB . Let $X \notin MB_G(T)$. X does not enter MB at line 2 (Theorem 5). If X does not satisfy the conditions at lines 3-6, then X does not enter MB at line 8. On the other hand, if X satisfies the conditions at lines 3-6, then it must be due to either $T \rightarrow Y \rightarrow X$ or $T \leftarrow Y \leftarrow X$ or $T \leftarrow Y \rightarrow X$. Therefore, X does not satisfy the condition at line 7 (faithfulness assumption). Consequently, X does not enter MB at line 8. $\square$

In practice, *AlgorithmMB* performs a test if it is reliable and skips it otherwise. *AlgorithmMB* follows the same criterion as *IAMB* and *MMMB* to decide whether a test is reliable or not. *AlgorithmMB* is data efficient because the number of instances required to identify $MB_G(T)$ does not depend on the size of $MB_G(T)$ but on the topology of G . For instance, if G is a tree, then *AlgorithmMB* does not need to perform any test that is conditioned on more than one node in order to identify $MB_G(T)$, no matter how large $MB_G(T)$ is. *AlgorithmMB* scales to databases with thousands of features because it does not require learning a complete BN. The experiments in Section 5 confirm it. Like *IAMB* and *MMMB*, if the assumptions in Theorem 6 do not hold, then *AlgorithmMB* may not return a MB but an approximation.

5 Experiments

In this section, we evaluate *AlgorithmMB* on synthetic and real data. We use *interIAMB* as benchmark (recall Section 3.1). We would have liked to include

Table 4. Results of the experiments with the Alarm and Pigs databases

Database	Instances	Algorithm	Precision	Recall	Distance	Time
Alarm	100	<i>interIAMB</i>	0.85±0.06	0.46±0.03	0.54±0.06	0±0
Alarm	100	<i>AlgorithmMB</i>	0.79±0.04	0.49±0.05	0.51±0.04	0±0
Alarm	200	<i>interIAMB</i>	0.87±0.04	0.59±0.04	0.42±0.04	0±0
Alarm	200	<i>AlgorithmMB</i>	0.94±0.03	0.56±0.05	0.38±0.06	0±0
Alarm	500	<i>interIAMB</i>	0.91±0.03	0.73±0.03	0.30±0.04	0±0
Alarm	500	<i>AlgorithmMB</i>	0.94±0.01	0.72±0.04	0.25±0.04	0±0
Alarm	1000	<i>interIAMB</i>	0.93±0.03	0.80±0.01	0.22±0.02	0±0
Alarm	1000	<i>AlgorithmMB</i>	0.99±0.01	0.79±0.01	0.17±0.02	0±0
Alarm	2000	<i>interIAMB</i>	0.92±0.04	0.83±0.01	0.21±0.04	0±0
Alarm	2000	<i>AlgorithmMB</i>	1.00±0.00	0.83±0.02	0.14±0.02	0±0
Alarm	5000	<i>interIAMB</i>	0.92±0.02	0.86±0.01	0.18±0.02	0±0
Alarm	5000	<i>AlgorithmMB</i>	1.00±0.00	0.86±0.02	0.11±0.02	1±0
Alarm	10000	<i>interIAMB</i>	0.92±0.04	0.90±0.01	0.14±0.03	0±0
Alarm	10000	<i>AlgorithmMB</i>	1.00±0.00	0.91±0.02	0.07±0.02	1±0
Alarm	20000	<i>interIAMB</i>	0.94±0.00	0.92±0.00	0.10±0.00	1±0
Alarm	20000	<i>AlgorithmMB</i>	1.00±0.00	0.92±0.00	0.05±0.00	3±0
Pigs	100	<i>interIAMB</i>	0.82±0.01	0.59±0.01	0.48±0.02	0±0
Pigs	100	<i>AlgorithmMB</i>	0.83±0.01	0.81±0.02	0.29±0.02	0±0
Pigs	200	<i>interIAMB</i>	0.80±0.00	0.82±0.00	0.37±0.00	0±0
Pigs	200	<i>AlgorithmMB</i>	0.97±0.01	0.96±0.01	0.07±0.01	1±0
Pigs	500	<i>interIAMB</i>	0.82±0.00	0.84±0.00	0.34±0.00	0±0
Pigs	500	<i>AlgorithmMB</i>	0.98±0.00	1.00±0.00	0.02±0.00	2±0

interIAMBnPC in the evaluation but we were unable to finish the implementation on time. We will include it in an extended version of this paper. We do not consider *GS* because *interIAMB* outperforms it [10]. We do not consider *MMMB* because we are not interested in incorrect algorithms.

5.1 Synthetic Data

These experiments evaluate the accuracy and data efficiency of *AlgorithmMB* wrt those of *interIAMB*. For this purpose, we consider databases sampled from two known BNs, namely the Alarm BN [4] and the Pigs BN [11]. These BNs have 37 and 441 nodes, respectively, and the largest MB consists of eight and 68 nodes, respectively. We run *interIAMB* and *AlgorithmMB* with each node in each BN as target and, then, report the average precision and recall over all the nodes for each BN. Precision is the number of true positives in the output divided by the number of nodes in the output. Recall is the number of true positives in the output divided by the number of true positives in the BN. We also combine precision and recall as $\sqrt{(1 - \text{precision})^2 + (1 - \text{recall})^2}$ to measure the Euclidean distance from perfect precision and recall. Finally, we also report the running time in seconds. Both algorithms are written in C++ and all the experiments are run on a Pentium 2.4 GHz, 512 MB RAM and Windows 2000. The significance level for the tests of conditional independence is 0.01.

Table 4 summarizes the results of the experiments with the Alarm and Pigs databases for different number of instances. Each entry in the table shows the average and standard deviation values over 10 databases (the same 10 databases for *interIAMB* and *AlgorithmMB*). For the Alarm databases, both algorithms achieve similar recall but *AlgorithmMB* scores higher precision and, thus, shorter distance than *interIAMB*. Therefore, *AlgorithmMB* usually returns

fewer false positives than *interIAMB*. The explanation is that *AlgorithmMB* performs more tests than *interIAMB* and this makes it harder for false positives to enter the output. See, for instance, the heuristic in *AlgorithmPCD* and the double check in *AlgorithmPC*. For this reason, we expect *interIAMBnPC* to perform better than *interIAMB* but worse than *AlgorithmMB*. For the Pigs databases where larger MBs exist, *AlgorithmMB* outperforms *interIAMB* in terms of precision, recall and distance. For instance, *AlgorithmMB* correctly identifies the MB of node 435 of the Pigs BN, which consists of 68 nodes, with only 500 instances, while *interIAMB* performs poorly for this node (precision=1.00, recall=0.04 and distance=0.96). The explanation is that, unlike *interIAMB*, *AlgorithmMB* does not need to condition on the whole MB to identify it. Note that *interIAMBnPC* could not have done better than *interIAMB* for this node. In fact, *interIAMB* and *interIAMBnPC* require a number of instances at least exponential in 68 for perfect precision and recall for this node. Consequently, we can conclude that *AlgorithmMB* is more accurate and data efficient than *interIAMB* and, seemingly, *interIAMBnPC*.

5.2 Real Data

These experiments evaluate the ability of *AlgorithmMB* wrt that of *interIAMB* to solve a real-world FSS problem involving thousands of features. Specifically, we consider the Thrombin database which was provided by DuPont Pharmaceuticals for the KDD Cup 2001 and is exemplary of the real-world drug design environment [1]. The database contains 2543 instances characterized by 139351 binary features. Each instance represents a drug compound tested for its ability to bind to a target site on thrombin, a key receptor in blood clotting. The features describe the three-dimensional properties of the compounds. Each compound is labelled with one out of two classes, either it binds to the target site or not. The task of the KDD Cup 2001 is to learn a classifier from 1909 given compounds in order to predict binding affinity. The accuracy of the classifier is evaluated wrt the remaining 634 compounds. The accuracy is computed as the average of the accuracy on true binding compounds and the accuracy on true non-binding compounds. The Thrombin database is particularly challenging for two reasons. First, the learning data are extremely imbalanced: Only 42 compounds out of 1909 bind. Second, the testing data are not sampled from the same probability distribution as the learning data, because the compounds in the testing data were synthesized based on the assay results recorded in the learning data. Better than 60 % accuracy is impressive according to [1].

As discussed in Section 1 and in [1], solving the FSS problem for the Thrombin database is crucial due to the excessive number of features. Since the truly relevant features for binding affinity are unknown, we cannot use the same performance criteria for *interIAMB* and *AlgorithmMB* as in Section 5.1. Instead, we run each algorithm on the learning data and, then, use only the features in the output to learn a naive Bayesian (NB) classifier [2], whose accuracy on the testing data is our performance criterion: The higher the accuracy the better the features selected and, thus, the algorithm. We also report the number of

Table 5. Results of the experiments with the Thrombin database

Algorithm	Features	Accuracy	Time
Winner KDD Cup 2001 with TANB	4.00	0.68	Not available
Winner KDD Cup 2001 with NB	4.00	0.67	Not available
<i>interIAMB</i>	8.00 ± 0.00	0.52 ± 0.02	3102 ± 69
<i>AlgorithmMB</i>	4.00 ± 1.00	0.60 ± 0.02	8631 ± 915

features selected and the running time of the algorithm in seconds. The rest of the experimental setting is the same as in Section 5.1.

Table 5 summarizes the results of the experiments with the Thrombin database. The table shows the average and standard deviation values over 10 runs of *interIAMB* and *AlgorithmMB* because the algorithms break ties at random and, thus, different runs can return different MBs. The table also shows the accuracy of the winner of the KDD Cup 2001, a tree augmented naive Bayesian (TANB) classifier [2] with the features 10695, 16794, 79651 and 91839 and only one augmenting edge between 10695 and 16794, as well as the accuracy of a NB classifier with the same features as the winning TANB. We were unable to learn a NB classifier with all the 139351 features. The winning TANB and NB are clearly more accurate than *interIAMB* and *AlgorithmMB*. The explanation may be that the score used to learn the winning TANB, the area under the ROC curve with a user-defined threshold to control complexity, works better than the tests of conditional independence in *interIAMB* and *AlgorithmMB* when the learning data are as imbalance as in the Thrombin database. This question is worth of further investigation, but it is out of the scope of this paper. What is more important in this paper is the performance of *AlgorithmMB* wrt that of *interIAMB*. The former is clearly more accurate than the latter, though it is slower because it performs more tests. It is worth mentioning that, while the best run of *interIAMB* reaches 54 % accuracy, two of the runs of *AlgorithmMB* achieve 63 % accuracy which, according to [1], is impressive. The features selected by *AlgorithmMB* in these two runs are 12810, 28852, 79651, 91839 and either 106279 or 109171. We note that no existing algorithm for learning BNs from data can handle such a high-dimensional database as the Thrombin database.

6 Discussion

We have introduced *AlgorithmMB*, an algorithm for learning the MB of a node from data without having to learn a complete BN. We have proved that *AlgorithmMB* is correct under the faithfulness assumption. We have shown that *AlgorithmMB* is scalable and data efficient and, thus, it can solve the FSS problem for databases with thousands of features but with many less instances. Since there is no algorithm for learning BNs from data that scales to such high-dimensional databases, it is very important to develop algorithms for learning MBs from data that, like *AlgorithmMB*, avoid learning a complete BN as an intermediate step. To our knowledge, the only work that has addressed the poor scalability of the existing algorithms for learning BNs from data is [3], where

Friedman et al. propose restricting the search for the parents of each node to some promising nodes that are heuristically selected. Therefore, Friedman et al. do not develop a scalable algorithm for learning BNs from data but some heuristics to use prior to running any existing algorithm. Unfortunately, Friedman et al. do not evaluate the heuristics for learning MBs from data. It is worth mentioning that learning the MB of each node can be a helpful intermediate step in the process of learning a BN from data [5]. As part of *AlgorithmMB*, we have introduced *AlgorithmPC*, an algorithm that returns the parents and children of a target node. In [7], we have reused this algorithm for growing BN models of gene networks from seed genes.

Acknowledgements

We thank Björn Brinne for his comments. This work is funded by the Swedish Foundation for Strategic Research (SSF) and Linköping Institute of Technology.

References

1. Cheng, J., Hatzis, C., Hayashi, H., Krogel, M. A., Morishita, S., Page, D., Sese, J.: KDD Cup 2001 Report. ACM SIGKDD Explorations **3** (2002) 1–18. See also <http://www.cs.wisc.edu/~dpage/kddcup2001/>
2. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian Network Classifiers. Machine Learning **29** (1997) 131–163
3. Friedman, N., Nachman, I., Peér, D.: Learning Bayesian Network Structure from Massive Datasets: The “Sparse Candidate” Algorithm. UAI (1999) 206–215
4. Herskovits, E. H.: Computer-Based Probabilistic-Network Construction. PhD Thesis, Stanford University (1991)
5. Margaritis, D., Thrun, S.: Bayesian Network Induction via Local Neighborhoods. NIPS (2000) 505–511
6. Neapolitan, R. E.: Learning Bayesian Networks. Prentice Hall (2003)
7. Peña, J. M., Björkegren, J., Tegnér, J.: Growing Bayesian Network Models of Gene Networks from Seed Genes. Submitted (2005)
8. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. Springer-Verlag (1993)
9. Tsamardinos, I., Aliferis, C. F.: Towards Principled Feature Selection: Relevancy, Filters and Wrappers. AI & Statistics (2003)
10. Tsamardinos, I., Aliferis, C. F., Statnikov, A.: Algorithms for Large Scale Markov Blanket Discovery. FLAIRS (2003) 376–380
11. Tsamardinos, I., Aliferis, C. F., Statnikov, A.: Time and Sample Efficient Discovery of Markov Blankets and Direct Causal Relations. KDD (2003) 673–678
12. Tsamardinos, I., Aliferis, C. F., Statnikov, A.: Time and Sample Efficient Discovery of Markov Blankets and Direct Causal Relations. Technical Report DSL TR-03-04, Vanderbilt University (2003)

Discriminative Learning of Bayesian Network Classifiers via the TM Algorithm

Guzmán Santafé, Jose A. Lozano, and Pedro Larrañaga

Intelligent Systems Group,
Department of Computer Science and Artificial Intelligence,
University of the Basque Country, Spain
{guzman, lozano, ccplamup}@si.ehu.es

Abstract. The learning of probabilistic classification models can be approached from either a generative or a discriminative point of view. Generative methods attempt to maximize the unconditional log-likelihood, while the aim of discriminative methods is to maximize the conditional log-likelihood. In the case of Bayesian network classifiers, the parameters of the model are usually learned by generative methods rather than discriminative ones. However, some numerical approaches to the discriminative learning of Bayesian network classifiers have recently appeared. This paper presents a new statistical approach to the discriminative learning of these classifiers by means of an adaptation of the TM algorithm [1]. In addition, we test the TM algorithm with different Bayesian classification models, providing empirical evidence of the performance of this method.

1 Introduction

Supervised classification is a part of machine learning which has a large number of applications in many tasks such as pattern recognition and medical diagnosis. In general, supervised classification assumes the existence of two different kinds of variables: the predictive variables, $\mathbf{X} = (X_1, \dots, X_n)$, and the class variable or response, C . A supervised classifier attempts to learn the relationship between the predictive and the class variables. Hence, it is able to assign a class value to a new data sample $\mathbf{x} = (x_1, \dots, x_n)$ whose response is unknown.

The learning of a classification model can be approached, among other paradigms, from either a generative or a discriminative point of view [2, 3, 4, 5]. Generative classifiers, also called informative classifiers, obtain the parameters of the model by maximizing the unconditional log-likelihood function. Models like discriminant analysis [6] or naïve Bayes [7] are typical examples of generative classifiers. On the other hand, discriminative classifiers obtain the parameters of the model by maximizing the conditional log-likelihood function (e.g. logistic regression [8]) or just model the class boundaries (e.g. neural networks [9]).

Bayesian networks [10, 11] are widely used for classification tasks due to their simplicity and accuracy. Usually, Bayesian network learners are generative but, recently, there has been a considerable growth of interest in the discriminative learning of Bayesian network classifiers [12, 13]. The use of a discriminative

learning for classification purposes seems more natural because the classification model directly maximizes the probability of the class given the predictive variables, which is what we use to classify new instances. However, generative classifiers can sometimes yield better performance than discriminative ones [4]. Normally, generative learning performs better in those cases where the classification model learned from a dataset is close to the one that has generated this dataset. On the other hand, when the learned model is different from the original one, generative classifiers normally perform worse than discriminative ones [3].

The aim of this paper is to propose a statistical approach to the discriminative learning of Bayesian network classifiers, in contrast to other more generic numerical optimization schemes [12, 13], via the adaptation of the TM algorithm. The TM algorithm [1] is a general iterative process that allows the maximization of the conditional log-likelihood in models where the unconditional log-likelihood function is easier to maximize, which is the case of Bayesian networks. We introduce the theoretical development of the algorithm in the context of Bayesian classification models. Additionally, we evaluate the performance of Bayesian network classifiers learned with the TM algorithm by comparing their estimated accuracy with the estimated accuracy of the classifiers learned by a classical generative method. This empirical evaluation is performed using simple models such as naïve Bayes [7] and tree augmented naïve Bayes (TAN) [14].

The rest of this paper is organized as follows. In Section 2, the general structure of the TM algorithm is described, and this structure is particularized to the exponential family of distributions. In Section 3, we adapt the TM algorithm to be used with Bayesian network classifiers. Section 4 provides empirical results of the performance of the TM algorithm and, finally, the conclusions yielded from the paper are exposed in Section 5.

2 The TM Algorithm by Edwards and Lauritzen

This section introduces the TM algorithm in the same way as [1] but bearing in mind the classification purpose of the model that we want to learn. Thus, we expect to give the reader a general and intuitive idea about how the TM algorithm works.

2.1 General Structure of the TM Algorithm

Let $\mathbf{X} = (X_1, \dots, X_n)$ be a vector where each X_i , with $i = 1, \dots, n$, is a predictive variable, and let C be the class variable. Since we are focusing on classification problems, we consider C a unidimensional variable, but in general both $\mathbf{X}$ and C could be multivariate variables.

We denote the unconditional, marginal and conditional log-likelihood functions as follows:

$$l(\boldsymbol{\theta}) = \log f(\mathbf{x}, c | \boldsymbol{\theta}) \quad , \quad l_{\mathbf{x}}(\boldsymbol{\theta}) = \log f(\mathbf{x} | \boldsymbol{\theta}) \quad , \quad l^{\mathbf{x}}(\boldsymbol{\theta}) = \log f(c | \mathbf{x}, \boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ is the parameter set of the unconditional probability distribution for the variable $(\mathbf{X}, C)$.

The foundations of the TM algorithm are based on the tilted unconditional log-likelihood function, $q(\boldsymbol{\theta}|\boldsymbol{\theta}_r)$. This function is an approximation to $l^{\mathbf{x}}(\boldsymbol{\theta})$, which we want to maximize, at point $\boldsymbol{\theta}_r$. Note that $l^{\mathbf{x}}(\boldsymbol{\theta})$ can be expressed in terms of the unconditional and the marginal log-likelihood:

$$l^{\mathbf{x}}(\boldsymbol{\theta}) = l(\boldsymbol{\theta}) - l_{\mathbf{x}}(\boldsymbol{\theta})$$

Therefore, if we expand $l_{\mathbf{x}}(\boldsymbol{\theta})$ in a first order Taylor series about the point $\boldsymbol{\theta}_r$, and then omit the terms which are constant with respect to $\boldsymbol{\theta}$, we can approximate $l^{\mathbf{x}}(\boldsymbol{\theta})$ by $q(\boldsymbol{\theta}|\boldsymbol{\theta}_r)$ as follows:

$$l^{\mathbf{x}}(\boldsymbol{\theta}) \approx q(\boldsymbol{\theta}|\boldsymbol{\theta}_r) = l(\boldsymbol{\theta}) - \boldsymbol{\theta}^T \dot{l}_{\mathbf{x}}(\boldsymbol{\theta}_r) \quad (1)$$

where $\dot{l}_{\mathbf{x}}(\boldsymbol{\theta}_r)$ is the derivative of $l_{\mathbf{x}}(\boldsymbol{\theta})$ at point $\boldsymbol{\theta}_r$.

The tilted unconditional log-likelihood function and the conditional log-likelihood have the same gradient at $\boldsymbol{\theta}_r$, thus, we can maximize $l^{\mathbf{x}}(\boldsymbol{\theta})$ by maximizing $q(\boldsymbol{\theta}|\boldsymbol{\theta}_r)$

Since the approximation of $l^{\mathbf{x}}(\boldsymbol{\theta})$ is at point $\boldsymbol{\theta}_r$, we need an iterative process in order to maximize the conditional log-likelihood. This process alternates between two steps, T and M. In the T step, the tilted unconditional log-likelihood described above is obtained. The second step of the algorithm, the M step, consists in maximizing the tilted unconditional log-likelihood function:

$$\boldsymbol{\theta}_{r+1} = \arg \max_{\boldsymbol{\theta}} q(\boldsymbol{\theta}|\boldsymbol{\theta}_r) \quad (2)$$

Under regularity conditions of the usual type and due to the fact that the expected score statistic for the conditional model is equal to 0, $\dot{l}_{\mathbf{x}}(\boldsymbol{\theta})$ can be calculated as the expectation of the score statistic for the unconditional model.

$$\dot{l}_{\mathbf{x}}(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}}\{\dot{l}_{\mathbf{x}}(\boldsymbol{\theta})|\mathbf{x}\} = E_{\boldsymbol{\theta}}\{\dot{l}(\boldsymbol{\theta}) - \dot{l}^{\mathbf{x}}(\boldsymbol{\theta})|\mathbf{x}\} = E_{\boldsymbol{\theta}}\{\dot{l}(\boldsymbol{\theta})|\mathbf{x}\}$$

Therefore, the M step involves the solution of the following equation:

$$E_{\boldsymbol{\theta}_r}\{\dot{l}(\boldsymbol{\theta}_r)|\mathbf{x}\} = \dot{l}(\boldsymbol{\theta}) \quad (3)$$

In summary, the relevance of the TM algorithm is that it allows us to obtain a model that maximizes the conditional log-likelihood, $l^{\mathbf{x}}(\boldsymbol{\theta})$, by using the unconditional log-likelihood, $l(\boldsymbol{\theta})$. This is very useful for models like Bayesian network classifiers, where the obtention of the unconditional (generative) model is much easier than the obtention of the conditional (discriminative) one.

The TM algorithm begins by making its initial parameters the ones which maximize the unconditional log-likelihood given the dataset. Then, both the T and the M steps are repeated until the value of the conditional log-likelihood converges. See [15] for details about the convergence of the TM algorithm.

2.2 The TM Algorithm for the Exponential Family

The TM algorithm can be easily particularized for probability distributions belonging to the exponential family. In this case, the unconditional log-likelihood is given by the following formula:

$$l(\boldsymbol{\theta}) = \boldsymbol{\alpha}^T u(c, \mathbf{x}) + \boldsymbol{\beta}^T v(\mathbf{x}) - \psi(\boldsymbol{\alpha}, \boldsymbol{\beta}) \quad (4)$$

where

$$\psi(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \log \int \exp\{\boldsymbol{\alpha}^T u(c, \mathbf{x}) + \boldsymbol{\beta}^T v(\mathbf{x})\} \mu(dc|\mathbf{x}) \mu(d\mathbf{x})$$

Let us introduce a new parametrization for $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\eta})$, with:

$$\boldsymbol{\eta} = \frac{\partial}{\partial \boldsymbol{\beta}} \psi(\boldsymbol{\alpha}, \boldsymbol{\beta})$$

Moreover, if we define two new random variables, $\mathbf{U} = u(C, \mathbf{X})$ and $\mathbf{V} = v(\mathbf{X})$, it can be demonstrated that the maximum likelihood parameters are $\hat{\boldsymbol{\theta}} = (\mathbf{u}, \mathbf{v})$ with $\mathbf{u} = E_{\boldsymbol{\theta}}\{\mathbf{U}\}$ and $\mathbf{v} = \boldsymbol{\eta} = E_{\boldsymbol{\theta}}\{\mathbf{V}\}$.

Following the general structure of the TM algorithm, Equation 3 has to be solved in order to maximize the approximation to the conditional log-likelihood given by $q(\boldsymbol{\theta}|\boldsymbol{\theta}_r)$. Thus, we have:

$$\begin{aligned} E_{\boldsymbol{\theta}} \left\{ \frac{\partial}{\partial \boldsymbol{\theta}} l(\boldsymbol{\theta}) \middle| \mathbf{x} \right\} &= E_{\boldsymbol{\theta}} \left\{ \mathbf{U} - \frac{\partial}{\partial \boldsymbol{\alpha}} \psi(\boldsymbol{\alpha}, \boldsymbol{\beta}), \mathbf{V}^T \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\eta}} - \frac{\partial}{\partial \boldsymbol{\beta}} \psi(\boldsymbol{\alpha}, \boldsymbol{\beta}) \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\eta}} \middle| \mathbf{x} \right\} \\ &= \left(E_{\boldsymbol{\theta}}\{\mathbf{U}|\mathbf{x}\} - E_{\boldsymbol{\theta}}\{\mathbf{U}\}, (E_{\boldsymbol{\theta}}\{\mathbf{V}\} - \boldsymbol{\eta})^T \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\eta}} \right) = (E_{\boldsymbol{\theta}}\{\mathbf{U}|\mathbf{x}\} - E_{\boldsymbol{\theta}}\{\mathbf{U}\}, 0) \end{aligned} \quad (5)$$

and also:

$$i(\boldsymbol{\theta}) = \left(\mathbf{U} - \frac{\partial}{\partial \boldsymbol{\alpha}} \psi(\boldsymbol{\alpha}, \boldsymbol{\eta}), \mathbf{V}^T \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\eta}} - \frac{\partial}{\partial \boldsymbol{\beta}} \psi(\boldsymbol{\alpha}, \boldsymbol{\beta}) \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\eta}} \right) = (\mathbf{U} - E_{\boldsymbol{\theta}}\{\mathbf{U}\}, 0) \quad (6)$$

Finally, the solution of Equation 3 gives the value of the sufficient statistics at the $r + 1$ -th iteration of the TM algorithm:

$$\begin{aligned} \mathbf{u}_{r+1} &= \mathbf{u}_r + \mathbf{u}_0 - E_{\boldsymbol{\theta}_r}\{\mathbf{U}|\mathbf{x}\} \\ \boldsymbol{\theta}_{r+1} &= \hat{\boldsymbol{\theta}}(\mathbf{u}_{r+1}, \mathbf{v}) \end{aligned} \quad (7)$$

where the initial sufficient statistics, $\mathbf{u}_0$ and $\mathbf{v}$, are given by the maximum likelihood estimators obtained from the data set. Moreover, $\hat{\boldsymbol{\theta}}(\mathbf{u}_{r+1}, \mathbf{v})$ denotes the maximum likelihood estimations of $\boldsymbol{\theta}$ obtained from sufficient statistics $\mathbf{u}_{r+1}$ and $\mathbf{v}$.

Generally, it may happen that an iteration of the TM algorithm yields an illegal set of parameters $\boldsymbol{\theta}$ or that the conditional log-likelihood decreases from one iteration to another. These situations must be corrected by applying a linear search. Thus, the sufficient statistics at step $r + 1$ are calculated as:

$$\mathbf{u}_{r+1} = \mathbf{u}_r + \lambda(\mathbf{u}_0 - E_{\boldsymbol{\theta}_r}\{\mathbf{U}|\mathbf{x}\}), \quad \text{with } \lambda \in (0, 1) \quad (8)$$

being λ the one that maximizes the conditional log-likelihood.

3 The TM Algorithm for Bayesian Classifiers

In this section we show how the TM algorithm can be adapted to the Bayesian classification models considered in this paper. Even when Bayesian networks belong to the exponential family, the adaptation of the calculations shown in Section 2.2 is not trivial. As an example, Section 3.1 shows the calculations needed

to apply the TM algorithm to a naïve Bayes model with multinomial variables. Calculations for the TAN model are similar. Therefore, for these models, Section 3.2 only shows the sufficient statistics, $\mathcal{U}$ and $\mathcal{V}$, used in the algorithm. See [16] for more details about the adaptation of the TM algorithm to Bayesian network classifiers with dichotomic and multinomial variables.

3.1 The TM Algorithm for a Naïve Bayes with Multinomial Variables

We assume that each variable can take multiple states, therefore $C \in \{0, \dots, v_0\}$ and $X_i \in \{0, \dots, v_i\}$ with $v_0 + 1$ and $v_i + 1$ as the number of possible states for variables C and X_i , respectively.

The general algorithm for probability distributions of the exponential family requires the expression of the unconditional log-likelihood via Equation 4. This can be achieved by writing the naïve Bayes unconditional model as follows:

$$p(c, \mathbf{x}) = \frac{1}{(p(c))^{n-1}} \prod_{i=1}^n p(x_i, c) \quad (9)$$

In order to identify the sufficient statistics for the TM algorithm, we can rewrite the unconditional model as follows:

$$p(c, \mathbf{x}) = \left[\prod_{j=0}^{v_0} (p(C = j))^{w_j \prod_{l=0}^{j-1} (c-l) \prod_{l=j+1}^{v_0} (l-c)} \right]^{-(n-1)} \cdot \prod_{i=1}^n \prod_{j=0}^{v_0} \prod_{k=0}^{v_i} (p(C = j, X_i = k))^{w_{jk}^i \prod_{l=0}^{j-1} (c-l) \prod_{l=j+1}^{v_0} (l-c) \prod_{l=0}^{k-1} (x_i-l) \prod_{l=k+1}^{v_i} (l-x_i)}$$

where w_j and w_{jk}^i are the following constants:

$$w_j = \frac{1}{\prod_{l=0}^{j-1} (j-l) \prod_{l=j+1}^{v_0} (l-j)}, \quad w_{jk}^i = \frac{1}{\prod_{l=0}^{j-1} (j-l) \prod_{l=j+1}^{v_0} (l-j) \prod_{l=0}^{k-1} (k-l) \prod_{l=k+1}^{v_i} (l-k)}$$

Note that the values of w_j and w_{jk}^i have no influence on the selection of the sufficient statistics for the TM algorithm.

If we have a dataset with N samples, the unconditional log-likelihood can be written using the previous equation as follows:

$$l(\theta) = \sum_{d=1}^N \left\{ - (n-1) \sum_{j=0}^{v_0} [w_j \prod_{l=0}^{j-1} (c^{(d)} - l) \prod_{l=j+1}^{v_0} (l - c^{(d)}) \log(p(C = j))] + \sum_{i=1}^n \sum_{j=0}^{v_0} \sum_{k=0}^{v_i} [w_{jk}^i \prod_{l=0}^{j-1} (c^{(d)} - l) \prod_{l=j+1}^{v_0} (l - c^{(d)}) \prod_{l=0}^{k-1} (c_i^{(d)} - l) \prod_{l=k+1}^{v_i} (l - c_i^{(d)}) \log(p(C = j, X_i = k))] \right\}$$

where $c^{(d)}$ and $x_i^{(d)}$ are the values of variables C and X_i in the d -th sample of the dataset, respectively.

A few transformations in Equation 10 can match its terms with the ones from Equation 4. We thus obtain the sufficient statistics $\mathcal{U} = (\mathcal{U}_1, \mathcal{U}_2)$ and $\mathcal{V}$:

$$\begin{aligned} \mathcal{U}_1 &= (M_0^s | s = 1, \dots, v_0) \\ \mathcal{U}_2 &= (M_{0x_i}^{st} | s = 1, \dots, v_0, \quad t = 1, \dots, v_i, \quad i = 1, \dots, n) \\ \mathcal{V} &= (M_{x_i}^t | t = 1, \dots, v_i, \quad i = 1, \dots, n) \end{aligned}$$

where M_0^s , $M_{0x_i}^{st}$ and $M_{x_i}^t$ terms from the former equation are defined as:

$$M_0^s = \sum_{d=1}^N (C^{(d)})^s, \quad M_{0x_i}^{st} = \sum_{d=1}^N (C^{(d)})^s (X_i^{(d)})^t, \quad M_{x_i}^d = \sum_{r=1}^N (X_i^{(d)})^r \quad (10)$$

It was shown in Section 2.2 that, at each iteration, the calculation of $E_{\theta}\{\mathcal{U}|\mathbf{x}\}$ is needed to update the sufficient statistics $\mathcal{U}$. This requires the following calculations:

$$E_{\theta_r}[M_0^s|\mathbf{x}] = \sum_{d=1}^N \sum_{c=0}^{v_0} p_{\theta_r}(C=c|\mathbf{X}=\mathbf{x}^{(d)}) c^s \quad (11)$$

$$E_{\theta_r}[M_{0x_i}^{st}|\mathbf{x}] = \sum_{d=1}^N \sum_{c=0}^{v_0} p_{\theta_r}(C=c|\mathbf{X}=\mathbf{x}^{(d)}) c^s (x_i^{(d)})^t$$

where $s = 1, \dots, v_0$; $t = 1, \dots, v_i$ and $i = 1, \dots, n$.

Since we assume that the structure of the model is a naïve Bayes, we need to obtain $p(C=c)$ and $p(X_i=l|C=c)$ to calculate $p(C=c|\mathbf{X}=\mathbf{x}^{(k)})$, where $c = 1, \dots, v_0$; $i = 1, \dots, n$ and $l = 1, \dots, v_i$. In order to obtain these probabilities, let us define a new set of sufficient statistics $\mathcal{N} = (N_0^c, N_i^l, N_{0i}^{cl} | c = 1, \dots, v_0; i = 1, \dots, n; l = 1, \dots, v_i)$. On the one hand, N_0^c counts the number of cases in which $C=c$, and N_i^l the number of cases in which $X_i=l$. On the other hand, N_{0i}^{cl} denotes the number of times that both $C=c$ and $X_i=l$ happen.

The sufficient statistics $\mathcal{N}$ are related to the sufficient statistics set, $(\mathcal{U}, \mathcal{V})$, from the TM algorithm. In the special case where all the variables are dichotomic, both sets of sufficient statistics are the same. However when the variables are multinomial, this relationship is given by linear systems of equations which can be obtained by means of Equation 10. Therefore, using these systems of equations we are able to obtain the values of $\mathcal{N}$ from $\mathcal{U}$ and vice versa.

As an example, we show how one of the linear systems of equations can be obtained from Equation 10. M_0^s , with $s = 1, \dots, v_0$, are sufficient statistics from the set $\mathcal{U}$ and, as mentioned in Equation 10, $M_0^s = \sum_{d=1}^N (C^{(d)})^s$. Since $\sum_{d=1}^N (C^{(d)}) = 0 \cdot N_0^0 + \dots + v_0 \cdot N_0^{v_0}$, the system of equation that relates both $\mathcal{U}$ and $\mathcal{N}$ for the variable C can be written in the matrix form as follows:

$$\underbrace{\begin{pmatrix} 1 & 2 & \dots & v_0 \\ 1 & 2^2 & \dots & v_0^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^{v_0} & \dots & v_0^{v_0} \end{pmatrix}}_{\mathcal{COEFFS}^*} \underbrace{\begin{pmatrix} N_0^1 \\ N_0^2 \\ \vdots \\ N_0^{v_0} \end{pmatrix}}_{\mathcal{N}^*} = \underbrace{\begin{pmatrix} M_0^1 \\ M_0^2 \\ \vdots \\ M_0^{v_0} \end{pmatrix}}_{\mathcal{U}^*} \quad (12)$$

Once we have obtained the values of the statistics in $\mathcal{N}$, we are able to calculate $p(C=c)$ and $p(X_i=l|C=c)$ by:

$$p(C=c) = \frac{N_0^c}{N}, \quad p(X_i=l|C=c) = \frac{N_{0i}^{cl}}{N_0^c}$$

and therefore calculate the value of $E_{\theta}\{\mathcal{U}|\mathbf{x}\}$. Finally, we are able to iterate the algorithm and thus obtain the new value for the statistic $\mathcal{U}$ (see Equation 7). These $p(C=c)$ and $p(X_i=l|C=c)$ are also the parameters θ of the naïve

```

Obtain  $\mathbf{n}_0$  from the dataset
Calculate  $\mathbf{u}_0$  from  $\mathbf{n}_0$ 
while stopping criterion is not met
    Calculate  $E_{\theta_r}\{\mathcal{U}|\mathbf{x}\}$ 
    Update  $\mathbf{u}$ :
         $\mathbf{u}_{r+1} = \mathbf{u}_r + \mathbf{u}_0 - E_{\theta_r}\{\mathcal{U}|\mathbf{x}\}$ 
    Calculate  $\mathbf{n}_{r+1}$  from  $\mathbf{u}_{r+1}$ 
    Calculate  $\theta_{r+1}$  from  $\mathbf{n}_{r+1}$ 
    if illegal  $\theta_{r+1}$  or conditional log-likelihood decreases
        Find the best legal  $\theta_{r+1}$  via linear search
    end if
end while

```

Fig. 1. General pseudocode for the discriminative learning of Bayesian classifiers. Note that $\mathbf{n}_r$ and $\mathbf{u}_r$ are the values of the statistics in $\mathcal{N}$ and $\mathcal{U}$ at iteration r , respectively

Bayes classifier that we are learning. Hence, we have to calculate $\mathcal{N}$ in order to obtain θ . A general pseudo-algorithm for the discriminative learning of Bayesian classifiers is given in Figure 1.

The process of maximizing the conditional log-likelihood with the TM algorithm looks computationally hard because we have to solve several linear systems of equations at each iteration. However, from one iteration of the algorithm to another one, in the systems of equations, only the values in $\mathcal{U}^*$ change (see Equation 12). Therefore, we can obtain the LU transformation of $\mathbf{COEFFS}^*$, which is constant throughout the algorithm. Thus, the solution of the systems of equations at each iteration is quite simple. Moreover, the LU transformation is also the same for every problem with the same number of variables and the same number of states per variable. Hence, it may be feasible to calculate these transformations and store the solutions for future use.

3.2 The TM Algorithm for TAN

In this section we introduce the adaptation of the TM algorithm in order to maximize the conditional log-likelihood with TAN models where the variables are assumed to be multinomial. The development of the TM algorithm for TAN models assumes that the structure of the model is already known. Therefore, before performing the discriminative learning of a TAN model, we need to set its structure.

The adaptation of the TM algorithm for TAN is similar to the adaptation for a naïve Bayes model shown above. Hence, we only provide the sufficient statistics that the TM algorithm uses.

In the case of TAN models, we need to differentiate between two kinds of predictive variables: the one which has only one parent, that is, the root of the tree formed by the predictive variables, and the rest of predictive variables, which have two parents: the class and another predictive variable. We assume, without loss of generality, that the root variable is the first one, X_1 . If we develop the $l(\theta)$ function for a TAN model with multinomial variables in a similar way to Equation 4, we can identify the following set of sufficient statistics $\mathcal{U} = (\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3)$ and $\mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2)$, where:

$$\begin{aligned}\mathcal{U}_1 &= (M_0^w | w = 1, 2, \dots, v_0) \\ \mathcal{U}_2 &= (M_{0x_i}^{wt} | w = 1, 2, \dots, v_0; t = 1, 2, \dots, v_i; i = 1, \dots, n) \\ \mathcal{U}_3 &= (M_{0x_i x_{j(i)}}^{wtz} | w = 1, \dots, v_0; t = 1, 2, \dots, v_i; z = 1, 2, \dots, v_{j(i)}; i = s + 1, \dots, n) \\ \mathcal{V}_1 &= (M_{x_i}^t | t = 1, \dots, v_i) \\ \mathcal{V}_2 &= (M_{x_i x_{j(i)}}^{tz} | t = 0, 1, \dots, v_i; z = 0, 1, \dots, v_{j(i)}; i = s + 1 \dots, n)\end{aligned}$$

with M_c^w , $M_{cx_i}^{wt}$, $M_{cx_i x_{j(i)}}^{wtz}$, $M_{x_i}^t$ and $M_{x_i x_{j(i)}}^{tz}$ defined as follows:

$$\begin{aligned}M_c^w &= \sum_{k=1}^N (C^{(k)})^w, \quad M_{cx_i}^{wt} = \sum_{k=1}^N (C^{(k)})^w (X_i^{(k)})^t, \quad M_{cx_i x_{j(i)}}^{wtz} = \sum_{k=1}^N (C^{(k)})^w (X_i^{(k)})^t (X_{j(i)}^{(k)})^z \\ M_{x_i}^t &= \sum_{k=1}^N (X_i^{(k)})^t, \quad M_{x_i x_{j(i)}}^{tz} = \sum_{k=1}^N (X_i^{(k)})^t (X_{j(i)}^{(k)})^z\end{aligned}$$

The adaptation of the TM algorithm for TAN models is equal to the one shown in Figure 1 but using the set of sufficient statistics described above.

4 Experimental Results

In this section we present an empirical test which attempts to illustrate the performance of the TM algorithm applied to Bayesian classification models such as naïve Bayes and TAN.

In the case of naïve Bayes models, the structure does not depend on the data, that is, a naïve Bayes structure may only differ from another one in the number of predictive variables. However, the structure of TAN models is learned from the data using the algorithm proposed by [14], which takes the conditional mutual information of two variables given the class into account.

We have evaluated the TM algorithm for the discriminative learning of Bayesian classifiers using sixteen datasets obtained from the UCI repository [17]. Moreover, we use the Corral and Mofn-3-7-10 datasets, which were developed by [18] to evaluate methods for subset selection, and the Tips dataset [19]. Tips is a medical dataset to identify the subgroup of patients surviving within the first six months after the *transjugular intrahepatic portosystemic shunt* (TIPS) placement, a non-surgical method to avoid portal hypertension.

Table 1. Estimated accuracy obtained in the experiments with naïve Bayes and TAN models

	<i>NB</i>	<i>NB-TM</i>	<i>NB</i> <i>vs.</i> <i>NB-TM</i>	<i>TAN</i>	<i>TAN-TM</i>	<i>TAN</i> <i>vs.</i> <i>TAN-TM</i>
Australian	85.65± 2.61	88.41±2.67	0.112	86.08±2.88	88.98±3.53	○ 0.094
Breast	97.37± 1.64	98.98±0.74	● 0.036	97.37±1.64	95.46±1.41	● 0.016
Chess	87.77± 0.91	95.15±0.41	● 0.009	92.40±1.73	96.81±0.49	● 0.009
Cleve	83.14± 4.89	87.53±4.72	○ 0.072	82.77±1.61	87.85±3.24	● 0.043
Corral	86.77± 9.27	90.61±6.27	0.197	100.00±0.00	99.20±1.60	0.317
Crx	86.68± 4.70	88.52±1.59	0.600	86.06±1.33	89.59±1.56	○ 0.075
Flare	92.12± 2.16	95.12±1.21	● 0.015	95.78±2.79	96.72±1.23	0.527
German	75.40± 3.50	78.90±4.00	○ 0.059	72.80±2.22	84.00±0.89	● 0.009
Glass	74.31± 7.32	76.18±6.92	0.344	72.90±2.74	81.75±3.88	● 0.045
Heart	83.33± 6.73	86.67±4.44	0.390	72.90±2.74	81.75±3.87	● 0.036
Hepatitis	85.00±10.15	93.75±5.56	0.316	87.50±6.84	100.00±0.00	● 0.004
Iris	94.67± 3.40	95.33±3.40	0.746	93.33±2.11	96.00±2.49	0.142
Lymphography	83.77± 4.97	91.22±3.49	0.141	79.08±2.28	98.98±1.65	● 0.008
Mofn-3-7-10	86.63± 2.53	100.00±0.00	● 0.005	90.86±1.79	100.00±0.00	● 0.005
Pima	77.96± 1.31	79.95±1.47	0.136	79.17±3.72	79.82±3.72	0.136
Soybean-large	96.26± 1.64	97.51±1.42	● 0.014	98.58±0.71	99.29±0.66	● 0.014
Tips	88.78± 4.65	100.00±0.00	● 0.019	89.87±6.20	100.00±0.00	● 0.005
Vehicle	61.94± 1.58	78.61±1.51	● 0.009	71.63±4.19	83.46±3.72	● 0.009
Vote	89.88± 2.45	98.39±1.17	● 0.008	93.56±1.55	99.08±0.86	● 0.008

The discriminative learning of Bayesian network classifiers described in the paper does not deal with missing data or continuous variables. Therefore, a pre-processing step was needed before using the datasets. On the one hand, every data sample which contained missing data was removed. On the other hand, variables with continuous values were discretized using the method described by [20], which is a variant of the Fayyad and Irani's [21] discretization method. The accuracy of the classifiers is measured by five-fold cross validation and it is based on the percentage of successful predictions. The same pre-processing and validation methodology has been used before in the literature for the generative [14] or discriminative [12] learning of Bayesian network classifiers using all the datasets that have been used in this paper, except for Tips.

The TM algorithm iteratively maximizes the conditional log-likelihood and it stops when a certain criterion is met. In the experiments, the algorithm stops when the difference between the conditional log-likelihood value in two consecutive steps is less than 0.001. On the other hand, as pointed out in Section 2.2, the TM calculations may lead the parameters of the model to illegal values. These situations are solved by applying a linear search where we look for λ in interval $(0, 1)$ with a 0.01 increment (see Equation 8).

Table 1 shows the estimated accuracy for the naïve Bayes (NB) and TAN classifiers learned using both generative and discriminative approaches. The generative approach that we use is the classical learning of Bayesian classifiers using the maximum likelihood parameters. In contrast, the discriminative learning is carried out using the TM algorithm proposed in this paper. In order to com-

Table 2. Conditional log-likelihood values for the experiments with naïve Bayes and TAN models

	<i>NB</i>	<i>NB-TM</i>	<i>TAN</i>	<i>TAN-TM</i>
Australian	-291.71	-190.38	-195.97	-195.97
Breast	-136.74	-22.71	-22.13	-10.41
Chess	-917.76	-478.22	-591.46	-307.04
Cleve	-124.65	-93.24	-96.36	-82.27
Corral	-37.58	-25.17	-10.19	-3.28
Crx	-251.71	-177.90	-173.71	-173.71
Flare	-287.91	-216.18	-152.76	-137.43
German	-488.91	-456.81	-409.05	-378.12
Glass	-141.86	-141.86	-117.60	-116.96
Heart	-112.91	-86.11	-88.21	-73.66
Hepatitis	-23.29	-9.27	-9.12	-2.61
Iris	-21.52	-13.85	-17.90	-16.57
Lymphography	-46.94	-28.06	-25.90	-13.17
Mofn-3-7-10	-269.32	-3.75	-232.03	-44.74
Pima	-361.36	-340.55	-33.24	331.22
Soybean-large	-108.15	-43.27	-22.14	-22.14
Tips	-45.66	-0.12	-2.77	-0.03
Vehicle	-1487.18	-355.30	-360.27	-297.16
Vote	-257.63	-13.66	-49.55	-13.88

pare the estimated accuracy for both discriminative and generative models we perform a Mann-Whitney test [22], whose results are also shown in Table 1. In addition to the Mann-Whitney p-value, we mark with • those experiments where the difference between generative and discriminative models is significant at the 95% level, and with ◦ if it is significant only at the 90% level. The TM algorithm improves the estimated accuracy for naïve Bayes in all the datasets and, in the case of TAN models, only in Breast and Corral does the generative model obtain a higher estimated accuracy. This may be due to a worse performance of discriminative learning when the structure of the classifier is correct [3], that is the structure that perfectly models the relationship between the variables, and TAN is a structure a bit more complex that can model the dataset better. However, even if the estimated accuracy is usually higher in discriminative models, the difference with respect to generative models is not always significant. In most of the cases if the improvement obtained by the discriminative method is not significant at the 95% level, it is because of a high standard deviation. A cause of this high standard deviation may be the small number of folds used in the cross-validation process. For instance, leaving-one-out cross-validation, used to measure the estimated accuracy of a naïve Bayes and a TAN learned from a dataset such as Corral, leads to a decrease in the standard deviation while the estimated accuracy does not change very much. Nevertheless, we have decided to maintain the cross-validation schema in order to agree with the one used by [12]. Thus we have a point of reference for the result obtained in our experiment. Although it is difficult to compare the results of both papers because we do not have all the data need to perform a statistical test, TM-learning, whose results

are reported in this paper, seems to obtain slightly better results than the [12] method in most of the datasets.

On the other hand, the results of Table 1 only measure the goodness of the TM algorithm indirectly. Actually, the aim of the algorithm is to maximize the conditional log-likelihood and not to maximize directly the estimated accuracy of the classifier. In Table 2, the improvement of the conditional log-likelihood score for the discriminative model with respect to the generative one is shown. As described in Sections 2 and 3, the TM algorithm begins with the same parameters obtained by the generative model (that is the maximum likelihood parameters) and, following an iterative process, it modifies these parameters to maximize the conditional log-likelihood. Note that the TM algorithm is able to obtain a model with higher value for the conditional log-likelihood score in all datasets except for the TAN model learned from Australian, Crx and Soybean-large. This is because, in these three cases, the parameters that maximize the unconditional log-likelihood also represent a maximum for the conditional log-likelihood score. This maximum is not necessarily a global maximum but may be a local one because of the possible non-concavity of the conditional log-likelihood score [13]. However, even when generative and discriminative TAN are the same models for Australian, Crx and Soybean-large, the difference between the estimated accuracies is significant at the 90% level. This is because the conditional log-likelihood value reported in Table 2 is obtained from a classifier which has been learned using the whole dataset. On the other hand, for the cross-validation process, whose results are shown in Table 1, we learn the classifiers using only part of the dataset. Therefore, for each fold, the generative and discriminative classifiers are not necessarily the same.

5 Conclusions

Bayesian classifiers are usually generative classifiers, that is, their parameter configuration attempts to maximize the unconditional log-likelihood function. As far as we know, all the techniques for the discriminative learning of Bayesian classification models are generic numerical optimization methods [12, 13]. This paper presents a new statistical approach to the discriminative learning of Bayesian network classifiers by adapting the TM algorithm proposed by [1]. We present a theoretical development of the TM algorithm to be used with naïve Bayes and TAN, therefore providing an efficient discriminative learning of these models. However, the fact that the discriminative learning maximizes the conditional log-likelihood does not necessarily lead to a better performance of these kind of classifiers. It depends on the dataset and the classifier selected to model this dataset. This idea has been also shown, for example, by [4] and it is reasserted by the results from the experiments that we include in Section 4.

Discriminative learning with the TM algorithm, as it has been presented in this paper, can only be used in supervised classification problems with no missing values, but it can be extended to deal with missing values and with other problems such as unsupervised classification by using a hybrid of the TM and EM algorithms. On the other hand, the same idea can be extended to structural

learning, that is, searching in the space of structures and parameters in order to find the model which maximizes the conditional log-likelihood function.

Acknowledgments

This work was supported in part by the Spanish Ministerio de Ciencia y Tecnología under TIC2001-2973-C05-03 grant, by the ETORTEK-BIOLAN SAIO-TEK S-PE04UN25 projects of the Basque Government, by the Navarra Government under PhD grant, and by the University of the Basque Country under grant 9/UPV 00140.226-15334/2003.

The authors thank the Clínica Universitaria de Navarra, Spain, for providing the Tips dataset.

References

1. Edwards, D., Lauritzen, S.L.: The TM algorithm for maximising a conditional likelihood function. *Biometrika* **88** (2001) 961–972
2. Dawid, A.P.: Properties of diagnostic data distributions. *Biometrics* **32** (1976) 647–658
3. Rubinstein, Y.D., Hastie, T.: Discriminative vs. informative learning. In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*. (1997) 49–53
4. Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: A comparison of logistic regression and naïve Bayes. In: *Proceedings of the Sixteenth Advances in Neural Information Processing Systems 14*. (2002)
5. Jebara, T.: *Machine Learning: Discriminative and Generative*. Kluwer Academic Publishers (2003)
6. Fisher, R.A.: The use of multiple measurement. *Annals of Eugenics* **7** (1936) 179–188
7. Duda, R., Hart, P.: *Pattern Classification and Scene Analysis*. John Wiley and Sons (1973)
8. Hosmer, D., Lemeshow, S.: *Applied Logistic Regression*. John Wiley and Sons (1989)
9. Bishop, C.: *Neural Networks for Pattern Recognition*. Oxford Press (1996)
10. Pearl, J.: *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann (1988)
11. Neapolitan, R.E.: *Learning Bayesian Networks*. Prentice Hall (2003)
12. Greiner, R., Zhou, W., Su, X., Shen, B.: Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. *Machine Learning* (2004) Accepted for publication.
13. Roos, T., Wettig, H., Grünwald, P., Myllymäki, P., Tirri, H.: On discriminative Bayesian network classifiers and logistic regression. *Machine Learning* (2004) Accepted for publication.
14. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. *Machine Learning* **29** (1997) 131–164
15. Sundberg, R.: The convergence rate of the TM algorithm of Edwards and Lauritzen. *Biometrika* **89** (2002) 478–483

16. Santafé, G., Lozano, J.A., Larrañaga, P.: El algoritmo TM para clasificadores Bayesianos (in Spanish). Technical Report EHU-KZAA-IK-2/04, University of the Basque Country (2004)
17. Blake, C., Merz, C.: UCI repository of machine learning databases. <http://www.ics.uci.edu/~mlearn> (1998)
18. Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artificial Intelligence* **97** (1997) 273–324
19. Inza, I., Merino, M., Larrañaga, P., Quiroga, J., Sierra, B., Giralá, M.: Feature subset selection by genetic algorithms and estimation of distribution algorithms. A case study in the survival of cirrhotic patients treated with TIPS. *Artificial Intelligence in Medicine* **23** (2001) 187–205
20. Dougherty, J.R., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: *Proceedings of the Twelfth International Conference on Machine Learning*. (1995) 194–202
21. Fayyad, U., Irani, K.: Multi-interval discretization of continuous-valued attributes for classification learning. In: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*. (1993) 1022–1027
22. Mann, H., Whitney, D.: On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics* **18** (1947) 50–60

Constrained Score+(Local)Search Methods for Learning Bayesian Networks

José A. Gámez and J. Miguel Puerta

Dpto. de Informática and SIMD-*i³A*,
Universidad de Castilla-La Mancha,
02071 – Albacete, Spain
{jgamez, jpuerta}@info-ab.uclm.es

Abstract. The dominant approach for learning Bayesian networks from data is based on the use of a scoring metric, that evaluates the fitness of any given candidate network to the data, and a search procedure, that explores the space of possible solutions. The most used method inside this family is (iterated) hill climbing, because its good trade-off between CPU requirements, accuracy of the obtained model, and ease of implementation. In this paper we focus on the search space of dags and in the use of hill climbing as search engine. Our proposal consists in the reduction of the candidate dags or neighbors to be explored at each iteration, making the method more efficient on CPU time, but without decreasing the quality of the model discovered. Thus, initially the parent set for each variable is not restricted and so all the neighbors are explored, but during this exploration we take advantage of locally consistent metrics properties and remove some nodes from the set of candidate parents, constraining in this way the process for subsequent iterations. We show the benefits of our proposal by carrying out several experiments in three different domains.

1 Introduction

Bayesian Networks (BNs) are graphical models able to represent and manipulate efficiently n -dimensional probability distributions [15]. A BN uses two components to codify qualitative and quantitative knowledge: (a) A directed acyclic graph (dag), $G = (\mathbf{V}, E)$, where the nodes in $\mathbf{V} = \{X_1, X_2, \dots, X_n\}$ represent the random variables from the problem we want to solve, and the topology of the graph (the arcs in E) encodes conditional (in)dependence relationships among the variables (by means of the presence or absence of direct connections between pairs of variables); (b) a set of conditional probability distributions drawn from the graph structure: For each variable $X_i \in \mathbf{V}$ we have a family of conditional probability distributions $P(X_i|pa_G(X_i))$, where $pa_G(X_i)$ represents any combination of the values of the variables in $Pa_G(X_i)$, and $Pa_G(X_i)$ is the parent set of X_i in G . From these conditional distributions we can recover the joint distribution over $\mathbf{V}$:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i|pa_G(X_i)) \quad (1)$$

This decomposition of the joint distribution gives rise to important savings in storage requirements and also allow to do probabilistic inference by means of (efficient) local propagation schemes [13].

We denote that variables in $\mathbf{X}$ are conditionally independent (through d-separation) of variables in $\mathbf{Y}$ given the set $\mathbf{Z}$, in a dag G as $\langle \mathbf{X}, \mathbf{Y} | \mathbf{Z} \rangle_G$. The same sentence but in a probability distribution p is denoted as $I_p(\mathbf{X}, \mathbf{Y} | \mathbf{Z})$. A dag G is an *I-map* of a probability distribution p if $\langle \mathbf{X}, \mathbf{Y} | \mathbf{Z} \rangle_G \implies I_p(\mathbf{X}, \mathbf{Y} | \mathbf{Z})$, and is *minimal* if no arc can be eliminated from G without violating the I-map condition. G is a *D-map* of p if $\langle \mathbf{X}, \mathbf{Y} | \mathbf{Z} \rangle_G \longleftarrow I_p(\mathbf{X}, \mathbf{Y} | \mathbf{Z})$. When a dag G is both an I-map and a D-map of p , it is said that G and p are *isomorphic* models. It is always possible to build a minimal I-map of any given probability distribution p , but some distributions do not admit an isomorphic model [15].

In general, when learning Bayesian networks from data our goal is to obtain a dag being a minimal I-map of the probability distribution encoded by the dataset. Somewhat generalizing, there are two main approaches for learning BNs:

- *Score+search methods*. In these algorithms a function f is used to score a network/dag with respect to the training data, and a search method is used to look for the network with best score. Different Bayesian and non-Bayesian scoring metrics can be used ([14], chapter 8; [12]). As learning BNs from data is a NP-Hard problem [10], many heuristics have been proposed to guide the search. In this paper we focus on the application of local search methods to the problem of learning Bayesian networks in the space of dags ([5, 12, 9, 8, 11]; [14], chapter 9).
- *Constraint-based methods*. The idea underlying these methods is to satisfy as much independences present in the data as possible ([16]; [14], chapter 10). Statistical hypotheses testing is used to determine the validity of conditional independence sentences. There also exist hybrid algorithms that combine this approach with the described above, e.g. [1].

The main goal of this work is to show how efficiency (CPU time requirements) of local search algorithms for learning Bayesian Networks can be enlarged without decreasing their accuracy. To achieve this goal we take advantage of some interesting properties which possess some of the scoring metrics usually used. In our approach we use hill climbing as local search, but the method can be extended to any local algorithm which use the classical neighborhood (arc addition, arc deletion and arc inversion).

The paper is structured as follows: We begin in Section 2 with some preliminaries about local search in the space of graphs. Then, in Section 3 we revise some metrics interesting properties that would be used in our work. Sections 4 and 5 constitute the core of this paper, because in them we develop the algorithms and experimentally evaluate them. Finally, in Section 6 we present our final conclusions and outline future research.

2 Learning BNs by Local Search

The problem of learning the structure of a Bayesian network can be stated as follows: Given a training dataset $D = \{\mathbf{v}^1, \dots, \mathbf{v}^m\}$ of instances (configurations of values) of V , find the dag G^* such that

$$G^* = \arg \max_{G \in \mathcal{G}_n} f(G : D) \quad (2)$$

where $f(G : D)$ is a scoring metric which evaluates the merit of any candidate dag G with respect to the dataset D , and $\mathcal{G}_n$ is the set containing all the dags with n nodes.

Local Search (concretely Hill-Climbing) methods traverse the search space starting in an initial solution and doing a finite number of steps. At each step the algorithm only considers local changes, i.e. neighbor dags, and chooses that resulting in the greatest improvement of f . The algorithm stops its execution when there is no local change yielding an improvement of f . Because of this greedy behavior the execution stops when the algorithm is trapped in a solution that most times locally maximizes f rather than globally maximizing it. Different strategies are used to try to escape from local optima: restarts, randomness, etc.

The effectiveness and efficiency of a local search procedure depends on several aspects, like the neighborhood structure considered, the starting solution or the ability of fast evaluation of candidate subgraphs (neighbors). The neighborhood structure considered is directly related with the operators used to generate neighbors by applying local changes. In BN learning, the usual choices for local changes in the space of dags are arc addition, arc deletion and arc reversal. Of course, except in arc deletion we have to take care of avoiding to introduce directed cycles in the graph. Thus, there are $O(n^2)$ possible changes, n being the number of variables. With respect to the starting solution, the empty network is usually considered although random starting points or perturbed local optima are also used, specially in the case of iterated local search.

Efficient evaluation of neighbors/dags is based on an important property of scoring metrics: *decomposability* in presence of full data. In the case of BNs decomposable metrics evaluate a given dag as the addition of its nodes family score, i.e., the subgraphs formed by a node and its parents in G . Formally, if f is decomposable then:

$$f(G : D) = \sum_{i=1}^n f_D(X_i, Pa_G(X_i)) \quad (3)$$

$$f_D(X_i, Pa_G(X_i)) = f_D(X_i, Pa_G(X_i) : N_{x_i, pa_G(X_i)}) \quad (4)$$

where $N_{x_i, pa_G(X_i)}$ are the statistics of the variables X_i and $Pa_G(X_i)$ in D , i.e., the number of instances in D that match each possible instantiation of X_i and $Pa_G(X_i)$.

Thus, if a decomposable metric is used, a procedure that changes only one arc at each move can efficiently evaluate the neighbor obtained by this change.

This kind of (local) methods can reuse the computations carried out at previous stages, and only the statistics corresponding to the variables whose parents has been modified need to be recomputed. It is clear that a hill climbing algorithm using the operators above described can take advantage of this operation mode, concretely it has to measure the following differences when evaluating the improvement obtained by a neighbor dag:

1. Addition of $X_j \rightarrow X_i$: $f_D(X_i, Pa_G(X_i) \cup \{X_j\}) - f_D(X_i, Pa_G(X_i))$
2. Deletion of $X_j \rightarrow X_i$: $f_D(X_i, Pa_G(X_i) \setminus \{X_j\}) - f_D(X_i, Pa_G(X_i))$
3. Reversal of $X_j \rightarrow X_i$: It is obtained as the sequence: deletion($X_j \rightarrow X_i$) plus addition($X_i \rightarrow X_j$), so we compute $[f_D(X_i, Pa_G(X_i) \setminus \{X_j\}) - f_D(X_i, Pa_G(X_i))] + [f_D(X_j, Pa_G(X_j) \cup \{X_i\}) - f_D(X_j, Pa_G(X_j))]$

3 Asymptotic Behavior of a Scoring Metric

In section 2 we have introduced the concept of scoring metric to evaluate a dag G with respect to a dataset D . In this section we review some (desirable) properties of scoring metrics.

Definition 1. A scoring metric f is score equivalent if for any pair of equivalent¹ dags G and G' , $f(G : D) = f(G' : D)$.

Definition 2. [9] Let D be a dataset containing m iid samples from some distribution p . Let G and H be two dags. Then, a scoring metric f is consistent if in the limit as m grows large, the following two properties hold:

1. If H contains p and G does not contain p , then $f(H : D) > f(G : D)$
2. if H and G contain p , but G is simpler than H (has less parameters), then $f(G : D) > f(H : D)$

Proposition 1. [9] BDe, MDL and BIC metrics are score equivalent and consistent.

Definition 3. Let D be a dataset containing m iid samples from some distribution p . Let G be any dag, and G' the dag obtained by adding edge $X_i \rightarrow X_j$ to G . A scoring metric is locally consistent if in the limit as m grows large, the following two conditions hold:

1. If $\neg I_p(X_i, X_j | Pa_G(X_j))$, then $f(G : D) < f(G' : D)$
2. If $I_p(X_i, X_j | Pa_G(X_j))$, then $f(G : D) > f(G' : D)$

Proposition 2. [9] BDe, MDL and BIC metrics are locally consistent.

From this result, we can (asymptotically) assume that the differences computed by a locally consistent scoring metric f can be used as conditional independence tests over the dataset D . To do this, it is enough to suppose that D constitutes a sample which is isomorphic² to a graph.

¹ Two dags are equivalent if they share the same skeleton and the same v-structures [17].

² In fact, Chickering [9] proves that the isomorphy condition can be relaxed.

4 CHC: A Constrained Hill Climbing for Learning BNs

In this section we describe our proposal, that is, a hill climbing method for learning Bayesian networks in which we restrict the number of neighbors to be explored, and so evaluated, at each iteration. First, let us to recall the operation mode of a (unconstrained) hill climbing (HC) with the operations (local changes) described in Section 2:

1. Initialization: Choose a dag G as the starting point.
2. Neighbors generated by addition: For every node X_i and every node $X_j \notin Pa_G(X_i)$ compute the difference d between G and $G \cup \{X_j \rightarrow X_i\}$ as described in Section 2. Of course, neighbors in which adding $X_j \rightarrow X_i$ induces a directed cycle are avoided. Store the change which maximizes d .
3. Neighbors generated by deletion: For every node X_i and every node $X_j \in Pa_G(X_i)$ compute the difference d between G and $G \setminus \{X_j \rightarrow X_i\}$ as described in Section 2. Store the change which maximizes d .
4. Neighbors generated by reversal: For every node X_i and every node $X_j \in Pa_G(X_i)$ compute the difference d between G and $G \setminus (X_j \rightarrow X_i) \cup (X_i \rightarrow X_j)$ as described in Section 2. Again, modifications inducing directed cycles are not taken into account. Store the change which maximizes d .
5. From the three changes stored in the previous steps takes the one which maximizes d . If $d \leq 0$ then stops the algorithm and return G , else modify G by applying the selected change and return to step 2.

Now, we take advantage of the properties described in Section 3, concretely Definition 3 and Proposition 2, to constrain the number of modifications to be explored at each iteration of the hill climbing procedure. We call this algorithm *Constrained Hill Climbing* (CHC) and below we describe it (the differences with respect to HC are underlined):

1. Initialization: Choose a dag G as the starting point. $\forall X_i$ do $FP(X_i) = \emptyset$.
2. Neighbors generated by addition: For every node X_i and every node $X_j \notin (Pa_G(X_i) \cup FP(X_i))$ compute the difference d between G and $G \cup \{X_j \rightarrow X_i\}$ as described in Section 2. If $d < 0$ then add $\{X_j\}$ to $FP(X_i)$. Of course, neighbors in which adding $X_j \rightarrow X_i$ induces a directed cycle are not taken into account. Store the change which maximizes d .
3. Neighbors generated by deletion: For every node X_i and every node $X_j \in Pa_G(X_i)$ compute the difference d between G and $G \setminus \{X_j \rightarrow X_i\}$ as described in Section 2. If $d > 0$ then add $\{X_j\}$ to $FP(X_i)$. Store the change which maximizes d .
4. Neighbors generated by reversal: For every node X_i and every node $X_j \in Pa_G(X_i)$, such that, $X_i \notin FP(X_j)$ compute the difference d between G and $G \setminus \{X_j \rightarrow X_i\} \cup \{X_i \rightarrow X_j\}$ as described in Section 2. In this case $d = d_1 + d_2$, where d_1 is the difference obtained by removing X_j as parent of X_i , and d_2 is the difference obtained by adding X_i as parent of X_j . If $d_1 > 0$ then add $\{X_j\}$ to $FP(X_i)$. If $d_2 < 0$ then add $\{X_i\}$ to $FP(X_j)$. Again, modifications inducing directed cycles are avoided. Store the change maximizing d .

5. From the three changes stored in the previous steps takes the one which maximizes d . If $d \leq 0$ then stops the algorithm and return G , else modify G by applying the selected change and return to step 2.

Thus, CHC restricts the neighborhood of a dag G by constraining the set of allowed parents for each node X_i . To do this, we associate a set of *forbidden parents* (FP) to each node. The content of $FP(X_i)$ is modified by using the information provided by the differences computed at the current step, concretely we use definition 3 (locally consistent metric) to update $FP(X_i)$:

- Addition. If when adding $X_j \rightarrow X_i$ we get $d < 0$, then (asymptotically) $I_p(X_i, X_j | Pa_G(X_i))$, and so we do not have to test anymore the addition of X_j as parent of X_i .
- Deletion. This case is analogous to addition. Now, if we get $d > 0$ when deleting $X_j \rightarrow X_i$, then again we have that (asymptotically) $I_p(X_i, X_j | Pa_G(X_i))$.

Here, we have used metric differences as a sort of conditional independence tests, but in a more general framework, we could use any conditional independence test to manage the FP set. The main reason because of we have used metric differences is to save CPU time.

As in HC, at each step of CHC we choose the best operation with respect to the improvement of f , so we can easily ensure monotonicity, i.e., $f(G : D) \leq f(G' : D)$, where G' is the neighbor of G which maximizes the difference d . As CHC stops when there is no neighbor of G which improves $f(G)$, and due to CHC monotonic behavior, termination is guaranteed.

There are, as expected, some differences between the behavior of CHC and HC because of constraining the set of allowed parents for each variable during the search. The first difference we can appreciate is that from the same starting point (dag) both algorithms can obtain different outputs. In fact, CHC relies on the conditions required in definitions 2 and 3, which ensure the asymptotic behavior but rarely hold in real datasets. Because of this, CHC can get stuck in a locally sub-optimal solution while HC gets stuck in a locally optimal solution. The second difference we can observe is related with the *kind* of output obtained by each algorithm. Thus, if $\hat{G}$ is the dag obtained by applying HC over a dag G_0 as our starting point, then the following proposition hold:

Proposition 3. *Let D be a dataset containing m i.i.d samples from some distribution p . Let $\hat{G}$ be the dag obtained by running (unconstrained) HC algorithm by taking a dag G_0 as the initial solution, i.e., $\hat{G} = HC(G_0)$. If the metric f used to evaluate dags in HC is locally consistent then $\hat{G}$ is a minimal I-map of p in the limit as m grows large.*

Proof sketch: First we prove that $\hat{G}$ is an I-map of p . Let us suppose the converse, i.e., $\hat{G}$ is not an I-map of p . Then there is at least a pair of variables X_i and X_j such that $\langle X_i, X_j | Pa_{\hat{G}}(X_i) \rangle_{\hat{G}}$ and $\neg I_p(X_i, X_j | Pa_{\hat{G}}(X_i))$. Thus, $\hat{G}$ cannot be a local optimum of f because the addition of arc $X_j \rightarrow X_i$ has positive difference. Now we prove the minimal condition. Again let us suppose the converse, that

is, there exists $X_j \in Pa_{\hat{G}}(X_i)$ such that $I_p(X_i, X_j | Pa_{\hat{G}}(X_i))$. If so, $\hat{G}$ cannot be a local optimum because there is (at least) a deletion operation with positive difference. $\square$

This result is not, somewhat, surprising regarding that HC algorithm traverses a subset of equivalence classes as defined by Chickering [9]. However, in CHC we cannot be sure that $\hat{G} = CHC(G_0)$ is, asymptotically, an I-map of p , as we show in a counterexample. Let us consider the dag G_t shown in Figure 1.a as our *true* model, and suppose that we get a dataset D by sampling from G_t . In the initial step of a hill climbing algorithm the six arcs $\{X1 \rightarrow X2, X2 \rightarrow X1, X2 \rightarrow X3, X3 \rightarrow X2, X3 \rightarrow X4, X4 \rightarrow X3\}$ should be those having greatest positive difference (with respect to the empty network) because there is a direct dependence relation in the pairs $(X1, X2)$, $(X2, X3)$ and $(X3, X4)$. Let us suppose that because of the sampling process the algorithm selects $X2 \rightarrow X3$ as the arc with greatest difference (Fig. 1.b). Notice also that after this step we (should) have $FP(X1) = \{X3, X4\}$, $FP(X2) = \emptyset$, $FP(X3) = \{X1\}$ and $FP(X4) = \{X1\}$. Figures 1.c and 1.d represents the next two likely steps carried out by CHC, being Figure 1.d the dag in which these algorithms get stuck (because of the restrictions in the parent sets) which clearly is not a minimal I-map of G_t . On the other hand, it is well known that when a (unconstrained) local search method makes a mistake in the direction of an arc, it usually compensates the error by *covering* such arc, which gives rise to the dag in Figure 1.e (a minimal I-map of G_t).

Because of the two problems reported above, we modify step 5 of CHC algorithm replacing [... If $d \leq 0$ then stops the algorithm and return G , else ...] by [... If $d \leq 0$ then return $HC(G)$, else ...]. In this way, an unconstrained local search is carried out by taking the output of CHC as starting point, which (likely) will improve the quality of the obtained solution with (hopefully) a smaller cost, because as we start the process in a very good point it is expected to need few iterations to get a locally optimal solution. We will re-take this point in the experiments analysis.

Additionally, we propose a second version of our CHC algorithm: CHC-2. The idea in which CHC-2 relies is as follows: in CHC we include X_j in $FP(X_i)$ when due to the obtained difference in f , we can (asymptotically) consider that $I(X_i, X_j | Pa_G(X_i))$ hold. However, because of this independence we can discard the undirected link, i.e., we can also include X_i in $FP(X_j)$. Besides, in CHC-2 the modifications over $FP(\cdot)$ can be exploited in the current iteration, e.g., we

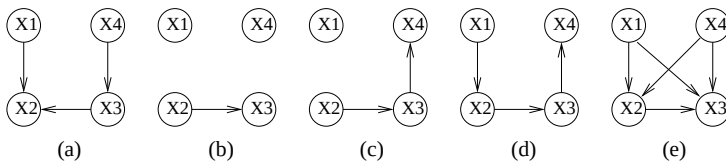


Fig. 1. A series of dags

can avoid to measure $\text{adding}(X_i \rightarrow X_j)$ if we obtained a negative difference when measuring $\text{adding}(X_j \rightarrow X_i)$.

To end with this section we briefly discuss about related work. Thus, the underlying idea in CHC and CHC-2 has been previously used *heuristically* in some others algoirithms, e.g., the sparse candidate [11]. However, the operation mode of that algorithm is quite far of our proposal because the sparse candidate is in fact an iterated HC that at each (outer) iteration restricts the number of candidate parents for each variable X_i to the k most promising ones, k being the same value for every variable. Perhaps, our approach is more related to classical constraint-based BN learning algorithm: PC [16], because it uses conditional independence tests $I(X, Y|Z)$ to guide the search. However, the main difference between such algorithm and our approach relies in the fact that we set the current parents as a d-separator set in G , while PC algorithm needs to perform tests with respect to all the possible subsets of $\text{adjacents}_G(X) \setminus \{Y\}$.

5 Experimental Results

In this section we evaluate experimentally the usefulness of the proposed algorithms: CHC and CHC-2. In the three cases (Hc, CHC and CHC-2) use a cache to store (and retrieve) the statistics computed during the search, in order to make the execution faster. We have selected three domains as our test suite:

- ALARM [3]. This domain is considered as a benchmark in BN learning literature. We use the first 3000 cases of the original 20000 cases dataset (sampled from the ALARM network). The ALARM network has 37 variables and 46 arcs. It is used for diagnosis in a medical domain.
- INSURANCE [4]. This domain has been used in other works to test BNs learning algorithms (e.g. [6, 2]). In this case we use three datasets (each one with 10000 cases) sampled from the INSURANCE network. For this domain we report the average of running the algorithms over the three datasets. The INSURANCE network has 27 variables and 52 edges. It is used to evaluate car insurance risks.
- SYNTHETIC. This domain has been constructed for this work. We generated a network with 91 variables and 162 edges. From the network we sampled a dataset with 10000 cases.

The logarithmic version of BDeu (uniform BDe) metric [12, 5] (sample size equal to 1) has been used as scoring metric in the three algorithms. All the algorithms have been run three times over each dataset, corresponding to the use of three different starting points: the empty network and the networks obtained by using algorithms PC [16] and K2SN [7].

Tables 1, 2 and 3 show the results obtained in our experiments. The following performance measures are reported (mean and standard deviation over the three initializations, and also the best run for accuracy parameters): the BDeu value of the final solution with respect to the data set; the number of arcs added (A), deleted (D) and reversed (R) in the learnt network with respect to the original one; the number of evaluated statistics (EstEv), i.e., the number of entries in

Table 1. Results obtained for ALARM network

CHC-2				CHC				HC			
	μ	σ	Best	μ	σ	Best		μ	σ	Best	
BDeu	-33125,37	11,12	-33119,0	-33125,37	11,12	-33119,0		-33125,37	11,12	-33119,0	
A	3,33	0,58	3	3,33	0,58	3		3,33	0,58	3	
D	2,00	0,00	2	2,00	0,00	2		2,00	0,00	2	
R	1,33	1,15	2	1,33	1,15	2		1,33	1,15	2	
EstEv	2867,33	1548,77		2973,00	1557,61			3389,67	1420,12		
TEst	29,88E03	76,52E02		26,51E03	13,36E03			85,38E03	62,53E03		
NVars	3,16	0,27		3,10	0,28			3,24	0,24		

Table 2. Results obtained for INSURANCE network

CHC-2				CHC				HC			
	μ	σ	Best	μ	σ	Best		μ	σ	Best	
BDeu	-134014,40	1378,78	-132617	-133958,04	1313,17	-132612		-134022,28	1317,45	-132612	
A	10,89	10,54	1	11,11	10,81	0		10,89	10,81	0	
D	11,89	3,62	9	11,78	3,73	8		11,89	3,79	8	
R	8,56	7,38	0	8,56	7,38	0		8,67	7,48	0	
EstEv	1901,89	1080,48		2006,22	1029,88			2233,00	1104,77		
TEst	20,73E03	11,23E03		20,16E03	11,37E03			45,09E03	24,76E03		
NVars	3,31	0,40		3,38	0,33			3,32	0,37		

Table 3. Results obtained for SYNTHETIC network

CHC-2				CHC				HC			
	μ	σ	Best	μ	σ	Best		μ	σ	Best	
BDeu	-422744,90	516,44	-422416	-423096,92	463,39	-422589		-423052,87	463,98	-422518	
A	26,67	11,55	20	26,67	9,87	22		30,67	9,50	21	
D	12,00	1,00	12	12,33	2,08	13		14,00	1,00	14	
R	20,67	12,66	16	21,67	10,02	14		23,67	10,26	15	
EstEv	22165,33	11298,25		22615,33	11281,92			27477,67	10107,36		
TEst	76,78E04	17,24E04		43,68E04	23,07E04			20,12E05	10,05E05		
NVars	3,70	0,19		3,53	0,36			3,77	0,35		

the cache; the number of accesses to the cache (TEst), i.e., the total number of evaluated statistics if cache is not used; and the average number of variables (NVars) in the computed statistics.

The following analysis can be done from the obtained results:

- With respect to the accuracy of the discovered networks, the three algorithms obtain similar figures in all the parameters (BDeu, A, D and R). Thus, apart of ALARM domain (where the three algorithms gets exactly the same results) we can observe that CHC-2 gets a better BDeu mean value than HC, although it has greater deviation. CHC improves BDeu values obtained by HC in INSURANCE domain (where it gets the best value) while HC improves CHC in SYNTHETIC domain. The same happens when parameters measuring the similarity to the original networks are used, i.e., A, D and R, the three algorithms obtains similar results except in the case of SYNTHETIC where the mean values for these parameters obtained by HC are considerable worse than those obtained by CHC-2 and CHC.

- With respect to algorithms CPU efficiency we can see that $\text{EstEv}(\text{CHC2}) < \text{EstEv}(\text{CHC}) < \text{EstEv}(\text{HC})$ and $\text{NVars}(\text{CHC}) < \text{NVars}(\text{CHC-2}) < \text{NVars}(\text{HC})$ in the three cases, so we think that this would be the expected trend for other domains (datasets). As the running time of a scoring-based learning algorithm is mostly spent in the evaluation of statistics from the database, and this time increases exponentially with the number of variables, we can approximate our algorithms (CPU) time complexity as a function of $\text{EstEv} \cdot 2^{\text{NVars}}$. Therefore, CHC and CHC-2 are considerably faster than HC.

To gain more insight in the behavior of the three algorithms over the selected domains, figures 2.a, 2.b and 2.c show a plot (for each algorithm when the empty network is taken as starting point) which relates accuracy (BDeu) with complexity (EstEv). In the plots each point represents an iteration and from them we can see that the same behavior is reproduced for the three domains and the following observations can be done:

- The greatest number of new statistics ($O(n^2)$) are computed at the first iteration because at this stage the cache is empty. In fact, CHC-2 computes less statistics at this stage because once $X_i \rightarrow X_j$ is discarded because of getting a negative difference, then $X_j \rightarrow X_i$ is not considered anymore.
- In the next iterations the number of *new* statistics to be computed is small because many of the required statistics retrieved from the cache. In the plots we can appreciate how CHC-2 and CHC evaluate fewer statistics than HC in these iterations due to the constrained set of allowed parents.
- On the other hand, once CHC and CHC-2 get firstly stucked, they run an unconstrained HC which increases considerably the number of *new* statistics evaluated. However, as we conjecture in Section 4 only a few iterations are needed (5/1/24 in CHC and 6/6/46 in CHC-2 for ALARM/INSURANCE/ SYNTHEIC) before convergence with respect to the number of iterations carried out during the constrained search (66/52/208 in CHC and 64/54/235 in CHC-2). Again, the large number of new statistics is computed at iteration 0 of the unconstrained HC, because a great deal of the $O(n^2)$ possible statistics has to be computed.
- From the plots we can also corroborate our suspicious (Section 4) about the quality of the solutions obtained by the constrained search (without running the posterior unconstrained HC) with respect to the networks obtained by HC. In fact, we always get the following order (with the empty network): $\text{BDeu}(\text{CHC-2}) < \text{BDeu}(\text{CHC}) < \text{BDeu}(\text{HC})$. However, in situations in which limited CPU time is available and anytime behavior is required, it is clear that the constraint-based algorithms are clearly advantageous.

Our last comment in this section is about the total number of statistics (TEst). Apart that HC requires by far much more statistics than CHC and CHC-2, it could be interpreted as a bit surprising that CHC-2 needs more total statistics than CHC. However this fact has its explanation in the greater number of iterations required by CHC-2 in its second stage (i.e. the unconstrained HC).

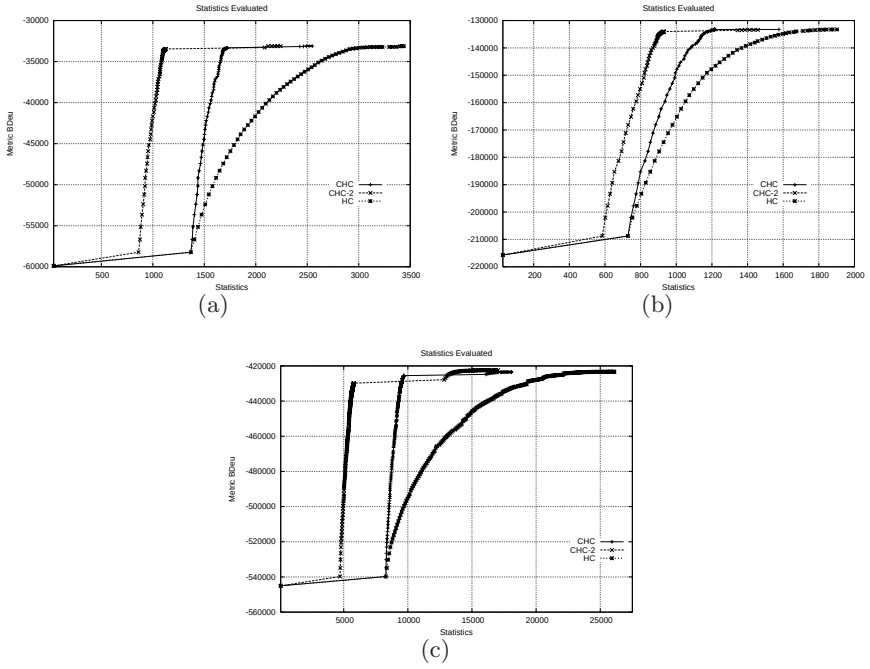


Fig. 2. A plot of BDeu against EstEv for (a) ALARM, (b) INSURANCE and (c) SYNTHETIC networks

Although this value is of minor importance (with respect to EstEv) because of the use of a cache, it can play an important role in at least two situations: (1) obviously if cache is not used, and (2) if the problem is so complex that all the *different* statistics cannot be simultaneously stored. In these cases the advantages of the constrained search over the unconstrained one are even more evident.

6 Concluding Remarks

In this paper we have proposed two constrained score-plus-(local)search algorithms for learning Bayesian networks. Both methods consist in the use of a hill climbing algorithm with the classical operators (addition, deletion and reversal) but in which we restrict the neighborhood by using a list of parents not allowable for each variable. The underlying idea of the method relies on the theoretical property of *local consistency* exhibited by some scoring metrics as BDe, MDL and BIC. Concretely we exploit the relation between the scores of a dag which includes or not an arc $X_j \rightarrow X_i$ and the (in)dependence of X_i and X_j given $Pa_G(X_i)$. The experiments show that if the solution obtained by the constrained search is improved by an unconstrained one, then we get solution as least as good as when only using unconstrained search, but more efficiently with respect to

CPU time. Also, the new algorithms seem to be even more appropriated as the complexity domain grows and if anytime behavior is required.

For future research, we plan to carry out a more systematic experimentation and extend the comparative analysis to other related approaches. Furthermore, different local search methods and search spaces (as PDAGs [9] and RPDAGs [2]) will be considered.

Acknowledgements

This work has been partially supported by Spanish Ministerio de Ciencia y Tecnología, Junta de Comunidades de Castilla La Mancha and FEDER under projects TIC2001-2973-CO5-05, TIN2004-06204-C03-03 and PBC-02-002.

References

1. S. Acid and L.M. de Campos. A hybrid methodology for learning belief networks: Benedict. *International Journal of Approximate Reasoning*, 27(3):235–262, 2001.
2. S. Acid and L.M. de Campos. Searching for Bayesian Network Structures in the Space of Restricted Acyclic Partially Directed Graphs. *Journal of Artificial Intelligence Research*, 18:445–490, 2003.
3. I.A. Beinlich, H.J. Suermondt, R.M. Chavez, and G.F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proc. of the 2nd European Conf. on Artificial Intelligence in Medicine*, 247–256, 1989.
4. J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29(2):213–244, 1997.
5. W. Buntine. Theory refinement of Bayesian networks. In *Proc. of the 7th Conf. on Uncertainty in Artificial Intelligence*, 52–60, 1991.
6. L.M. de Campos, J.M. Fernández-Luna, J.A. Gámez, and J.M. Puerta. Ant colony optimization for learning Bayesian networks. *International Journal of Approximate Reasoning*, 31:291–311, 2002.
7. L.M. de Campos and J.M. Puerta. Stochastic local and distributed search algorithms for learning belief networks. In *Proc. of 3rd Int. Symp. on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Model*, 109–115, 2001.
8. L.M. de Campos and J.M. Puerta. Stochastic local search algorithms for learning belief networks: Searching in the space of orderings. *LNAI*, 2143:228–239, 2001.
9. D.M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.
10. D.M. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks is NP-Complete. In D. Fisher and H. Lenz, Eds., *Learning from Data: Artificial Intelligence and Statistics V*, Springer-Verlag, 121–130, 1996.
11. N. Friedman, I. Nachman and D. Per. Learning Bayesian networks from massive datasets: The “sparse candidate” algorithm. In *Proc. of the 15th Conf. on Uncertainty in Artificial Intelligence*, 201–210, 1999.
12. D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–244, 1995.

13. F.V. Jensen. *Bayesian Networks and Decision Graphs*. Springer Verlag, 2001.
14. R. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003.
15. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, 1988.
16. P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Lecture Notes in Statistics 81, Springer Verlag, 1993.
17. T. Verma and J. Pearl. Equivalence and synthesis of causal models. In Proc. of the 6th Conf. on Uncertainty in Artificial Intelligence, pages 220-227, 1991.

On the Use of Restrictions for Learning Bayesian Networks

Luis M. de Campos and Javier G. Castellano

Departamento de Ciencias de la Computación e Inteligencia Artificial,
E.T.S.I. Informática, Universidad de Granada, 18071 – Granada, Spain
`{lci, fjgc}@decsai.ugr.es`

Abstract. In this paper we explore the use of several types of structural restrictions within algorithms for learning Bayesian networks. These restrictions may codify expert knowledge in a given domain, in such a way that a Bayesian network representing this domain should satisfy them. Our objective is to study whether the algorithms for automatically learning Bayesian networks from data can benefit from this prior knowledge to get better results. We formally define three types of restrictions: existence of arcs and/or edges, absence of arcs and/or edges, and ordering restrictions, and also study their interactions and how they can be managed within Bayesian network learning algorithms based on the score+search paradigm. Then we particularize our study to the classical local search algorithm with the operators of arc addition, arc removal and arc reversal, and carry out experiments using this algorithm on several data sets.

1 Introduction

Nowadays, Bayesian networks [15] constitute a widely accepted formalism for representing uncertain knowledge and for efficiently reasoning with it. A Bayesian network (BN) is a graphical representation of a joint probability distribution, which consists of a qualitative part, a directed acyclic graph (DAG), and a quantitative one, a collection of numerical parameters, usually conditional probability tables. There has been a lot of work in recent years on the automatic learning of Bayesian networks from data and, consequently, there are a great many learning algorithms, based on different methodologies. However, little attention has been paid to the use of additional expert knowledge, not present in the data, in combination with a given learning algorithm. This knowledge could help in the learning process and contribute to get more accurate results, and even reduce the search effort of the BN representing a given domain of knowledge.

In this paper we address this problem by defining several types of restrictions, that codify some kinds of expert knowledge, to be used in conjunction with algorithms for learning Bayesian networks. More precisely, we shall consider three types of restrictions: (1) existence of arcs and edges, (2) absence of arcs and edges, and (3) ordering restrictions. All of them will be considered “hard” restrictions (as opposed to “soft” restrictions [13]), in the sense that they are assumed to

be true for the BN representing the domain of knowledge, and therefore all the candidate BNs must necessarily satisfy them. The paper is structured as follows: in Section 2 we briefly give some preliminary basic concepts about learning the structure of Bayesian networks. Section 3 formally introduces the three types of restrictions that we are going to study. In Section 4 we describe how to represent the restrictions and how to manage them, including their self-consistency and the consistency of the restrictions with a given DAG. Section 5 studies how to combine the restrictions with learning algorithms based on the score+search paradigm, and particularizes this study to the case of algorithms based on local search. Section 6 discusses the experimental results. Finally, Section 7 contains the concluding remarks.

2 Notation and Preliminaries

Let us consider a finite set $\mathcal{V} = \{x_1, x_2, \dots, x_n\}$ of discrete random variables, each variable taking on values from a finite set. We shall use lower-case letters for variable names, and capital letters to denote sets of variables. The structure of a Bayesian network on this domain is a directed acyclic graph (DAG) $G = (\mathcal{V}, E_G)$, where E_G represents the set of arcs.

The problem of learning the structure of a BN from data is that given a training set $\mathcal{D}$ of instances of the variables in $\mathcal{V}$, find the network that, in some sense, best matches $\mathcal{D}$. The learning algorithms may be subdivided into two general approaches: methods based on conditional independence tests, and methods based on a scoring function and a search procedure (for references, see [2]). In this paper we are more interested in the algorithms based on the score+search paradigm, which attempt to find a graph that maximizes the selected score. All use a scoring function, usually defined as a measure of fit between the graph and the data, in combination with a search method in order to measure the goodness of each explored structure from the space of feasible solutions. Most of these algorithms use different search methods but the same search space: the space of DAGs¹. Our objective is to narrow this (hyper-exponential) search space by introducing several types of restrictions that the elements in this space must satisfy.

3 Types of Restrictions

We are going to study three types of restrictions on the DAG structures defined for the domain $\mathcal{V}$, namely existence, absence and ordering restrictions.

3.1 Existence or Arcs and/or Edges

Consider two subsets of pairs of variables $\mathcal{E}_a, \mathcal{E}_e \subseteq \mathcal{V} \times \mathcal{V}$, with $\mathcal{E}_a \cap \mathcal{E}_e = \emptyset$. They will be interpreted as follows:

¹ Although other alternatives are possible, as searching in a space of equivalence classes of DAGs or in a space of orderings, in this paper we shall focus only on the space of DAGs.

- $(x, y) \in \mathcal{E}_a$: the arc $x \rightarrow y$ must belong to any DAG in the search space.
- $(x, y) \in \mathcal{E}_e$: the edge (i.e. the arc without direction) $x-y$ must belong to any DAG in the search space. In other words, either the arc $x \rightarrow y$ or the arc $y \rightarrow x$ must appear in any DAG.

An example of the use of existence restrictions may be any BAN algorithm [5], a BN learning algorithm for classification, which fixes the naive Bayes structure (i.e. arcs from the class variable to all the attribute variables) and searches for the appropriate additional arcs, linking pairs of attribute variables.

3.2 Absence of Arcs and/or Edges

Now, consider the subsets $\mathcal{A}_a, \mathcal{A}_e \subseteq \mathcal{V} \times \mathcal{V}$, with $\mathcal{A}_a \cap \mathcal{A}_e = \emptyset$. Their meaning is the following:

- $(x, y) \in \mathcal{A}_a$: the arc $x \rightarrow y$ cannot be present in any DAG in the search space.
- $(x, y) \in \mathcal{A}_e$: the edge $x-y$ cannot appear in any DAG in the search space (i.e. neither the arc $x \rightarrow y$ nor the arc $y \rightarrow x$ can appear).

An example of the use of absence restrictions is a selective naive Bayesian classifier [14], which forbids arcs between attribute variables and also arcs from the attributes to the class variable.

3.3 Partial Ordering

Consider the subset $\mathcal{R}_o \subseteq \mathcal{V} \times \mathcal{V}$. In this case the interpretation is:

- $(x, y) \in \mathcal{R}_o$: all the DAGs in the search space have to satisfy that x precedes y in some total ordering of the variables compatible with the DAG structure.

We need some additional concepts to better understand the meaning of this kind of restriction. We shall say that a total ordering, σ , of the set of variables $\mathcal{V}$ is compatible with a partial ordering, μ , of the same set of variables if

$$\forall x, y \in \mathcal{V}, \text{ if } x <_{\mu} y \text{ then } x <_{\sigma} y,$$

i.e. if x precedes y in the ordering μ then also x precedes y in the ordering σ . Notice that a DAG determines a partial ordering on its variables: if there is a directed path from x to y in a DAG G , then x precedes y . Therefore, we can also say that a total ordering σ on the set $\mathcal{V}$ is compatible with a DAG $G = (\mathcal{V}, E)$ if

$$\forall x, y \in \mathcal{V}, \text{ if } x \rightarrow y \in E \text{ then } x <_{\sigma} y.$$

The ordering restrictions may represent, for example, temporal or functional precedence between variables. Notice that the restriction $(x, y) \in \mathcal{R}_o$ also means that there is not a directed path from y to x in any of the DAGs in the search space. Examples of use of ordering restrictions are the BN learning algorithms that require a fixed total ordering of the variables (as the K2 algorithm [7]).

4 Representing and Managing the Restrictions

In order to manage the restrictions it is useful to represent them graphically. So, the existence restrictions can be represented by means of a partially directed graph $G_e = (\mathcal{V}, E_e)$, where each element (x, y) in $\mathcal{E}_a$ is associated with the corresponding arc $x \rightarrow y \in E_e$, and each element (x, y) in $\mathcal{E}_e$ is associated with the edge $x-y \in E_e$. The absence restrictions are represented by means of another partially directed graph $G_a = (\mathcal{V}, E_a)$, where the elements (x, y) in $\mathcal{A}_a$ correspond with arcs $x \rightarrow y \in E_a$ and the elements (x, y) in $\mathcal{A}_e$ are associated with edges $x-y \in E_a$. Finally, the ordering restrictions are represented by using a directed graph $G_o = (\mathcal{V}, E_o)$, with (x, y) in $\mathcal{R}_o$ being associated with the arc $x \rightarrow y \in E_o$. Notice that, as we are assuming that the ordering restrictions form a partial ordering (i.e. the relation is transitive), we are not forced to include in G_o an arc for each element in $\mathcal{R}_o$. G_o may be any graph such that its transitive closure contains an arc for each element in $\mathcal{R}_o$. For example, to represent a total ordering restriction $x_1 < x_2 < \dots < x_n$ it suffices to include in G_o the $n-1$ arcs $x_i \rightarrow x_{i+1}$, $i = 1, \dots, n-1$, instead of having a complete graph with all the arcs $x_i \rightarrow x_j$, $\forall i < j$.

Now, let us formally define when a given DAG G is consistent with a set of restrictions (i.e. G verifies them):

Definition 1. Let $G = (\mathcal{V}, E)$ be a DAG and $G_e = (\mathcal{V}, E_e)$, $G_a = (\mathcal{V}, E_a)$ and $G_o = (\mathcal{V}, E_o)$ be the graphs representing the existence, absence and ordering restrictions, respectively. We say that

- G is consistent with the existence restrictions if and only if
 - $\forall x, y \in \mathcal{V}$, if $x \rightarrow y \in E_e$ then $x \rightarrow y \in E$, and
 - $\forall x, y \in \mathcal{V}$, if $x-y \in E_e$ then $x \rightarrow y \in E$ or $y \rightarrow x \in E$.
- G is consistent with the absence restrictions if and only if
 - $\forall x, y \in \mathcal{V}$, if $x \rightarrow y \in E_a$ then $x \rightarrow y \notin E$, and
 - $\forall x, y \in \mathcal{V}$, if $x-y \in E_a$ then $x \rightarrow y \notin E$ and $y \rightarrow x \notin E$.
- G is consistent with the ordering restrictions if and only if
 - there exists a total ordering σ of the variables in $\mathcal{V}$ compatible with both G and G_o .

Before using a set of restrictions we must be sure that we are not demanding conditions impossible to satisfy. In this sense, we shall say that a set of restrictions is self-consistent if there is some DAG that is consistent with them. Testing the self-consistency of each type of restriction separately is very simple²:

Proposition 1. Let $G_e = (\mathcal{V}, E_e)$, $G_a = (\mathcal{V}, E_a)$ and $G_o = (\mathcal{V}, E_o)$ be the graphs representing existence, absence and ordering restrictions, respectively. Then

- The set of existence restrictions is self-consistent if and only if the graph G_e has no directed cycle.

² The proofs of this and all the other propositions stated in the paper are not given, because of their relative simplicity and space limitations.

- *The set of absence restrictions is always self-consistent.*
- *The set of ordering restrictions is self-consistent if and only if G_o is a DAG.*

When several types of restrictions are considered simultaneously, some interactions can occur among each other. These interactions may give rise to inconsistencies. For example, the existence and absence of the same arcs; or the existence of some arcs that (as they implicitly also represent partial ordering restrictions) may contradict with ordering restrictions. For instance, $x \rightarrow v, v \rightarrow y \in E_e$ contradicts with $y \rightarrow z, z \rightarrow t, t \rightarrow x \in E_o$.

It also may happen that some absence or ordering restrictions force an existence restriction. For instance, if an arc must exist in either direction (i.e. $x-y \in E_e$) but an absence or ordering restriction indicates that some direction is forbidden (e.g. $x \rightarrow y \in E_a$ or $y \rightarrow x \in E_o$), then the other direction is forced ($x-y$ should be replaced by $y \rightarrow x$ in E_e). This can also produce interactions among the three types of restrictions, giving rise to inconsistencies. For example, if $y \rightarrow t, t \rightarrow x, x-z, z-y \in E_e, x \rightarrow z \in E_o$ and $y \rightarrow z \in E_a$, the absence and ordering restrictions force the orientation of the edges $x-z$ and $z-y$ which, together with the other existence restrictions, generate a directed cycle. The following result characterizes global self-consistency of the restrictions, in terms of simple operations on graphs.

Proposition 2. *Let $G_e = (\mathcal{V}, E_e)$, $G_a = (\mathcal{V}, E_a)$ and $G_o = (\mathcal{V}, E_o)$ be the graphs representing existence, absence and ordering restrictions, respectively. Let $G_{re} = (\mathcal{V}, E_{re})$ be the refined graph of existence restrictions³ defined as*

$$E_{re} = \{x \rightarrow y \mid x \rightarrow y \in E_e\} \cup \{y \rightarrow x \mid x-y \in E_e, x \rightarrow y \in E_a\} \cup \{x-y \mid x-y \in E_e, x \rightarrow y \notin E_a, y \rightarrow x \notin E_a\}$$

Then the three sets of restrictions are self-consistent if and only if

$$G_{re} \cap G_a = G_\emptyset \text{ and } G_{re} \cup G_o \text{ has no directed cycle,}$$

where $G_\emptyset$ is the empty graph (a graph having neither arcs nor edges), and both the union and the intersection of two partially directed graphs use the convention that $\{x \rightarrow y\} \cup \{x-y\} = \{x \rightarrow y\}$ and $\{x \rightarrow y\} \cap \{x-y\} = \{x \rightarrow y\}$.

Testing the consistency of a DAG with a set of restrictions can also be reduced to simple graph operations, as the following result shows:

Proposition 3. *Let $G_e = (\mathcal{V}, E_e)$, $G_a = (\mathcal{V}, E_a)$ and $G_o = (\mathcal{V}, E_o)$ be graphs representing self-consistent existence, absence and ordering restrictions, respectively, and let $G = (\mathcal{V}, E)$ a DAG. Then G is consistent with the restrictions if and only if*

$$G \cup G_e = G, G \cap G_a = G_\emptyset \text{ and } G \cup G_o \text{ is a DAG.}$$

³ This is the same graph G_e with the edges whose direction is forced by virtue of some absence restriction being replaced by the corresponding arcs.

5 Using the Restrictions for Learning

In case that we want to get a Bayesian network from data using a score+search learning algorithm and we have a set of (self-consistent) restrictions, it seems natural to use them to reduce the search space and force the algorithm to return a DAG consistent with the restrictions. A general mechanism to do it, which is valid for any algorithm, is very simple: each time the search process selects a candidate DAG G to be evaluated by the scoring function, we can use the result in the previous proposition to test whether G is consistent with the restrictions, and reject it otherwise.

However, this general procedure may be somewhat inefficient. It would be convenient to adapt it to the specific characteristics of the learning algorithm being used. We are going to do that for the case of the classical score+search learning algorithm based on local search [13], which uses the operators of arc insertion, arc deletion and arc reversal. We start from the current DAG G , which is consistent with the restrictions, and let G' be the DAG obtained from G by applying one of the operators. Let us see which are the conditions necessary and sufficient to assure that G' is also consistent with the restrictions.

Proposition 4. *Let $G_e = (\mathcal{V}, E_e)$, $G_a = (\mathcal{V}, E_a)$ and $G_o = (\mathcal{V}, E_o)$ be graphs representing self-consistent existence, absence and ordering restrictions, respectively, and let $G = (\mathcal{V}, E)$ a DAG consistent with the restrictions.*

- (a) *Arc insertion: Let $G' = (\mathcal{V}, E')$, $E' = E \cup \{x \rightarrow y\}$, with $x \rightarrow y \notin E$. Then G' is consistent with the restrictions if and only if*
 - $x \rightarrow y \notin E_a$ and $x - y \notin E_a$,
 - *there is not any directed path from y to x in $G \cup G_o$.*
- (b) *Arc deletion: Let $G' = (\mathcal{V}, E')$, $E' = E \setminus \{x \rightarrow y\}$, with $x \rightarrow y \in E$. Then G' is consistent with the restrictions if and only if*
 - $x \rightarrow y \notin E_e$ and $x - y \notin E_e$.
- (c) *Arc reversal: Let $G' = (\mathcal{V}, E')$, $E' = (E \setminus \{x \rightarrow y\}) \cup \{y \rightarrow x\}$, with $x \rightarrow y \in E$. Then G' is consistent with the restrictions if and only if*
 - $x \rightarrow y \notin E_e$, $y \rightarrow x \notin E_a$ and $x \rightarrow y \notin E_o$,
 - *if we exclude the arc $x \rightarrow y$, there is not any other directed path from x to y in $G \cup G_o$.*

Notice that the conditions about the absence of directed paths between x and y in the previous proposition have also to be checked by the algorithm that does not consider the restrictions (using in this case the DAG G instead of $G \cup G_o$), so that the extra cost of managing the restrictions is quite reduced: two or three tests about the absence of either an arc or an edge from a graph.

It is also interesting to notice that other score+search learning algorithms, more sophisticated than a simple local search, can also be easily extended to efficiently deal with the restrictions. There are many BN learning algorithms that perform a search more powerful than local search but use the same basic operators, as variable neighborhood search [10], tabu search [2] or GRASP⁴ [9],

⁴ Greedy Randomized Adaptive Search Procedure.

or even a subset of them (arc insertion), as ant colony optimization [8]. These algorithms can be used together with the restrictions with almost no additional modification.

Another question to be considered is the initialization of the search process. In general, the learning algorithms start from one or several initial DAGs that, in our case, must be consistent with the restrictions. A very common starting point is the empty DAG $G_\emptyset$. In our case $G_\emptyset$ should be replaced by the graph G_e or, even better, by the graph G_{re} . However, as G_{re} is not necessarily a DAG, it must be transformed into a DAG. An easy way to do it is to iteratively select an edge $x-y \in E_{re}$, randomly choose an orientation and test whether the restrictions are still self-consistent (choosing the opposite orientation if the test is negative). This process is based on the following result:

Proposition 5. *Let $G_e = (\mathcal{V}, E_e)$, $G_a = (\mathcal{V}, E_a)$ and $G_o = (\mathcal{V}, E_o)$ be graphs representing self-consistent existence, absence and ordering restrictions, respectively, and let $G_{re} = (\mathcal{V}, E_{re})$ be the refined graph of existence restrictions. Let $x-y \in E_{re}$ and define the graph $G_{e(x \rightarrow y)} = (\mathcal{V}, (E_e \setminus \{x-y\}) \cup \{x \rightarrow y\})$. Then $G_{e(x \rightarrow y)}$, G_a and G_o are still self-consistent if and only if there is not a directed path from y to x in $G_{re} \cup G_o$. Moreover, either $G_{e(x \rightarrow y)}$ or $G_{e(y \rightarrow x)}$, together with G_a and G_o , are self-consistent.*

In other cases the search algorithm is initialized with one (or several) random DAGs. The process of selecting a random DAG, checking the restrictions and iterating until the generated DAG satisfies the restrictions may be time-consuming, specially when there are many restrictions. In these cases it would be quite useful to have a repair operator, i.e. a method to transform any DAG into one verifying the restrictions. This method can also be useful for learning algorithms using population-based search processes (as genetic algorithms and EDAs).

6 Experimental Results

In this section we shall describe the experiments carried out to test the effect of using restrictions on BN learning algorithms, and the obtained results. We have selected four different problems. The Alarm network (left hand side of Figure 1) displays the relevant variables and relationships for the Alarm Monitoring System [3], a diagnostic application for patient monitoring. This network contains 37 variables and 46 arcs. Insurance [4] is a network for evaluating car insurance risks. The Insurance network (Figure 2) contains 27 variables and 52 arcs. Hailfinder [1] is a normative system that forecasts severe summer hail in northeastern Colorado. The Hailfinder network contains 56 variables and 66 arcs. Asia (right hand side of Figure 1) is a small Bayesian network that calculates the probability of a patient having tuberculosis, lung cancer or bronchitis respectively based on different factors. All these networks have been widely used in specialist literature for comparative purposes.

For Alarm, the input data commonly used are subsets of a standard database containing 20000 cases. In our experiments, we have used a subset containing

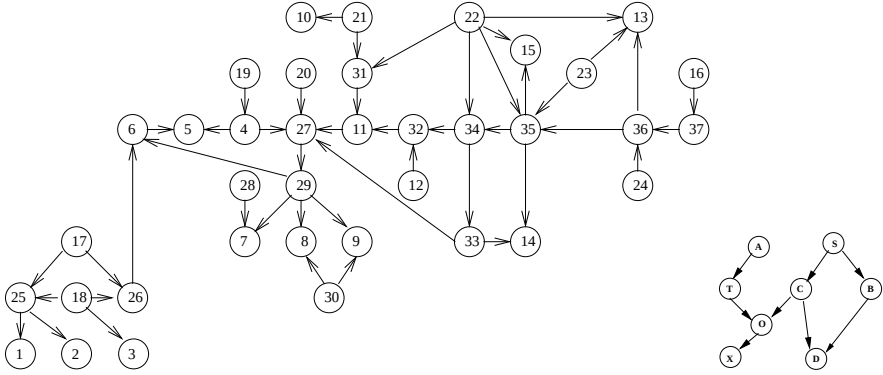


Fig. 1. The Alarm (left) and the Asia (right) networks

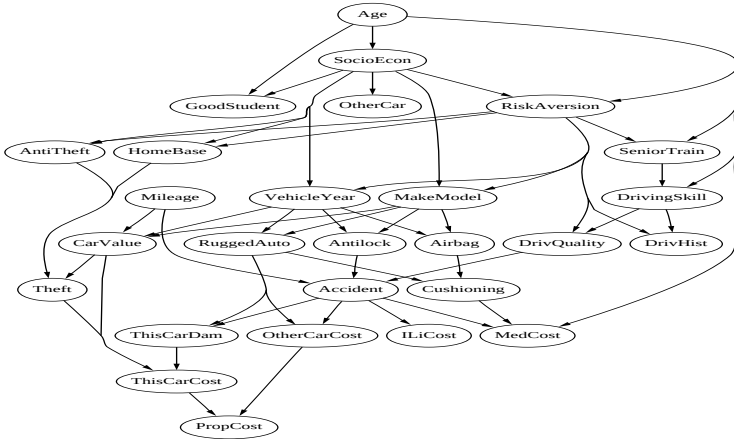


Fig. 2. The Insurance network

the first 10000 cases. In each of the other three problems, a database containing 10000 cases generated from the corresponding network has been used.

The score+search learning algorithm considered is the previously mentioned classical local search (with addition, removal and reversal of arcs), using the BDeu scoring function [13], with the parameter representing the equivalent sample size set to 1 and a uniform structure prior. The collected performance measures are the scoring value of the obtained network (BDeu) and three measures of the structural difference between the learned network and the true one: the number of added arcs (A), the number of deleted arcs (D) and the number of inverted arcs (I) in the learned network with respect to the true network. To eliminate fictitious differences or similarities between the two networks, caused by different but equivalent subDAG structures, before comparing the two networks we have converted them to their corresponding completed PDAG (also

called essential graph) representation⁵, using the algorithm proposed in [6]. The percentages of running time of the algorithm using restrictions (T) with respect to the running time of the algorithm without using them have also been computed. All the implementations have been carried out within the Elvira System [12], a Java tool to construct probabilistic decision support systems, which works with Bayesian networks and influence diagrams.

For each dataset we have randomly selected fixed percentages of restrictions of each type, extracted from the whole set of restrictions corresponding to the true network. More precisely, if $G = (\mathcal{V}, E)$ is the true network, then each arc $x \rightarrow y \in E$ is a possible existence restriction (we may select the restriction $x \rightarrow y \in E_e$ if this arc is also present in the completed PDAG representation of G ; otherwise we would use the restriction $x \neg y \in E_e$); each arc $x \rightarrow y \notin E$ is a possible absence restriction (in case that also $y \rightarrow x \notin E$ we randomly select whether to use the restriction $x \rightarrow y \in E_a$ or $x \neg y \in E_a$); finally, if there is a directed path from x to y in completed PDAG representation of G then $x \rightarrow y \in E_o$ is a possible ordering restriction. The selected percentages have been 10%, 20%, 30% and 40%. We have run the learning algorithm for each percentage of restrictions of each type alone, and also using the three types of restrictions together.

The results in Tables 1–4 represent the average values of the performance measures across 50 iterations (i.e. 50 random subsets of restrictions for each percentage and each dataset). For comparative purposes, these tables display also the results obtained by the learning algorithm without using restrictions (0%), its running time and the scoring value of the true network.

First, let us analyze the results from the perspective of the structural differences. What it was expected is that the number of deleted arcs, added arcs and inverted arcs decreases as the number of existence, absence and ordering restrictions, respectively, increases. This behaviour is indeed observed in the results. Moreover, another less obvious effect, almost systematically observed in the experiments (except in Asia), is that the use of any of the three types of restrictions also tends to decrease the other measures of structural difference. For example, the existence restrictions decrease the number of deleted arcs, but also the number of added and inverted arcs.

With respect to the analysis of the results from the perspective of the scoring function, we have to distinguish Hailfinder from the other three datasets, the reason being that in the first case the learning algorithm, without using restrictions, finds a network with a score much better than the true Hailfinder network. The true Insurance network is also worse in score than the learned one but at a much lesser extend, whereas the true Asia and Alarm networks are better than the learned ones. This is important because the use of restrictions tries to guide the search process towards the true network. On the one hand, in the last three cases, both the existence and the ordering restrictions lead to better network

⁵ A completed PDAG is a partially directed acyclic graph which is a canonical representation of all the DAGs belonging to the same equivalence class of DAGs.

Table 1. Results obtained for Asia

%	G_e, G_a, G_o					only G_e					only G_a					only G_o				
	BDeu	A	D	I	T	BDeu	A	D	I	T	BDeu	A	D	I	T	BDeu	A	D	I	T
10%	-2258.48	1.8	0.9	1.3	76	-2257.42	1.8	0.8	2.4	76	-2260.59	1.8	1.0	2.2	76	-2257.65	1.9	1.0	2.4	77
20%	-2256.95	1.5	0.8	0.2	56	-2256.94	1.8	0.8	1.6	57	-2260.34	1.6	1.1	1.1	56	-2257.37	1.9	1.1	1.9	58
30%	-2256.71	1.1	0.5	0.0	43	-2256.69	1.9	0.5	0.8	44	-2258.96	1.4	1.1	0.7	43	-2256.76	2.0	1.1	1.0	42
40%	-2256.87	0.7	0.5	0.0	28	-2256.61	1.9	0.5	0.4	29	-2260.00	0.9	1.0	0.4	28	-2256.59	2.0	1.1	0.8	30
0%	-2257.90	2	1	3		running time: 0.51 sec.					BDeu true network: -2257.55									

Table 2. Results obtained for Alarm

%	G_e, G_a, G_o					only G_e					only G_a					only G_o				
	BDeu	A	D	I	T	BDeu	A	D	I	T	BDeu	A	D	I	T	BDeu	A	D	I	T
10%	-108551	1.6	1.2	1.4	77	-108666	3.0	1.5	2.0	78	-108773	4.1	1.7	2.4	77	-108758	4.1	1.7	3.1	78
20%	-108550	0.9	1.0	1.0	60	-108613	2.4	1.4	2.3	61	-108806	3.4	1.7	2.5	60	-108739	3.7	1.3	2.5	61
30%	-108513	0.3	0.8	0.4	46	-108562	1.8	1.0	1.9	47	-108788	2.6	1.5	2.3	46	-108718	3.3	1.1	2.1	47
40%	-108504	0.2	0.7	0.3	33	-108486	0.9	0.8	2.0	35	-108782	1.8	1.3	1.8	34	-108675	2.8	1.1	1.9	35
0%	-108828	5	2	3		running time: 2.53 min.					BDeu true network: -108452									

Table 3. Results obtained for Insurance

%	G_e, G_a, G_o					only G_e					only G_a					only G_o				
	BDeu	A	D	I	T	BDeu	A	D	I	T	BDeu	A	D	I	T	BDeu	A	D	I	T
10%	-1323693	18.79	0.79			-1324064	28.810	0.779			-1324104	69.710	0.679			-1324175	3.10	0.10	0.579	
20%	-1322711	47.46	4.58			-1324293	48.0	8.5	59		-1324343	0.9	3	8.5	59	-1323624	6	9.8	9.4	59
30%	-1322250	56.04	4.42			-1323131	86.4	6	8	43	-1325092	59.3	8.4	43		-1323093	8	9.8	8.4	44
40%	-1322330	25.13	8.31			-1324091	65.7	5.2	32		-1323081	48.9	6.4	31		-1322573	3	9.5	7.5	33
0%	-132488	6	10	11		running time: 1.60 min.					BDeu true network: -132512									

structures. For Hailfinder, the convergence towards the true network results in worse networks. On the other hand, the use of absence restrictions seems to be self-defeating: the obtained networks frequently are worse in score than the one obtained without using restrictions. We believe that the explanation of this behaviour lies in the following fact: when a local search-based learning algorithm mistakes the direction of some arc connecting two nodes⁶, then the algorithm tends to ‘cross’ the parents of these nodes to compensate the wrong orientation; if some of these ‘crossed’ arcs are used as absence restrictions, then the algorithm cannot compensate the mistake and has to stop in a worse configuration. These results suggest that perhaps it is not a good idea to limit the search space using absence restrictions. Instead, once the algorithm, using only existence and ordering restrictions, has found a local maximum, we could delete all the forbidden arcs and run another local search.

Finally, with respect to the efficiency of the learning algorithm, it can be observed that the running times decrease considerably when using the restrictions, these times being progressively lesser as the number of restrictions increases.

In order to test the behavior of the restrictions in more realistic situations, where the number of available cases is much smaller (and therefore the expert knowledge that the restrictions represent is less probable to be already embedded in the data), we have also carried out experiments (20 iterations) with data sets containing only 500 cases. The results are displayed in Table 5. We can observe,

⁶ This situation may be quite frequent at early stages of the search process.

Table 4. Results obtained for Hailfinder

%	G_e, G_a, G_o					only G_e					only G_a					only G_o				
	BDeu	A	D	I	T	BDeu	A	D	I	T	BDeu	A	D	I	T	BDeu	A	D	I	T
10%	-498306	13.110	0.11	680		-497977	14.810	0.415	280		-498146	16.111	0.517	380		-498245	17.412	0.715	881	
20%	-498354	10.1 8.4	4.7 64			-498098	13.2 9.1	8.7 64			-498475	14.211	0.214	364		-498444	17.112	0.712	565	
30%	-498424	7.0 6.6	3.2 51			-498219	12.1 8.0	5.5 52			-498535	11.210	0.412	451		-498726	16.812	0.95	53	
40%	-498550	5.0 5.4	1.0 41			-498347	11.0 6.9	3.7 42			-498516	8.5 9.3	8.9 41			-498722	15.912	0.5	7.5	43
0%	-497904	17	12	19		running time: 6.53 min.					BDeu true network: -503095									

Table 5. Results obtained using data sets with only 500 cases

	G_e, G_a, G_o					only G_e					only G_a					only G_o				
%	BDeu	A	D	I		BDeu	A	D	I		BDeu	A	D	I		BDeu	A	D	I	
Asia																				
10%	-1075.94	0.0	0.8	0.6		-1075.94	0.0	0.8	0.6		-1075.36	0.0	1.0	0.0		-1075.36	0.0	1.0	0.0	
20%	-1075.74	0.0	0.6	0.3		-1075.87	0.0	0.6	0.4		-1075.36	0.0	1.0	0.0		-1075.36	0.0	1.0	0.0	
30%	-1075.75	0.0	0.6	0.3		-1075.75	0.0	0.6	0.3		-1075.36	0.0	1.0	0.0		-1075.36	0.0	1.0	0.0	
40%	-1075.51	0.0	0.6	0.0		-1075.51	0.0	0.6	0.0		-1075.36	0.0	1.0	0.0		-1075.36	0.0	1.0	0.0	
0%	-1075.36	0	1	0		BDeu true network: -1075.69														
Alarm																				
10%	-5990	9.7	4.0	12.7		-5972	10.4	3.8	15.6		-5998	11.0	5.0	19.3		-6002	13.2	4.7	14.6	
20%	-5971	6.2	3.3	7.6		-5959	9.1	3.2	12.1		-5990	9.2	4.6	14.2		-6005	12.4	5.0	12.2	
30%	-5946	4.4	2.2	5.8		-5949	7.7	2.6	10.2		-5984	7.3	4.6	10.8		-6003	11.3	4.9	10.2	
40%	-5943	3.2	1.6	4.4		-5946	7.1	2.0	8.4		-5989	6.5	4.6	8.8		-5986	9.8	4.6	8.5	
0%	-5986	11	5	22		BDeu true network: -5935														
Insurance																				
10%	-7262	4.8	17.6	8.8		-7270	5.6	18.2	9.0		-7274	7.2	20.6	7.1		-7270	8.0	20.9	7.8	
20%	-7280	3.4	15.2	7.8		-7286	4.6	16.2	8.7		-7270	5.9	19.6	7.4		-7270	7.9	20.7	7.4	
30%	-7310	2.0	12.2	6.0		-7325	3.6	13.4	8.0		-7264	4.6	18.7	6.2		-7268	7.8	20.6	7.2	
40%	-7353	1.4	9.9	6.2		-7358	3.1	11.2	6.8		-7273	3.9	18.0	5.8		-7265	7.8	20.4	6.9	
0%	-7270	8	21	8		BDeu true network: -7592														
Hailfinder																				
10%	-27270	13.322	211.1			-27231	15.422	513.0			-27202	14.723	811.4			-27183	16.825	211.3		
20%	-27408	11.019	9.9	9.2		-27330	14.219	912.1			-27244	12.523	310.4			-27190	16.425	310.7		
30%	-27582	8.7	17.2	8.0		-27461	13.317	412.2			-27280	10.322	2 9.8			-27198	16.325	610.2		
40%	-27781	7.0	14.2	7.6		-27649	12.414	611.4			-27309	8.0	21.2	9.2		-27204	16.226	0 9.6		
0%	-27171	17	25	12		BDeu true network: -29347														

in general, the same behavior as in the previous experiments, although in this case it must be taken into account that in all the databases (except Alarm) the true networks have a score worse than the learned ones without using restrictions.

7 Concluding Remarks

We have formally defined three types of structural restrictions for Bayesian networks, namely existence, absence and ordering restrictions, and studied their use in combination with BN learning algorithms that use scoring functions and search methods. We have illustrated it for the specific case of a learning algorithm using local search. The experimental results show that the use of additional knowledge in form of restrictions may lead to improved network structures in less time. For future work we plan to study the use of restrictions within score+search-based learning algorithms that do not search directly in the DAG space [2, 11] or within algorithms based on independence tests [16]. Finally, we would like to study another type of restriction, namely conditional independence relationships between variables that must be true.

Acknowledgments. This work has been supported by the Spanish Ministerio de Ciencia y Tecnología and Junta de Comunidades de Castilla-La Mancha, under Projects TIC2001-2973-CO5-01 and PBC-02-002, respectively.

References

1. Abramson, B., Brown, J., Murphy, A., & Winkler, R. L. (1996). Hailfinder: A Bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12, 57–71.
2. Acid, S., & de Campos, L.M. (2003). Searching for Bayesian network structures in the space of restricted acyclic partially directed graphs. *Journal of Artificial Intelligence Research*, 18, 445–490.
3. Beinlich, I. A., Suermondt, H. J., Chavez, R. M., & Cooper, G. F. (1989). The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the European Conference on Artificial Intelligence in Medicine*, 247–256.
4. Binder, J., Koller, D., Russell, S., & Kanazawa, K. (1997). Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29, 213–244.
5. Cheng, J., & Greiner, R. (1999). Comparing Bayesian network classifiers. In *Proceedings of the Fifteenth UAI Conference*, 101–108.
6. Chickering, D.M. (1995). A transformational characterization of equivalent Bayesian network structures. In *Proceedings of the Eleventh UAI Conference*, 87–98.
7. Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9, 309–348.
8. de Campos, L. M., Fernández-Luna, J. M., Gámez, J. A., & Puerta, J. M. (2002). Ant colony optimization for learning Bayesian networks. *International Journal of Approximate Reasoning*, 31, 291–311.
9. de Campos, L. M., Fernández-Luna, J. M., & Puerta, J. M. (2002). Local search methods for learning Bayesian networks using a modified neighborhood in the space of dags. *Lecture Notes in Computer Science*, 2527, 182–192.
10. de Campos, L. M., & Puerta, J. M. (2001). Stochastic local and distributed search algorithms for learning belief networks. In *Proceedings of the III International Symposium on Adaptive Systems: Evolutionary Computation and Probabilistic Graphical Model*, 109–115.
11. de Campos, L. M., & Puerta, J. M. (2001). Stochastic local search algorithms for learning belief networks: Searching in the space of orderings. *Lecture Notes in Artificial Intelligence*, 2143, 228–239.
12. Elvira Consortium. (2002). Elvira: an environment for probabilistic graphical models. In J.A. Gámez, A. Salmerón (Eds.), *Proceedings of the 1st European Workshop on Probabilistic Graphical Models*, 222–230.
13. Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20, 197–243.
14. Langley, P., & Sage, S. (1994). Induction of selective Bayesian classifiers. In *Proceedings of the Tenth UAI Conference*, 399–406.
15. Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo: Morgan Kaufmann.
16. Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, Prediction and Search*. Lecture Notes in Statistics 81, New York: Springer Verlag.

Foundation for the New Algorithm Learning Pseudo-Independent Models

Jae-Hyuck Lee

Department of Computing and Information Science,
University of Guelph, Guelph, Ontario, Canada N1G 2W1

Abstract. A type of problem domains known as *pseudo-independent (PI)* models poses difficulty for common learning methods, which are based on the single-link lookahead search. To learn this type of domain models, a method called the multiple-link lookahead search is needed. An improved result can be obtained by incorporating model complexity into a scoring metric to explicitly trade off model accuracy for complexity and vice versa during selection of the best model among candidates at each learning step. Previous studies found the complexity formulae for full PI models (the simplest type of PI models) and for atomic PI models (PI models without submodels). This study presents the complexity formula for non-atomic PI models, which are more complex than full or atomic PI models, yet more general. Together with the previous results, this study completes the major theoretical work for the new learning algorithm that combines complexity and accuracy.

1 Introduction

Learning probabilistic networks [5, 2, 4, 3] has been an active area of research recently. The task of learning networks is NP-hard [1]. Therefore, learning algorithms use a heuristic search, and the common search method is the single-link lookahead which generates network structures that differ by a single link at each level of the search.

Pseudo-independent (PI) models [12] are a class of probabilistic domain models where a group of marginally independent variables shows collective dependency. PI models cannot be learned by the single-link lookahead search because the underlying collective dependency cannot be recovered. Incorrectly learned models introduce silent errors when used for decision making. To learn PI models, the more sophisticated search method called *multi-link lookahead* [13] should be used. It was implemented in the learning algorithm called *RML* [8]. The algorithm is equipped with the Kullback-Leibler cross entropy as a scoring metric for the goodness-of-fit to data.

The scoring metric of the learning algorithm can be improved by incorporating model complexity to explicitly trade off model accuracy for complexity and vice versa [5, 9]. Therefore, obtaining the complexity of PI models becomes an issue.

Model complexity is defined by the number of parameters required to fully specify a model. A PI model can be *full* or *partial* as defined precisely in the next section. In previous work [7], a formula was presented for estimating the number of parameters in *full PI models*, the simpler type compared with partial PI models. However, the result

was very complex, and did not show the structural dependence relationships among parameters. The new concise formula for full PI models was later presented [11]. It was developed by employing a new perspective in viewing the dependence among parameters, called the *hypercube* method [10]. A PI model can have PI submodels embedded in the domain. Based on the existence of submodels, a PI model can be either *atomic*, which means the domain has no embedded PI submodels, or *non-atomic*, which means the domain has embedded PI submodels. Recently, the formula for complexity of atomic PI models was presented in [6].

In this paper, the complexity formula for non-atomic PI models is presented. It is shown that a non-atomic PI model can be decomposed into its submodels. Based on this result, the complexity formula for non-atomic PI models is derived from integrating the formulae [11] [6] previously found for full and atomic PI models. This study, together with previous studies [9] [11] [6], provides the theoretical work for the foundation of the new probabilistic learning algorithm that combines complexity and accuracy [9].

2 Background

Let V be a set of n discrete variables $X_1, \dots, X_n$ (in what follows, domains of finite and discrete variables will be focused on). Each variable X_i has a finite space $S_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,D_i}\}$ of cardinality D_i . The space of a set V of variables is defined by the Cartesian product of the spaces of all variables in V , that is, $S_V = S_1 \times \dots \times S_n$ (or $\prod_i S_i$). Thus, S_V contains the tuples made of all possible combinations of values of the variables in V . Each tuple is called a *configuration* of V , denoted by $(x_1, \dots, x_n)$.

Let $P(X_i)$ denote the probability function over X_i and $P(x_i)$ denote the probability value $P(X_i = x_i)$. The following axiom of probability is called the *total probability law*:

$$P(S_i) = 1, \text{ or } P(x_{i,1}) + P(x_{i,2}) + \dots + P(x_{i,D_i}) = 1. \quad (1)$$

For two subsets A and B of V such that $P(B) > 0$, a conditional probability function is defined as

$$P(A | B) = \frac{P(A, B)}{P(B)}. \quad (2)$$

A *probabilistic domain model (PDM)* $\mathcal{M}$ over V defines the probability values of every configuration for every subset $A \subset V$. $P(V)$ or $P(X_1, \dots, X_n)$ refers to the *joint probability distribution (JPD)* function over $X_1, \dots, X_n$, and $P(x_1, \dots, x_n)$ refers to the joint probability value of the configuration $(x_1, \dots, x_n)$. The probability function $P(A)$ over any proper subsets $A \subset V$ refers to the *marginal probability distribution (MPD)* function over A . If $A = \{X_1, \dots, X_m\}$ ($A \subset V$), then $P(x_1, \dots, x_m)$ refers to the marginal probability value.

A set of probability values that directly specifies a PDM is called *parameters* of the PDM. A joint probability value $P(x_1, \dots, x_n)$ is referred to as a *joint parameter* or *joint* and a marginal probability value $P(x_1, \dots, x_m)$ as a *marginal parameter* or *marginal*. Among parameters associated with a PDM, some parameters can be derived from others by using constraints such as the total probability law (Eq. (1)). Such derivable parameters are called *constrained* or *dependent* parameters while underivable parameters are

called *unconstrained*, *free*, or *independent* parameters. The number of independent parameters of a PDM is called the *model complexity* of the PDM, denoted as ω .

When no information of the constraints on a general PDM is given, the PDM should be specified only by joint parameters. The following ω_g gives the number of joint parameters required: Let $\mathcal{M}$ be a general PDM over $V = \{X_1, \dots, X_n\}$. Then the number of independent parameters of $\mathcal{M}$ is upper-bounded by

$$\omega_g = \left(\prod_{i=1}^n D_i \right) - 1. \quad (3)$$

One joint is dependent since it can be derived from others by the total probability law (Eq. (1)).

For any three disjoint subsets of variables A , B and C in V , A and B are called *conditionally independent* given C , denoted by $I(A, B \mid C)$, iff $P(A|B, C) = P(A|C)$ for all values in A , B and C such that $P(B, C) > 0$. Given subsets of variables $A, B, C, D \subseteq V$, the following property of conditional independence is called *Composition*:

$$I(A, B \mid C) \wedge I(A, D \mid C) \Rightarrow I(A, B \cup D \mid C). \quad (4)$$

Two disjoint subsets A and B are said to be *marginally independent*, denoted by $I(A, B \mid \emptyset)$, iff $P(A \mid B) = P(A)$ for all values A and B such that $P(B) > 0$. If two subsets of variables are marginally independent, no dependency exists between them. Hence, each subset can be modelled independently without losing information. If each variable X_i in a set A is marginally independent of the rest, the variables in A are said to be *marginally independent*. The probability distribution over a set of marginally independent variables can be written as the product of the marginal of each variable, that is, $P(A) = \prod_{X_i \in A} P(X_i)$.

Variables in a set A are called *generally dependent* if $P(B \mid A \setminus B) \neq P(B)$ for every proper subset $B \subset A$. If a subset of variables is generally dependent, a proper subsets cannot be modelled independently without losing information. Variables in A are *collectively dependent* if, for each proper subset $B \subset A$, there exists no proper subset $C \subset A \setminus B$ that satisfies $P(B \mid A \setminus B) = P(B \mid C)$. Collective dependence prevents conditional independence and modelling through proper subsets of variables.

A *pseudo-independent (PI)* model is a PDM where proper subsets of a set of collectively dependent variables display marginal independence [13].

Definition 1 (Full PI model). A PDM over a set V ($|V| \geq 3$) of variables is a *full PI model* if the following properties (called *axioms of full PI models*) hold:

(S_I) Variables in any proper subset of V are marginally independent.

(S_{II}) Variables in V are collectively dependent.

The complexity of full PI models is given as follows:

Theorem 2 (Complexity of full PI models [11]). Let a PDM $\mathcal{M}$ be a full PI model over $V = \{X_1, \dots, X_n\}$. Then the number of independent parameters of $\mathcal{M}$ is upper-bounded by

$$\omega_f = \prod_{i=1}^n (D_i - 1) + \sum_{i=1}^n (D_i - 1). \quad (5)$$

The axiom (S_I) of marginal independence is relaxed in partial PI models, as is defined through *marginally-independent partition*. The following concept of *marginally-independent subsets* is required to define *marginally-independent partition*.

Definition 3 (Marginally-independent subsets). Let V be a set of variables. Disjoint nonempty subsets $B_1 \dots B_m$ ($m \geq 2$) of V are *marginally independent subsets* if every pair $X \in B_i$ and $Y \in B_j$ for any two subsets B_i and B_j of B are marginally independent.

Definition 4 (Marginally independent partition). Let V be a set of variables and $B = \{B_1, \dots, B_m\}$ ($m \geq 2$) be a partition of V . B is a *marginally independent partition* if $B_1, \dots, B_m$ are marginally independent subsets. $B_i \in B$ is referred to as a *marginally-independent block* if the partition B is assumed.

A marginally independent partition of V groups variables in V into m marginally independent blocks. The property of marginally independent blocks is that if a subset A is formed by taking one element from different blocks, then variables in A are always marginally independent.

In a *partial PI model*, it is not necessary that every proper subset is marginally independent.

Definition 5 (Partial PI model). A PDM over a set V ($|V| \geq 3$) of variables is a *partial PI model* on V if the following properties (called *axioms of partial PI models*) holds:

(S'_I) V can be partitioned into two or more marginally independent blocks.

(S_{II}) Variables in V are collectively dependent.

The following definitions on *maximum marginally independent partition* is needed later for obtaining the complexity of partial PI models:

Definition 6 (Maximum partition and minimal blocks). Let $B = \{B_1, \dots, B_m\}$ be a marginally independent partition of a partial PI model over V . B is called a *maximum marginally independent partition* if there exists no marginally independent partition B' of V such that $|B| < |B'|$. The blocks of a *maximum marginally independent partition* are referred to as the *minimal marginally-independent blocks* or *minimal blocks*.

Partial PI models can contain (recursively-embedded) PI submodels and can be very complex. In [6], we analyzed the pure type of partial PI models as the basis for more complex models, called *atomic PI models*, which contain no embedded PI submodels.

Definition 7 (Atomic PI models). A PDM $\mathcal{M}$ over a set V ($|V| \geq 3$) of variables is an *atomic PI model* if $\mathcal{M}$ is either a full or partial PI model, and no collective dependency exists in any proper subsets of V .

The recent study [6] showed the complexity of atomic PI models is given as follows:

Theorem 8 (Complexity of atomic PI models [6]). Let a PDM $\mathcal{M}$ be an *atomic PI model* over $V = \{X_1, \dots, X_n\}$ ($n \geq 3$), where Composition (Statement (4)) holds in every proper subset. Let $D_1, \dots, D_n$ denote the cardinality of the space of each variable. Let a marginally-independent partition of V be denoted by $B = \{B_1, \dots, B_m\}$

($m \geq 2$), and the cardinality of the space of each block $B_1, \dots, B_m$ be denoted by $D_{(B_1)}, \dots, D_{(B_m)}$, respectively. Then, the number ω_{ap} of parameters required for specifying the JPD of $\mathcal{M}$ is upper-bounded by

$$\omega_{ap} = \prod_{i=1}^n (D_i - 1) + \sum_{j=1}^m (D_{(B_j)} - 1). \quad (6)$$

It is possible that a PI model exists as a subdomain over a subset of variables in a domain. This type of PI models is called *embedded PI submodels*.

Definition 9 (Embedded PI submodel). In a PDM $\mathcal{M}$ over a set V , a proper subset $C \subset V$ of variables is an *embedded PI model* over C if the following properties (called *axioms of embedded PI models*) hold:

(S_{III}) C forms a partial PI model.

(S'_I) The marginally-independent partition $B_{(C)} = \{B_{(C)1}, \dots, B_{(C)r}\}$ of C extends into V , that is, a set of marginally-independent blocks $B_1, \dots, B_r$ of V exist such that $B_{(C)i} \subseteq B_i$ ($i = 1, \dots, r$).

A PI submodel can contain one or more embedded PI submodels within itself. These are called *recursively embedded PI submodels*.

3 Complexity of Non-atomic PI Models

This section starts with defining terms related to non-atomic PI models.

Definition 10 (Non-atomic PI models). A PDM $\mathcal{M}$ over a set V ($|V| \geq 4$) of variables is a *non-atomic PI model* if $\mathcal{M}$ is either a full or partial PI model and contains at least one PI submodel M over a subset $A \subset V$, called *subdomain*.

To refer to the location of each submodel or each subdomain with respect to how deep it is embedded in, the term called *depth* of embedding is introduced:

Definition 11 (Depth of embedding). Given a PDM $\mathcal{M}$ over a set V of variables, the *depth* of $\mathcal{M}$ is defined as 0. Suppose a PI submodel M embedded in $\mathcal{M}$ has its *depth* d . Suppose another PI submodel $\tilde{M}$ is embedded in M and there exists no other PI submodel M^* such that M^* is embedded in M and $\tilde{M}$ is embedded in M^* . Then the *depth* of $\tilde{M}$ is defined as $d + 1$.

Non-atomic PI models consist of marginally-independent blocks (Definition 4) as atomic PI models do since both types of PI models are partial PI models. On the other hand, the former contain PI subdomains unlike the latter. Therefore, the relationship between PI subdomains and marginally-independent blocks in a non-atomic PI domain should be analyzed to utilize the result from the previous study on atomic PI models [6]. The following lemma reveals the relationship:

Lemma 12 (Relation between PI subdomains and minimal blocks). Let a partial PI model $\mathcal{M}$ over a set V and of depth d contain embedded PI submodels $M_1, \dots, M_p$ of depth $d + 1$ over subsets $A_1, \dots, A_p$, respectively. Any two of $A_1, \dots, A_p$ may be either

disjoint, or one may intersect another, but one cannot be included in another; In other words, they are incomparable. Let the maximum marginally-independent partition of V be denoted by $B^M = \{B_1^M, \dots, B_m^M\}$. For a given minimal block $B_j^M \in \{B_1^M, \dots, B_m^M\}$, either of the following, but not both, must hold in relation to PI subdomains $A_1, \dots, A_p$:

- **Relation 1.** For some $A_i \in \{A_1, \dots, A_p\}$, $B_j^M \subset A_i$;

- **Relation 2.** For every $A_i \in \{A_1, \dots, A_p\}$, $B_j^M \bowtie A_i$, where $\bowtie$ denotes “incomparable”; Hence, $B_j^M \bowtie A_i$ implies $(B_j^M \cap A_i = \emptyset) \vee ((B_j^M \cap A_i \neq \emptyset) \wedge (B_j^M \not\subseteq A_i) \wedge (B_j^M \not\supset A_i))$.

Such as B_j^M in Relation 2, a minimal block that is incomparable of any PI subdomains is referred to as an MIP-subset.

Lemma 12 implies that every minimal block in a domain of depth d is either a subset of at least one PI subdomains of depth $d + 1$ or an MIP-subset of the domain of depth d .

The following presents the units of complexity computation on non-atomic PI models. These units are referred to as *components* of the domain, which consist of PI subdomains and MIP-subsets of the same depth.

Definition 13 (Components of a non-atomic PI domain). Let a partial PI model $\mathcal{M}$ over a set $V = \{X_1, \dots, X_n\}$ ($n \geq 4$) of depth d contain embedded PI subdomains $A_1, \dots, A_p$ of depth $d + 1$ ($p \geq 1$) and MIP-subsets (if any exist) $B_1, \dots, B_q$ of V ($q \geq 1$). Both a PI subdomain $A_i \in \{A_1, \dots, A_p\}$ and a MIP-subset $B_j \in \{B_1, \dots, B_q\}$ are referred to as *components* of V , notated by $C_k \in \{C_1, \dots, C_r\}$ ($r = p + q$).

Figure 1 depicts a non-atomic PI model $\mathcal{M}$ over the domain $\{X_1, X_2, X_3, X_4, X_5, X_6\}$ of depth 0 that contains one non-atomic PI subdomain $C_1 = \{X_1, X_2, X_3, X_4, X_5\}$ and one MIP-subset $C_2 = \{X_6\}$ of depth 1. C_1 contains two atomic PI subdomains of depth 2 which are $\dot{C}_1 = \{X_1, X_2, X_3\}$ and $\dot{C}_2 = \{X_1, X_2, X_4\}$, and one MIP-subset $\dot{C}_3 = \{X_5\}$ of depth 2. The components of $\mathcal{M}$ are C_1 and C_2 ; and the components of C_1 are $\dot{C}_1$, $\dot{C}_2$, and $\dot{C}_3$.

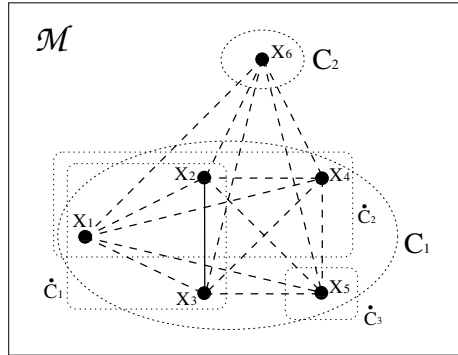


Fig. 1. A partial PI model with one PI subdomain C_1 that contains two recursively embedded PI subdomains $\dot{C}_1$ and $\dot{C}_2$

The following lemma states that eliminating one variable from a non-atomic PI domain removes collective dependency from the domain unless the new domain is formed itself into a single PI domain.

Lemma 14 (Removing one variable from a non-atomic PI domain). *Let a partial PI model $\mathcal{M}$ over a set $V = \{X_1, \dots, X_n\}$ ($n \geq 4$) of depth d contain embedded PI subdomains $A_1, \dots, A_p$ of depth $d + 1$ and MIP-subsets (if any exist) $B_1, \dots, B_q$. Let $C_1, \dots, C_r$ denote the components of V such that $\{C_1, \dots, C_r\} = \{A_1, \dots, A_p, B_1, \dots, B_q\}$ ($r = p + q$ and $\bigcup_{i=1}^r C_i = V$). For an arbitrary variable $X_\alpha \in V$, let V' denote $V \setminus \{X_\alpha\}$. Likewise, let C'_i denote $C_i \setminus \{X_\alpha\}$; that is, C'_i is the same as C_i except that if C_i contains X_α , $C'_i = C_i \setminus \{X_\alpha\}$. Then no collective dependency among V' exists unless V' is formed itself into a single PI domain.*

Lemma 15 says that the conditional independence relation holds among components of the same depth in a domain with at least one variable removed. Due to this lemma, a domain can be decomposed into the product of its conditional factors or components, as will be shown later in Lemma 16.

Lemma 15 (Conditional independence among multiple subdomains). *Let a partial PI model $\mathcal{M}$ of depth d over a set $V = \{X_1, \dots, X_n\}$ ($n \geq 4$) contain embedded PI subdomains $A_1, \dots, A_p$ of depth $d + 1$ and MIP-subsets (if any exist) $B_1, \dots, B_q$. Let $C_1, \dots, C_r$ denote the components of V such that $\{C_1, \dots, C_r\} = \{A_1, \dots, A_p, B_1, \dots, B_q\}$ ($r = p + q$ and $\bigcup_{i=1}^r C_i = V$). For an arbitrary variable $X_\alpha \in V$, let V' denote $V \setminus \{X_\alpha\}$. Likewise, let C'_i denote $C_i \setminus \{X_\alpha\}$; that is, C'_i is the same as $C_i \setminus \{X_\alpha\}$ except that if C_i contains X_α , $C'_i = C_i \setminus \{X_\alpha\}$. Then*

$$I\left(C'_i \setminus (C'_i \cap (\bigcup_{j=1, j \neq i}^r C'_j)), V' \setminus C'_i \mid C'_i \cap (\bigcup_{j=1, j \neq i}^r C'_j)\right). \quad (7)$$

Proof. In case that V consists of a PI subdomain C_1 and an MIP-subset $C_2 = \{X_\alpha\}$, Statement (7) becomes $I(C_1, \emptyset \mid \emptyset)$, which is trivial. Therefore, hereinafter this case will be excluded, and $V' \setminus C'_i$ is assumed to be non-empty.

The following is proof by contradiction: Suppose Statement (7) is not true. Then, since $\neg I(X, Y \mid Z) \iff D(X, Y \mid Z)$, the following must be true:

$$D\left(C'_i \setminus (C'_i \cap (\bigcup_{j=1, j \neq i}^r C'_j)), V' \setminus C'_i \mid C'_i \cap (\bigcup_{j=1, j \neq i}^r C'_j)\right). \quad (8)$$

In order for Statement (8) to be true, either of the following must be true:

- Case 1. There exists marginal dependence between at least a pair of variables $Y \in C'_i \setminus (C'_i \cap (\bigcup_{j=1, j \neq i}^r C'_j))$ and $Z \in V' \setminus C'_i$;

- Case 2. There exists collective dependence among variables in a subset E ($E \subset V'$) that consists of a subset $Q \subseteq C'_i \setminus (C'_i \cap (\bigcup_{j=1, j \neq i}^r C'_j))$ and a subset $R \subseteq V' \setminus C'_i$.

However, the following proves neither of the above is true: Consider Case 1. Since Y and Z are marginally dependent, they must be included in a minimal block. Let C^* denote the component that contains this minimal block. Since $Z \in V' \setminus C'_i$, C^* must belong to $\bigcup_{j=1, j \neq i}^r C'_j$. In addition, C^* must share Y with $C'_i \setminus (C'_i \cap (\bigcup_{j=1, j \neq i}^r C'_j))$. However, this is impossible because $C'_i \setminus (C'_i \cap (\bigcup_{j=1, j \neq i}^r C'_j))$ cannot share any variables that belong to $\bigcup_{j=1, j \neq i}^r C'_j$.

Consider Case 2. This collective dependence of E implies that E is a PI subdomain, which is to be denoted by C^* . Then C^* must be of depth greater than d that is the depth of V since $C^* \subseteq V'$ and V' has no collective dependency. In addition, the depth of C^* must be less than $d + 2$ because C^* cannot be contained in any PI subdomains of depth $d + 1$ due to the two subsets Q and R of C^* that belong to two distinct components of depth $d + 1$. Therefore, the depth of C^* must be $d + 1$. Since $R \subseteq V' \setminus C'_i$, C^* must be one of C'_j ($j = 1, \dots, r$ and $j \neq i$). In addition, C^* shares Q with $C'_i \setminus (C'_i \cap (\bigcup_{j=1, j \neq i}^r C'_j))$. However, this is impossible because any members of C'_j ($j = 1, \dots, r$ and $j \neq i$) cannot share a subset with $C'_i \setminus (C'_i \cap (\bigcup_{j=1, j \neq i}^r C'_j))$.

As shown above, neither case can be true, and thus Statement 7 must be true. $\square$

The following lemma states that the joint probability of a non-atomic PI domain can be decomposed into the marginal probability of each component by using Lemma 15. Due to this lemma, the total number of independent marginal parameters needed for using a hypercube method can be obtained from the sum of the marginals of each component.

Lemma 16 (JPD as the product of its conditional factors). *Let a partial PI model $\mathcal{M}$ over a set $V = \{X_1, \dots, X_n\}$ ($n \geq 4$) of depth d contain embedded PI subdomains $A_1, \dots, A_p$ of depth $d + 1$ and MIP-subsets (if any exist) $B_1, \dots, B_q$. Let $C_1, \dots, C_r$ denote the components of V such that $\{C_1, \dots, C_r\} = \{A_1, \dots, A_p, B_1, \dots, B_q\}$ ($r = p + q$ and $\bigcup_{i=1}^r C_i = V$). For an arbitrary variable $X_\alpha \in V$, let V' denote $V \setminus \{X_\alpha\}$. Likewise, let C'_i denote $C_i \setminus \{X_\alpha\}$; that is, C'_i is the same as C_i except that if C_i contains X_α , $C'_i = C_i \setminus \{X_\alpha\}$. Then*

$$P(V') = \left(\prod_{i=1}^{r-1} P(C'_i \setminus (C'_i \cap (\bigcup_{j=i+1}^r C'_j)) \mid C'_i \cap (\bigcup_{j=i+1}^r C'_j)) \right) \cdot P(C'_r). \quad (9)$$

Proof. This proof shows how $P(V')$ can be decomposed into the product of its conditional factors, resulting in Eq. (9).

First, decompose $P(C'_1, \dots, C'_r)$: Applying Eq. (15) to $P(C'_1, \dots, C'_r)$ for $C'_i = C'_1$ gives $I(C'_1 \setminus (C'_1 \cap \{C'_2, \dots, C'_r\}), \{C'_2, \dots, C'_r\} \mid C'_1 \cap \{C'_2, \dots, C'_r\})$. Since $P(X, Y, Z) = P(X \mid Z)P(Y, Z)$,

$$\begin{aligned} P(C'_1, \dots, C'_r) &= P(C'_1 \setminus (C'_1 \cap \{C'_2, \dots, C'_r\}) \mid C'_1 \cap \{C'_2, \dots, C'_r\}) \\ &\quad P(\{C'_2, \dots, C'_r\}, C'_1 \cap \{C'_2, \dots, C'_r\}) \\ &= P(C'_1 \setminus (C'_1 \cap \{C'_2, \dots, C'_r\}) \mid C'_1 \cap \{C'_2, \dots, C'_r\}) P(C'_2, \dots, C'_r). \end{aligned} \quad (10)$$

Next, decompose $P(C'_2, \dots, C'_r)$: Applying Eq. (15) to $P(C'_2, \dots, C'_r)$ for $C'_i = C'_2$ gives $I(C'_2 \setminus (C'_2 \cap \{C'_3, \dots, C'_r\}), \{C'_3, \dots, C'_r\} \mid C'_2 \cap \{C'_3, \dots, C'_r\})$. By the same process as the previous step,

$$P(C'_2, \dots, C'_r) = P(C'_2 \setminus (C'_2 \cap \{C'_3, \dots, C'_r\}) \mid C'_2 \cap \{C'_3, \dots, C'_r\}) P(C'_3, \dots, C'_r). \quad (11)$$

Substitute this result into Eq. (10). Then

$$\begin{aligned} P(C'_1, \dots, C'_r) &= P(C'_1 \setminus (C'_1 \cap \{C'_2, \dots, C'_r\}) \mid C'_1 \cap \{C'_2, \dots, C'_r\}) \\ &\quad P(C'_2 \setminus (C'_2 \cap \{C'_3, \dots, C'_r\}) \mid C'_2 \cap \{C'_3, \dots, C'_r\}) P(C'_3, \dots, C'_r). \end{aligned} \quad (12)$$

Similarly, repeat this process of decomposition from $i = 3$ to $i = r - 1$ and back-substitute recursively. Then the result will be

$$P(C'_1, \dots, C'_r) = P(C'_1 \setminus (C'_1 \cap \{C'_2, \dots, C'_r\}) \mid C'_1 \cap \{C'_2, \dots, C'_r\}) \cdots P(C'_{r-1} \setminus (C'_{r-1} \cap C'_r) \mid C'_{r-1} \cap C'_r) P(C'_r). \quad (13)$$

□

Corollary 17, which follows from Lemma 16, gives the constraint relationship among the joint parameters of a non-atomic PI domain. By this constraint (Eq. (15)), a set of joint parameters can be derived from other joints plus necessary marginals.

Corollary 17 (Joint-marginal equality and joint constraint). *Let a partial PI model $\mathcal{M}$ over a set $V = \{X_1, \dots, X_n\}$ ($n \geq 4$) of depth d contain embedded PI subdomains $A_1, \dots, A_p$ of depth $d + 1$ and MIP-subsets (if any exist) $B_1, \dots, B_q$. Let $C_1, \dots, C_r$ denote the components of V such that $\{C_1, \dots, C_r\} = \{A_1, \dots, A_p, B_1, \dots, B_q\}$ ($r = p + q$ and $\bigcup_{i=1}^r C_i = V$). For an arbitrary variable $X_\alpha \in V$, let V' denote $V \setminus \{X_\alpha\}$. Likewise, let C'_i denote $C_i \setminus \{X_\alpha\}$; that is, C'_i is the same as C_i except that if C_i contains X_α , $C'_i = C_i \setminus \{X_\alpha\}$. Then the following, called joint-marginal equality, holds:*

$$\sum_{k=1}^{D_\alpha} P(X_1, \dots, x_{\alpha,k}, \dots, X_n) = \left(\prod_{i=1}^{r-1} \frac{P(C'_i)}{P(C'_i \cap (\bigcup_{j=i+1}^r C'_j))} \right) \cdot P(C'_r). \quad (14)$$

Eq. (14) implies the following constraint upon the joint parameters of V : For a t -th value of X_α ($1 \leq t \leq D_\alpha$), denoted by $x_{\alpha,t}$,

$$P(X_1, \dots, x_{\alpha,t}, \dots, X_n) = \left(\prod_{i=1}^{r-1} \frac{P(C'_i)}{P(C'_i \cap (\bigcup_{j=i+1}^r C'_j))} \right) \cdot P(C'_r) - \sum_{k=1, k \neq t}^{D_\alpha} P(X_1, \dots, x_{\alpha,k}, \dots, X_n). \quad (15)$$

Proof. Eq. (15) directly follows from Eq. (14), and therefore only Eq. (14) needs to be proved. The summation $\sum_{k=1}^{D_\alpha} P(X_1, \dots, x_{\alpha,k}, \dots, X_n)$ on the left-hand side represents marginalization on X_α . The result from this is $P(X_1, \dots, X_{\alpha-1}, X_{\alpha+1}, \dots, X_n)$ or $P(V \setminus \{X_\alpha\})$, which is equivalent to $P(V')$ and has no collective dependency on V' . (Same as the proof for Lemma 15, the trivial case that gives $I(C_1, \emptyset \mid \emptyset)$ or $P(X_1, \dots, X_{\alpha-1}, X_{\alpha+1}, \dots, X_n) = P(C_1)$ is excluded.) Therefore, by Lemma 16,

$$P(X_1, \dots, X_{\alpha-1}, X_{\alpha+1}, \dots, X_n) = \left(\prod_{i=1}^{r-1} P(C'_i \setminus (C'_i \cap (\bigcup_{j=i+1}^r C'_j)) \mid C'_i \cap (\bigcup_{j=i+1}^r C'_j)) \right) \cdot P(C'_r). \quad (16)$$

By the definition of Conditional Probability (Eq. (2)), the right-hand side of Eq. (16) can be written in the form of Eq. (14). □

The constraint Eq. (15) expresses a set of joint parameters (on the left-hand side) can be derived from a set of marginal parameters (the first term on the right) and other joints (the second term). Then the total number of independent parameters required for specifying a domain is the number of marginal parameters (the first term) plus joint parameters (the second term) needed to be provided in order to use Eq. (15) for deriving

as many joints as possible. Since derived joints (on the left) also can be used as parameters at the second term on the right for deriving new joints, it is important to determine the number of independent joints (or the joints that are underivable) by a systematic method; For this purpose, a hypercube method is used.

The following is the general result on the total number of independent parameters of non-atomic PI models, obtained from the hypercube method.

Theorem 18 (Complexity of non-atomic PI models). *Let $\mathcal{M}$ be a partial PI model over a set $V = \{X_1, \dots, X_n\}$ ($n \geq 4$) of depth 0. Let the maximum marginally-independent partition of V be denoted by $B = \{B_1, \dots, B_m\}$ ($m \geq 3$) and the cardinality of the space of each minimal block $B_1, \dots, B_m$ be denoted by $D_{(B_1)}, \dots, D_{(B_m)}$, respectively. Let all embedded PI subdomains regardless of their depths be denoted by $\hat{A}_1, \dots, \hat{A}_t$. Let each PI subdomain $\hat{A}_\beta$ ($\beta = 1, \dots, t$) be defined by $\hat{A}_\beta = \{X_k \mid X_k \in \hat{A}_\beta, k = 1, \dots, n\}$ and the cardinality of each X_k be denoted by D_k . Then the number ω_{np} of parameters required for specifying the JPD of $\mathcal{M}$ is upper-bounded by*

$$\omega_{np} = \prod_{i=1}^n (D_i - 1) + \sum_{j=1}^m (D_{(B_j)} - 1) + \sum_{\beta=1}^t \left(\prod_{X_k \in \hat{A}_\beta} (D_k - 1) \right). \quad (17)$$

Proof. Before proving this theorem, the following is a brief explanation of the result. The first term on the right is the cardinality of the joint space of $\mathcal{M}$ over the set of variables except the space of each variable is reduced by one. This term represents the number of independent joint parameters of $\mathcal{M}$ and is the same for full or atomic PI models (Eq. (5) for full PI models and Eq. (6) for atomic PI models). The second term is the number of marginal parameters for specifying the joint space of each minimal marginally-independent block. The third term corresponds to the number of joint parameters required for specifying every PI subdomain regardless of its depth.

Eq. (17) is a form of recursive definition on every non-atomic PI (sub)domain from depth 0 to the greatest depth. The recursive definition consists of the number $\prod_{i=1}^n (D_i - 1)$ of joint parameters over the domain variables from $X_1, \dots, X_n$ in each non-atomic PI (sub)domain and the number of marginal parameters for every component in the (sub)domain. Therefore, to prove Theorem 17, what needs to be shown is that all joint parameters of $P(V)$ of depth 0 can be derived from $(\sum_{j=1}^m (D_{(B_j)} - 1) + \sum_{\beta=1}^t (\prod_{X_k \in \hat{A}_\beta} (D_k - 1)))$ marginals plus $\prod_{i=1}^n (D_i - 1)$ joints by using Eq. (15).

First, assume the joint probabilities $P(C_1), \dots, P(C_r)$ of components $C_1, \dots, C_r$ in V can be specified by $\sum_{j=1}^m (D_{(B_j)} - 1) + \sum_{\beta=1}^t (\prod_{X_k \in \hat{A}_\beta} (D_k - 1))$ marginals. (This assumption is to be proved later.) Then any $P(C'_1), \dots, P(C'_r)$ and $P(C'_i \cap (\bigcup_{j=i+1}^r C'_j))$ for $i = 1, \dots, r-1$ in Eq. (15) can be derived by marginalization from the corresponding $P(C_1), \dots, P(C_r)$.

Next, with this marginals and $\prod_{i=1}^n (D_i - 1)$ joints, show all joint parameters of $P(V)$ can be derived. For this purpose, a hypercube is constructed and Eq. (15) is applied to each group of relevant cells systematically. Once a cell (which corresponds to a joint parameter) is determined to be derivable from other cells plus the marginals, it is eliminated from further consideration. Due to the page limit, detailed description about eliminating all derivable cells from the hypercube is omitted. However, the procedure is very similar to that for full PI models [11] or atomic PI models [6]. By using Eq. (15),

hyperplanes at $X_1 = x_{1,D_1}, X_2 = x_{2,D_2}, \dots, X_n = x_{n,D_n}$ can be eliminated because for each X_i , all cells on the hyperplane at $X_i = x_{i,D_i}$ can be derived from cells outside the hyperplane and the marginal parameters already specified. The remaining cells form a reduced hypercube whose length along the X_i axis is $D_i - 1$ ($i = 1, 2, \dots, n$). Therefore, the total number of the remaining cells, which represent underivable cells, is $\prod_{i=1}^n (D_i - 1)$. Since this result is independent of the depth of the domain, this result on the number of independent joints on a PI (sub)domain can be applied to any non-atomic PI subdomains of any depths.

Finally, it is needed to show that all independent parameters of $P(C_1), \dots, P(C_r)$ can be specified by $\sum_{j=1}^m (D_{(B_j)} - 1) + \sum_{\beta=1}^t (\prod_{X_k \in \hat{A}_\beta} (D_k - 1))$ marginals. The following are the three types of components C_i and the corresponding number $\omega_{(C_i)}$ of independent parameters:

- (a) If C_i is an MIP-subset, then $\omega_{(C_i)} = (\prod D) - 1$ by Eq. (3);
- (b) If C_i is an atomic PI subdomain, then $\omega_{(C_i)} = \prod(D - 1) + \sum(D_{(B)} - 1)$ by Eq. (6);
- (c) If C_i is a non-atomic PI subdomain, then $\omega_{(C_i)} = \prod(D - 1) + \sum \omega_{(C)}$, where C is each component in C_i .

Both $(\prod D) - 1$ in (a) and $\sum(D_{(B)} - 1)$ in (b) are included in $\sum_{j=1}^m (D_{(B_j)} - 1)$ marginals since MIP-subsets are minimal blocks; and both $\prod(D - 1)$ in (b) and $\prod(D - 1)$ in (c) are included in $\sum_{\beta=1}^t (\prod_{X_k \in \hat{A}_\beta} (D_k - 1))$ marginals. Therefore, the two marginal parameter terms of Eq. (17) is sufficient for specifying all independent parameters of every component in V . $\square$

Note Theorem 18 also holds for atomic PI models since they are a special case of non-atomic PI models. This can be easily proved by removing the third term on the right from Eq. (17) because atomic PI models have no PI submodels, resulting in Eq. (6).

The following is an example that shows how to apply Theorem 18:

Example 19 (Applying Eq. (17)). Consider a non-atomic PI model $\mathcal{M}$ with one PI submodel that has two embedded PI submodels, as shown in Figure 1. The domain consists of 6 variables from X_1 to X_6 ; X_1, X_2, X_3 are binary; X_4 is ternary; X_5 is 5-nary; and X_6 is 6-nary.

The number of independent marginal parameters for specifying every minimal block is 15, given by $(2 - 1) + (2 \cdot 2 - 1) + (3 - 1) + (5 - 1) + (6 - 1)$. With the marginals given above, the number of independent joint parameters needed for specifying the PI subdomain C_1 of depth 1 and the PI subdomains $\dot{C}_1$ and $\dot{C}_2$ of depth 2 is 11, given by $[(2 - 1)(2 - 1)(2 - 1)(3 - 1)(5 - 1)] + [(2 - 1)(2 - 1)(2 - 1)] + [(2 - 1)(2 - 1)(3 - 1)]$. Therefore, the total number of independent marginal parameters for specifying C_1 of depth 1, and $\dot{C}_1$ and $\dot{C}_2$ of depth 2 is 26, given by $15 + 11$. The number of independent joint parameters of $\mathcal{M}$ is 40, given by $(2 - 1)(2 - 1)(2 - 1)(3 - 1)(5 - 1)(6 - 1)$. Therefore, the total number of parameters for specifying $\mathcal{M}$ in this example is $26 + 40 = 66$. Compare with this number with the number of parameters for specifying a general PDM over the same set of variables by using the total probability law (Eq. (1)), giving $2 \cdot 2 \cdot 2 \cdot 3 \cdot 5 \cdot 6 - 1 = 719$. This shows the complexity of a non-atomic PI model is significantly less than that of a general PDM. $\square$

4 Conclusion

This research presented the complexity formula (Eq. (17)) for non-atomic PI models. Together with the previous results about complexity of full PI models [11] and atomic PI models [6], and the scoring metric [9], this study concludes all major theoretical work for the new learning algorithm that combines complexity and accuracy.

Acknowledgments

The author is grateful to Yang Xiang and the anonymous reviewers for their comments on this work. This research is partially supported by NSERC of Canada.

References

1. D. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks: search methods and experimental results. In *Proceedings of 5th Conference on Artificial Intelligence and Statistics*, pages 112–128, Ft. Lauderdale, 1995. Society for AI and Statistics.
2. G.F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.
3. N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. In G.F. Cooper and S. Moral, editors, *Proceedings of 14th Conference on Uncertainty in Artificial Intelligence*, pages 139–147, Madison, Wisconsin, 1998. Morgan Kaufmann.
4. D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.
5. W. Lam and F. Bacchus. Learning Bayesian networks: an approach based on the MDL principle. *Computational Intelligence*, 10(3):269–293, 1994.
6. J. Lee and Y. Xiang. Model complexity of pseudo-independent models. In *Proceedings of 16th Florida Artificial Intelligence Research Society Conference*, 2005. Forthcoming.
7. Y. Xiang. Towards understanding of pseudo-independent domains. In *Poster Proceedings of 10th International Symposium on Methodologies for Intelligent Systems*, Charlotte, 1997.
8. Y. Xiang, J. Hu, N. Cercone, and H. Hamilton. Learning pseudo-independent models: analytical and experimental results. In H. Hamilton, editor, *Advances in Artificial Intelligence*, pages 227–239. Springer, 2000.
9. Y. Xiang and J. Lee. Local score computation in learning belief networks. In E. Stroulia and S. Matwin, editors, *Advances in Artificial Intelligence*, pages 152–161. Springer, 2001.
10. Y. Xiang, J. Lee, and N. Cercone. Parameterization of pseudo-independent models. In *Proceedings of 16th Florida Artificial Intelligence Research Society Conference*, pages 521–525, St. Augustine, 2003.
11. Y. Xiang, J. Lee, and N. Cercone. Towards better scoring metrics for pseudo-independent models. *International Journal of Intelligent Systems*, 20, 2004.
12. Y. Xiang, S.K.M. Wong, and N. Cercone. Critical remarks on single link search in learning belief networks. In *Proceedings of 12th Conference on Uncertainty in Artificial Intelligence*, pages 564–571, Portland, 1996.
13. Y. Xiang, S.K.M. Wong, and N. Cercone. A ‘microscopic’ study of minimum entropy search in learning decomposable Markov networks. *Machine Learning*, 26(1):65–92, 1997.

Optimal Threshold Policies for Operation of a Dedicated-Platform with Imperfect State Information - A POMDP Framework

Arsalan Farrokh and Vikram Krishnamurthy

University of British Columbia, Vancouver, BC, Canada
{arsalanf, vikramk}@ece.ubc.ca

Abstract. We consider the general problem of optimal stochastic control of a dedicated-platform that processes one primary function or task (target-task). The dedicated-platform has two modes of action at each period of time: it can attempt to process the target-task at the given period of time, or suspend the target-task for later completion. We formulate the optimal trade-off between the processing cost and the latency in completion of the target-task as a Partially Observable Markov Decision Process (POMDP). By reformulating this POMDP as a Markovian search problem, we prove that the optimal control policies are threshold in nature. Threshold policies are computationally efficient and inexpensive to implement in real time systems. Numerical results demonstrate the effectiveness of these threshold based operating algorithms as compared to non-optimal heuristic algorithms.

Keywords: Partially Observable Markov Decision Process (POMDP), optimal threshold policies, dynamic programming, Bellman equation, two-state Markov chain, optimal search, overlook probability, sufficient statistics, Hidden Markov Model (HMM).

1 Introduction

Many applications in manufacturing, personal telecommunications and defense involve a dedicated-platform that utilizes the system resources in order to process or execute one primary function or task (target-task). Conservation of the system resources and minimization of the latency in completing the target-task are important issues in efficient operation of this dedicated-platform. In a real-time system, the target-task may become stochastically active and inactive (a task must be active in order to be processed successfully). The dedicated-platform must then adapt its operation to the dynamic of the target-task: The dedicated-platform attempts to process (utilizes the system resource) only when the target-task is active and hence the task can be completed with a non-zero probability.

In this paper, we consider the problem of optimally operating a dedicated-platform when the dynamic of the target-task is *not* directly observed. The only

information available to the controller is whether the task is successfully processed or not. Associated with each processing attempt is a cost that represents limited resources of the system, independent of whether the task is active or inactive. On the other hand, the latency to successfully process the target task also incurs a cost representing the task completion delay. Therefore, there is a strong motivation to devise a novel algorithm to attempt or suspend processing to minimize the average cost up to the completion of the target-task (successful processing). In this view, we present a computationally efficient control algorithm that achieves an optimal trade-off between the processing cost and the latency in completing the target task.

Main results: The main results of this paper are organized as follows:

(i) In Section 2, we introduce a stochastic model for the operation of the dedicated-platform. We assume the controller decisions as whether to attempt or suspend the task are made at discrete times and the cost at each discrete time is only dependent on the action taken for that time. The *state* of the target-task, i.e. whether the target-task is active or inactive, is assumed to be a two-state Markov chain. Since we assume the task dynamic is not directly observed, the state of the target-task is described by a Hidden Markov model.

(ii) In Section 3, we use the model in Section 2 to formulate the dedicated-platform control problem as an optimal search problem with a POMDP framework. The optimal solutions to Markovian search problems are generally complex and computationally intractable. However, there are special structures of the search problems whose optimal solutions are shown to be threshold in nature and hence efficiently computable. We show that in our case, the control of the dedicated-platform can be formulated as a search problem with a special structure described in [1],[2] for which the optimal policy is threshold in nature.

(iii) In Section 4, we adapt the results of the Markovian search problem in [1] to present the optimal threshold policies for the control of the dedicated-platform. We show that depending on the system parameters, the operating control systems can be categorized into three different classes. The optimal policy for each system has a different threshold level.

(iv) In section 5, we use numerical examples to demonstrate the performance improvement that can be obtained by applying the optimal threshold policies as compared to heuristic algorithms.

Literature review: Several papers consider the search problem formulation used in this paper. Ross [2] first conjectured the existence of threshold policies for this search problem. Weber [3] solves the continuous time version of the problem. Recently, MacPhee and Jordan [1] prove the Ross' conjecture for an overwhelming proportion of transition rules and system parameters. Also, a useful overview of general search problems is given by Benkoski, Monticino and Weisinger [4]. In our paper we mainly rely on [1] to derive optimal control algorithms for a dedicated-platform. A similar formulation is also used in [5] to find optimal retransmission policies in the Gilbert-Elliot fading channels.

2 System Model

In this section a stochastic model is presented to describe the operation dynamic of our dedicated-platform. We outline our model by the following five elements:

(i) Time: The time axis is divided into slots of equal duration denoted by ΔT . By convention, discrete time k , $k \in Z_+$ is the the time interval $[k\Delta T, (k+1)\Delta T)$, where Z_+ is the set of non-negative integers. We assume that the attempt or suspend decisions by the controller are made in discrete times $k \in Z_+$.

(ii) Markovian target-task: Assume a dynamic where the target-task is active or inactive based on a two state Markov chain. Note that the task must be active in order to be processed successfully. Define the target-task state space as:

$$S = \{\text{Active} \triangleq 1, \text{Inactive} \triangleq 2\}. \quad (1)$$

Also define $s_k \in S$ as:

$$s_k = \text{State of the target-task at discrete time } k. \quad (2)$$

We assume s_k is a two-state irreducible Markov chain with the transition matrix A , where:

$$A = \begin{bmatrix} a & 1-a \\ 1-h & h \end{bmatrix} \quad (3)$$

Here, $a < 1$ is the probability that an active task remains in the “Active” state and $h < 1$ is the probability that an inactive task remains in the “Inactive” state.

(iii) Actions: At each discrete time $k \in Z_+$, the controller makes a decision as whether to attempt to process, or suspend the target-task. Define the action space U as:

$$U = \{\text{Attempt to process} \triangleq \text{At}, \text{Suspend processing} \triangleq \text{Su}\}. \quad (4)$$

Also, define the action $u_k \in U = \{\text{At}, \text{Su}\}$:

$$u_k \triangleq \text{Action taken by the controller at time } k. \quad (5)$$

(iv) Observations: The state of the target-task is not directly observed and hence it is described by a Hidden Markov Model (HMM). At each time, the controller can only observe whether the completion of the target-task is successful or not. For example, if for a given discrete time the target-task is inactive, the controller observes at the next discrete time that the completion is not successful. Define the observation space Y :

$$Y = \{\text{Completion Affirmed} \triangleq \text{AFF}, \text{Completion Not Affirmed} \triangleq \text{NAF}\}, \quad (6)$$

and define the observation $y_k \in Y$:

$$y_k \triangleq \text{Observation by the controller at time } k. \quad (7)$$

If the target-task is inactive at time k (i.e., $s_k = 2$) or if there is no attempt to process (i.e., $u_k = \text{Su}$) then $y_{k+1} = \text{NAF}$. On the other hand, if at time k the target-task is “Active” ($s_k = 1$) and the attempt is made to process the task ($u_k = \text{At}$), then the target-task will be successfully completed at time $k + 1$ with probability p_d which represents the processing precision:

$$p_d \triangleq \text{Probability that the active task is completed upon processing.} \quad (8)$$

We then have:

$$\begin{aligned} \mathbf{P}(y_{k+1} = \text{NAF} | s_k = 1, u_k = \text{At}) &= 1 - p_d \\ \mathbf{P}(y_{k+1} = \text{NAF} | s_k = 2, u_k = \text{At}) &= 1 \\ \mathbf{P}(y_{k+1} = \text{NAF} | s_k = 1, u_k = \text{Su}) &= 1 \\ \mathbf{P}(y_{k+1} = \text{NAF} | s_k = 2, u_k = \text{Su}) &= 1 \end{aligned} \quad (9)$$

(v) Cost: We assume the cost at each discrete time $k \in Z_+$ only depends on the action $u_k \in U$. In particular, associated with each processing ($u_k = \text{At}$) a cost c_1 incurs (independent of the current state or observation). c_1 represents the cost in utilizing the dedicated-platform and the limitations in the system resources. Also, each suspension of processing incurs a cost c_2 which represents the cost associated with the latency in completing the target-task. Let $g : U \rightarrow \{c_1, c_2\}$ be a function that maps the action space to the corresponding costs. We then have:

$$c_1 = g(\text{At}), \quad c_2 = g(\text{Su}). \quad (10)$$

3 Formulation as a Markovian Search Problem - A POMDP Framework

In this section we formulate the dedicated-platform control problem as a special Markovian search problem studied in [1], [2]. This search problem is proven to have optimal solutions that are threshold in nature and hence efficiently computable. The Markovian search problem described in [1], [2] is as follows:

Markovian search problem : *Consider an object that moves randomly between two sites. The movement is modeled with a two-state Markov chain. One of the sites is searched at each discrete times $k \in Z_+$ till the object is found. Associated with each search of site $i \in \{1, 2\}$ there is a cost C_i and an overlook probability α_i (α_i is the probability that the object is not found while it is in the searched site i). The aim is to find the object with minimum average cost.*

It is roughly seen that the structure of the above Markovian search problem fits into the framework that we have so far developed for finding an optimal control policy of the dedicated-platform. The movement of the object corresponds to the activation and deactivation of the target-task. Object in site 1 corresponds to an active target-task and object in site 2 corresponds to an inactive target-task. Searching site 1 corresponds to processing the task and searching site 2

corresponds to suspending the task. Finding the object corresponds to the completion of the target-task. Also, denoting the overlook probabilities in searching the two sites by α_1 and α_2 , we have:

$$\alpha_1 = 1 - p_d, \quad \alpha_2 = 1, \quad (11)$$

where precision p_d is defined in (8). Note that $\alpha_2 = 1$ since if we suspend processing, almost surely the task will not be completed.

At this point, we formulate the optimal control problem as a POMDP and derive its dynamic programming equations. We observe that the optimality equation of this POMDP has the exact same structure as a Markovian search problem in [1].

Let I_k be the information available to the controller at discrete time $k \in \mathbb{Z}_+$. This information consists of observations up to time k and actions up to time $k - 1$:

$$I_k = (y_1, \dots, y_k, u_1, \dots, u_{k-1}), \quad I_1 = y_1, \quad (12)$$

where y_k and u_k are observations and actions defined in (7) and (5), respectively. Since upon the completion of the target-task, the control is terminated, throughout the following analysis we assume $y_k = NAF$ for $0 < k < N$, where N is the stopping time denoting the discrete-time that the target-task is completed. At each discrete time k the controller takes an action based on the available information I_k . However, since the dimension of I_k is growing in time, we summarize I_k in quantities denoted by *sufficient statistics* which are of smaller dimension than I_k and yet embody all of the essential content of I_k as far as the control is concerned [6]. By checking the conditions in [6], it can be easily shown that a sufficient statistic can be given by the conditional probability distribution $P_{s_k|I_k}$ of the target-task state s_k , given the information vector I_k . $P_{s_k|I_k}$ then summarizes all the information available to the controller at time k . We have:

$$P_{s_k|I_k} = [p_k \quad q_k]', \quad (13)$$

where:

$$p_k = \mathbf{P}(s_k = 1|I_k), \quad q_k = \mathbf{P}(s_k = 2|I_k) \quad (14)$$

where $\mathbf{P}$ denotes the probability measure. p_k is the probability that the target-task is in the “Active” state at time k based on all the available information (e.g. knowing that the task is not yet completed) at time k . Since $p_k + q_k = 1$, we can further reduce the dimension of the sufficient statistics by choosing p_k as the *information state*. p_k then contains all the information relevant to the platform control at time k .

To complete the formulation of the problem, we need to describe the evolution of the information state. Let ϕ be a function that describe the evolution of the information state. We then have:

$$\phi(p_k, u_k) \triangleq p_{k+1} = \mathbf{P}(s_{k+1} = 1|I_{k+1}), \quad (15)$$

where $u_k \in U = \{\text{At}, \text{Su}\}$ is the action at time k . By expanding the R.H.S in (15) and applying basic probability rules, we have:

$$\begin{aligned} \phi(p_k, \text{At}) &= \mathbf{P}(s_{k+1} = 1 \mid u_k = \text{At}, y_{k+1} = \text{NAF}, I_k) \\ &= \frac{\mathbf{P}(s_{k+1} = 1, y_{k+1} = \text{NAF} \mid u_k = \text{At}, I_k)}{\mathbf{P}(y_{k+1} = \text{NAF} \mid u_k = \text{At}, I_k)} \\ &= \frac{\mathbf{P}(y_{k+1} = \text{NAF} \mid s_{k+1} = 1, u_k = \text{At}, I_k) \mathbf{P}(s_{k+1} = 1 \mid u_k = \text{At}, I_k)}{\mathbf{P}(y_{k+1} = \text{NAF} \mid u_k = \text{At}, I_k)} \end{aligned} \quad (16)$$

Evaluate the R.H.S in (16) by conditioning on s_k :

$$\phi(p_k, \text{At}) = \frac{a(1 - p_d)p_k + (1 - h)(1 - p_k)}{(1 - p_d)p_k + (1 - p_k)} \quad (17)$$

where a and h are the target-task transition probabilities defined in (3). Similar calculations give the expression for the updated state if we suspend processing:

$$\begin{aligned} \phi(p_k, \text{Su}) &= \mathbf{P}(s_{k+1} = 1 \mid u_k = \text{Su}, y_{k+1} = \text{NAF}, I_k) \\ &= (a + h - 1)p_k + 1 - h \end{aligned} \quad (18)$$

Equations (17) and (18) collectively describe the evolution of the information state p_k . Now, we formulate the optimality equation and show it has the same structure as the search problem in [1].

Let $\mathbf{u} = \{u_1, u_2, \dots\}$ be a sequence of the actions $u_k \in U$ taken by the controller at discrete times $k \in \mathbb{Z}_+$. Define $\mathbf{u}_{(n)} = \{u_n, u_{n+1}, \dots\}$. Let $V(p_1; \mathbf{u})$ to be the average cost of completing the target-task using the policy $\mathbf{u}$ with initial information state p_1 . Clearly, $V(\cdot, \mathbf{u})$ satisfies:

$$V(p_1, \mathbf{u}) = g(u_1) + V(\phi(p_1, u_1); \mathbf{u}_{(2)}) \mathbf{P}(y_2 = \text{NAF} \mid u_1, I_1) \quad (19)$$

where $g(\cdot)$, defined in (10), gives the cost associated with each action and $\phi(p_k, u_k)$ is given by (17) and (18). The term $\mathbf{P}(y_2 = \text{NAF} \mid u_1, I_1)$ is needed since it is the probability that the completion of the task is not affirmed and controlling decisions will still continue at time 2. Now, let $V(p_1)$ denote the *minimum* average cost starting with initial state p_1 . Then:

$$V(p_1) = \inf_{\mathbf{u}} V(p_1; \mathbf{u}), \quad (20)$$

and $V(p_1)$ satisfies the Bellman dynamic programming functional equation:

$$V(p_1) = \min\{c_1 + V(\phi(p_1, \text{At}))((1 - p_d)p_1 + 1 - p_1); c_2 + V(\phi(p_1, \text{Su}))\}, \quad (21)$$

where c_1 and c_2 are the costs given in (10). It is well known [2] that this functional equation has a unique bounded solution and furthermore V is concave in p .

The Bellman equation in (21) has the exact same structure as the the optimality equation in [1] with $\alpha_1 = 1 - p_d$ and $\alpha_2 = 1$. We therefore conclude that the problem of optimal platform control has been formulated as a Markovian search problem described in [1].

4 Optimal Policies for the Operation of the Dedicated-Platform

Generally, a value iteration algorithm as described in [7] can be used to solve the Bellman equation in (21). However, this method is often computationally complex and inefficient. In this section we obtain the solution to the optimality equation in (21) by using the special structure of the equivalent Markovian search problem described in [1]. For this Markovian search problem Ross [2] conjectured the existence of optimal threshold policies. In 1995, MacPhee and Jordan proved this conjecture for an overwhelming proportion of the possible transition matrices, search costs and overlook probabilities. By adapting the results in [1] we show that the optimal policy for the dedicated-platform control is a threshold policy: *The controller attempts to process if the probability of the target-task being active is greater or equal than a certain threshold level, otherwise processing is suspended.*

The following theorem states the existence of an optimal threshold policy for the dedicated-platform control problem:

Theorem 1. *Let p_k be the state information at time k in the dedicated-platform control problem. Then there exists a threshold value, δ , such that for any $k \in \mathbb{Z}_+$, if $p_k \geq \delta$, the optimal action at time k is to attempt to process the target-task, and if $p_k < \delta$, the optimal action at time k is to suspend processing.*

Proof. By observing the corresponding optimality equations, we established in Section 3 that our platform control problem is equivalent to a special form of a two-state Markovian search in [1]. The reader is then referred to [1] to see the details of the proof for the equivalent search problem.

It is shown in [1] that depending on the system parameters $\{a, h, c_1, c_2, p_d\}$, there are three different threshold levels, δ . In particular, whether which threshold level is applicable depends explicitly on the *fixed points* of the evolution equations in (17) and (18). Let P_{At} be the fixed point of $\phi(\cdot, \text{At})$ in equation (17) and P_{Su} be the fixed point of $\phi(\cdot, \text{Su})$ in equation (18). P_{At} and P_{Su} are then given by:

$$P_{\text{At}} = \frac{2 - ((1 - p_d)a + h) - \sqrt{((1 - p_d)a + h)^2 - 4(1 - p_d)\mu}}{2p_d}, \quad (22)$$

$$P_{\text{Su}} = \frac{1 - h}{1 - \mu},$$

where we have defined:

$$\mu \triangleq a + h - 1. \quad (23)$$

μ is an eigenvalue of the target-task transition matrix, A (defined in (3)), and hence can be regarded as a measure of the *memory* in the target-task state transitions (e.g. if $\mu = 0$ then the target-task state transitions are i.i.d). Also, to express the main result of this section we need to define the following mappings

of the information state by two different consecutive actions (i.e. $\{At, Su\}$ and $\{Su, At\}$):

$$\begin{aligned} P_{At,Su}(\cdot) &\triangleq \phi(\phi(\cdot, At), Su) \\ P_{Su,At}(\cdot) &\triangleq \phi(\phi(\cdot, Su), At), \end{aligned} \quad (24)$$

where $\phi(\cdot, u \in \{At, Su\})$ is defined in (15) and is given by equations (17) and (18).

The following proposition states the main result of [1] adapted to our platform control problem:

Proposition 1. *The platform control system is categorized into three different classes - Class 1, Class 2 and 3. Each class has a different threshold value δ_1 , δ_2 and δ_3 . Class membership rules are as follows:*

$$\text{Class 1 : } \delta_1 < P_{At} \quad (25)$$

$$\text{Class 2 : } P_{At} < \delta_2 < P_{Su} \quad \text{and} \quad P_{Su,At}(\delta_2) < \delta_2 < P_{At,Su}(\delta_2)$$

$$\text{Class 3 : } P_{At} < \delta_3 < P_{Su} \quad \text{and} \quad \{\delta_3 > P_{At,Su}(\delta_3) \quad \text{or} \quad \delta_3 < P_{Su,At}(\delta_3)\},$$

where the fixed points P_{At} and P_{Su} are given in (22) and $P_{At,Su}$ and $P_{Su,At}$ are defined in (24). The Threshold levels for class 1 and class 2 are given by:

$$\delta_1 = \frac{(1-h)(c_1 - c_2)}{(1-\mu)c_1}. \quad (26)$$

$$\delta_2 = \frac{(1-h)(c_1 - \mu c_2)}{(1-\mu)(c_1 + c_2)}, \quad (27)$$

where μ is defined in (23). The threshold level for Class 3, δ_3 , cannot be obtained in closed form but as explained in [1], δ_3 can be numerically computed by applying multiple compositions of $\phi(\cdot, At)$ and $\phi(\cdot, Su)$.

The following is an important conceptual consequence of the above proposition:

Corollary 1. *Each class is uniquely determined by the system parameters $\{a, h, c_1, c_2, p_d\}$. Furthermore, the system belongs to one and only one of the classes.*

At this point, we have obtained the optimal threshold policies for our dedicated-platform control problem. By analyzing the properties of these policies, we observe that for Class 1 systems, the optimal control policy is to suspend processing till the information state p_k exceeds the threshold δ_1 . After that, the controller successively attempts to process up to the completion of the target-task. This is because in Class 1, once p_k exceeds the threshold, the updated information state will be still above the threshold δ_1 after each attempt. In the case of Class 2 and 3, the optimal policy may have a more complex form, i.e., the optimal actions may vary between successive attempts and suspensions. In the next section we justify our results by numerical examples to demonstrate the performance improvement that can be obtained by the optimal threshold policies as compared to heuristic algorithms.

5 Numerical Examples

The purpose of this section is to evaluate by numerical experiments the performance of the optimal threshold policy in terms of the incurred average cost up to the completion of the target-task. We consider three different scenarios, whereby different costs and different processing precisions p_d are selected. Also, we examine three different control policies: optimal threshold policy, persistent attempt and Suspend-M. The persistent attempt is the most aggressive method where the controller chooses to process at each discrete time until the target-task is completed. Suspend-M denotes a method that controller waits for M discrete-time after an unsuccessful attempt before attempting to process the target-task again [8]. The number M generally increases with the state transition memory as described in [8]. We assume the stationary distribution of the target-task states is $\pi = [1/2 \quad 1/2]$ so in a long term the target-task can be active or inactive with equal probabilities. The stationary distribution of matrix A , defined in (3), is simply calculated as $[\frac{1-a}{1-\mu} \quad \frac{1-h}{1-\mu}]$. Therefore, we have:

$$\frac{1-a}{1-\mu} = \frac{1-h}{1-\mu} = \frac{1}{2} \quad (28)$$

where $\mu = a + h - 1$. The above gives $a = h$ which is also obvious from the symmetry of our assumption.

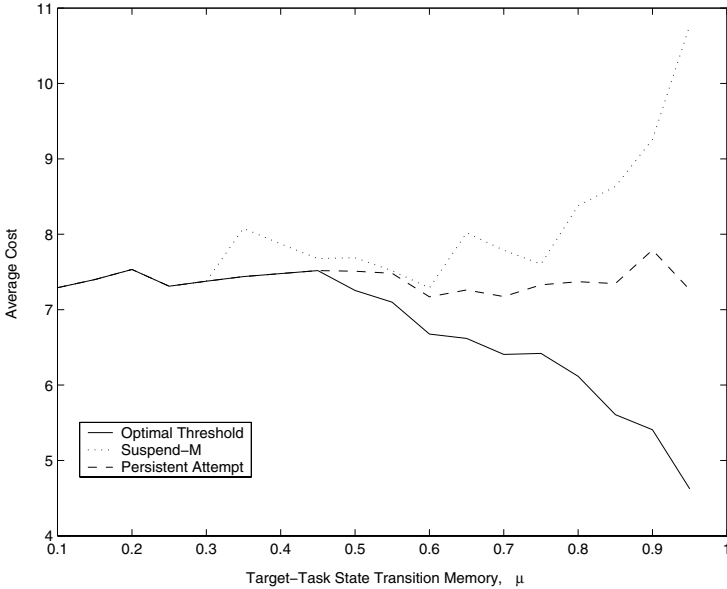


Fig. 1. Average cost vs. target-task transition memory: $a = h$, $c_1 = 4$, $c_2 = 1$, $p_d = 0.7$

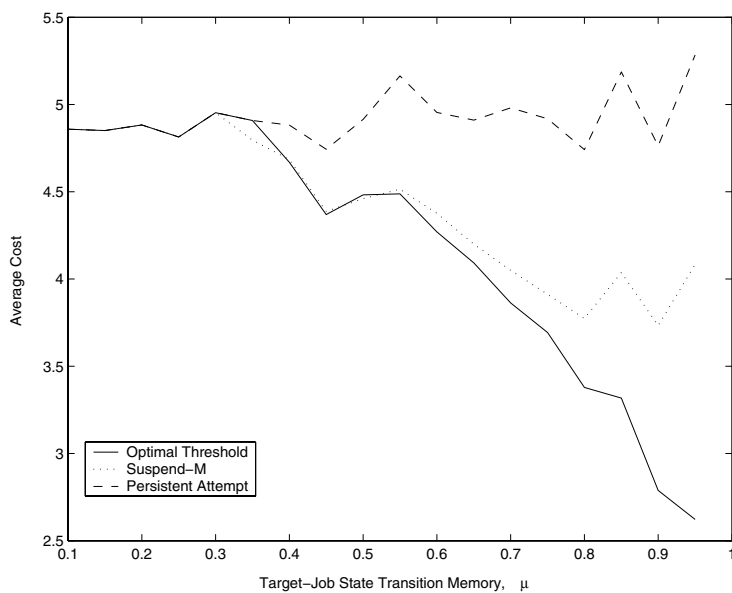


Fig. 2. Average cost vs. target-task transition memory: $a = h$, $c_1 = 4$, $c_2 = 1$, $p_d = 0.9$

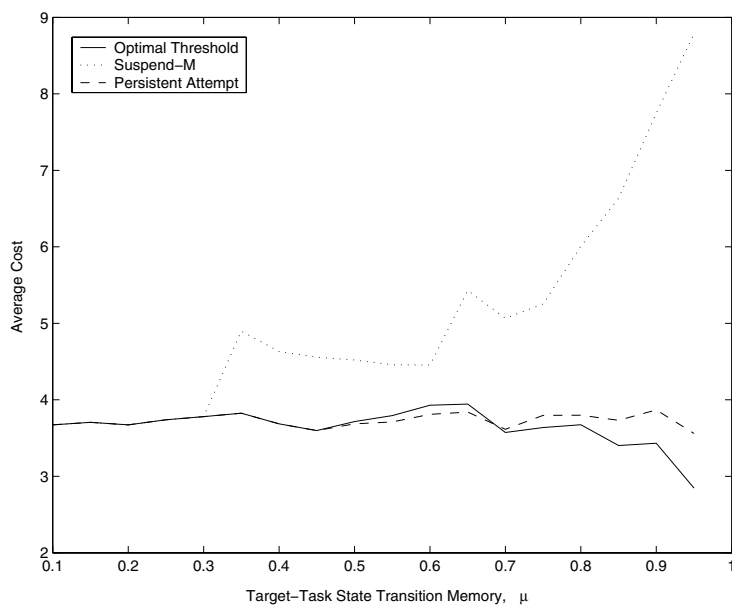


Fig. 3. Average cost vs. target-task transition memory: $a = h$, $c_1 = 2$, $c_2 = 1$, $p_d = 0.7$

The results for $c_1 = 4$, $c_2 = 1$, and $p_d = 0.7$ are shown in Fig 1. It is clear that the threshold policy gives the best performance. In the case that the processing precision increases to $p_d = 0.9$, as shown in Fig 2, the Suspend-M policy gives a better performance. However, the threshold policy still gives the lowest average cost. By reducing the cost of processing attempt to $c_1 = 2$, as shown in Fig 3, the persistent attempt policy gives a close performance to the optimal policy. In all cases when the memory, μ , increases, the Suspend-M policy shows a degraded performance while the persistent attempt policy shows much less variations.

6 Conclusion

We have derived stochastic control algorithms to achieve the optimal trade-off between the processing cost and the latency in completing the target-task by a dedicated-platform. The structural results in Makovian target search problems have been used to derive optimal threshold control policies. The resulting threshold policies are efficiently computable and easy to implement. We have shown by numerical examples that these policies outperform non-optimal heuristic algorithms in terms of the average task completion cost.

References

1. I. MacPhee and B. Jordan, "Optimal search for a moving target," *Probability in the Engineering and Information Sciences*, vol. 9, pp. 159–182, 1995.
2. S. Ross, *Introduction to Stochastic Dynamic Programming*. Academic Press, 2000.
3. R. R. Weber, "Optimal search for a randomly moving object," *Journal of Applied Probability*, vol. 23, pp. 708–717, 1986.
4. S. J. Benkoski, M. G. Monticino, and J. R. Weisinger, "A survey of the search theory literature," *Naval Research Logistics*, vol. 38, pp. 469–494, 1991.
5. L. A. Johnston and V. Krishnamurthy, "Optimality of threshold transmission policies in Gilbert Elliott fading channels," in *IEEE International Conference on Communications, ICC '03*, vol. 2, pp. 1233–1237, May 2003.
6. D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd ed., 2000.
7. A. Lovejoy, "A survey of algorithmic methods for partially observed Markov decision processes," *Annals of Operations Research*, vol. 28, pp. 47–66, 1991.
8. D. Zhang and K. M. Wasserman, "Energy efficient data communication over fading channels," *IEEE Wireless Communications and Networking Conference*, pp. 986–991, 2000.

APPSSAT: Approximate Probabilistic Planning Using Stochastic Satisfiability

Stephen M. Majercik

Bowdoin College, Brunswick ME 04011, USA

smajerci@bowdoin.edu

<http://www.bowdoin.edu/~smajerci>

Abstract. We describe APPSSAT, an approximate probabilistic contingent planner based on ZANDER, a probabilistic contingent planner that operates by converting the planning problem to a stochastic satisfiability (SSAT) problem and solving that problem instead [1]. The values of some of the variables in an SSAT instance are probabilistically determined; APPSSAT considers the most likely instantiations of these variables (the most probable situations facing the agent) and attempts to construct an approximation of the optimal plan that succeeds under those circumstances, improving that plan as time permits. Given more time, less likely instantiations/situations are considered and the plan is revised as necessary. In some cases, a plan constructed to address a relatively low percentage of possible situations will succeed for situations not explicitly considered as well, and may return an optimal or near-optimal plan. This means that APPSSAT can sometimes find optimal plans faster than ZANDER. And the anytime quality of APPSSAT means that suboptimal plans could be efficiently derived in larger time-critical domains in which ZANDER might not have sufficient time to calculate the optimal plan. We describe some preliminary experimental results and suggest further work needed to bring APPSSAT closer to attacking real-world problems.

1 Introduction

Previous research has extended the planning-as-satisfiability paradigm to support probabilistic contingent planning; in [1], it was shown that a probabilistic, partially observable, finite-horizon, contingent planning problem can be encoded as a stochastic satisfiability (SSAT) [2] instance such that the solution to the SSAT instance yields a contingent plan with the highest probability of reaching a goal state. This has been used to construct ZANDER, a competitive probabilistic contingent planner [1]. APPSSAT is a probabilistic contingent planner based on ZANDER that produces an approximate contingent plan and improves that plan as time permits. APPSSAT does this by considering the most probable situations facing the agent and constructing a plan, if possible, that succeeds under those circumstances. Given more time, less likely situations are considered and the plan is revised as necessary.

Other researchers have explored the possibility of using approximation to speed the planning process. In “anytime synthetic projection” a set of control rules establishes a base plan which has a certain probability of achieving the goal [3]. Time permitting, the probability of achieving the goal is incrementally increased by identifying failure situations that are likely to be encountered by the current plan and synthesizing additional control rules to handle these situations. Similarly, MAHINUR is a probabilistic partial-order planner that creates a base plan with some probability of success and then improves that plan [4].

Exploring approximation techniques in Markov decision processes (MDPs) and partially observable Markov decision processes (POMDPs) is a very active area of research. In [5] value functions are represented using decision trees and these decision trees are pruned so that the leaves represent ranges of values, thereby approximating the value function. Evidence that the value function of a factored MDP can often be well approximated using a factored value function has been presented in [6], and it is shown that this approximation technique can be used as a subroutine in a policy iteration process to solve factored MDPs [7]. A method for choosing, with high probability, approximately optimal actions in an infinite-horizon discounted Markov decision process using truncated action sequences and random sampling is described in [8]. In [9] the authors transform a POMDP into a simpler, fully observable POMDP in which it is assumed an oracle tells the agent what region its current state is in. This POMDP is easier to solve and they use its solution to construct an approximate solution for the original POMDP.

In Section 2, we describe stochastic satisfiability. In Section 3, we describe how ZANDER uses stochastic satisfiability to solve probabilistic planning problems. In Section 4, we describe the APPSSAT algorithm for approximate planning and in Section 5 we describe some preliminary experimental results. We conclude with a discussion of further work.

2 Stochastic Satisfiability

SSAT, suggested in [10] and explored further in [2], is a generalization of satisfiability (SAT) that is similar to quantified Boolean formulae (QBF). The ordered variables of the Boolean formula in an SSAT problem, instead of being existentially or universally quantified, are existentially or randomly quantified. Randomly quantified variables are **true** with a certain probability, and an SSAT instance is satisfiable with some probability that depends on the ordering of and interplay between the existential and randomized variables. The goal is to choose values for the existentially quantified variables that maximize the probability of satisfying the formula.

More formally, an SSAT problem $\Phi = Q_1v_1 \dots Q_nv_n\phi$ is specified by a sequence $Q_1v_1 \dots Q_nv_n$ that orders a set of n Boolean variables $V = \{v_1, \dots, v_n\}$ and specifies the quantifier Q_i associated with each variable v_i , and a formula ϕ that is a Boolean formula constructed from these variables. More specifically, the prefix $Q_1v_1 \dots Q_nv_n$ associates a quantifier Q_i , either existential ($\exists_i$) or randomized

($\mathfrak{A}_i^{\pi_i}$), with the variable v_i . The value of an existentially quantified variable can be set arbitrarily by a solver, but the value of a randomly quantified variable is determined stochastically by π_i , an arbitrary rational probability that specifies the probability that v_i will be **true**. (In the basic SSAT problem described in [2], every randomized variable is **true** with probability 0.5, but it is noted that the probabilities associated with randomized variables can be arbitrary rational numbers.) In this paper, we will use $x_1, x_2, \dots$ for existentially quantified variables and $y_1, y_2, \dots$ for randomly quantified variables.

The matrix ϕ is assumed to be in conjunctive normal form (CNF), i.e. a set of m conjuncted clauses, where each clause is a set of distinct disjuncted literals. A literal l is either a variable v (a **positive** literal) or its negation $\neg v$ (a **negative** literal). For a literal l , $|l|$ is the variable v underlying that literal and $\bar{l}$ is the “opposite” of l , i.e. if l is v , $\bar{l}$ is $\neg v$; if l is $\neg v$, $\bar{l}$ is v ; A literal l is **true** if it is positive and $|l|$ has the value **true**, or if it is negative and $|l|$ has the value **false**. A literal is **false** if $|l|$ is existentially (randomly) quantified. The probability that a randomly quantified variable v has the value **true** (**false**) is denoted $Pr[v]$ ($Pr[\neg v]$). The probability that a randomized literal l is **true** is denoted $Pr[l]$. As in a SAT problem, a clause is satisfied if at least one literal is **true**, and unsatisfied, or **false**, if all its literals are **false**. The formula is satisfied if all its clauses are satisfied.

The solution of an SSAT instance is an assignment of truth values to the existentially quantified variables that yields the maximum probability of satisfaction, denoted $Pr[\Phi]$. Since the values of existentially quantified variables can be made contingent on the values of randomly quantified variables that appear earlier in the prefix, the solution is, in general, a **contingent assignment** that specifies the optimal assignment to each existentially quantified variable x_i for each possible instantiation of the randomly quantified variables that precede x_i in the prefix. A simple example will help clarify this idea before we define $Pr[\Phi]$ formally. Suppose we have the following SSAT problem:

$$\exists x_1, \mathfrak{A}^{0.7} y_1, \exists x_2 \{ \{x_1, y_1\}, \{x_1, \overline{y_1}\}, \{y_1, x_2\}, \{\overline{y_1}, \overline{x_2}\} \} . \quad (1)$$

The form of the solution is a noncontingent assignment for x_1 plus two contingent assignments for x_2 , one for the case when y_1 is **true** and one for the case when y_1 is **false**. In this problem, x_1 should be set to **true** (if x_1 is **false**, the first two clauses become $\{\{y_1\}, \{\overline{y_1}\}\}$, which specify that y_1 must be both **true** and **false**), and x_2 should be set to **true** (**false**) if y_1 is **false** (**true**). Since it is possible to satisfy the formula for both values of y_1 , $Pr[\Phi] = 1.0$. If we add the clause $\{\overline{y_1}, x_2\}$ to this instance, however, the maximum probability of satisfaction drops to 0.3: x_1 should still be set to **true**, and when y_1 is **false**, x_2 should still be set to **true**. When y_1 is **true**, however, we have the clauses $\{\{\overline{x_2}\}, \{x_2\}\}$, which insist on contradictory values for x_2 . Hence, it is possible to satisfy the formula only when y_1 is **false**, and, since $Pr[\neg y_1] = 0.3$, the probability of satisfaction, $Pr[\Phi]$, is 0.3.

We will need the following additional notation to define $Pr[\Phi]$ formally. A partial assignment α of the variables V is a sequence of $k \leq n$ literals $l_1; l_2; \dots; l_k$

such that no two literals in α have the same underlying variable. Given l_i and l_j in an assignment α , $i < j$ implies that the assignment to $|l_i|$ was made before the assignment to $|l_j|$. A positive (negative) literal v ($-v$) in an assignment α indicates that the variable v has the value **true** (**false**). The notation $\Phi(\alpha)$ denotes the SSAT problem Φ' remaining when the partial assignment α has been applied to Φ (i.e. clauses with **true** literals have been removed from the matrix, **false** literals have been removed from the remaining clauses in the matrix, and all variables and associated quantifiers not in the remaining clauses have been removed from the prefix) and $\phi(\alpha)$ denotes ϕ' , the matrix remaining when α has been applied. Similarly, given a set of literals L , such that no two literals in L have the same underlying variable, the notation $\Phi(L)$ denotes the SSAT problem Φ' remaining when the assignments indicated by the literals in L have been applied to Φ , and $\phi(L)$ denotes ϕ' , the matrix remaining when the assignments indicated by the literals in L have been applied. A literal $l \notin \alpha$ is **active** if some clause in $\phi(\alpha)$ contains l ; otherwise it is **inactive**.

Given an SSAT problem Φ , the maximum probability of satisfaction of Φ , denoted $Pr[\Phi]$, is defined according to the following recursive rules:

1. If ϕ contains an empty clause, $Pr[\Phi] = 0.0$.
2. If ϕ is the empty set of clauses, $Pr[\Phi] = 1.0$.
3. If the leftmost quantifier in the prefix of Φ is existential and the variable thus quantified is v , then $Pr[\Phi] = \max(Pr[\Phi(v)], Pr[\Phi(-v)])$.
4. If the leftmost quantifier in ϕ is randomized and the variable thus quantified is v , then $Pr[\Phi] = (Pr[\Phi(v)] \times Pr[v]) + (Pr[\Phi(-v)] \times Pr[-v])$.

These rules express the intuition that a solver can select the value for an existentially quantified variable that yields the subproblem with the higher probability of satisfaction, whereas a randomly quantified variable forces the solver to take the probability weighted average of the two possible results.

There are simplifications that allow an algorithm implementing this recursive definition to avoid the often infeasible task of enumerating all possible assignments. A solver can interrupt the normal left-to-right evaluation of quantifiers to take advantage of **active** and **inactive** literals. A literal l is **active** if it is the only literal in some clause; in this case, $|l|$ must be assigned the value that makes l **true**. A literal l is **inactive** if l is active and $\bar{l}$ is inactive; if l is an existential pure literal, $|l|$ can be set to make l **true** without changing $Pr[\Phi]$. These simplifications modify the rules given above for determining $Pr[\Phi]$, but we omit a restatement of the modified rules, instead describing an algorithm to solve SSAT instances based on the modified rules (Fig. 1). Note that both ZANDER and APPSSAT construct and return the optimal solution tree (plan), but we omit the details of solution tree construction in the algorithm description.

3 ZANDER

ZANDER works on partially observable probabilistic propositional planning domains consisting of a finite set of distinct **variables**, any of which may be


```

SolveSSAT ( $\Phi$ )
  if  $\phi$  contains an empty clause: return 0.0;
  if  $\phi$  is the empty set of clauses: return 1.0;
  if some  $l$  in  $\Phi$  is an existential unit literal:
    return SolveSSAT( $\Phi(l)$ );
  if some  $l$  in  $\Phi$  is a randomized unit literal:
    return SolveSSAT( $\Phi(l)$ ) * Pr[ $l$ ];
  if some  $l$  in  $\Phi$  is an existential pure literal:
    return SolveSSAT( $\Phi(l)$ );
  if the leftmost quantifier in  $\Phi$  is  $\exists$  and its variable is  $v$ :
    return max(SolveSSAT( $\Phi(v)$ ), SolveSSAT( $\Phi(-v)$ ));
  if the leftmost quantifier in  $\Phi$  is  $\forall$  and its variable is  $v$ :
    return (SolveSSAT( $\Phi(v)$ ) * Pr[ $v$ ]) + (SolveSSAT( $\Phi(-v)$ ) * Pr[ $-v$ ]);

```

Fig. 1. The basic algorithm for solving SSAT instances

true or false at any discrete time t . A $\langle \text{assignment} \rangle$ is an assignment of truth values to these propositions. A possibly probabilistic $\langle \text{plan} \rangle$ is specified by a set of decision trees, one for each proposition. $\langle \text{states} \rangle$ are specified by a partial assignment to the set of propositions; any state that extends this partial assignment is a goal state. Each of a finite set of $\langle \text{actions} \rangle$ probabilistically transforms a state at time t into a state at time $t + 1$ and so induces a probability distribution over the set of all states at time $t + 1$. A subset of the set of propositions is the set of $\langle \text{observable propositions} \rangle$. The task is to find an action for each step t as a function of the value of observable propositions for steps before t that maximizes the probability of reaching a goal state.

ZANDER translates the planning problem into an SSAT problem. Fig. 2 shows an example of such an SSAT plan encoding (where all unit and pure literals have been removed as described above and the effects propagated). In this problem, a part must be painted, but the paint action succeeds only with probability 0.7 and it is an error to try to paint the part if it is already painted. The agent has two time steps, so the best plan is to paint the part at $t = 1$ and observe whether the action was successful, painting again (at $t = 2$) if it was not, and doing nothing (noop) otherwise.

$$\exists pa_1 \exists no_1 \forall opd_1 \exists pa_2 \exists no_2 \forall cvp_1^{0.7} \forall cvp_2^{0.7} \exists pd_1$$

$$\begin{aligned}
& \{ \{ pa_1, no_1 \} \quad , \{ \overline{pa_1}, \overline{pd_1}, opd_1 \} , \{ pa_2, no_2 \} \quad , \\
& \{ \overline{pa_1}, \overline{no_1} \} \quad , \{ \overline{pa_1}, pd_1, \overline{opd_1} \} , \{ \overline{pa_2}, \overline{no_2} \} \quad , \\
& \{ \overline{pa_1}, cvp_1^{0.7}, pd_1 \} , \{ pa_1, \overline{pd_1} \} \quad , \{ \overline{pa_2}, cvp_2^{0.7}, pd_1 \} , \\
& \{ \overline{pa_1}, cvp_1^{0.7}, \overline{pd_1} \} , \{ \overline{no_1}, \overline{opd_1} \} \quad , \{ \overline{pa_2}, \overline{pd_1} \} \quad , \\
& \quad , \{ pa_1, \overline{opd_1} \} \quad , \{ pa_2, pd_1 \} \}
\end{aligned}$$

Fig. 2. An example of an SSAT plan encoding, where pa_1 = (paint at $t = 1$), no_1 = (noop at $t = 1$), opd_1 = (observe painted after the action at $t = 1$), pa_2 = (paint at $t = 2$), no_2 = (noop at $t = 2$), $cvp_1^{0.7}$ = (chance variable associated with pa_1), $cvp_2^{0.7}$ = (chance variable associated with pa_2), and pd_1 = (painted at $t = 1$)

The variables in an SSAT plan encoding fall into three segments [1]: the action-observation segment (variables pa_1 , no_1 , opd_1 , pa_2 , no_2 in Fig. 2), the domain uncertainty segment (variables $cvp_1^{0.7}$, $cvp_2^{0.7}$ in Fig. 2), and a segment representing the result of the actions taken given the domain uncertainty (variable pd_1 in Fig. 2). The action-observation segment is an alternating sequence of existentially quantified variable blocks (one for each action choice) and randomly quantified variable blocks (one for each set of possible observations at a time step). If Fig. 2, pa_1 and no_1 constitute the first existentially quantified action block, opd_1 is the first (and only) randomly quantified observation block, and pa_2 and no_2 constitute the second existentially quantified action block. We will refer to an instantiation of these variables as an *action-observation path*. The domain uncertainty segment is a single block containing all the randomly quantified variables that modulate the impact of the actions on the observation and state variables. The result segment is a single block containing all the existentially quantified state variables. Essentially, ZANDER uses the solver described in Section 2 to find the optimal *action-observation tree*. An action-observation tree is composed of action-observation paths whose assignments are mutually consistent and that specify the assignments to existentially quantified action variables for all possible settings of the observation variables. The optimal action-observation tree is the one that maximizes the probability of satisfaction (i.e. the probability that the plan will reach the goal) [1]. In what follows, we will refer to existentially and randomly quantified variables as *action* and *observation* variables, respectively.

4 APPSSAT

Before we describe APPSSAT it is worth looking at *randevalssat*, a previous approach to approximation in this framework. This algorithm illuminates some of the problems associated with formulating such an algorithm and explains some of the choices we made in developing APPSSAT. The *randevalssat* algorithm uses stochastic local search in a reduced plan space [2]. It uses random sampling to select a subset of possible chance variable instantiations (thus limiting the size of the contingent plans considered) and stochastic local search to find the best size-bounded plan. There are two problems with this approach. First, since chance variables are used to describe observations, a random sample of the chance variables describes an observation sequence as well as an instantiation of the uncertainty in the domain, and the observation sequence thus produced may not be *internally consistent*, and these inconsistencies can make it impossible to find a plan, even if one exists. Second, this algorithm returns a partial policy, that specifies actions only for those situations represented by paths in the random sampling of chance variables. APPSSAT addresses these two problems by:

1. designating each observation variable as a special type of variable, termed a *contingent* variable, rather than a chance variable, and
2. evaluating the approximate plan's performance under all circumstances, not just those used to generate the plan.

The introduction of branch variables violates the pure SSAT form of the plan encoding, but is justified, we think, for the sake of conceptual clarity. We could achieve the same end in the pure SSAT form by making observation variables chance variables (as in [1]), and not including them when the possible chance variable assignments are enumerated. But, rather than taking this circuitous route, we have chosen to acknowledge the special role played by observation variables; these variables indicate a potential branch in a contingent plan (hence the name). As such, the value of an observation variable node in the assignment tree described above is the *majority* of the values of its children. This introduces a minor modification into the ZANDER approach and has the benefit of clarifying the role of the observation variables.

APPSSAT incrementally constructs the optimal action-observation tree (described in Section 3) by generating the instantiations of the chance variables in descending order of probability, finding all choice (action) variable assignments that are consistent with each chance variable instantiation in turn, and updating the probabilities of the possible action-observation paths as it processes these chance variable instantiations. APPSSAT can stop this process after any number of chance variable assignments have been considered and extract and evaluate the best plan (action-observation tree) for the chance variable assignments that have been considered so far (thus yielding an *anytime* algorithm). The current best plan is extracted by finding the action-observation tree whose action-observation path probabilities sum to the highest probability. (Note that this probability is a lower bound on the true probability of success of the plan represented by the tree.) The probability of success of that plan is found by evaluating the full assignment tree using that plan.

If the probability of success of this plan is sufficient (probability 1.0 or exceeding a user-specified threshold), APPSSAT halts and return the plan and probability; otherwise, APPSSAT continues processing chance variable assignments. Note that the probability of success of the just-extracted plan can be used as a new lower threshold in subsequent plan evaluations, often allowing additional pruning to be done. The quality of the plan produced increases (if the optimal success probability has not already been attained) with the available computation time. See Fig. 3 for a description of the algorithm.

Because the chance variable instantiations are investigated in descending order of probability, a plan with a relatively high percentage of the optimal success probability can potentially be found quickly. An exception is a domain in which the high probability situations are hopeless and the best that can be done is to construct a plan that addresses some number of lower probability situations. Even here, the basic SSAT heuristics used will allow APPSSAT to quickly discover that no plan is possible for the high-probability situations, and lead it to focus on the low-probability situations for which a plan is feasible. Of course, if *all* chance variable assignments are considered, the plan extracted is the optimal plan, but, as we shall see, the optimal plan may sometimes be produced even after only a relatively small fraction of the chance variable assignments have been considered.

```

APPSSAT ( $\Phi$ , k, d,  $\pi_{\text{thresh}}$  )
  k = number of chance variable instantiations to be considered;
  d = number of chance variable instantiations processed per iteration;
   $\pi_{\text{thresh}}$  = minimum acceptable probability of satisfaction (plan success);
  pc = current plan, initially empty;
   $\pi_{\text{pc}}$  = probability of success of the current plan, initially 0.0;
  w = function that maps action-observation paths to probabilities,
      initially all 0.0;
  i = 0;
  while (i < k/d  $\wedge$   $\pi_{\text{pc}}$  <  $\pi_{\text{thresh}}$ );
    for j = (i * d) + 1 to (i * d) + d:
      cij = jth chance variable instantiation in descending order
            of probability;
      Pr[cij] = probability of chance variable instantiation cij;
      for each action-observation path (aop) that is consistent with cij:
        w(aop) = w(aop) + Pr[cij];
      pc = current best plan;
       $\pi_{\text{pc}}$  = Pr[pc reaches the goal];
  return pc and  $\pi_{\text{pc}}$ 

```

Fig. 3. The APPSSAT algorithm for solving SSAT instances

Unlike ZANDER, which, in effect, looks at chance variable instantiations at a particular time step based on the instantiation of variables (particularly action variables) at previous times steps, APPSSAT, by enumerating complete instantiations of the chance variables in descending order of probability, examines the most likely outcomes of all actions at all time steps. Because it is not taking variable independencies into account, it does so somewhat inefficiently. At the same time, however, by instantiating all the chance variables at the same time, APPSSAT reduces the SSAT problem to a much simpler SAT problem. Although this approach will also entail the repeated solving of a number of sub-problems with one or more chance variable settings changed, the conjecture is that solving a large number of SAT problems will take less time than solving a large number of SSAT problems. Obviously, this will depend on the relative number of problems involved, but we have chosen to explore the approach embodied in APPSSAT first.

In the current implementation of APPSSAT, the user specifies k , the total number of chance variable instantiations to be considered, d , the interval of chance variable instantiations processed after which the current plan should be extracted and evaluated (the default is 5% of the total number of chance variable assignments), and π_{thresh} , the minimum acceptable probability of satisfaction (plan success). If the algorithm finds a plan whose probability meets or exceeds π_{thresh} , it halts and returns that plan. Otherwise, it returns the best plan after all k chance variable instantiations have been processed.

All of the operations in APPSSAT can be performed as or more efficiently than the operations necessary in the ZANDER framework. The chance variable instantiations can be generated in time linear in the number of instantiations

using a priority queue. Finding all consistent action-observation paths amounts to a depth-first search of the assignment tree checking for satisfiability using pruning heuristics (the central operation of ZANDER). Note also that once an action-observation path is instantiated, checking whether it can be extended to a satisfying assignment amounts to a series of fast unit literal propagations. In fact, once the chance variables have all been set, the remaining variables are all choice variables and the search for all action-observation paths that lead to satisfying assignments can be accomplished by any efficient SAT solver that finds all satisfying assignments. Extracting the current best plan involves a depth-first search of the action-observation tree, which is sped up by the fact that satisfiability does not have to be checked. Finally, plan evaluation requires a depth-first search of the entire assignment tree, but heuristics speed up the search, and the resulting probability of success can be used as a lower threshold if the search continues, thus potentially speeding up subsequent computation.

5 Results

Preliminary results are mixed but indicate that APPSSAT has some potential as an approximation technique. In some cases, it outperforms ZANDER, in spite of the burden of the additional approximation machinery. And, in those cases where its performance is poorer, there is potential for improvement (see Further Work). We tested APPSSAT on three domains that ZANDER was tested on in [1]. The TIGER problem contains uncertain initial conditions and a noisy observation; the agent needs the entire observation history in order to act correctly. The COFFEE-ROBOT problem is a larger problem (7 actions, 2 observation variables, and 8 state propositions in each of 6 time steps) with uncertain initial conditions, but perfect causal actions and observations. Finally, the GO (GENERAL OPERATIONS) problem has no uncertainty in the initial conditions, but requires that probabilistic actions be interleaved with perfect observations. All experiments were conducted on an 866 MHz Dell Precision 620 with 256 Mbytes of RAM, running Linux 7.1.

In the 4-step TIGER problem, ZANDER found the optimal plan (0.93925 probability of success) in 0.01 CPU seconds. APPSSAT requires 0.42 CPU seconds to find the same plan (extracting and evaluating the current plan after every 5% of chance variable instantiations). This is, however, if we insist on forcing APPSSAT to look for the best possible plan (and, thus, to process all 512 chance variable instantiations), which seems somewhat out of keeping with the notion of APPSSAT as an approximation technique. If we run APPSSAT on this problem under similar assumptions, but specify $\pi_{\text{thresh}} = 0.90$ (we will accept any plan with a success probability of 0.90 or higher), APPSSAT returns a plan in 0.02 CPU seconds. The plan returned is, in fact, the optimal plan, and is found after examining the first 18 chance variable instantiations.

Table 1 provides an indication of what kind of approximation would be available if less time were available than what would be necessary to compute the

Table 1. Probability of success increases with number of chance variable instantiations

4-STEP TIGER			6-STEP COFFEE-ROBOT			7-STEP GO		
NCVI	SECS	PROB	NCVI	SECS	PROB	NCVI	SECS	PROB
1	0.0	0.307062	1	2.24	0.5	1	1.06	0.1250
2	0.0	0.614125	2	4.98	0.5	2	1.20	0.1250
3	0.0	0.614125	3	9.12	1.0	3	1.51	0.1250
4	0.0	0.668312	4	15.07	1.0	4	1.74	0.1250
5	0.01	0.668312	–	–	–	5	1.98	0.1250
6	0.01	0.722500	–	–	–	6	2.17	0.1250
7	0.01	0.722500	–	–	–	7	2.47	0.1250
8	0.01	0.722500	–	–	–	8	2.67	0.1250
9	0.01	0.776687	–	–	–	9	2.92	0.1250
10	0.01	0.776687	–	–	–	10	3.07	0.125
11	0.01	0.830875	–	–	–	11	3.36	0.1875
12	0.01	0.830875	–	–	–	12	3.62	0.1875
13	0.01	0.885062	–	–	–	13	3.83	0.1875
14	0.01	0.885062	–	–	–	14	4.03	0.1875
15	0.01	0.885062	–	–	–	15	4.26	0.1875
16	0.02	0.885062	–	–	–	16	4.47	0.1875
17	0.02	0.885062	–	–	–	17	4.83	0.1875
18	0.02	0.939250	–	–	–	18	4.97	0.1875
–	–	–	–	–	–	19	5.16	0.2500
–	–	–	–	–	–	20	5.44	0.2500

NCVI = number of chance variable instantiations

SECS = time in CPU seconds

PROB = probability of plan success

optimal plan. This table shows how computation time and probability of plan success increases with the number of chance variable instantiations considered until the optimal plan is reached at 18 chance variable instantiations.

The 6-step COFFEE-ROBOT problem provides an interesting counterpoint to the TIGER problem in that APPSSAT does better than ZANDER. ZANDER is able to find the optimal plan (success probability 1.0) in 19.34 CPU seconds, while APPSSAT can find the same plan in 9.12 CPU seconds. There are only 4 chance variable instantiations in the COFFEE-ROBOT problem and, since extraction and evaluation of the plan at intervals of 5% would result in intervals of less than one, the algorithm defaults to extracting and evaluating the plan after each chance variable instantiation is considered. Although one might conjecture that this constant plan extraction and evaluation is a waste of time, in this case it leads to the discovery of an optimal plan (success probability of 1.0) after processing the first 3 chance variable instantiations, and the resulting solution time of 9.12 CPU seconds (including plan extraction and evaluation time) is less than the solution time if we force APPSSAT to wait until all four chance variable instantiations have been considered before extracting and evaluating the best plan (15.07 CPU seconds).

This illustrates an interesting tradeoff. In the latter case, although APPSSAT does not extract and evaluate the plan after each chance variable instantiation, it does an extra chance variable instantiation, and this turns out to take more time than the extra plan extractions and evaluations. This is not surprising since checking a chance variable instantiation involves solving a SAT problem to find all possible satisfying assignments, while extracting and evaluating the plan requires only depth-first search. This suggests that we should be biased toward more frequent plan extraction and evaluation; more work is needed to determine if some optimal frequency can be automatically determined for a given problem. Table 1 provides an indication of how computation time and probability of plan success increases with the number of chance variable instantiations considered for the COFFEE-ROBOT problem. Interestingly, although the probability mass of the chance variables is spread uniformly across the four chance variable instantiations, APPSSAT is still able to find the optimal plan without considering all the chance variable instantiations.

The 7-step GO problem shows that this is not necessarily the case when, as in the GO problem, the probability mass is spread uniformly over many more (2^{21}) chance variable instantiations. In this problem, ZANDER is able to find the optimal plan (success probability 0.773437) in 2.48 CPU seconds. Because of the large number of chance variable instantiations to be processed, APPSSAT cannot approach this speed. APPSSAT needs about 566 CPU seconds to process 3000 (0.14%) of the total chance variable instantiations, yielding a plan with success probability of 0.648438. Table 1 provides an indication of how computation time and probability of plan success increases with the number of chance variable instantiations considered for the GO problem.

As the size of the problem increases, however, to the point where ZANDER might not be able to return an optimal plan in sufficient time, APPSSAT may be useful if it can return a plan with some probability of success in less time than it would take ZANDER to find the optimal plan. We tested this conjecture on the 10-step GO problem ($2^{30} = 1073741824$ chance variable instantiations). Here, ZANDER needed 405.35 CPU seconds to find the optimal plan (success probability 0.945313). APPSSAT was able to find a plan in somewhat less time (324.92 CPU seconds to process 20 chance variable instantiations), but this plan has a success probability of only 0.1875.

6 Further Work

We need to improve the efficiency of APPSSAT if it is to be a viable approximation technique, and there are a number of techniques we are in the process of implementing that should help us to achieve this goal. First, we are implementing an incremental approach: every time a new action-observation path is added, APPSSAT would incorporate that path into the current plan, checking to see if it changes that plan by checking values stored in that path from that point to the root. Whenever this process indicates that the plan has changed, the plan extraction and evaluation process will be initiated.

Second, when APPSSAT is processing the chance variable instantiations in descending order, in many cases the difference between two adjacent instantiations is small. We can probably take advantage of this to find the action-observation paths that satisfy the new chance variable instantiation more quickly.

Third, since we are repeatedly running a SAT solver to find action-observation paths that lead to satisfying assignments for the chance variable assignments, and since two chance variable assignments will frequently generate the same satisfying action-observation path, it seems likely that we could speed up this process considerably by incorporating learning into APPSSAT. (We also note that we could improve performance by taking advantage of the speed available from current state-of-the-art SAT solvers.)

Finally, we are investigating whether plan simulation (instead of exact calculation of the plan success probability) would be a more efficient way of evaluating the current plan.

References

1. Majercik, S.M., Littman, M.L.: Contingent planning under uncertainty via stochastic satisfiability. *Artificial Intelligence* **147** (2003) 119–162
2. Littman, M.L., Majercik, S.M., Pitassi, T.: Stochastic Boolean satisfiability. *Journal of Automated Reasoning* **27** (2001) 251–296
3. Drummond, M., Bresina, J.: Anytime synthetic projection: Maximizing the probability of goal satisfaction. In: *Proceedings of the Eighth National Conference on Artificial Intelligence*, Morgan Kaufmann (1990) 138–144
4. Onder, N., Pollack, M.E.: Contingency selection in plan generation. In: *Proceedings of the Fourth European Conference on Planning*. (1997) 364–376
5. Boutilier, C., Dearden, R.: Approximating value trees in structured dynamic programming. In: *Proceedings of the Thirteenth International Conference on Machine Learning*. (1996) 56–62
6. Koller, D., Parr, R.: Computing factored value functions for policies in structured MDPs. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, The AAAI Press/The MIT Press (1999) 1332–1339
7. Koller, D., Parr, R.: Policy iteration for factored MDPs. In: *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI 2000)*. (2000) 326–334
8. Kearns, M.J., Mansour, Y., Ng, A.Y.: A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine Learning* **49** (2002) 193–208
9. Zhang, N.L., Lin, W.: A model approximation scheme for planning in partially observable stochastic domains. *Journal of Artificial Intelligence Research* **7** (1997) 199–230
10. Papadimitriou, C.H.: Games against nature. *Journal of Computer Systems Science* **31** (1985) 288–301

Racing for Conditional Independence Inference

Remco R. Bouckaert¹ and Milan Studený^{2,*}

¹ Computer Science Department,
University of Waikato & Xtal Mountain Information Technology, New Zealand
`remco@cs.waikato.ac.nz`, `rrb@xm.co.nz`

² Institute of Information Theory and Automation,
Academy of Sciences of the Czech Republic, Prague, Czech Republic
`studený@utia.cas.cz`

Abstract. In this article, we consider the computational aspects of deciding whether a conditional independence statement t is implied by a list of conditional independence statements L using the implication related to the method of structural imsets. We present two methods which have the interesting complementary properties that one method performs well to prove that t is implied by L , while the other performs well to prove that t is not implied by L . However, both methods do not perform well the opposite. This gives rise to a parallel algorithm in which both methods race against each other in order to determine effectively whether t is or is not implied.

Some empirical evidence is provided that suggest this racing algorithms method performs a lot better than an existing method based on so-called skeletal characterization of the respective implication. Furthermore, the method is able to handle more than five variables.

1 Introduction

Conditional independence (CI) is a crucial notion in many calculi for dealing with knowledge and uncertainty in artificial intelligence [2, 3]. A powerful formalism for describing probabilistic CI structures is provided by the method of structural imsets [7]. In this algebraic approach, CI structures are described by certain vectors whose components are integers, called $\mathbf{imset}$. An important question is to decide whether a CI statement is implied by a set of CI statements. The method of structural imsets offers a sufficient condition for the probabilistic implication of CI statements. The offered inference mechanism is based on linear algebraic operations with imsets. The basic idea is that every CI statement can be translated into a simple imset and the respective algebraic relation between imsets, called $\mathbf{imset}$, forces the probabilistic implication of CI statements. Techniques were developed in [5] to test the

* The work of the second author has been supported by the grant GAČR n. 201/04/0393.

independence implication through systematic calculation when there are up to five variables involved.

For reasoning about CI statements with more than five variables one may resort to making severe assumptions. For example, one can assume that the CI structure is graph isomorphic for a class of graphs such as directed acyclic graphs (DAG) [3, 8], undirected graphs (UG) [2], chain graphs (CG) [1], etc. Then CI inference from a set of CI statements of a special form, a so-called $\mathcal{I}$, can be made as follows. The list is used to construct a graph and CI statements are read from the graph through the respective graphical separation criterion. However, the assumption that the CI structure is graph isomorphic may be too strong in many cases and only special input lists can be processed anyway. Using the method of structural imsets, many more CI structures can be described than with DAGs, UGs or CGs. However, the computational effort required when more than five variables are involved is not clear at present.

Fortunately, structural imsets have some properties that we can exploit. First, a relatively easy sufficient condition for independence implication is that the respective linear combination of imsets can be decomposed into so-called $\mathcal{I}$. The existence of this decomposition can be found relatively quickly. On the other hand, to prove that the decomposition does not exist requires trying all decompositions, which often takes a long time. Second, there exists a method to show that the independence implication does not hold. It suffices to find a certain vector, called $\mathcal{I}$, such that its inner product with the respective combination of structural imsets is negative. These supermodular functions can be generated randomly. This only allows us to disprove independence implication of imsets, not to disprove probabilistic implication of respective CI statements. However, if the obtained supermodular function is a multiple of a multiinformation function of a probability distribution [7] then it also allows us to disprove probabilistic implication of respective CI statements. Thus, we have one method that allows us to find a proof that a statement is implied, and one method to find a proof that a statement is not implied. However, both methods perform poorly in proving their opposite outcome. This gives rise to a race: both methods are started at the same time and the method that returns first also returns a proof whether the statement of interest is implied or not.

The following section introduces formal terminology and the fundamentals of CI inference using imsets. The racing algorithms are described in Section 3 where many more smaller optimizations are described as well. Section 4 presents experiments that were performed to get an impression of the run-times of various variants of inference algorithms. We conclude with some final comments and directions for further research.

2 Terminology

Let N be a set of variables $\{x_1, \dots, x_n\}$ ($n \geq 1$), as will be assumed throughout the paper. Let X and Y be subsets of N . We use XY to denote the union of

X and Y and $X \setminus Y$ to denote the set of variables that are in X but not in Y . Further, let x be a variable in N , then x will also denote the singleton $\{x\}$.

2.1 Conditional Independence

Let P be a discrete probability distribution over N and X, Y, Z pairwise disjoint subsets of N . We say that X is independent of Y given Z if $P(\mathbf{x}|\mathbf{yz}) = P(\mathbf{x}|\mathbf{z})$ for all configurations $\mathbf{x}, \mathbf{y}, \mathbf{z}$ of values for X, Y, Z with $P(\mathbf{yz}) > 0$. We write then $X \perp\!\!\!\perp Y | Z [P]$ or just $X \perp\!\!\!\perp Y | Z$, and call it a conditional independence statement. It is well-known that CI follows some simple rules, known as the semi-graphoid axioms, defined as follows ($X, Y, Z, W \subseteq N$ are pairwise disjoint):

Symmetry	$X \perp\!\!\!\perp Y Z$	$\Rightarrow Y \perp\!\!\!\perp X Z,$
Decomposition	$X \perp\!\!\!\perp WY Z$	$\Rightarrow X \perp\!\!\!\perp Y Z,$
Weak union	$X \perp\!\!\!\perp WY Z$	$\Rightarrow X \perp\!\!\!\perp W YZ,$
Contraction	$X \perp\!\!\!\perp W YZ \ \& \ X \perp\!\!\!\perp Y Z$	$\Rightarrow X \perp\!\!\!\perp WY Z.$

The problem we address in this paper is the following: Let L be a set of CI statements, called an *input list*, and t is a CI statement $X \perp\!\!\!\perp Y | Z$. Does L imply t ? More formally, is it true that for any discrete distribution P for which all statements in L hold necessarily t holds as well? This is the *inference problem* of those CI statements. The semi-graphoid axioms do not cover this implication. For example,

$$\begin{aligned} X \perp\!\!\!\perp Y | WZ \ \& \ W \perp\!\!\!\perp Z | X \ \& \ W \perp\!\!\!\perp Z | Y \ \& \ X \perp\!\!\!\perp Y | \emptyset \quad \Leftrightarrow \\ \Leftrightarrow \quad W \perp\!\!\!\perp Z | XY \ \& \ X \perp\!\!\!\perp Y | Z \ \& \ X \perp\!\!\!\perp Y | W \ \& \ W \perp\!\!\!\perp Z | \emptyset \end{aligned}$$

is also a valid rule [7]. In fact, there is no complete finite set of rules of this kind describing relationships between probabilistic CI statements [4]. A more powerful formalism to describe the properties of CI is provided by the method of structural imsets.

2.2 Imsets

An *integer-valued multiset* over N (abbreviation for **integer-valued multiset**) is an integer-valued function on the power set of N . It can be viewed as a vector whose components, indexed by subsets of N , are integers. Given $X \subseteq N$, we use δ_X to denote the identifier imset, that is, $\delta_X(X) = 1$ and $\delta_X(Y) = 0$ for all $Y \subseteq N$, $Y \neq X$. An imset associated with a CI statement $X \perp\!\!\!\perp Y | Z$ is $u_{\langle X, Y | Z \rangle} = \delta_{XYZ} + \delta_Z - \delta_{XZ} - \delta_{YZ}$. The imset associated with an input list L is then $u_L = \sum_{t \in L} u_t$.

The basic technique for inference of a statement t from an input list L using the method of structural imsets is based on the following property. If $n \cdot u_L$ (for some natural number $n \in \mathbb{N}$) can be written as u_t plus the sum of some imsets associated with CI statements then t is implied by L . This can be derived from results of [7]. For example, if L consists of a single statement $X \perp\!\!\!\perp WY | Z$ and t is $X \perp\!\!\!\perp Y | Z$, we have (with $n = 1$)

$$\begin{aligned}
n \cdot u_L &= \delta_{WXYZ} + \delta_Z - \delta_{XZ} - \delta_{WYZ} \\
&= (\delta_{XYZ} + \delta_Z - \delta_{XZ} - \delta_{YZ}) + (\delta_{WXYZ} + \delta_{YZ} - \delta_{XYZ} - \delta_{WYZ}) \\
&= u_t + u_{\langle X, W | YZ \rangle}.
\end{aligned}$$

Thus, $X \perp\!\!\!\perp WY \mid Z$ implies t and we have derived the decomposition rule of the semi-graphoid axioms. Realize that any statement in the decomposition on the right-hand side can be swapped with t , so those statements are implied too. This means that above we have derived weak union as well.

An $\langle x, y \mid z \rangle$ is an imset associated with an elementary CI statement $x \perp\!\!\!\perp y \mid Z$, namely $u_{\langle x, y | Z \rangle} = \delta_{xyz} + \delta_z - \delta_{xz} - \delta_{yz}$. It is convenient to denote the set of elementary imsets over N by $\mathcal{E}(N)$ or simply $\mathcal{E}$. A $\langle x, y \mid z \rangle$ is an imset u that can be decomposed into elementary imsets when multiplied by positive natural number, that is,

$$n \cdot u = \sum_{v \in \mathcal{E}} k_v \cdot v$$

for some $n \in \mathbb{N}$ and $k_v \in \mathbb{Z}^+$. Note that every structural imset induces a whole CI structure through an algebraic criterion, which is omitted here. The attraction of the method of structural imsets is that every discrete probabilistic CI structure can be described in this way [7].

Let u, v be structural imsets over N . We say that $u \prec v$ and write $u \rightarrow v$ if there exists $k \in \mathbb{N}$ such that $k \cdot u - v$ is a structural imset. This terminology is motivated by the fact that $u \rightarrow v$ actually means that u encodes more CI statements than v – see Lemma 6.1 in [7]. If $v \in \mathcal{E}$ then the constant $k \in \mathbb{N}$ can be supposed lesser than a limit $k_{\max}$ depending on the number of variables $|N|$ – see Lemma 4 in [6]. However, the value of the exact limit $k_{\max}$ for $|N| \geq 6$ is not known. It follows from results of [5] that $k_{\max} = 1$ if $|N| \leq 4$ and $k_{\max} = 7$ if $|N| = 5$. In our computer programs for $|N| \geq 6$ we need a limit for k . Instead of the unknown exact theoretical limit $k_{\max}$ we use the number $2^{|N|}$. Although we have not a proof of that we believe that $k_{\max} \leq 2^{|N|}$. Now, we can reformulate our inference problem. Given an elementary CI statement t and an input list (of elementary CI statements) L we are going to test whether $u_L \rightarrow u_t$. This is a sufficient condition for probabilistic implication of t by L . However, in general, it is not a necessary condition for it.

3 Algorithms

This section introduces algorithms for testing the implication $u_L \rightarrow u_t$. In Section 3.1, we revisit a method based on skeletal characterization of structural imsets from [7] and optimize the method. In Section 3.2, an algorithm for verification of $u_L \rightarrow u_t$ is presented based on searching a decomposition of $k \cdot u_L - u_t$ into elementary imsets. Section 3.3 concentrates on a method of disproving $u_L \rightarrow u_t$ by exploiting properties of supermodular functions. Section 3.4 combines the two previous methods by letting them race against each other and the one that returns its outcome first has a proof whether $u_L \rightarrow u_t$ or not.

3.1 Skeletal Characterization of Independence Implication

We will only consider the implementation details here. Technical details and motivation of this approach can be found in § 6.2.2 of [7]. This skeletal characterization is based on a particular set of imsets called the ℓ -skeleton, denoted as $\mathcal{K}_\ell^\circ(N)$. It follows from Lemma 6.2 in [7] that, for this particular set of imsets, we have $u_L \rightarrow u_t$ iff

$$\text{for all } m \in \mathcal{K}_\ell^\circ(N) \text{ if } \langle m, u_t \rangle > 0 \text{ then } \langle m, u_L \rangle > 0. \quad (1)$$

Recall that the inner product $\langle m, u \rangle$ of a function $m : \mathcal{P}(N) \rightarrow \mathbb{R}$ and an imset u is defined by $\sum_{S \subseteq N} m(S) \cdot u(S)$. Thus, to conclude $u_L \rightarrow u_t$, we just need to check the conditions in (1) for all imsets in the ℓ -skeleton.¹ It can be used to check which elementary imsets over five variables are implied in this sense by a user defining the input list.

The ℓ -skeleton for five variables consists of 117978 imsets, which break into 1319 permutational types with each involving at most 120 imsets. So, checking whether $u_L \rightarrow u_t$ requires at most 117978 operations [5]. However, if t is not implied by L , we might find out far earlier that (1) does not hold for a particular imset in $\mathcal{K}_\ell^\circ(N)$. By ordering skeletal imsets such that imsets that are more likely to cause violation in (1) are tried earlier, the required time can be minimized. The likelihood of violating (1) by $m \in \mathcal{K}_\ell^\circ(N)$ grows with the number of zeros in $\{\langle m, v \rangle; v \in \mathcal{E}\}$. Thus, sorting skeletal imsets on basis of this criterion helps to speed up the inference. The second auxiliary criterion is the number of sets $S \subseteq N$ with $u(S) = 0$.

Unfortunately, the skeletal characterization approach is hard to extend to more than five variables. First, because finding all elements of the ℓ -skeleton for more than five variables is computationally infeasible. Second, because it appears that the size of the ℓ -skeleton grows extremely fast with a growing number of variables. Therefore, we will consider different approaches to perform the inference in the rest of the paper.

3.2 Verification Algorithm

If an imset u is a combination of elementary imsets $u = \sum_{v \in \mathcal{E}} k_v \cdot v$, $k_v \in \mathbb{Z}^+$ then we say that it is a $\mathbb{Z}^+$ -combination. This is a sufficient condition for an imset to be structural and it is an open question if it is also a necessary condition [7]. The method to verify $u_L \rightarrow u_t$ presented in this section is based on testing whether $u \equiv k \cdot u_L - u_t$ is a combinatorial imset for some $k \in \mathbb{N}$.

Testing whether u is combinatorial can be done recursively, by checking, for each $v \in \mathcal{E}$, whether $u - v$ is combinatorial. Obviously, this naive approach is computationally demanding and it requires some guidance and extra tests in order to reduce the search space.

¹ An applet at http://www.utia.cas.cz/user_data/studený/VerifyView.html uses this method.

There are a number of sanity checks we can apply, before starting the search. First of all, let t be $X \perp\!\!\!\perp Y \mid Z$, then $u_L \rightarrow u_t$ implies there exists $W \supseteq XYZ$ with $u_L(W) > 0$. This can be shown by Proposition 4.4 from [7] where we use $m^{A\dagger}$ with $A = XYZ$. Another sanity check is as follows. Whenever u is a structural imset and $S \subseteq N$ a maximal set with respect to inclusion satisfying $u(S) \neq 0$ then $u(S) > 0$. Likewise, $u(S) > 0$ for any minimal set satisfying $u(S) \neq 0$ – see Lemma 6.5 in [7].

To guide the search, for each elementary imset $v \in \mathcal{E}$, we define the deviance of v from a non-zero imset u as follows. Let $\text{maxcard}(u)$ be the cardinality of the largest set $S \subseteq N$ for which $u(S) \neq 0$. It follows from the notes above that if u is structural then $u(S) \geq 0$ whenever $|S| = \text{maxcard}(u)$. Then, with $v = u_{\langle x, y \mid z \rangle}$,

$$\text{dev}(v|u) = \begin{cases} \infty & |xyz| < \text{maxcard}(u) \text{ or } u(xyz) \leq 0, \\ \sum_{S \subseteq N} |v(S) - u(S)| & \text{otherwise.} \end{cases}$$

Thus, the deviance of v from a combinatorial imset u is finite only if δ_{xyz} has a positive coefficient in u and no set larger than $|xyz|$ has a positive coefficient in u . We pick the elementary imset with the lowest deviance first. Observe that if u is a non-zero combinatorial imset then $v \in \mathcal{E}$ with finite $\text{dev}(v|u)$ exists.

The deviance is defined in such a way that the elementary imsets that cancel as many of the coefficients in u as possible are tried before the imsets that cancel out fewer of the coefficients. For example, let $u = u_{\langle x, wy \mid z \rangle} + u_{\langle x, y \mid z \rangle} = \delta_{xywz} + 2\delta_z - 2\delta_{xz} - \delta_{wyz} + \delta_{xyz} - \delta_{yz}$ and $v_1 = u_{\langle x, w \mid yz \rangle} = \delta_{xywz} + \delta_{yz} - \delta_{xyz} - \delta_{wyz}$ then $\text{dev}(v_1|u) = 8$ while $v_2 = u_{\langle w, z \mid xy \rangle} = \delta_{xywz} + \delta_{xy} - \delta_{wxy} - \delta_{xyz}$ has the deviance $\text{dev}(v_2|u) = 10$. Furthermore $v_3 = u_{\langle x, y \mid rz \rangle}$ has infinite deviance since $|xyz| = 3$ while $\text{maxcard}(u) = 4$. Finally, $v_4 = u_{\langle w, y \mid rz \rangle}$ has infinite deviance as $u(rwyz) = 0$. Therefore, v_1 will be tried before v_2 , while v_3 and v_4 will not be tried at all in this cycle.

Thus, the deviance leads our search in a direction where we can hope to find a proper decomposition. Obviously, if t is not implied by L , the verification algorithm can spend a long time searching through the complete space of possible partial decompositions.

3.3 Falsification Algorithm

Falsification is based on supermodular functions. A function $m : \mathcal{P}(N) \rightarrow \mathbb{R}$ such that, for all $X, Y \subseteq N$,

$$m(XY) + m(X \cap Y) - m(X) - m(Y) \geq 0.$$

Note that an equivalent definition is that $\langle m, v \rangle \geq 0$ for every $v \in \mathcal{E}$. For example, δ_N is a supermodular function. By a *supermodular imset* we understand an imset which is a supermodular function.

Theorem 1. *If u is a supermodular imset and $\langle m, u \rangle \geq 0$ for every supermodular imset m such that $\sum_{S, S \supseteq K} u(S) = 0$ for every $K \subseteq N$ with $|K| \leq 1$, then u is a structural imset.*

The necessity of the conditions is easy for they both hold for elementary imsets and can be extended to structural imsets. The sufficiency follows from Theorem 5.1 in [7] which claims that the same holds for a finite subset of the class of supermodular functions, namely the ℓ -skeleton $\mathcal{K}_\ell^\circ(N)$.

Thus, we can exploit Theorem 1 to disprove $u_L \rightarrow u_t$ by constructing non-negative supermodular imsets randomly and taking their inner products with $k \cdot u_L - u_t$. If u_t is elementary and, for all $1 \leq k \leq k_{\max}$, the inner product is negative then we can conclude that $\neg(u_L \rightarrow u_t)$. A random supermodular imset m can be generated by first generating a 'base' imset m_{base} and then by modifying it to ensure the resulting imset is supermodular. We randomly select the size n of the base, then randomly select n different subsets $S_1, \dots, S_n$ of N and assign $m_{base} = \sum_{S \in \{S_1, \dots, S_n\}} k_S \cdot \delta_S$ where k_S are randomly selected integers in the range from 1 to $2^{|N|}$. Selecting larger values of the coefficients k_S would not make difference. On the other hand, they also would not help.

Now, m_{base} needs to be modified to ensure that the obtained function m is supermodular. We perform the following operation on m_{base} . Let $S_1, \dots, S_{2^{|N|}}$ be an ordering of the subsets of N with $S_j \subseteq S_i \Rightarrow j \leq i$. For $i = 1, \dots, 2^{|N|}$ define $m(S_i)$ to be the maximum of $m_{base}(S_i)$ and $m(S_i \setminus x) + m(S_i \setminus y) - m(S_i \setminus xy)$ for all $x, y \in S_i$. This ensures that $\langle m, v \rangle \geq 0$ for all $v \in \mathcal{E}$ and we have constructed an imset m which is supermodular.

Note that this technique can be used to disprove $u_L \rightarrow u_t$ but it cannot be used to prove it.

3.4 Racing Algorithms for a Proof

Typically, the verification algorithm from Section 3.2 can quickly find a decomposition of $k \cdot u_L - u_t$ into $\sum_{v \in \mathcal{E}} k_v \cdot v$, which proves that t is implied by L . Nevertheless, if $\neg(u_L \rightarrow u_t)$, the verification algorithm may spend a long time before it exhausts the whole space of possible decompositions of $k \cdot u_L - u_t$. However, the falsification algorithm from Section 3.3 can find a supermodular imset m with $\langle m, k \cdot u_L - u_t \rangle < 0$, which proves u_t is not implied by u_L . On the other hand, it will not be able to prove that $u_L \rightarrow u_t$.

We can combine the two algorithms by starting two threads, one with the verification algorithm and one with the falsification algorithm. The one that finds

```

Algorithm: Racing for inference with structural imsets
Input: Input list  $L$ , CI statement  $t$ 
1: thread1 = new RaceThread(Verify( $L$ ,  $t$ , proof))
2: thread2 = new RaceThread(Falsify( $L$ ,  $t$ , proof), thread1)
4: thread1.start(); thread2.start()
5: thread1.join() // wait for thread1 to stop
// if thread2 finished first, it will stop thread1
6: thread2.stop()
return proof
    
```

Fig. 1. Racing algorithm

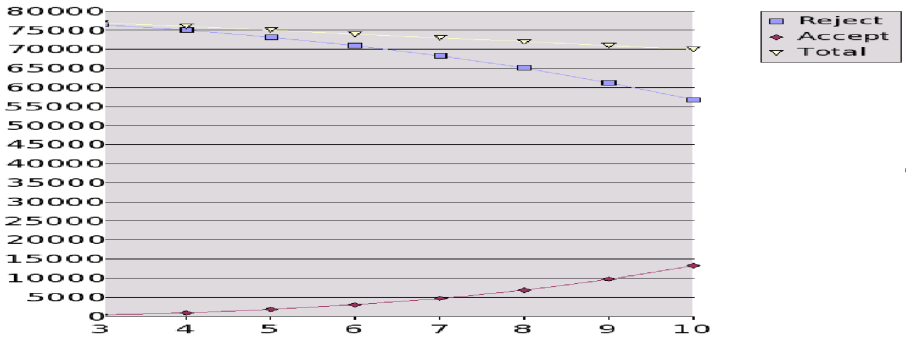


Fig. 2. Total number of rejects and accepts per experiment over 5 variables for various input list sizes. The size of the input list is shown on the x-axis. The number of rejects, accepts and total of unknown elementary statements is shown on the y-axis

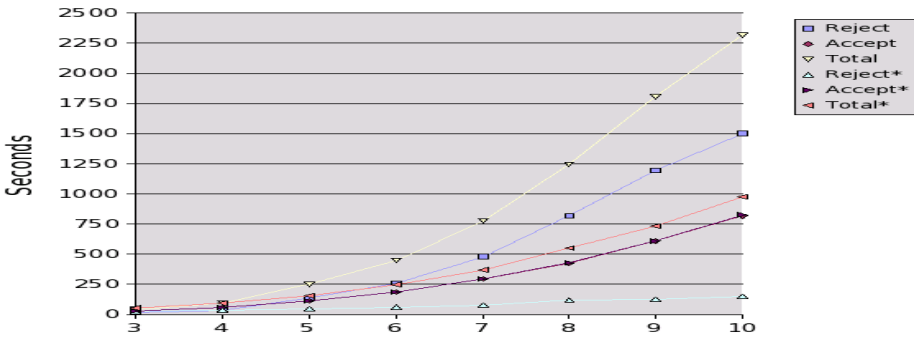


Fig. 3. Original skeleton-based testing compared with sorted skeleton-based testing. Sequences marked with asterisk are results for the sorted testing

a proof first, returns its outcome and stops the other thread. Figure 1 illustrates the algorithm.

4 Experiments

We would like to judge the algorithms above on computational speed. However, it is hard to get a general impression of the performance of the algorithms, because it depends on the distribution of inference problems, which is unknown.

Still, we think we can get a representative impression of the relative performance of the algorithms by generating inference problems randomly and measuring the computation speed. We generated inference problems over five variables so that we can compare the performance of the skeleton-based algorithm from Section 3.1 with the others. A thousand input lists each were generated by randomly selecting 3,4 up to 10 elementary CI statements, giving a total of 8000 input lists. The algorithms described in Section 3 were applied to this class of lists with each of the elementary CI

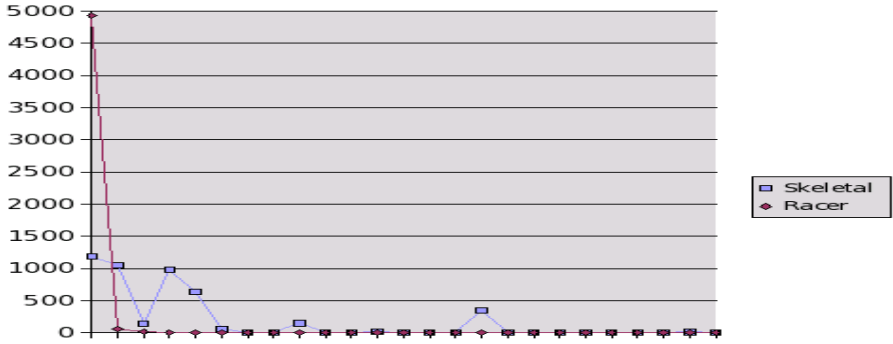


Fig. 4. Distribution of reject times of sorted skeleton-based method and racing algorithms method for input lists of size 10. The x-axis shows time, and the y-axis the number of elementary statements rejected in that time

statements that were not in the list. This gave 1000×77 inference problems for input lists with 3 statements, 1000×76 inference problems for input lists with 4 statements, etc. In total, this created $1000 \times ([80 - 3] + [80 - 4] + \dots + [80 - 10]) = 588.000$ inference problems over five variables.

Figure 2 shows the total number of elementary CI statements that are implied (labeled by **Accept**) and not implied (labeled by **Reject**) grouped by the number of elementary CI statements (3, 4 up to 10) in the input list. Naturally, the number of implied statements increases with increased input list size.

Figure 3 shows the total run-times for running the experiments comparing skeleton-based testing with sorted skeleton-based testing. We distinguish between run-time for accepts, rejects and total because the run-time for accepts is not influenced by the order of skeletal imsets as all of them need to be inspected. Indeed, run-times for accepts hardly differed (run-times only slightly differ due to the fact that at random intervals garbage collection and other processes were

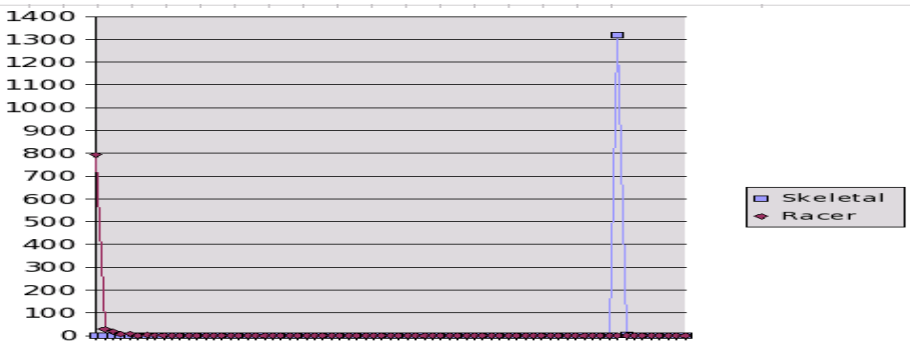


Fig. 5. Distribution of accept times of the sorted skeleton-based method and the racing algorithms method for input lists of size 10. The x-axis shows time, and the y-axis the number of elementary statements accepted in that time

Table 1. Number of fails of the falsification algorithm with two different methods of generating random base imsets and various input list sizes (times $1000 \times k_{\max}$)

$ L $	Rnd 1	Rnd 2						
	1	1	2	3	4	5	20	
3	1	0	0	0	0	0	0	
4	19	2	0	0	0	0	0	
5	57	18	3	6	2	3	1	
6	147	50	37	24	18	16	5	
7	243	92	61	39	46	42	21	
8	429	189	144	124	109	95	48	
9	423	195	138	112	97	92	46	
10	547	299	239	201	192	193	110	

performed). Run-times for rejects are reduced by about one order of magnitude so that total run-times are about halved. Thus, sorting the skeleton indeed helps significantly.

Figure 4 shows the striking difference in reject times for the racing algorithms method from Section 3.4 and the skeleton-based method from Section 3.1, which clearly favors the new method. Only input lists of size 10 are shown, but the shapes for input lists of other size are the same.

Unfortunately, the distribution of accept times shows a different picture, as illustrated in Figure 5. The graph for skeleton-based method shows just one peak around 6 seconds per elementary CI statement, because that is how long it approximately takes to visit all skeletal imsets. The graph for the racing algorithms² shows a peak close to 10 milliseconds, that drops off pretty quickly. Shapes for input lists of other size look very similar, though the tail gets thinner with decreasing size of input lists.

An alternative approach is to only run the falsification algorithm and run it long enough that the complete space of elementary statements is covered. Table 1 shows the number of fails³ of the falsification algorithm.

Two methods of generating random 'base' imsets are compared. The first method draws weights from the interval 1 to 32 for randomly selected subsets, while the second always selects 1. The second method appears far more effective in identifying rejections as one can judge from the number of fails in the columns labeled 1 in Table 1. We also looked at the impact of the number of randomly selected supermodular imsets on the number of fails. Increasing this number decreases the failure rate, but the rate only drops very slowly. Even when generating the same number of supermodular functions as the number of skeletal imsets in the skeleton-based method, not all statements are correctly classified.

² It is actually enlargement of the graph for the verification algorithm since the falsification thread cannot return acceptance.

³ These are those elementary CI statements that are not implied by the input list but the algorithm did not succeed to identify them in a fixed time limit.

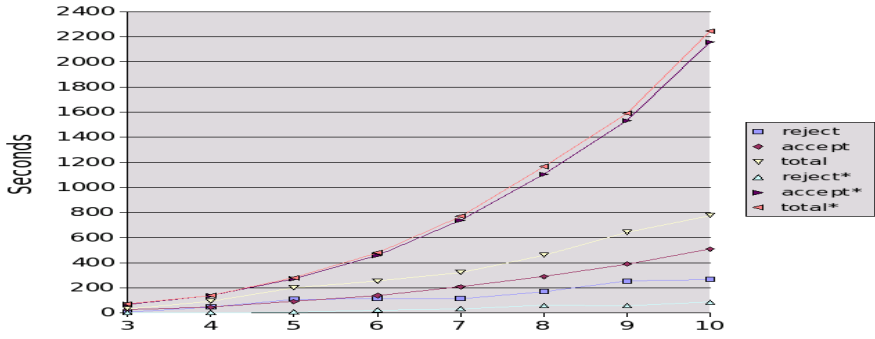


Fig. 6. Racing algorithms vs. sole falsification algorithm. Sequences marked with asterisk are results for the falsification

Figure 6 shows run-times of the racing algorithms method compared with pure falsification algorithm (without the verification part). While reject times are about a third on average for pure falsification, non-reject times are about four times larger than the accept times of the combined algorithm.

The same experiments as for five variables were performed with six variables, but obviously the skeleton-based algorithm was not applied on these problems. Apart from longer run-times of the algorithms, all observation as for five variables were confirmed.

5 Conclusions

We considered the computational aspects of performing CI inference using the method of structural imset, that is, deciding whether a CI statement t follows from an input list L of CI statements in that sense. The existing skeleton-based algorithm [5] that allows inference with up to five variables was improved. We presented an algorithm for creating a constructive proof that t follows from L . Unfortunately, this method does not perform well if t is not implied by L . Fortunately, we can prove t is not implied by L by randomly generating supermodular functions and testing whether the inner product based on L and t is negative. But this method cannot be used to give a conclusive proof that t is implied by L . Together, these methods can race against each other on the same problem.⁴

Empirical evidence suggests the mode of the run-time of the racing algorithms method is an order of magnitude less than the skeleton-based method. Furthermore, the new method also works well for problems with more than five variables, unlike the old one. An analysis of accept times of the new method indicates that the verification algorithm sometimes cannot find the decomposition efficiently. This suggests that it can benefit from further guidance.

⁴ An applet is available at <http://www.cs.waikato.ac.nz/~remco/ci/index.html>

Some questions remain open, in particular finding an upper estimate on $k_{\max}$ (see Section 2.2) for six and more variables. A good upper estimate can decrease the computational effort in proving t is not implied by L .

Though the falsification algorithm cannot give a conclusive proof that an statement t is implied by L , we found that it was often very good at finding all elementary CI statements that are not implied by L in our experiments. This suggests that one can have some confidence that the falsification algorithm can identify statements that are implied by L . Deriving theoretical bounds on the probability that the falsification algorithm actually correctly identifies such statements would be interesting, since this would allow us to quantify our confidence.

References

1. R.R. Bouckaert and M. Studený, Chain graphs: semantics and expressiveness, in Symbolic and Quantitative Approaches to Reasoning and Uncertainty (C. Froidevaux, J. Kohlas eds.), Lecture Notes in AI 946, Springer-Verlag 1995, 67-76.
2. R.G. Cowell, S.L. Lauritzen, A.P. Dawid, D.J. Spiegelhalter, Probabilistic Networks and Expert Systems, Springer-Verlag, New York, 1999.
3. J. Pearl, Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann, San Mateo, 1988.
4. M. Studený, Conditional independence relations have no finite complete characterization, in Information Theory, Statistical Decision Functions and Random Processes vol. B (S. Kubík, J.Á. Věšek eds.), Kluwer, Dordrecht, 1999, 377-396.
5. M. Studený, R.R. Bouckaert, T. Kočka, Extreme supermodular set functions over five variables, research report n. 1977, Institute of Information Theory and Automation, Prague, January 2000.
6. M. Studený, Structural imsets: an algebraic method for describing conditional independence structures, in Proceedings of IPMU 2004 (B. Bouchon-Meunier, G. Coletti, R.R. Yager eds.), 1323-1330.
7. M. Studený, Probabilistic Conditional Independence Structures, Springer-Verlag, London, 2005.
8. T. Verma and J. Pearl, Causal networks: semantics and expressiveness, in Uncertainty in Artificial Intelligence 4 (R.D. Shachter, T.S. Lewitt, L.N. Kanal, J.F. Lemmer eds.), North-Holland, Amsterdam, 1990, 69-76.

Causality, Simpson's Paradox, and Context-Specific Independence

M.J. Sanscartier and E. Neufeld

Department of Computer Science, University of Saskatchewan,
57 Campus Drive, Saskatoon, Saskatchewan, Canada S7K 5A9
mjs696@mail.usask.ca, eric@cs.usask.ca

Abstract. Cognitive psychologist Patricia Cheng suggests that erroneous causal inference is perhaps too often incorrectly attributed to problems with the *process* of inference rather than the *data* on which the inference is carried out. In this paper, we discuss the role of incomplete data in making faulty inferences and where those problems arise. We focus on one of two potential problems in the data we call '*unmeasured-in*' and '*unmeasured-out*' and address a generalization of the causal knowledge in the hope of detecting independencies hidden inside variables, causing the system to behave less than adequately.

The interpretation of the data can be more representative of the problem domain by examining subsets of values for variables in the data. We show how to do this with a generalized form of statistical independence that can resolve relevance problems in the causal model. The most interesting finding is how the examination of contexts can formalize the paradoxical statements in Simpson's paradox and how a simple detection method can eliminate the problem.

1 Introduction

The study of causes and effects in the world is predominant in the aim for a better understanding of human reasoning about everyday events. It is an ongoing quest for genuine causal relationships explaining different phenomena. Esposito et al. [8] state that no genuine causal inference is possible unless we can *cleverly manipulate the variables* in the domain of interest or we are *given all causally relevant factors*. The former is concerned with the *process* of inference, while the latter has to do with the *data*. However, in AI research, the search for a model that can represent and infer causes focuses primarily on the inference engine and pays little attention to the input data on which the inference is carried out. While the AI literature addresses the algorithmic portion of causal induction, cognitive psychologists Cheng and Novik [5] have emphasized the importance in making the distinction between inference problems arising strictly from the mechanism of inference and the integrity of the data under investigation.

It is clear that if the algorithm is not provided the correct data as input, it is impossible to obtain correct output. Thus, on the input data side of the question, the errors that lead to incorrect output are measurement errors. There are two

scenarios where data is unmeasured and therefore incomplete. One scenario is when the relevant information is simply not in the model. We call this scenario ‘*unmeasured-out*’. Alternately, it could be hidden inside a variable, typically by means of an independency that holds in a particular context. We call this scenario ‘*unmeasured-in*’. This arises in the Pearl/SGS treatment [11, 10] where causality is inferred from relations among variables, rather than variables among events.

When relevant independencies lie within variables, erroneous inference is almost inevitable, as we are considering uniformity in a non-uniform set. In the extreme case, that type of error may lead to an instance of *Simpson’s paradox*.

The data problem leading to *Simpson’s paradox* can be approached and formalized with a known independency in Artificial Intelligence (AI), namely, *context-specific independence* (CSI) [1]. Besides formalizing the problem, a simple known detection method [2] can discover such hidden relationships and correct a flawed causal model by dividing it into a set of incrementally more accurate causal models with different topologies depending on the context of variable values.

The remainder of the paper is organized as follows. Section 2 discusses in more detail where the process and data problems arise, and which sub-category of the issue we wish to address. In Section 3, we provide some definitions and terminology relevant to causal models and present an example. In the following section, we discuss Simpson’s paradox and give an example of such an instance in a causal model. In Section 5, we discuss *context-specific independence* (CSI) and show the relationship with the data problem, once again, through an example. We then offer a formal method for accounting for the independencies hidden below the surface. Finally, we use a CSI detection method to construct the refined models, avoiding the data problem altogether.

2 Inference Process Versus Input Data

2.1 Inference Process

On the algorithmic side of things, it is important to have an algorithm capable of determining genuine causation. One such algorithm by Pearl and Verma [13] allows for the discovery of genuine causes in uncontrolled observations and it also provides a mechanism for distinguishing between genuine causal influences and spurious covariations.

The algorithm outputs a graph with four types of links joining nodes. A directed arrow indicates a causal relationship between the two joined variables, while a double-headed arrow indicates a spurious association between two joined variables. Directed arrows can be marked to indicate potential or genuine causation. In other words, the double-headed arrow shows *where* spurious associations can be found, without saying *what* causes the spurious association. Finally, an undirected link between nodes indicates insufficient information to make a conclusion about the nature of the relationship between the variables.

Although the algorithm gives intuition on the causal relationships in the data, it cannot determine *what* the spurious cause is, as it lies outside the set

of available variables. Also, since the algorithm uses *probabilistic conditional independencies* [11] among variables as input, portions of the data containing independencies specific to a subset of the values will be ignored.

2.2 Input Data

As mentioned previously, there are two incomplete data related scenarios, namely 'unmeasured-in' and 'unmeasured-out'. In an 'unmeasured-out' situation, the inference mechanism described above may discover spurious associations in the data. However, the engine can't provide the user with the factor or set of factors that is a common cause to the spurious association: "No causes in, no causes out" [4]. The expert or the user must then decide what common cause could be leading to the spurious association. However, in an 'unmeasured-in' situation, the measurement error can lead to instances of *Simpson's paradox* [15]. We provide a solution to this by considering the context of variables.

3 Causal Models

Several authors express causal models in probabilistic terms because, as argued by Suppes [17], most causal statements in everyday conversation are a reflection of probabilistic and not categorical relations. For that reason, probability theory should provide an adequate framework for reasoning with causal knowledge [9, 14]. Pearl's *causal models* provide the mechanism and structure needed to allow for a representation of causal knowledge based on the presence and absence of *probabilistic conditional independencies* (CIs).

3.1 Definitions and Terminology

Definition 1: A causal model [13] of a set of random variables R can be represented by a directed acyclic graph (DAG), where each node corresponds to an element in R and edges denote direct causal relationships between pairs of elements of R .

The direct causal relations in the causal model can be expressed in terms of *probabilistic conditional independencies* (CIs) [11].

Definition 2: Let $R = \{A_1, A_2, \dots, A_n\}$ denote a finite set of discrete variables, where each variable $A \in R$ takes on values from a finite domain V_A . We use capital letters, such as A, B, C , for variable names and lowercase letters a, b, c to denote outcomes of those variables.

Let X and Y be two disjoint subsets of variables in R and let $Z = R - \{X \cup Y\}$. We say that Y and Z are *conditionally independent* given X , denoted $I(Y, X, Z)$ if, given any $x \in V_x$, $y \in V_y$, then for all $z \in V_z$

$$p(y|x, z) = p(y|x), \text{ whenever } p(x, z) > 0.$$

With the causal model alone, we can express portions of the causal knowledge based on the CIs in the model. The conditional probabilities resulting from the

CI's defined in the model can be formally expressed for all configurations in the Cartesian product of the domains of the variables for which we are storing conditional probabilities.

Definition 3: Let X and Y be two subsets of variables in R such that $p(y) > 0$. We define the *conditional probability distribution* (CPD) of X given $Y = y$ as:

$$p(x|y) = \frac{p(x, y)}{p(y)}, \text{ which implies } p(x, y) = p(y) \cdot p(x|y) \quad (1)$$

for all configurations in $V_x \times V_y$.

Definition 4: A *causal theory* is a pair $T = \langle D, \theta_D \rangle$ consisting of a DAG D along with a set of CPDs θ_D consistent with D . To each variable $A \in R$, there is an attached CPD $p(A_i|Y_i \dots Y_n)$ describing the state of a variable A_i given the state of its parents $Y_i \dots Y_n$.

3.2 Example of Causal Model

The causal model in Fig. 1 describes the causal relationship between the variables $(M)elanoma$, $(S)unscreen$, and $Skin-(T)ype$. According to the DAG, wearing sunscreen has a direct causal influence on the incidence of melanoma, and skin-type has a direct causal influence on wearing sunscreen, and on the incidence of melanoma. The corresponding causal theory attaches to variables M , S , and T respectively the following CPDs:

$$p(M|S, T), p(S|T), \text{ and } p(T). \quad (2)$$

Although the causal model in Fig. 1 seems reasonable and intuitive, a recent study showed that sunscreen users might be at risk of melanoma [7]. In subsequent sections, we show how such erroneous conclusions could faultily penetrate into the system. Although the notion of causation is frequently associated with concepts of necessity and functional dependence, “causal expressions often tolerate exceptions, primarily due to missing variables and coarse descriptions” [13]. As described in Section 2, those exceptions stem from particularities in the data. In the following section, we describe the data problem of *Simpson's paradox* and relate it to this example of a causal model.

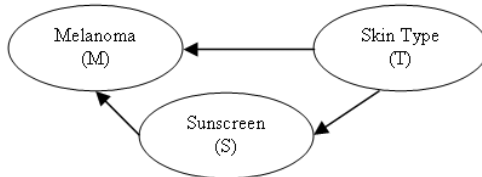


Fig. 1. Causal model describing the causal relationship between use of sunscreen, skin-type, and incidence of melanoma

4 Simpson's Paradox

Simpson [15] makes a point about a particularity of a subset of combinations of fractions that makes intuitively implausible relationships seem mathematically correct.

4.1 Description of Simpson's Reversal of Inequalities

Simpson's paradox occurs when arithmetic inequalities are reversed when we aggregate individual proportions. The result is called *Simpson's reversal of inequalities*. Below is a generalization of the type of expression that results in such reversal:

$$\frac{a_1/b_1 < a_2/b_2}{c_1/d_1 < c_2/d_2} \quad \frac{(a_1 + c_1)/(b_1 + d_1) > (a_2 + c_2)/(b_2 + d_2)}$$

Cohen and Nagel [6] introduce a classic example of Simpson's paradox. They gathered data about death rates from tuberculosis in Richmond, Virginia and New York, New York and found the following propositions held true:

For African Americans, the death rate was *lower* in Richmond than in New York. For Caucasians, the death rate was also *lower* in Richmond than in New York. However, for the total combined population of both African Americans and Caucasians, the death rate was *higher* in Richmond than in New York. However, scrutiny of the data reveals that Caucasians are naturally less likely to get tuberculosis. This is true for Caucasians regardless of whether they live in Richmond or in New York. At the time of the survey, there were more Caucasians than African Americans living in New York, therefore a higher proportion of the New York population was less at risk. The reverse held true for Richmond, which caused the seemingly paradoxical scenario. A complete example in Section 4.2 uses numbers to support such statements.

Cartwright [3] used Simpson's paradox to support claims that causal laws and causal capacities are required by scientific inquiry and by theories of rational choice. As Pearl notes in his survey of the statistical literature on Simpson's paradox, statisticians had an aversion to talk of causal relations and causal inference that was based on the belief that the concept of causation was unsuited to and unnecessary for scientific methods of inquiry and theory construction [12].

In the next subsection, we instantiate the variables from Fig. 1 to show how faulty conclusions and counterintuitive associations can be obtained from mathematically sound equations. We then show how Simpson's paradox can be understood in terms of independencies hidden in specific contexts in the data.

4.2 Example of Erroneous Causal Models Due to Simpson's Paradox

The department of health is attempting to promote the use of sunscreen as a measure to prevent being exposed to the disease melanoma. The promotion encourages both dark-skinned people and light-skinned people to wear sunscreen.

However, statistics gathered from a typical sample of the population, shows some puzzling and questionable results.

For the remainder of this example, we assume the domains of variables $(M)elanoma$, $Skin-(T)ype$, and use of $(S)unscreen$ to be binary. The variables may take on the following sets of values respectively: $\{(y)es, (n)o\}$, $\{(l)ight, (d)ark\}$, and $\{(y)es, (n)o\}$. The numbers here are contrived to illustrate the example.

In the sample set, 50 people with dark skin wore sunscreen and only 10 got melanoma. On the other hand, out of 80 dark-skinned people *not* wearing sunscreen, 20 got melanoma. Of all dark-skinned people in the sample set, 20% of those who wore sunscreen got melanoma, while 25% of those who didn't wear sunscreen were victims of the disease.

In the light-skinned portion of the sample set, out of 80 who wore sunscreen, 60 got melanoma, while 40 out of 50 people who didn't wear sunscreen got sick. In total, 75% of light-skinned people who wore sunscreen got melanoma, while 80% of those who didnt protect their skin were affected.

Yet, altogether 130 people wore sunscreen and 130 people didn't wear sunscreen. Of the 130 people who did in fact wear sunscreen, 70 got melanoma and of the 130 people who didn't wear sunscreen, 60 people got the disease. The percentage of people who did wear sunscreen and still got melanoma is greater than the percentage of people who didn't wear sunscreen and got melanoma. Table 1 shows Simpson's reversal of inequalities in the above example.

This illustration of the problem gives rise to perplexity. How can it be that both dark skin and light skin favor the use of sunscreen and yet overall, *not* wearing sunscreen is better than wearing sunscreen? The sample sizes are equal for both groups, sunscreen (130) and no sunscreen (130), and also for light skin (130) and dark skin (130). In addition, the problem doesn't arise due to small sample size, as it is fairly large and the problem remains for any multiple of the numbers. Also, as we increase the sample size, we only solidify the reversal of inequalities. For a factor of 1 million for example, we can add or remove a fair number from each of the millions and keep Simpson's reversal of inequalities to hold.

The answer to this bewildering example is nothing more than the fact that a greater proportion of the group not wearing sunscreen is naturally less likely to get melanoma. In other words, it is less likely for the dark-skinned person to get melanoma independent of their use of sunscreen. In the example, of the people not wearing sunscreen and getting melanoma, more have dark skin then light skin, and the reverse is true for those who wear sunscreen. Of those with dark

Table 1. Simpson's reversal of inequalities in the *Sunscreen*, *Skin-Type*, and *Melanoma* problem

	Sunscreen		No Sunscreen
Dark Skin	10/50 (20%)	<	20/80 (25%)
Light Skin	60/80 (75%)	<	40/50 (80%)
All Subjects	70/130 ($\approx 53.8\%$)	>	60/130 ($\approx 46.2\%$)

skin, only 30 out of 130 got melanoma, whereas 100 out of 130 light-skinned people got melanoma, where there were more people wearing sunscreen.

More formally, in the context where the skin-type is dark, wearing sunscreen and getting melanoma are independent. We can formalize Simpson’s paradox using *context-specific independence* (CSI) [1].

5 Context-Specific Independence(CSI)

Boutilier et al. [1] formalize the notion of context-specific independence. Without CSI, it is only possible to establish a causal relationship between two variables if a certain set of CIs is absent for all values of a variable in the distribution. With CSI, we can recognize CIs that hold for a subset of values of a variable in a distribution.

5.1 An Independence Holding in Specific Contexts Only

CSI is a CI that holds only in a particular context. Discovery of CSI can help us build more specific causal models instead of a single causal model ignoring particular subsets of values. CSI is defined as follows.

Definition 6: Let X, Y, Z, C be pairwise disjoint subsets of variables in R , and let $c \in V_c$. We say that Y and Z are *conditionally independent* given X in *context* $C = c$ [1], denoted $I_{C=c}(Y, X, Z)$ if,

$$p(y|x, z, c) = p(y|x, c), \text{ whenever } p(x, z, c) > 0.$$

Note that since we are dealing with partial CPDs, a more general operator than the multiplication operator is necessary for manipulating CPDs containing CSIs. This operator, formalized by Zhang and Poole [18] is called the *union-product* operator and we represent it with the symbol $\odot$.

Common sense tells us that wearing sunscreen decreases the incidence of melanoma. Therefore, we expect [7] that there is a negative association between sunscreen and melanoma. An increase in the number of people who wear sunscreen should cause a decrease in the incidence of melanoma. However, data associated with Fig. 1 shows this is not necessarily the case.

However, this seemingly intuitive association is only true when variable *Skin-Type* = *light*. Since the prior likelihood of melanoma for dark skinned people is quite low, it will not make much difference if they wear sunscreen or not. Formally, in the context *Skin-Type* = *dark*, the variables *Sunscreen* and *Melanoma* are independent. If that CSI is not considered, the inference may yield some misleading results. The system behaves very differently for variable *Skin-Type* = *dark* and *Skin-Type* = *light*.

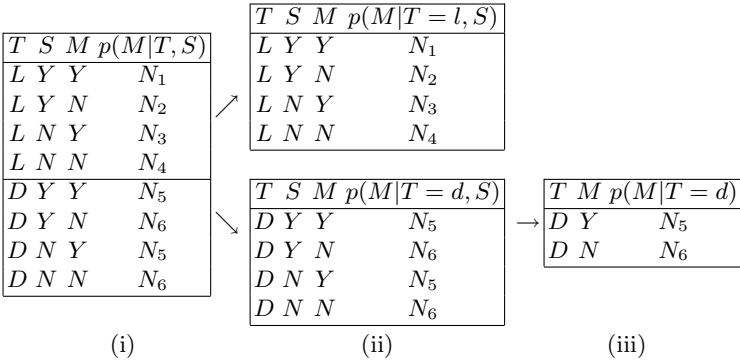
5.2 Formalization with CSI

As we just saw, there are situations where CI is too restrictive to capture independencies that hold only in certain contexts. Although those independencies

Table 2. CPD for $p(M|T, S)$, the probability of *Melanoma* given *Skin-Type* and *Sunscreen*

T	S	M	$p(M T, S)$
L	Y	Y	N_1
L	Y	N	N_2
L	N	Y	N_3
L	N	N	N_4
D	Y	Y	N_5
D	Y	N	N_6
D	N	Y	N_5
D	N	N	N_6

Table 3. CSI decomposition of CPD $p(M|T, S)$ in Table 2



are not visible when all contexts of the data are considered, the presence of independencies that are only true in certain contexts will affect the causal model, and perhaps yield causal links that either do not exist in reality, or are much stronger than what the model shows if context was considered. Also, consideration of CSI may improve causal inference even in cases where the relationships do not result in paradoxical statements.

Consider the following expression, which follows directly from Equation (1)

$$\begin{aligned}
 p(T, S, M) &= p(T) \cdot p(S|T) \cdot p(M|S, T) \\
 &= p(T) \cdot \frac{p(S, T)}{p(T)} \cdot \frac{p(M, S, T)}{p(S, T)}.
 \end{aligned}
 \tag{3}$$

By eliminating common terms in Equation (3), we see that the LHS and the RHS are identical. From the indirect specification of the causal model in Fig. 1, in Equation (2), and in the identity above, it is fair to state that the multiplication of CPDs $p(T)$, $p(S|T)$, and $p(M|S, T)$ define the complete causal model in terms of the available information. However, using CSI, we previously established that given *Skin-Type* = *dark*, variables *Melanoma* and *Sunscreen* are

conditionally independent. The associated CPD is shown in Table 2, and the CSI decomposition for that CPD is presented in Table 3.

Using Zhang and Poole's *union-product* operator for inference with CSI, the CPD $p(M|S, T)$ can be decomposed as follows:

$$\begin{aligned} p(M|S, T) &= p(M|S, T = l) \odot p(M|S, T = d) \\ &= p(M|S, T = l) \odot p(M|T = d) \end{aligned}$$

By substitution, we obtain the following final decomposition of the available causal model.

$$p(T, S, M) = p(T) \cdot p(S|T) \cdot p(M|S, T = l) \odot p(M|T = d)$$

Note that S is not included in the CPD for M when $T = d$.

6 A CSI Detection Method

To eliminate the problem formalized in the previous section, it is possible to detect CSI in the input data and therefore build a set of representative causal models for relevant subsets of the data instead of one causal model based on only CI. One detection method, namely the CPD-Tree algorithm [2] allows for decomposition of the CPDs based on CSI, where the detection is entirely performed from data.

The detection method is straightforward. Initially, we express the CPD as a tree, as in Fig. 2 (left), which is taken from the CPD $p(M|S, T)$. The detection algorithm summarizes as follows:

1. If all children of a node A in the tree are identical, then replace A by one of its offspring.
2. Delete all other children of node A .

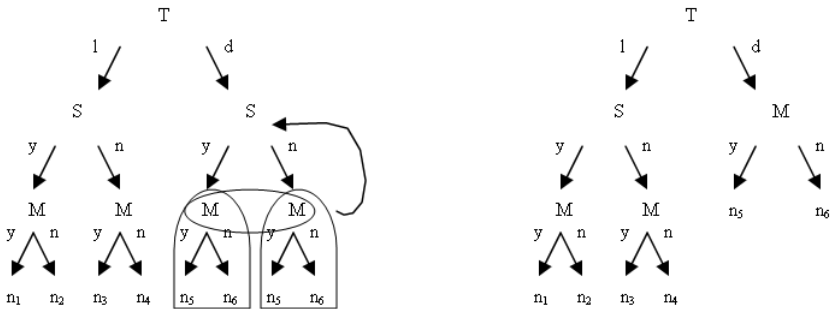


Fig. 2. CPD-Trees for CSI detection from data



Fig. 3. Resulting causal models after CSI detection with CPD-Trees

Fig. 2 (right) shows the tree after CSI detection. The resulting figure, where *Skin-Type* = d , does not mention sunscreen. Given variable *Skin-Type* = d , variables *Melanoma* and *Sunscreen* are conditionally independent. From the now known independencies, the resulting CPDs for $p(M|S, T)$ are the two CPDs in Table 3, and therefore the resulting causal models for the contexts *Skin-Type* = *light* and *Skin-Type* = *dark* respectively are shown in Fig. 3.

In summary, the detection of CSI results into two causal models, each expressing different independencies based on contexts of the data, therefore capturing the problems with the paradoxical data and repairing it with the detection method.

7 Conclusions and Future Work

We showed that statistical inference methods show much promise for improvement of the current state of causal models. We presented a method for formalizing the paradoxical data in Simpson's paradox and for building causal models considering more relevant particularities about the data.

For future work, it would be interesting to see if we can and generalize this formalization using contextual weak independence [16]. Other work by Cheng and Novick shows promise both for assessing judgment of causal models and for providing cognitive validity to such decisions.

References

1. C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in bayesian networks. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 115–123, 1996.
2. C.J. Butz and M.J. Sanscartier. A method for detecting context-specific independence in conditional probability tables. In *Third International Conference on Rough Sets and Current Trends in Computing*, pages 344–348, 2002.
3. N. Cartwright. Causal laws and effective strategies. *Nous*, 13(4):419–437, 1979.
4. N. Cartwright. *Nature, Capacities and their Measurements*. Clarendon Press, Oxford, 1989.
5. P.W. Cheng and L.R. Novick. A probabilistic contrast of causal induction. *Journal of Personality and Social Psychology*, 58:545–567, 1990.
6. M.R. Cohen and E. Nagel. *An Introduction to Logic and Scientific Method*. Brace and Co., New York: Harcourt, 1934.

7. L.K. Dennis, L.F. Beane Freeman, and M.J. Vanbeek. Sunscreen use and the risk for melanoma: a quantitative review. *Annals of Internal Medicine*, 139(12):966–978, 2003.
8. F. Esposito, D. Malerba, and G. Semeraro. Discovering probabilistic causal relationships: A comparison between two methods. *Lecture Notes in Statistics: Selecting Models from Data*, 89, 1994.
9. I.J. Good. A causal calculus. *British Journal for Philosophy of Science*, 11, 1983.
10. Sprites P., Glymour C., and Scheines R. Causation, prediction and search. *Lecture Notes in Statistics*, 81, 1993.
11. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco USA, 1988.
12. J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, USA, 2000.
13. J. Pearl and T.S. Verma. A theory of inferred causation. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pages 441–452. Morgan Kaufmann, 1991.
14. H. Reichenbach. *The Direction of Time*. University of California Press, Berkeley, 1956.
15. E.H. Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society*, 13(B):238–241, 1951.
16. S.K.M.Wong and C.J. Butz. Contextual weak independence in bayesian networks. In *Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 670–679, 1999.
17. P. Suppes. *A Probabilistic Theory of Causation*. North Holland, Amsterdam, 1970.
18. N. Zhang and D. Poole. On the role of context-specific independence in probabilistic reasoning. In *Sixteenth International Joint Conference on Artificial Intelligence*, pages 1288–1293, 1999.

A Qualitative Characterisation of Causal Independence Models Using Boolean Polynomials

Marcel van Gerven, Peter Lucas, and Theo van der Weide

Institute for Computing and Information Sciences,
Radboud University Nijmegen,
Toernooiveld 1, 6525 ED Nijmegen, The Netherlands
{marcelge, peter1, th.p.vanderweide}@cs.ru.nl

Abstract. Causal independence models offer a high level starting point for the design of Bayesian networks but are not maximally exploited as their behaviour is often unclear. One approach is to employ qualitative probabilistic network theory in order to derive a qualitative characterisation of causal independence models. In this paper we exploit polynomial forms of Boolean functions to systematically analyse causal independence models, giving rise to the notion of a polynomial causal independence model. The advantage of the approach is that it allows understanding qualitative probabilistic behaviour in terms of algebraic structure.

1 Introduction

Since the end of the 1980s, Bayesian networks have gained a lot of attention as models for reasoning with uncertainty. A Bayesian network is essentially a graphical specification of independence assumptions underlying a joint probability distribution, allowing for the compact representation of probabilistic information in terms of local probability tables [8]. However, in many cases the amount of probabilistic information required is still too large. The theory of *qualitative probabilistic networks*, CI for short, offers one way to reduce this amount of probabilistic information [4]. Basically, a probability table is specified in terms of a linear number of parameters $P(I_k | C_k)$, as schematically indicated in Fig. 1.a, which are combined by means of a combination function f . A well-known example of a CI model is the noisy OR model, which is employed to model the disjunctive interaction of multiple independent causes of an effect [1, 5].

In principle, the choice of the combination function is free and can be any of the 2^{2^n} possible Boolean functions. Given the attractive nature of the properties of causal independence models, it is regrettable that only few of the possible CI models are used in practice. This is caused by the fact that it is often unclear with what behaviour a particular CI model is endowed. In [7], *qualitative probabilistic networks* (QPN) theory [10] was adopted in order to characterise the behaviour of *qualitative probabilistic networks* CI models [4]. Such a qualitative characterisation may then be matched to the behaviour that is dictated by the domain (Fig. 1.b). In this paper,

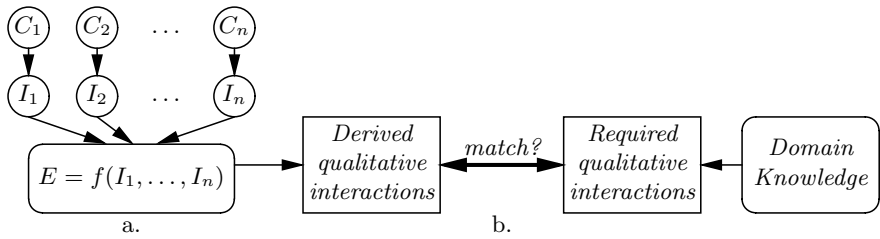


Fig. 1. Comparing the observed qualitative behaviour of a CI model with the desired qualitative behaviour as specified by a domain expert

we provide an alternative, systematic characterisation of Boolean combination functions in terms of their polynomial form. The resulting models are called *qualitative independence models*. On the basis of this canonical representation, a number of important qualitative properties of CI models are derived.

2 Preliminaries

In order to illustrate the theory we introduce a CI model for the domain of medical oncology. Carcinoid tumours synthesise various compounds which leads to a complex symptomatology. Patients may be diagnosed by performing a radioactive scan and can be treated by means of radiotherapy. Patients that are known to have a carcinoid tumour but have a negative radioactive scan (i.e. the tumour does not show up on the scan) will have a decreased probability of survival. This is a counter-intuitive result, which is due to the fact that given a negative radioactive scan, radiotherapy will not be effective. The CI model in Fig. 2 represents this interaction, where I_{Tu} (Tu) denotes whether or not the tumour has been identified during surgery, I_{Sc} (Sc) denotes whether a radioactive scan is positive or negative and I_{Th} (Th) denotes whether radiotherapy was or was not performed. The main task in building a CI model is then to estimate $P(I_{Tu} | Tu)$, $P(I_{Sc} | Sc)$ and $P(I_{Th} | Th)$, and to determine the combination function $f(I_{Tu}, I_{Sc}, I_{Th})$ that models the interaction between these factors with respect to I_{Pr} (Pr), where $Pr = \top$ refers to a good prognosis and $Pr = \perp$ refers to a poor prognosis. We will refer to this example as the *carcinoid tumour model*.

Bayesian networks provide for a concise factorisation of a joint probability distribution over random variables. A *Bayesian network* B is defined as a pair $B = (G, P)$, where G is an acyclic digraph with vertices $V(G)$ and arcs $A(G)$ and P is a joint probability distribution over a set $\mathbf{X}$ of random variables. It is assumed that there is a one-to-one correspondence between the vertices $V(G)$ and the random variables $\mathbf{X}$ such that $P(\mathbf{X})$ factorises according to the structure of the acyclic digraph G . To simplify notation, we will use vertices $V(G)$ and random variables in $\mathbf{X}$ interchangeably, where the interpretation will be clear from context. In this paper it is assumed that all random variables are binary and we use v_i to denote $V_i = \top$ and $\bar{v}_i$ to denote $V_i = \perp$.

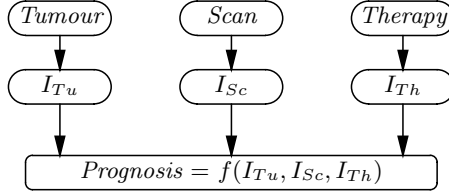


Fig. 2. Prognosis of carcinoid cancer using a CI model

CI is the notion that causes $\mathbf{C}$ are independently contributing to the occurrence of an effect E through some pattern of interaction. As indicated in Fig. 1.a, intermediate variables $\mathbf{I}$ are used not only to connect causal variables $\mathbf{C}$ to the effect variable E , but also in defining the combination function f . In this paper it is assumed that the interaction among causes is represented by means of a function $f: \mathbb{B}^n \rightarrow \mathbb{B}$ over the domain $\mathbb{B} = \{\perp, \top\}$ with $\perp < \top$. We assign Boolean values to a set $\mathbf{S}$ of Boolean variables by means of a valuation v , which is a function $v: \mathbf{S} \rightarrow \mathbb{B}$ assigning either $\top$ or $\perp$ to each variable in $\mathbf{S}$. We use $\sum_{\mathbf{I}} g(\mathbf{I}) = \sum_{(I_1, \dots, I_n) \in \mathbb{B}^n} g(I_1, \dots, I_n)$ to denote a summation over all valuations of $\mathbf{I}$. A CI model is then defined as follows.

Definition 1 (Causal independence model). Let $B = (G, P)$ be a Bayesian network with $V(G) = \mathbf{C} \cup \mathbf{I} \cup \{E\}$, where $\mathbf{C}$ is the set of causes, $\mathbf{I}$ is the set of intermediate variables, and E is the effect. Let $\mathbf{C} \cap \mathbf{I} = \emptyset$ and $E \notin \mathbf{C} \cup \mathbf{I}$. Let $A(G) = \{(C, I_C) \mid C \in \mathbf{C}\} \cup \{(I, E) \mid I \in \mathbf{I}\}$ be the set of arcs in G . Then B is a causal independence (CI) model if there exists a function $f: \mathbb{B}^n \rightarrow \mathbb{B}$, such that

$$P(e \mid \mathbf{C}) = \sum_{\mathbf{I}} f(\mathbf{I}) \prod_{C \in \mathbf{C}} P(I_C \mid C). \quad (1)$$

We use $P[f]$ to denote this probability function and assume that $P(i_C \mid \bar{c}) = 0$ and $P(i_C \mid c) > 0$, where an intermediate variable I_C can be thought to inhibit the occurrence of a cause C whenever $P(i_C \mid c) < 1$.

Qualitative probabilistic networks (QPNs) were introduced by Wellman [10] and are a qualitative abstraction of ordinary Bayesian networks. In the following, let (G, P) be a Bayesian network, let $A, B, C \in V(G)$ represent binary random variables and let (A, C) and (B, C) be arcs in G .

A qualitative influence expresses how the value of one vertex influences the probability of observing values for another vertex. Let X denote $\pi_G(C) \setminus \{A\}$. We say that there is a qualitative influence of A on C if

$$P(c \mid a, x) - P(c \mid \bar{a}, x) \geq 0$$

for all valuations $x \in \mathbb{B}^{|X|}$. Analogously, qualitative influences are defined analogously, replacing $\geq$ by $\leq$ and $=$ respectively. If there are valuations $x, x' \in \mathbb{B}^{|X|}$ such that $P(c \mid a, x) - P(c \mid \bar{a}, x) > 0$ and $P(c \mid a, x') - P(c \mid \bar{a}, x') < 0$ then we say that the qualitative influence is mixed. If none of these cases hold

(i.e. when there is incomplete information about the probability distribution) then we say that the qualitative influence is *additive synergy*.

An additive synergy expresses how the interaction between two variables influences the probability of observing values for a third vertex. Let X denote $\pi_G(C) \setminus \{A, B\}$. There is a *positive additive synergy* of A and B on C if

$$P(c \mid a, b, x) + P(c \mid \bar{a}, \bar{b}, x) - P(c \mid \bar{a}, b, x) - P(c \mid a, \bar{b}, x) \geq 0$$

for all valuations $x \in \mathbb{B}^{|X|}$. Negative, zero, non-monotonic and ambiguous additive synergies are defined analogous to qualitative influences.

A product synergy expresses how upon observation of a common child of two vertices, observing the value of one parent vertex influences the probability of observing a value for the other parent vertex. The original definition of a product synergy is as follows [6]. Let X denote $\pi_G(C) \setminus \{A, B\}$. We say that there is a *positive product synergy* of A and B with regard to the value c_0 of variable C if

$$P(c_0 \mid a, b, x)P(c_0 \mid \bar{a}, \bar{b}, x) - P(c_0 \mid \bar{a}, b, x)P(c_0 \mid a, \bar{b}, x) \geq 0$$

for all valuations $x \in \mathbb{B}^{|X|}$. Again, the other types of product synergies are defined analogous to the corresponding types of qualitative influences. Modifications to product synergies have been made after the observation that this definition is incomplete when parent vertices in X are uninstantiated [2]. However, since we are considering the CI model in isolation; i.e. we assume that a cause C is independent of $\mathbf{C} \setminus \{C\}$, we are entitled to use the original definition of the product synergy in the qualitative analysis of CI models.

In this paper, CI models are analysed by rewriting the combination function in terms of well-formed formulas (wffs) of propositional logic [3]. We will make use of the following concepts. Let b be a Boolean variable. A *literal* refers to b or its negation $\neg b$. In the following we will also write a conjunction of literals as a set of literals $\bigcup_{l \in m} \{l\}$ where we interpret the empty set as $\top$. A *conjunction* of literals $m \equiv \bigwedge_{l \in m} l$ is a conjunction of literals l . Throughout, we will use a *disjunction* of monomials as a set of monomials $\bigcup_{m \in p} \{m\}$ where we interpret the empty set as $\perp$. A *disjunction* of monomials $p \equiv \bigvee_{m \in p} m$ stands for a disjunction of monomials m . We will use the equivalent notation $p = \bigvee_{m \in p} \bigwedge_{l \in m} l \equiv \{\{l_{11}, \dots, l_{1n_1}\}, \dots, \{l_{k1}, \dots, l_{kn_k}\}\}$ to denote a Boolean polynomial. We use m^+ to denote the set of positive literals in m , such that if $l \in m^+$, then $l = b$ and m^- to denote the set of negative literals in m , such that if $l \in m^-$ then $l = \neg b$. Since a monomial may consist of positive and negative literals, we may write $m \equiv \bigwedge_{l \in m^+} l \wedge \bigwedge_{l \in m^-} \neg l$.

The relation between Boolean functions and well-formed formulas is made explicit by the fact that any Boolean function can be realised by a well-formed formula. This is guaranteed by the fact that any Boolean function can be realised by a Boolean polynomial which is in *disjunctive normal form* (DNF) [3]. A Boolean polynomial p is in DNF if every monomial in p contains the same Boolean variables and every two distinct monomials are mutually exclusive. A disadvantage of the disjunctive normal form is that in the worst case, we need to specify 2^n different monomials for an n -ary Boolean function. Therefore, often

the notion of *Boolean polynomial* is employed, where we find a more compact Boolean polynomial p' that is logically equivalent to the disjunctive normal form p of some Boolean function f [9]. In this paper, we will use Boolean functions f and wffs ϕ that realise f interchangeably. Particularly, we will not distinguish between combination functions of CI models that are specified in terms of either f or ϕ , where we assume a bijection $B : \mathbf{C} \rightarrow \mathbf{B}$ between the cause variables $\mathbf{C}$ and the Boolean variables in $\mathbf{B}$, which we abbreviate by b_C . We will use the notion of substitution to write $f_\phi(\mathbf{I})$ more compactly as $\phi(\mathbf{I})$.

Definition 2 (Substitution). $\phi[t_1/x_1, \dots, t_n/x_n]$ is the Boolean polynomial obtained from ϕ by replacing all occurrences of x_i in ϕ by t_i , $1 \leq i \leq n$. We write $\phi(\mathbf{I})$ for $\phi[b_{C_1}/I_{C_1}, \dots, b_{C_n}/I_{C_n}]$, $\mathbf{C} = \{C_1, \dots, C_n\}$.

Consider for instance the carcinoid example. At some point it is postulated that the combination function $f(I_{Tu}, I_{Sc}, I_{Th})$ might be realised by the DNF: $(\neg b_{Tu} \wedge \neg b_{Sc} \wedge \neg b_{Th}) \vee (\neg b_{Tu} \wedge \neg b_{Sc} \wedge b_{Th}) \vee (\neg b_{Tu} \wedge b_{Sc} \wedge b_{Th}) \vee (b_{Tu} \wedge b_{Sc} \wedge b_{Th})$, expressing the background knowledge about the causal mechanism underlying the model. This DNF p is equivalent to the minimal polynomial $p' = (\neg b_{Tu} \wedge \neg b_{Sc}) \vee (b_{Sc} \wedge b_{Th})$. We may then write $p'(i_{Tu}, \bar{i}_{Sc}, i_{Th})$ to denote the substitution of b_{Tu} by $\top$, b_{Sc} by $\perp$ and b_{Th} by $\top$ in p' , which evaluates to $(\perp \wedge \top) \vee (\perp \wedge \top) = \perp$.

3 Polynomial CI Models

In this section, we introduce *polynomial CI models*. These models enable us to zoom in on the characteristics of Boolean functions mediating a CI model. In the next section, we will derive the qualitative properties of these polynomial CI models. We will first prove a number of general properties of CI models. For the sake of readability we will often write $P[\phi]$ instead of $P[\phi](e \mid \mathbf{C})$, and if we state a property of $P[\phi]$ then the property holds for all valuations of $\mathbf{C}$. We list most properties without proof due to space considerations.

Lemma 1. $P[\neg\phi] = 1 - P[\phi]$

Lemma 2. $P[\phi \vee \psi] = 1 - P[\neg\phi \wedge \neg\psi] = P[\phi] + P[\psi] - P[\phi \wedge \psi]$

Lemma 3. $\phi \wedge \psi = \perp \iff P[\phi \vee \psi] = P[\phi + \psi] = P[\phi] + P[\psi]$

Lemma 4. $P[\phi - \psi] = P[\phi] - P[\psi]$

Lemma 5. $P[\phi \wedge \psi] \leq P[\phi]$

In general, we can model the behaviour of an combination function in terms of any equivalent wff using the basis functions $\vee, \wedge$ and $\neg$, but in this paper, we will resort to the use of Boolean polynomials. We will use $l_m(C)$ to refer to a literal in a monomial m that is associated with a cause variable C , where $l_m(C) = b_C$ if $b_C \in m$, $l_m(C) = \neg b_C$ if $\neg b_C \in m$ and $l_m(C) = \top$ otherwise.

We refer to a CI model that employs a Boolean polynomial p as its combination function as a *polynomial CI model*. The probability of observing an effect E given causes $\mathbf{C}$ for such a model is determined by the following proposition.

Proposition 1. *Let p be a polynomial in $\mathcal{P}$. Then, for any $e \in \mathcal{E}$ and $\mathbf{C} \in \mathcal{C}$,*

$$P[p](e | \mathbf{C}) = 1 - \sum_{\mathbf{I} \in m^+} \prod_{m \in p} (1 - \prod_{l \in m^+} l(\mathbf{I}) \prod_{l \in m^-} l(\mathbf{I})) P(\mathbf{I} | \mathbf{C}). \quad (2)$$

By DeMorgan's law, p is equivalent to $\neg \bigwedge_{m \in p} \neg m$. From lemma 1 it then follows that $P[p](e | \mathbf{C}) = P[\neg \bigwedge_{m \in p} \neg m](e | \mathbf{C}) = 1 - P[\bigwedge_{m \in p} \neg m](e | \mathbf{C})$. Due to the analogy between Boolean algebra and ordinary logic we may write $\bigwedge_{m \in p} \neg m$ as $\prod_{m \in p} (1 - m(\mathbf{I}))$. Likewise, and using the equivalence of m and $\bigwedge_{l \in m^+} l \wedge \bigwedge_{l \in m^-} \neg l$ we may write $m(\mathbf{I})$ as $\prod_{l \in m^+} l(\mathbf{I}) \prod_{l \in m^-} \neg l(\mathbf{I})$. By plugging this in into the previous equation we obtain the required result. $\square$

The use of Boolean polynomials instead of Boolean functions is valid since any Boolean function can be realised by a Boolean polynomial in DNF. The properties of the DNF lead to a different form of Equation (2).

Proposition 2. *Let p be a polynomial in $\mathcal{P}$. Then, for any $e \in \mathcal{E}$ and $\mathbf{C} \in \mathcal{C}$,*

Let p be such that in $\forall m, m' \in p: m \neq m' \Rightarrow m \wedge m' \equiv \perp$. Then, according to lemma 3, $P[m_1 \vee \dots \vee m_k](e | \mathbf{C})$ equals $\sum_{m \in p} P[m](e | \mathbf{C})$. $\square$

We may compute the probability that a monomial yields $\top$ given a valuation of the causes $\mathbf{C}$ by

$$P[m](e | \mathbf{C}) = \prod_{l_m(\mathbf{C}) \in m^+} P(i_C | C) \prod_{l_m(\mathbf{C}) \in m^-} P(\bar{i}_C | C). \quad (3)$$

We list the following two properties of polynomial CI models, as they are used in the proof of qualitative properties in the next section.

Proposition 3. *Let B be a polynomial in $\mathcal{B}$. Then, for any $e \in \mathcal{E}$ and $\mathbf{C} \in \mathcal{C}$,*

Let B be a polynomial in $\mathcal{B}$. Then, for any $e \in \mathcal{E}$ and $\mathbf{C} \in \mathcal{C}$,

4 Qualitative Behaviour of Polynomial CI Models

CI models will now be described qualitatively in terms of concepts taken from QPN theory. Note that we can assume that the causes are direct parents of E as the intermediate variables are marginalised out in the final computation of $P[f](e | \mathbf{C})$ (cf. Equation (1)). For our analysis, we assume some fixed CI model over a set $\mathbf{C}$ of n cause variables, in which we focus on the interaction between different cause variables C and C' and the effect variable E , where we abbreviate I_C by I and $I_{C'}$ by I' . Throughout this paper we will use $\mathbf{C}^1$ to denote $\mathbf{C} \setminus \{C\}$ and $\mathbf{C}^2$ to denote $\mathbf{C} \setminus \{C, C'\}$. Likewise, we will use $\mathbf{I}^1$ to denote $\mathbf{I} \setminus \{I\}$ and $\mathbf{I}^2$ to denote $\mathbf{I} \setminus \{I, I'\}$. We use $\mathbf{c}$ to denote a valuation of $\mathbf{C}^1$ or $\mathbf{C}^2$, where the

interpretation will be clear from context. We will also use the notion of a $f_{x_1, \dots, x_k}(\mathbf{x})$ with $x_1, \dots, x_k \in \mathbf{x}$ to denote the function $f(\mathbf{x})$ where x_i is set to v_i for $1 \leq i \leq k$. For example, let I and I' be the intermediate variables as defined above and let $f(I, I')$ be a Boolean function. Then, the curry $f_{\bar{i}}(I')$ is the function $f(\perp, I')$. In the following sections we will analyse the different types of qualitative interactions in CI models. We remark that the listed conditions are sufficient but may not be necessary. We will therefore use the ambiguous category to collect those interactions for which the qualitative behaviour is uncertain.

4.1 Qualitative Influences

A qualitative influence σ_C between a cause C and effect E denotes how the observation of C influences the observation of the effect e . The sign of a qualitative influence for a CI model mediated by f is then determined by the sign of

$$\delta_C(\mathbf{C}^1) = P[f](e \mid c, \mathbf{C}^1) - P[f](e \mid \bar{c}, \mathbf{C}^1) \quad (4)$$

such that there is a σ_C qualitative influence ($\sigma_C = +$) if the sign of $\delta_C(\mathbf{C}^1)$ is zero or positive for every valuation of $\mathbf{C}^1$. $\sigma_C = -$), $\sigma_C = 0$), $\sigma_C = ?$) and $\sigma_C = \sim$) are defined analogously. The analysis requires that we isolate the contribution of a cause variable C with respect to the effect E . By writing

$$P[f](e \mid C, \mathbf{C}^1) = P[f_{\bar{i}}](e \mid \mathbf{C}^1) + P(i \mid C)P[\Delta_C(f)](e \mid \mathbf{C}^1) \quad (5)$$

where $\Delta_C(f)$ denotes the $f_i - f_{\bar{i}}$, we obtain this isolation. Additionally, we isolate the contribution of a variable I to the results of a Boolean function f . To this end, we use the following notation regarding the isolation of one Boolean variable associated with a cause variable C and a polynomial p . $q_C \equiv \{m \setminus \{l_m(C)\} \mid m \in p, l_m(C) \in m^+\}$ represents those monomials where $l_m(C)$ is positive, $q_{\bar{C}} \equiv \{m \setminus \{l_m(C)\} \mid m \in p, l_m(C) \in m^-\}$ represents those monomials where $l_m(C)$ is negative and $q_{\dot{C}} \equiv \{m \mid m \in p, l_m(C) \notin m\}$ represents those monomials where $l_m(C)$ is absent. Let $X \in \{C, \bar{C}, \dot{C}\}$. We use $p_X \equiv \{m \setminus \{l_m(C)\} \mid m \in q_X\}$ to denote q_X from which $l_m(C)$ is removed and $\bar{p}_X \equiv \{m \setminus \{l_m(C)\} \mid m \in p, m \notin q_X\}$ to denote those monomials that do not occur in q_X , where again $l_m(C)$ is removed from the monomials. For instance, in the minimal polynomial $p = (\neg b_{Tu} \wedge \neg b_{Sc}) \vee (b_{Sc} \wedge b_{Th})$ of the carcinoid example we have $p_{Tu} = \{\{-b_{Sc}\}\}$, $p_{Sc} = \{\{b_{Th}\}\}$ and $p_{Th} = \{\{\neg b_{Tu}, \neg b_{Sc}\}\}$. Using this notation, we can decompose a Boolean polynomial p as follows:

$$p(I, \mathbf{I}^1) = ((I \wedge p_C) \vee (\neg I \wedge p_{\bar{C}}) \vee p_{\dot{C}})(\mathbf{I}^1). \quad (6)$$

If we substitute (5) into (4) and under the assumption that $P(i \mid c) > P(i \mid \bar{c})$ we obtain $P[\Delta_C(f)](e \mid \mathbf{C}^1)$ as the specialisation of (4) to qualitative influences in CI models. We may further specialise this to polynomial CI models. The difference $\Delta_C(f)$ is non-zero if either $f_i(\mathbf{I}^1) = \top$ and $f_{\bar{i}}(\mathbf{I}^1) = \perp$ or $f_i(\mathbf{I}^1) = \top$ and $f_{\bar{i}}(\mathbf{I}^1) = \perp$. With the use of (6), this leads to $\Delta_C(f) = (p_C \wedge \neg \bar{p}_C) - (p_{\bar{C}} \wedge \neg \bar{p}_C)$.

Table 1. Determining the qualitative influences for the carcinoid example

Condition	Tumour	Scan	Therapy
1	b_{Sc}	$b_{Tu} \vee b_{Th}$	$\top$
2	$\top$	$\neg b_{Th} \vee \neg b_{Tu}$	$\neg b_{Sc}$
σ_C	$-$	$?$	$+$

Then, using lemma 4, the sign of the qualitative influence for polynomial CI models, is determined by the sign of

$$d_C(\mathbf{C}^1) = P[p_C \wedge \neg \bar{p}_C](e \mid \mathbf{C}^1) - P[p_{\bar{C}} \wedge \neg \bar{p}_{\bar{C}}](e \mid \mathbf{C}^1). \quad (7)$$

Lemma 6 then lists a sufficient condition for observing a positive value of $d_C(\mathbf{C}^1)$.

Lemma 6. $\exists m \in p_C \forall m' \in \bar{p}_C : m^+ \wedge \neg m'^+ \implies \exists c \in \mathbb{B}^{n-1} : d_C(c) > 0$

This follows from the observation that according to lemmas 3 and 5, we can find a valuation of causes such that $P[p_{\bar{C}} \wedge \neg \bar{p}_{\bar{C}}](e \mid c) = 0$, reducing (7) to $P[p_C \wedge \neg \bar{p}_C](e \mid C)$, which is larger then zero for some valuation of causes and intermediate variables. The same reasoning holds for negative values of $d_C(\mathbf{C}^1)$.

Lemma 7. $\exists m \in p_C \forall m' \in \bar{p}_C : m^+ \wedge \neg m'^+ \implies \exists c \in \mathbb{B}^{n-1} : d_C(c) < 0$

We may use Equation (7) to derive the following proposition, characterising the qualitative influences for polynomial CI models.

Proposition 5. $\text{sign}(d_C(\mathbf{C}^1)) = \text{sign}(P[p_C \wedge \neg \bar{p}_C](e \mid \mathbf{C}^1) - P[p_{\bar{C}} \wedge \neg \bar{p}_{\bar{C}}](e \mid \mathbf{C}^1))$

- 1. $p_C \Rightarrow \bar{p}_{\bar{C}} \implies \sigma_C = +$
- 2. $p_C \Rightarrow \bar{p}_C \implies \sigma_C = -$
- 3. $p_C \Rightarrow p_{\bar{C}} \implies \sigma_C = 0$
- 4. $p_C \Rightarrow p_C \implies \sigma_C = \sim$
- 5. $p_C \Rightarrow ? \implies \sigma_C = ?$

We prove just case (1), since case (2) proceeds analogously and the rest follows directly from the definitions of the different types of qualitative influences. Case (1) states that $p_{\bar{C}} \Rightarrow \bar{p}_{\bar{C}}$, which is equal to $\neg p_{\bar{C}} \vee \bar{p}_{\bar{C}}$ or $\neg(p_{\bar{C}} \wedge \neg \bar{p}_{\bar{C}})$. But then (7) reduces to $P[p_C \wedge \neg \bar{p}_C](e \mid \mathbf{C}^1) - P[\perp](e \mid \mathbf{C}^1) \geq 0$, since $P[\perp](e \mid \mathbf{C}^1) = 0$. Therefore, the sign of the qualitative influence is positive.

We illustrate these results with the carcinoid example. Using proposition 5 we can easily determine the signs of the qualitative influences. The conditions of proposition 5 and the outcomes for the clinical variables are listed in Table 1. Recall the conventions that the empty monomial $\emptyset$ is equal to $\top$, whereas the empty polynomial $\emptyset$ is equal to $\perp$. For instance, we determine condition 2 for the clinical variable b_{Sc} by $p_{Tu} \Rightarrow p_{\bar{Tu}} \vee p_{\bar{Th}}$, which is equal to $\perp \Rightarrow \neg b_{Sc} \vee (b_{Sc} \wedge b_{Th})$, or $\top$. Table 1 represents the situations in which a qualitative influence is positive, negative or ambiguous. The results show that observing a tumour has a negative effect on patient prognosis. The qualitative influence

of a scan on prognosis cannot be determined by proposition 5 alone. We may then use lemmas 6 and 7 to determine whether there is a non-monotonicity present. However, the condition $\exists_{m \in p_{Sc}} \forall_{m' \in \bar{p}_{Sc}} : m^+ \wedge \neg m'^+$ does not hold since $b_{Th} \wedge \neg \top = \perp$. This implies that the qualitative influence of a scan on patient prognosis is of the ambiguous type. b_{Th} has a positive qualitative influence on patient prognosis. Note that if the scan is negative then the influence of therapy on prognosis is zero, since a therapy is only fruitful when the scan is positive.

4.2 Additive Synergies

Additive synergies express how two cause variables jointly influence the probability of observing the effect. The additive synergy $\sigma_{C,C'}$ between two causes C and C' is determined by

$$\delta_{C,C'}(\mathbf{C}^2) = P[f](e \mid c, c', \mathbf{C}^2) + P[f](e \mid \bar{c}, \bar{c}', \mathbf{C}^2) - P[f](e \mid \bar{c}, c', \mathbf{C}^2) - P[f](e \mid c, \bar{c}', \mathbf{C}^2) \quad (8)$$

where the different types of additive synergies are defined similarly to the different types of qualitative influences. The analysis requires an isolation of C and C' . We apply the decomposition (5) twice and obtain by straight computation:

$$P[f] = P(i \mid C)P(i' \mid C')P[\Delta_{C,C'}(f)] + P[f_{i,\bar{i}'}] + P(i \mid C)P[\Delta_C(f_{i'})] + P(i' \mid C')P[\Delta_{C'}(f_i)], \quad (9)$$

where the $\Delta_{C,C'}(f) = f_{i,i'} + f_{i,\bar{i}'} - f_{\bar{i},i'} - f_{\bar{i},\bar{i}'}$, can also be expressed as $\Delta_{C'}(f_i) - \Delta_{C'}(f_{\bar{i}})$ or $\Delta_C(f_{i'}) - \Delta_C(f_{\bar{i}'})$. With regard to the analysis of Boolean variables associated with C and C' we introduce the following notation. Let $X \in \{C, \bar{C}, \dot{C}\}$ and $Y \in \{C', \bar{C}', \dot{C}'\}$. Then $p_{X,Y} \equiv (p_X)_Y$ refers to polynomials in which both X and Y are present, $p_{X|Y} \equiv p_{X,Y} \cup p_{\dot{C},Y} \cup p_{X,\dot{C}'}$ refers to polynomials in which both or either of X and Y are present and $p_{X,Y} \equiv p_{X|Y} \cup p_{\dot{C},\dot{C}'}$ refers to polynomials in which both, either or none of X and Y are present. We use $\bar{p}_{X,Y} \equiv \{m \setminus \{l_m(C), l_m(C')\} \mid m \in p, m \notin q_X \cap q_Y\}$ to refer to the complement $q_{X,Y}$ from which literals $l_m(C)$ and $l_m(C')$ are removed. For instance, for the minimal polynomial associated with the running example we have $p_{Tu,Sc} = \{\emptyset\}$, $p_{\bar{T}u|Sc} = \{\{b_{Th}\}\}$, $p_{\bar{S}c;Th} = \{\{\neg b_{Tu}\}\}$ and $\bar{p}_{Tu,Th} = \{\{\neg b_{Sc}\}, \{b_{Sc}\}\}$. Now we can decompose a Boolean polynomial p as follows:

$$p(I, I', \mathbf{I}^2) = ((I \wedge I' \wedge p_{C,C'}) \vee (\neg I \wedge I' \wedge p_{\bar{C},C'}) \vee (I \wedge \neg I' \wedge p_{C;\bar{C}'}) \vee (\neg I \wedge \neg I' \wedge p_{\bar{C};\bar{C}'}))(\mathbf{I}^2). \quad (10)$$

By inserting (9) into (8), and under the assumptions that $P(i \mid c) > P(i \mid \bar{c})$ and $P(i' \mid c') > P(i' \mid \bar{c}')$ we obtain $P[\Delta_{C,C'}(f)](e \mid \mathbf{C}^2)$ for computing the sign of the additive synergy in CI models. In terms of polynomials, we can write $\Delta_{C,C'}(f)$ using (10) as: $p_{C,C'} + p_{\bar{C},\bar{C}'} - p_{\bar{C},C'} - p_{C;\bar{C}'}$. This difference is positive if either $p_1 = p_{C|C'} \wedge p_{\bar{C}|\bar{C}'} \wedge \neg(p_{\bar{C},C'} \wedge p_{C;\bar{C}'})$ or $p_2 = p_{C,C'} \wedge \neg p_{\bar{C},C'}$ or $p_3 = p_{\bar{C},\bar{C}'} \wedge \neg p_{\bar{C},\bar{C}'}$.

hold. The difference is negative if either $p_4 = p_{\bar{C}|C'} \wedge p_{C|\bar{C}'} \wedge \neg(p_{C;C'} \wedge p_{\bar{C};\bar{C}'})$ or $p_5 = p_{\bar{C},C'} \wedge \neg \bar{p}_{\bar{C},C'}$ or $p_6 = p_{C,\bar{C}'} \wedge \neg \bar{p}_{C,\bar{C}'}$ holds. As these cases are mutually exclusive, this results in the following equation:

$$d_{C,C'}(\mathbf{C}^2) = P[p_1](e \mid \mathbf{C}^2) + P[p_2](e \mid \mathbf{C}^2) + P[p_3](e \mid \mathbf{C}^2) - P[p_4](e \mid \mathbf{C}^2) - P[p_5](e \mid \mathbf{C}^2) - P[p_6](e \mid \mathbf{C}^2). \quad (11)$$

We proceed by examining the positive and negative contributions to (11). We use $(U, V) \in \{(C, C'), (\bar{C}, \bar{C}')\}$ and $(X, Y) \in \{(C, C'), (C', C)\}$ in the following.

Lemma 8. $\exists_{\mathbf{c} \in \mathbb{B}^{n-2}} : d_{C,C'}(\mathbf{c}) > 0$

$$\begin{aligned} \exists_{m \in p_{U,V}} \forall_{m' \in \bar{p}_{U,V}} : m^+ \wedge \neg m'^+ \\ \exists_{m_u \in p_{C|C'}, m_v \in p_{C|\bar{C}'}} \forall_{m \in p_{X,Y}} : m_u^+ \wedge m_v^+ \wedge \neg m^+ \end{aligned}$$

This lemma can be proved using the same line of thought as the proof of lemma 6. The second case is just the decomposition of p_1 .

Lemma 9. $\exists_{\mathbf{c} \in \mathbb{B}^{n-2}} : d_{C,C'}(\mathbf{c}) < 0$

$$\begin{aligned} \exists_{m \in p_{X,Y}} \forall_{m' \in \bar{p}_{X,Y}} : m^+ \wedge \neg m'^+ \\ \exists_{m_u \in p_{\bar{C}|C'}, m_v \in p_{C|\bar{C}'}} \forall_{m \in p_{U,V}} : m_u^+ \wedge m_v^+ \wedge \neg m^+ \end{aligned}$$

The characterisation of additive synergies is analogous to that of qualitative influences and follows from Equation (11).

Proposition 6.

$$\begin{aligned} \sigma_{\bar{C},C'} &= + & \sigma_{C,\bar{C}'} &= + & \sigma_{C|C'} \wedge p_{C|\bar{C}'} &= + \\ \sigma_{C,C'} &= + & \sigma_{\bar{C},\bar{C}'} &= + & \sigma_{C|C'} \wedge p_{\bar{C}|\bar{C}'} &= + \\ \sigma_{C,C'} &= - & \sigma_{\bar{C},\bar{C}'} &= - & \sigma_{C|C'} \wedge p_{C|\bar{C}'} &= - \\ \sigma_{C,C'} &= 0 & \sigma_{\bar{C},\bar{C}'} &= 0 & \sigma_{C|C'} \wedge p_{\bar{C}|\bar{C}'} &= 0 \\ \sigma_{C,C'} &= \sim & \sigma_{\bar{C},\bar{C}'} &= \sim & \sigma_{C|C'} \wedge p_{C|\bar{C}'} &= \sim \\ \sigma_{C,C'} &=? & \sigma_{\bar{C},\bar{C}'} &=? & \sigma_{C|C'} \wedge p_{C|\bar{C}'} &=? \end{aligned}$$

We determine the signs of the additive synergies for the carcinoid example using this proposition. $\bar{C}$ and C' are then found to exhibit a positive additive synergy. This is because observing a tumour and a positive scan or not observing a tumour and having a negative scan is in general better for prognosis than observing one of both. A positive additive synergy between C and C' is caused by the fact that they also amplify each other; i.e. a positive scan and the administration of therapy will yield a better prognosis than when either one of both is present. A zero additive synergy between C and $\bar{C}'$ is caused by the fact that b_{Sc} renders both independent; i.e. if a scan is negative, then the prognosis is dependent on C only, whereas if a scan is positive, then the prognosis is dependent on C' only.

4.3 Product Synergies

Product synergies describe the dependence between two causes when the value of the effect variable is observed. The sign $\sigma_{p,q}^e$ of a product synergy between C and C' is determined by

$$\delta_{C,C'}^E(\mathbf{C}^2) = P[f](E \mid c, c', \mathbf{C}^2)P[f](E \mid \bar{c}, \bar{c}', \mathbf{C}^2) - P[f](E \mid \bar{c}, c', \mathbf{C}^2)P[f](E \mid c, \bar{c}', \mathbf{C}^2) \quad (12)$$

where the different types of product synergies are defined similarly to the different types of qualitative influences. For binary variables, $\sigma_{C,C'}^{\bar{e}}$ is fully determined by $\sigma_{C,C'}^e$ and $\sigma_{C,C'}$ through the equation $\delta_{C,C'}^{\bar{e}}(\mathbf{C}^2) = \delta_{C,C'}^e(\mathbf{C}^2) - \delta_{C,C'}(\mathbf{C}^2)$ and we will therefore restrict ourselves to the case where $E = \top$. According to (9) and under the standard assumptions, we can compute the product synergy by:

$$P[\Delta_{C,C'}(f)](e \mid \mathbf{C}^2)P[f_{\bar{i},\bar{i}'}](e \mid \mathbf{C}^2) - P[\Delta_C(f_{i'})](e \mid \mathbf{C}^2)P[\Delta_{C'}(f_{\bar{i}})](e \mid \mathbf{C}^2).$$

As $\Delta_C(f_{i'}) = f_{i,i'} - f_{\bar{i},i'}$, $\Delta_{C'}(f_{\bar{i}}) = f_{\bar{i},i'} - f_{\bar{i},\bar{i}'}$, and $\Delta_{C,C'}(f) = f_{i,i'} + f_{\bar{i},\bar{i}'} - f_{\bar{i},i'} - f_{i,\bar{i}'}$ we can alternatively write this as

$$P[f_{i,i'}](e \mid \mathbf{C}^2)P[f_{\bar{i},\bar{i}'}](e \mid \mathbf{C}^2) - P[f_{\bar{i},i'}](e \mid \mathbf{C}^2)P[f_{i,\bar{i}'}](e \mid \mathbf{C}^2),$$

which, with the use (10), reduces for polynomial CI models to

$$d_{C,C'}^e(\mathbf{C}^2) = P[p_{C,C'}](e \mid \mathbf{C}^2)P[p_{\bar{C},\bar{C}'}](e \mid \mathbf{C}^2) - P[p_{\bar{C},C'}](e \mid \mathbf{C}^2)P[p_{C,\bar{C}'}](e \mid \mathbf{C}^2). \quad (13)$$

Again, we determine conditions for which $d_{C,C'}^e(\mathbf{C}^2)$ is positive or negative. The lemmas follow from (13) and their proof is analogous to that of lemma 6. We use $(X, Y) \in \{(C, C'), (C', C)\}$ in the following.

Lemma 10. $\exists_{\mathbf{c} \in \mathbb{B}^{n-2}} : d_{C,C'}^e(\mathbf{c}) > 0$ $\iff \exists_{m_u \in p_{X,Y}, m_v \in p_{\bar{X},\bar{Y}}} \forall_{m \in p_{\bar{X},Y}} : m_u^+ \wedge m_v^+ \wedge \neg m^+$

$$\exists_{m_u \in p_{\bar{X},Y}, m_v \in p_{\bar{X},\bar{Y}}} \forall_{m \in p_{X,\bar{Y}}} : m_u^+ \wedge m_v^+ \wedge \neg m^+$$

Lemma 11. $\exists_{\mathbf{c} \in \mathbb{B}^{n-2}} : d_{C,C'}^e(\mathbf{c}) < 0$ $\iff \exists_{m_u \in p_{\bar{X},Y}, m_v \in p_{X,\bar{Y}}} \forall_{m \in p_{\bar{X},Y}} : m_u^+ \wedge m_v^+ \wedge \neg m^+$

$$\exists_{m_u \in p_{\bar{X},Y}, m_v \in p_{X,\bar{Y}}} \forall_{m \in p_{\bar{X},\bar{Y}}} : m_u^+ \wedge m_v^+ \wedge \neg m^+$$

The characterisation of product synergies is analogous to that of qualitative influences and additive synergies and follows from Equation (13).

Proposition 7. $\sigma_{C,C'}^e = +$ $\iff p_{\bar{X},\bar{Y}} \vee p_{\bar{X},Y} \Rightarrow p_{X,Y} \text{ and } p_{X,\bar{Y}} \vee p_{\bar{X},\bar{Y}} \Rightarrow p_{\bar{X},\bar{Y}}$

$$\neg p_{U,V} \text{ and } (U, V) \in \{(\bar{C}, C'), (C, \bar{C}')\} \implies \sigma_{C,C'}^e = +$$

$$\sigma_{C,C'}^e = - \iff p_{\bar{X},Y} \vee p_{X,Y} \Rightarrow p_{X,\bar{Y}} \text{ and } p_{\bar{X},\bar{Y}} \vee p_{\bar{X},Y} \Rightarrow p_{\bar{X},Y}$$

$$\neg p_{U,V} \text{ and } (U, V) \in \{(C, C'), (\bar{C}, \bar{C}')\} \implies \sigma_{C,C'}^e = -$$

$$\begin{array}{l} \sigma_{C,C'}^e = 0 \\ \sigma_{C,C'}^e = \sim \\ \sigma_{C,C'}^e = ? \end{array}$$

For product synergies we use proposition 7 to determine the signs of the product synergies for the carcinoid example. We find a positive product synergy between scan and therapy , which is caused by the fact that given a good prognosis, it is more likely that a tumour is accompanied by a positive scan rather than that a tumour is accompanied by a negative scan. The positive product synergy between scan and therapy , is caused by the fact that given a good prognosis, it is more likely that a positive scan is accompanied by therapy rather than that a positive scan is not accompanied by therapy. The positive product synergy between scan and therapy , is caused by the fact that given a good prognosis, it is more likely that the tumour is present and therapy is given rather than that the tumour is present and no therapy is given.

5 Conclusions

In this paper we analysed the qualitative properties of Boolean CI models. Polynomial CI models, where the combination function is rewritten in terms of a Boolean polynomial, were introduced. They enable the analysis of a CI model's qualitative characteristics by examining the structure of the Boolean polynomial. Qualitative influences, additive synergies and product synergies were examined and conditions under which positive, negative, zero, non-monotonic and ambiguous signs are observed were determined. This facilitates the use of CI models in the construction of Bayesian networks since one can determine whether a particular model fulfils a qualitative specification of cause-effect interactions. The carcinoid example illustrated the usefulness of the theory in practice.

References

1. F. J. Díez. Parameter adjustment in Bayes networks. the generalized noisy or-gate. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, 1993. Morgan Kaufmann Publishers.
2. M. J. Druzdzel and M. Henrion. Intercausal reasoning with uninstantiated ancestor nodes. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 317–325. Morgan Kaufmann Publishers, San Mateo, California, 1993.
3. H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, Inc., 1972.
4. D. Heckerman and J. Breese. Causal independence for probability assessment and inference using Bayesian networks. *IEEE, Systems, Man, and Cybernetics*, 26:826–831, 1996.
5. M. Henrion. Some practical issues in constructing belief networks. In *Proceedings of the Third Conference on Uncertainty in Artificial Intelligence*, pages 161–173. Elsevier, Amsterdam, 1989.

6. M. Henrion and M. J. Druzdzel. Qualitative propagation and scenario-based approaches to explanation in probabilistic reasoning. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 17–32, 1991.
7. P.J.F. Lucas. Bayesian network modelling by qualitative patterns. *Artificial Intelligence*, 163:233–263, 2005.
8. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.
9. I. Wegener. *The Complexity of Boolean Functions*. John Wiley & Sons, New York, 1987.
10. M.P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44:257–303, 1990.

On the Notion of Dominance of Fuzzy Choice Functions and Its Application in Multicriteria Decision Making

Irina Georgescu

Turku Centre for Computer Science,
Åbo Akademi University,
Institute for Advanced Management Systems Research,
Lemminkäisenkatu 14, FIN-20520 Turku, Finland
Irina.Georgescu@abo.fi

Abstract. The aim of this paper is twofold: The first objective is to study the degree of dominance of fuzzy choice functions, a notion that generalizes Banerjee's concept of dominance. The second objective is to use the degree of dominance as a tool for solving multicriteria decision making problems. These types of problems describe concrete economic situations where partial information or human subjectivity appears. The mathematical modelling is done by formulating fuzzy choice problems where criteria are represented by fuzzy available sets of alternatives.

1 Introduction

The revealed preference theory was introduced by Samuelson in 1938 [14] in order to express the rational behaviour of a consumer by means of the optimization of an underlying preference relation. The elaboration of the theory in an axiomatic framework was the contribution of Arrow [1], Richter [12], Sen [15] and many others.

Fuzzy preference relations are a topic a vast literature has been dedicated to. Most authors admit that the preferences that appear in social choice are vague (hence modelled through fuzzy binary relations), but the act of choice is exact (hence choice functions are crisp) ([3], [4], [5]). They study crisp choice functions associated with a fuzzy preference relation.

In [2] Banerjee admits the vagueness of the act of choice and studies choice functions with a fuzzy behaviour. The domain of a Banerjee choice function C is made of all non-empty finite subsets of a set of alternatives X and its range is made of non-zero fuzzy subsets of X .

In [8], [9] we have considered choice functions C for which the domain and the range are made of fuzzy subsets of X . Banerjee fuzzifies only the range of a choice function; we use a fuzzification of both the domain and the range of a choice function. In our case, the available sets of alternatives are fuzzy subsets of X . In this way appears the notion of availability degree of an alternative x with respect to an available set S . The availability degree might be useful when

the decision-maker possesses partial information on the alternative x or when a criterion limits the possibility of choosing x . Therefore the available sets can be considered criteria in decision making.

Papers [2], [17] develop a theory of fuzzy revealed preference for a class of fuzzy choice functions. Papers [8], [9] study a larger class of fuzzy choice functions with respect to rationality and revealed preference.

The aim of this paper is to provide a procedure for ranking the alternatives according to fuzzy revealed preference. For this we introduce the degree of dominance of a fuzzy choice function, notion that refines the dominance from [2], [17]. This concept is derived from the fuzzy choice and not from the fuzzy preference. A problem of choice using the formulation of papers [8], [9] can be assimilated to a multicriteria decision problem. The criteria are mathematically modelled by the available sets of alternatives and the degree of dominance offers a hierarchy of alternatives for each criterion.

The paper is organized as follows. Section 2 is concerned with introductory aspects on fuzzy sets and fuzzy relations. Section 3 introduces some basic issues on fuzzy revealed preference. Section 4 recalls the Banerjee's concept of dominance. Section 5 introduces the degree of dominance and the main results around it. Three congruence axioms FC^*1 , FC^*2 and FC^*3 are studied; they extend the congruence axioms $FC1$, $FC2$ and $FC3$ from [2], [17]. A new revealed preference axiom $WAFRP_D$ is formulated and the equivalence $WAFRP_D \Leftrightarrow FC^*1$ is proved. The last section presents a mathematical model for a concrete problem of multicriteria decision making.

2 Preliminaries

In this section we shall recall some properties of the Gödel t-norm and its residuum, as well as some basic definitions on fuzzy sets [6], [10].

Let $[0, 1]$ be the unit interval. For any $a, b \in [0, 1]$ we shall denote $a \vee b = \max(a, b)$; $a \wedge b = \min(a, b)$. More generally, for any $\{a_i\}_{i \in I} \subseteq [0, 1]$ we denote $\bigvee_{i \in I} a_i = \sup\{a_i | i \in I\}$; $\bigwedge_{i \in I} a_i = \inf\{a_i | i \in I\}$.

Then $([0, 1], \vee, \wedge, 0, 1)$ becomes a distributive complete lattice. The binary operation $\wedge$ is a continuous t-norm, called Gödel t-norm [6], [10].

The $\rightarrow$ of the Gödel t-norm $\wedge$ is defined by

$$a \rightarrow b = \bigvee \{c \in [0, 1] | a \wedge c \leq b\} = \begin{cases} 1 & \text{if } a \leq b \\ b & \text{if } a > b \end{cases}$$

The corresponding biresiduum is defined by $a \leftrightarrow b = (a \rightarrow b) \wedge (b \rightarrow a)$.

Let X be a non-empty set. A fuzzy subset of X is a function $A : X \rightarrow [0, 1]$. Denote by $\mathcal{F}(X)$ the family of fuzzy subsets of X . By identifying a (crisp) subset A of X with its characteristic function, the set $\mathcal{P}(X)$ of subsets of X can be considered a subset of $\mathcal{F}(X)$.

A fuzzy subset A of X is normal if $A(x) = 1$ for some $x \in X$. The support of $A \in \mathcal{F}(X)$ is supp

$A = \{x \in X | A(x) > 0\}$. For any $x_1, \dots, x_n \in X$ denote by $[x_1, \dots, x_n]$ the characteristic function of the set $\{x_1, \dots, x_n\}$.

A fuzzy subset R is a fuzzy subset of X^2 , i.e. a function $R : X^2 \rightarrow [0, 1]$; for $x, y \in X$ the real number $R(x, y)$ is the degree of x with respect to y .

If R, Q are two fuzzy preference relations on X then the fuzzy preference relation $R \circ Q$ is the fuzzy preference relation defined by $R \circ Q = \bigvee \{R(x, z) \wedge Q(z, y) | z \in X\}$ for any $x, y \in X$.

If $A, B \in \mathcal{F}(X)$ then we denote

$$I(A, B) = \bigwedge_{x \in X} (A(x) \rightarrow B(x)); E(A, B) = \bigwedge_{x \in X} (A(x) \leftrightarrow B(x)).$$

$I(A, B)$ is called the truth value of A in B and $E(A, B)$ the truth value of A and B . Intuitively $I(A, B)$ expresses the truth value of the statement "A is included in B." and $E(A, B)$ expresses the truth value of the statement "A and B contain the same elements." (see [6]). We remark that $A \subseteq B$ if and only if $I(A, B) = 1$ and $A = B$ if and only if $E(A, B) = 1$.

3 Fuzzy Revealed Preference

Revealed preference is a concept introduced by Samuelson in 1938 [14] in the attempt to postulate the rationality of a consumer's behaviour in terms of a preference relation associated to a demand function. Revealed preferences are patterns that can be inferred indirectly by observing a consumer's behaviour. The consumer reveals by choices his preferences, hence the term revealed preference.

To study fuzzy revealed preferences and fuzzy choice functions associated to them is a natural problem. A vast literature has been dedicated to the case when preferences are fuzzy but the act of choice is exact [3], [4], [5]. In [2] Banerjee lifts this condition putting forth the idea of fuzzy choice functions (see also [16]). We give a short description of Banerjee's framework.

Let X be a non-empty set of alternatives, $\mathcal{H}$ the family of all non-empty finite subsets of X and $\mathcal{F}$ the family of non-zero fuzzy subsets of X with finite support. A Banerjee fuzzy choice function is a function $C : \mathcal{H} \rightarrow \mathcal{F}$ such that $\text{supp } C(S) \subseteq S$ for any $S \in \mathcal{H}$.

According to the previous definition the domain $\mathcal{H}$ of a Banerjee fuzzy choice function is the family of all non-empty finite subsets of X . In [8] and [9] we have developed a theory of fuzzy revealed preferences and fuzzy functions associated to them in an extended form, generalizing Banerjee's.

A fuzzy choice space is a pair $\langle X, \mathcal{B} \rangle$ where X is a non-empty set and $\mathcal{B}$ is a non-empty family of non-zero fuzzy subsets of X . A fuzzy consumer ($=$ fuzzy consumer) on $\langle X, \mathcal{B} \rangle$ is a function $C : \mathcal{B} \rightarrow \mathcal{F}(X)$ such that for each $S \in \mathcal{B}$, $C(S)$ is non-zero and $C(S) \subseteq S$.

Now we introduce the fuzzy revealed preference relation R associated to a fuzzy choice function $C : \mathcal{B} \rightarrow \mathcal{F}(X)$: $R(x, y) = \bigvee_{S \in \mathcal{B}} (C(S)(x) \wedge S(y))$ for any $x, y \in X$.

R is the fuzzy form of the revealed preference relation originally introduced by Samuelson in [14] and studied in an axiomatic framework in [1], [15] etc.

Conversely, to a fuzzy preference relation Q one assigns a fuzzy choice function C defined by $C(S)(x) = S(x) \wedge \bigwedge_{y \in X} [S(y) \rightarrow Q(x, y)]$ for any $S \in \mathcal{B}$ and $x \in X$. $C(S)(x)$ is the degree of truth of the statement "x is one of the Q -greatest alternatives satisfying criterion S ".

4 Banerjee's Concept of Dominance

Banerjee's paper [2] deals with the revealed preference theory for his fuzzy choice functions. He studies three congruence axioms $FC1$, $FC2$, $FC3$. In [17], Wang establishes the connection between $FC1$, $FC2$, $FC3$. These three axioms are formulated in terms of dominance of an alternative x in an available set S of alternatives.

In the literature of fuzzy preference relations there are several ways to define the dominance (see [11]). In general the dominance is related to a fuzzy preference relation [7]. The concept of dominance in [2] is related to the act of choice and is expressed in terms of the fuzzy choice function. For a fuzzy preference relation there exist a lot of ways to define the degree of dominance of an alternative [2], [3], [4], [5], [7], [11].

Let C be a fuzzy choice function, $S \in \mathcal{H}$ and $x \in S$. x is said to be *dominant* in S if $C(S)(y) \leq C(S)(x)$ for any $y \in S$. The dominance of x in S means that x has a higher potentiality of being chosen than the other elements of S . It is obvious that this definition of dominance is related to the act of choice, not to a preference relation.

Banerjee also considers a second type of dominance, associated to a fuzzy preference relation.

Let R be a fuzzy preference relation on X , $S \in \mathcal{H}$ and $x \in X$. x is said to be *relation dominant* in S in terms of R if $R(x, y) \geq R(y, x)$ for all $y \in S$.

Let $S \in \mathcal{H}$, $S = \{x_1, \dots, x_n\}$. The *restriction* of R to S is $R|_S = (R(x_i, x_j))_{n \times n}$. Then we have the *composition* $R|_S \circ C(S) = \bigvee_{j=1}^n (R(x_i, x_j) \wedge C(S)(x_j))$.

In [2] Banerjee introduced the following *axioms* for a fuzzy choice function C :

FC1 For any $S \in \mathcal{H}$ and $x, y \in S$, if y is dominant in S then $C(S)(x) = R(x, y)$.

FC2 For any $S \in \mathcal{H}$ and $x, y \in S$, if y is dominant in S and $R(y, x) \leq R(x, y)$ then x is dominant in S .

FC3 For any $S \in \mathcal{H}$, $\alpha \in (0, 1]$ and $x, y \in S$, $\alpha \leq C(S)(y)$ and $\alpha \leq R(x, y)$ imply $\alpha \leq C(S)(x)$.

In [17], Wang proved that $FC3$ holds iff for any $S \in \mathcal{H}$, $R|_S \circ C(S) \subseteq C(S)$. Then $FC3$ is equivalent with any of the following statements:

- For any $S \in \mathcal{H}$ and $x \in S$, $\bigvee_{y \in S} (R(x, y) \wedge C(S)(y)) \leq C(S)(x)$;
- For any $S \in \mathcal{H}$ and $x, y \in S$, $R(x, y) \wedge C(S)(y) \leq C(S)(x)$.

In [17] it is proved that $FC1$ implies $FC2$, $FC3$ implies $FC2$ and $FC1$, $FC3$ are independent.

Some results from Sect. 5 are based on the following hypotheses:

- (H1) Every $S \in \mathcal{B}$ and $C(S)$ are normal fuzzy subsets of X ;
- (H₂) $\mathcal{B}$ includes all fuzzy sets $[x_1, \dots, x_n]$, $n \geq 1$ and $x_1, \dots, x_n \in X$.

5 Degree of Dominance and Congruence Axioms

In this section we shall define a notion of degree of dominance in the framework of the fuzzy choice functions introduced above. This kind of dominance is attached to a fuzzy choice function and not to a fuzzy preference relation. It shows to what extent, as the result of the act of choice, an alternative has a dominant position among others.

As seen in the previous section, the concept of dominance appears essentially in the expression of congruence axioms $FC1$ - $FC3$. We define now the *degree of dominance* of an alternative x with respect to a fuzzy subset S . This will be a real number that shows the position of x among the other alternatives.

We fix a fuzzy choice function $C : \mathcal{B} \rightarrow \mathcal{F}(X)$.

Definition 1. The *degree of dominance* of x in S is given by

$$\begin{aligned} D_S(x) &= S(x) \wedge \bigwedge_{y \in X} [C(S)(y) \rightarrow C(S)(x)] \\ &= S(x) \wedge [(\bigvee_{y \in X} C(S)(y)) \rightarrow C(S)(x)]. \end{aligned}$$

If $D_S(x) = 1$ then we say that x is *dominant* in S .

Let S be a crisp subset of X . Identifying S with its characteristic function we have the equivalences:

$$\begin{aligned} D_S(x) = 1 \text{ iff } & S(x) = 1 \text{ and } C(S)(y) \leq C(S)(x) \text{ for any } y \in X \\ & \text{iff } x \in S \text{ and } C(S)(y) \leq C(S)(x) \text{ for any } y \in S. \end{aligned}$$

This shows that in this case we obtain exactly the notion of dominance of Banerjee.

In accordance with Definition 1, x is dominant in S iff $S(x) = 1$ and $\bigvee_{y \in X} C(S)(y) = C(S)(x)$.

Assume that C satisfies (H1), i.e. $C(S)(y_0) = 1$ for some $y_0 \in X$. In this case $\bigvee_{y \in X} C(S)(y) = 1$ therefore $D_S(x) = C(S)(x)$.

Lemma 1. $[x, y] \in \mathcal{B}$ $D_{[x, y]}(x) = C([x, y])(y) \rightarrow C([x, y])(x)$

Proposition 1. For any $S \in \mathcal{B}$ and $x, y \in X$,
 • $C(S)(x) \leq D_S(x) \leq S(x)$.
 • $S(x) \wedge D_S(y) \wedge [C(S)(y) \rightarrow C(S)(x)] \leq D_S(x)$

By Proposition 6, $D_S(x) > 0$ for some $x \in X$. Then the assignment $S \mapsto D_S$ is a fuzzy choice function $D : \mathcal{B} \rightarrow \mathcal{F}(X)$. According to Remark 4, if C satisfies (H1) then $C = D$. It implies that the study of the degree of dominance is interesting for the case when hypothesis (H1) does not hold.

For $S \in \mathcal{B}$ and $x \in X$ we define the sequence $(D_S^n(x))_{n \geq 1}$ by induction:

$$D_S^1(x) = D_S(x); D_S^{n+1}(x) = S(x) \wedge \bigwedge_{y \in X} [D_S^n(y) \rightarrow D_S^n(x)].$$

By Proposition 6 (i) we have $C(S)(x) \leq D_S^1(x) \leq \dots \leq D_S^n(x) \leq \dots \leq D_S^\infty(x) \leq S(x)$, where $D_S^\infty(x) = \bigvee_{n=1}^{\infty} D_S^n(x)$. The assignments $S \mapsto D_S^n$, $n \geq 1$ and $S \mapsto D_S^\infty$ provide new fuzzy choice functions.

The following definition generalizes Banerjee's notion of dominant relation in S in terms of R .

Definition 2. For any Q of x in S in terms of Q is defined by

$$D_S^Q(x) = S(x) \wedge \bigwedge_{y \in X} [(S(y) \wedge Q(y, x)) \rightarrow Q(x, y)]$$

If $D_S^Q(x) = 1$ then we say that x is S -dominant in terms of Q .

The congruence axioms $FC1$, $FC2$, $FC3$ play an important role in Banerjee's theory of revealed preference. The formulation of $FC1$, $FC2$ uses the notion of dominance and $FC3$ is a generalization of Weak Congruence Axiom (WCA).

Now we introduce the congruence axioms FC^*1 , FC^*2 , FC^*3 which are refinements of axioms $FC1$, $FC2$, $FC3$. Axioms FC^*1 and FC^*2 are formulated in terms of degree of dominance. FC^*3 is Weak Fuzzy Congruence Axiom ($WFCA$) defined in [8], [9].

FC^*1 For any $S \in \mathcal{B}$ and $x, y \in X$ the following inequality holds:

$$S(x) \wedge D_S(y) \leq R(x, y) \rightarrow C(S)(x).$$

FC^*2 For any $S \in \mathcal{B}$ and $x, y \in X$ the following inequality holds:

$$S(x) \wedge D_S(y) \wedge (R(y, x) \rightarrow R(x, y)) \leq D_S(x).$$

FC^*3 For any $S \in \mathcal{B}$ and $x, y \in X$ the following inequality holds:

$$S(x) \wedge C(S)(y) \wedge R(x, y) \leq C(S)(x).$$

The form FC^*1 is derived from FC^*3 by replacing $D_S(y)$ by $C(S)(y)$. By Remarks 4 and 7, $D_S(x)$ (resp. $D_S(y)$) can be viewed as a substitute of $C(S)(x)$ (resp. $C(S)(y)$).

If hypothesis (H1) holds, then by Remark 4, $D_S(y) = C(S)(y)$ axioms FC^*1 and FC^*3 are equivalent.

Notice that FC^*3 appears under the name $WFCA$ (Weak Fuzzy Congruence Axiom).

Proposition 2. $FC^*1 \Rightarrow FC^*3$

Proposition 3. $FC^*3 \Rightarrow FC^*2$

Proposition 4. $FC^*1 \Rightarrow D_S(x) \leq D_S^R(x), S \in \mathcal{B}, x \in X$

Theorem 1. $FC^*1, FC^*2, FC^*3, H2 \Leftrightarrow C$
 $FC^*1, FC^*2, FC^*3, S \in \mathcal{B}, x \in X$
 $D_S(x) = S(x) \wedge \bigwedge_{y \in X} [S(y) \rightarrow D_{[x,y]}(x)]$

The formulation of axiom FC^*3 has Lemma 2.1 in [17] as starting point. The following result establishes the equivalence of FC^*3 with a direct generalization of $FC3$.

Proposition 5. $FC^*3 \Leftrightarrow$
 $S \in \mathcal{B}, x, y \in X, \alpha \in (0, 1]$
 $S(x) \wedge S(y) \wedge [\alpha \rightarrow C(S)(y)] \wedge [\alpha \rightarrow R(x, y)] \leq \alpha \rightarrow C(S)(x)$

Definition 3. C
 $\langle X, \mathcal{B} \rangle$
 $R_2(x, y) = \bigwedge_{S \in \mathcal{B}} [(S(x) \wedge D_S(y)) \rightarrow C(S)(x)]$

Let C be a fuzzy choice function, $S \in \mathcal{B}$ and $x, y \in X$. By the definition of fuzzy revealed preference R

$$\begin{aligned} R(x, y) \wedge S(x) \wedge D_S(y) &= \left[\bigvee_{T \in \mathcal{B}} (C(T)(x) \wedge T(y)) \right] \wedge S(x) \wedge D_S(y) \\ &= \bigvee_{T \in \mathcal{B}} [S(x) \wedge T(y) \wedge C(T)(x) \wedge D_S(y)]. \end{aligned}$$

Then FC^*1 is equivalent to the following statement

- For any $S, T \in \mathcal{B}$ and $x, y \in X$
 $S(x) \wedge T(y) \wedge C(T)(x) \wedge D_S(y) \leq C(S)(x).$

In [9] the following revealed preference axiom was considered:

$WAFRP^o$ For any $S, T \in \mathcal{B}$ and $x, y \in X$ the following inequality holds:
 $[S(x) \wedge C(T)(x)] \wedge [T(x) \wedge C(S)(x)] \leq E(S \cap C(T), T \cap C(S)).$

In [9] it was proved that $WAFRP^o$ and $FC^*3 = WFCA$ are equivalent.

A problem is if we can find a similar result for condition FC^*1 . In order to obtain an answer to this problem we introduce the following axiom:

$WAFRP_D$ For any $x, y \in X$ and $S, T \in \mathcal{B}$,
 $[S(x) \wedge C(T)(x)] \wedge [T(y) \wedge D_S(y)] \leq I(S \cap C(T), T \cap C(S)).$

Theorem 2. Let $C : \mathcal{B} \rightarrow \mathcal{F}(X)$ be a fuzzy choice function. Then the following conditions are equivalent:

- i. C is FC^*1 .
- ii. $R \subseteq R_2$.
- iii. C is $WAFR_{PD}$.

6 An Application to Multicriteria Decision Making

In making a choice, a set of alternatives and a set of criteria are usually needed.

According to [18], the alternatives and the criteria are defined as follows:

”Alternatives are usually activities, objects, projects, or models of behaviour among which a choice is possible”.

”Criteria are measures, rules and standards that guide decision making. Since decision making is conducted by selecting or formulating different attributes, objectives or goals, all three categories can be referred as criteria. That is, criteria are all those attributes, objectives or goals which have been judged relevant in a given decision situation by a particular decision maker (individual or group)”.

In this section we shall present one possible application of fuzzy revealed preference theory. It represents a model of decision making based on the ranking of alternatives according to fuzzy choices. An agent’s decision is based on the ranking of alternatives according to different criteria. This ranking is obtained by using fuzzy choice problems and the instrument by which it is established is the degree of dominance associated to a fuzzy choice function. In defining this fuzzy choice function the revealed preference theory is applied.

A producer manufactures m types of products $P_1, \dots, P_m$. n companies $x_1, \dots, x_n$ are interested in selling his products. The sales obtained in year T are given in the following table:

	P_1	P_2	$\dots$	P_m
x_1	a_{11}	a_{12}	$\dots$	a_{1m}
x_2	a_{21}	a_{22}	$\dots$	a_{2m}
$\dots$				
x_n	a_{n1}	a_{n2}	$\dots$	a_{nm}

where a_{ij} denotes the number of units of product P_j sold by company x_i in year T . For the year $T + 1$ the producer would like to increase the number of sales with the n companies. The companies give an estimation of the sales for year $T + 1$ contained in a matrix (c_{ij}) with n rows and m columns; c_{ij} denotes the number of units of product P_j that the company x_i estimates to sell in year $T + 1$.

Each product has to be sold by those companies that have an efficient sales market. In choosing these companies, an analysis will require two aspects:

- (a) the sales a_{ij} for year T ;
- (b) the estimated sales c_{ij} for year $T + 1$.

The sales for year T can be considered results of the act of choice, or more clearly, values of a choice function, and the preferences will be given by the revealed preference relation associated to these choice functions. With the resulting preference relation and the estimated sale for the year $T+1$, a fuzzy choice function can be defined. This choice function will be used to rank the companies with respect to each type of product. Dividing the values a_{ij} and c_{ij} respectively by a power of 10 conveniently chosen we may assume that $0 \leq a_{ij}, c_{ij} \leq 1$ for each $i = 1, \dots, n$ and $j = 1, \dots, m$.

In establishing the mathematical model the following steps are needed:

(A) To build a fuzzy choice function from the sales of year T .

The set of alternatives is $X = \{x_1, \dots, x_n\}$.

For each $j = 1, \dots, m$ denote by S_j the subset of X whose elements are those companies that have had "good" sales for product P_j in year T . Only the companies whose sales are greater than a threshold e_j are considered.

If $\mathcal{H} = \{S_1, \dots, S_m\}$ then $\langle X, \mathcal{H} \rangle$ is a fuzzy choice space (we will identify S_j with its characteristic function). The sales (a_{ij}) of year T lead to a choice function $C' : \mathcal{H} \rightarrow \mathcal{F}(X)$ defined by:

$$(1) C'(S_j)(x_i) = a_{ij}$$

for each $j = 1, \dots, m$ and $x_i \in S_j$.

This context is similar to Banerjee [2]. There $\mathcal{H}$ contains all non-empty finite subsets of X .

(B) The choice function C' gives a fuzzy revealed preference relation R on X :

$$(2) R(x_i, x_j) = \bigvee \{C'(S_k)(x_i) | x_i, x_j \in S_k\} = \bigvee \{a_{ik} | x_i, x_j \in S_k\}$$

for any $x_i, x_j \in X$.

$R(x_i, x_j)$ represents the degree to which alternative x_i is preferred to alternative x_j as a consequence of current sales.

Since in most cases R is not reflexive, we replace it by its reflexive closure R' .

(C) From the fuzzy revealed preference matrix R' and the matrix c_{ij} of estimated sales one can define a fuzzy choice function C , whose values will estimate the potential sales for the year $T+1$. Starting from C one will rank the alternatives for each type of product.

The set of alternatives is $X = \{x_1, \dots, x_n\}$. For each $j = 1, \dots, m$ A_j will denote the fuzzy subset of X given by

$$(3) A_j(x_i) = c_{ij} \text{ for any } i = 1, \dots, n.$$

Take $\mathcal{A} = \{A_1, \dots, A_m\}$. One obtains the fuzzy choice space $\langle X, \mathcal{A} \rangle$. The choice function $C : \mathcal{A} \rightarrow \mathcal{F}(X)$ is defined by

$$(4) C(A_j)(x_i) = A_j(x_i) \wedge \bigwedge_{k=1}^n [A_j(x_k) \rightarrow R'(x_i, x_k)]$$

$$= c_{ij} \wedge \bigwedge_{k=1}^n [c_{ik} \rightarrow R'(x_i, x_k)]$$

for any $i = 1, \dots, n$ and $j = 1, \dots, m$.

Applying the degree of dominance for the fuzzy choice function C one will obtain a ranking of the companies with respect to each product. This ranking

gives the information that the mathematical model described above offers to the producer with respect to the sales activity for the following year.

We present next the algorithm of this problem.

The input data are:

m = the number of types of products

n = the number of companies

a_{ij} = the matrix of sales for year T

c_{ij} = the matrix of estimated sales for year $T + 1$

$(e_1, \dots, e_m)$ = the threshold vector

Assume $0 \leq a_{ij} \leq 1$, $0 \leq c_{ij} \leq 1$ for any $i = 1, \dots, n$ and $j = 1, \dots, m$.

From the mathematical model we can derive the following steps:

Step 1. Determine the subsets $S_1, \dots, S_m$ of $X = \{x_1, \dots, x_n\}$ by

$S_k = \{x_i \in X | a_{ik} \geq e_k\}$, $k = 1, \dots, m$.

Step 2. Compute the matrix of revealed preferences $R = (R(x_i, x_j))$ by

$$R(x_i, x_j) = \bigvee_{x_i, x_j \in S_k} a_{ik}.$$

Replace R with its reflexive closure R' .

Step 3. Determine the fuzzy sets $A_1, \dots, A_m$

$A_j = \frac{c_{1j}}{x_1} + \dots + \frac{c_{nj}}{x_n}$ for $j = 1, \dots, m$

Step 4. Obtain the choice function C applying (3)

Step 5. Determine the degrees of dominance $D_{A_j}(x_i)$, $i = 1, \dots, n$ and $j = 1, \dots, m$.

Step 6. Rank the set of alternatives with respect to each product P_j by ranking the set $\{D_{A_j}(x_1), \dots, D_{A_j}(x_n)\}$.

For a better understanding of this model we present a numerical illustration. Consider the initial data $m = 3$ products and $n = 5$ companies willing to sell these products.

The sales for year T are given in the following table:

	P_1	P_2	P_3
x_1	0.3	0.6	0.7
x_2	0.8	0.1	0.5
x_3	0.7	0.6	0.1
x_4	0.1	0.8	0.7
x_5	0.8	0.1	0.7

The estimated sales for year $T + 1$ are given in the following table:

	P_1	P_2	P_3
x_1	0.5	0.7	0.7
x_2	0.8	0.3	0.6
x_3	0.8	0.7	0.2
x_4	0.2	0.8	0.8
x_5	0.8	0.2	0.8

The thresholds are $e_1 = e_2 = e_3 = 0.2$.

We follow now the steps described above.

Step 1. The subsets S_1, S_2, S_3 of X are:

$$S_1 = \{x_1, x_2, x_3, x_5\}, S_2 = \{x_1, x_3, x_4\}, S_3 = \{x_1, x_2, x_4, x_5\}.$$

Step 2. We compute the matrix of revealed preferences R . Then we replace it by its reflexive closure R' .

$$R = \begin{pmatrix} 0.7 & 0.7 & 0.6 & 0.7 & 0.7 \\ 0.8 & 0.8 & 0.8 & 0.5 & 0.8 \\ 0.7 & 0.7 & 0.7 & 0.6 & 0.7 \\ 0.8 & 0.8 & 0.8 & 0.8 & 0.7 \\ 0.8 & 0.8 & 0.8 & 0.7 & 0.8 \end{pmatrix}; R' = \begin{pmatrix} 1 & 0.7 & 0.6 & 0.7 & 0.7 \\ 0.8 & 1 & 0.8 & 0.5 & 0.8 \\ 0.7 & 0.7 & 1 & 0.6 & 0.7 \\ 0.8 & 0.8 & 0.8 & 1 & 0.7 \\ 0.8 & 0.8 & 0.8 & 0.7 & 1 \end{pmatrix}.$$

$$\text{For example, } R(x_1, x_2) = \bigvee_{x_1, x_2 \in S_k} a_{1k} = a_{11} \vee a_{13} = 0.3 \vee 0.7 = 0.7.$$

Step 3. The fuzzy sets A_1, A_2, A_3 are:

$$\begin{aligned} A_1 &= \frac{0.5}{x_1} + \frac{0.8}{x_2} + \frac{0.8}{x_3} + \frac{0.2}{x_4} + \frac{0.8}{x_5}; \\ A_2 &= \frac{0.7}{x_1} + \frac{0.3}{x_2} + \frac{0.7}{x_3} + \frac{0.8}{x_4} + \frac{0.2}{x_5}; \\ A_3 &= \frac{0.7}{x_1} + \frac{0.6}{x_2} + \frac{0.2}{x_3} + \frac{0.8}{x_4} + \frac{0.8}{x_5}. \end{aligned}$$

Step 4. The corresponding fuzzy choice functions are:

$$\begin{aligned} C(A_1)(x) &= \frac{0.5}{x_1} + \frac{0.8}{x_2} + \frac{0.7}{x_3} + \frac{0.2}{x_4} + \frac{0.8}{x_5} \\ C(A_2)(x) &= \frac{0.6}{x_1} + \frac{0.3}{x_2} + \frac{0.6}{x_3} + \frac{0.8}{x_4} + \frac{0.2}{x_5} \\ C(A_3)(x) &= \frac{0.7}{x_1} + \frac{0.5}{x_2} + \frac{0.2}{x_3} + \frac{0.7}{x_4} + \frac{0.7}{x_5}. \end{aligned}$$

Step 5. The corresponding degrees of dominance are represented in the table:

$D_{A_i}(x_i)$	x_1	x_2	x_3	x_4	x_5
A_1	0.5	0.8	0.7	0.2	0.8
A_2	0.6	0.3	0.6	0.8	0.2
A_3	0.7	0.5	0.2	0.8	0.7

The table of degrees of dominance establishes the ranking of alternatives according to each criterion.

According to criterion A_1 ,

$$D_{A_1}(x_4) < D_{A_1}(x_1) < D_{A_1}(x_3) < D_{A_1}(x_2) = D_{A_1}(x_5).$$

According to criterion A_2 ,

$$D_{A_2}(x_5) < D_{A_2}(x_2) < D_{A_2}(x_1) = D_{A_2}(x_3) < D_{A_2}(x_4).$$

According to criterion A_3 ,

$$D_{A_3}(x_3) < D_{A_3}(x_2) < D_{A_3}(x_1) = D_{A_3}(x_5) < D_{A_3}(x_4).$$

7 Concluding Remarks

This paper completes the results of [8], [9]. Our main contribution is to introduce the concept of degree of dominance of an alternative, as a method of ranking

the alternatives according to different criteria. These criteria can be taken as the available sets of alternatives.

The degree of dominance of an alternative x in an available set S of alternatives reflects x 's position towards the other alternatives (with respect to S). This notion expresses the dominance of an alternative with regard to the act of choice, not to a preference relation. With the degree of dominance one can build a hierarchy of alternatives for each available set S . If one defines a concept of aggregated degree of dominance (that unifies the degrees of dominance with regard to various available sets) one obtains an overall hierarchy of alternatives.

References

1. Arrow K.J.: Rational Choice Functions and Orderings. *Economica* **26** (1959) 121-127
2. Banerjee A.: Fuzzy Choice Functions, Revealed Preference and Rationality. *Fuzzy Sets Syst.* **70** (1995) 31-43
3. Barrett C.R., Pattanaik P.K., Salles M.: On the Structure of Fuzzy Social Welfare Functions. *Fuzzy Sets Syst.* **19** (1986) 1-11
4. Barrett C.R., Pattanaik P.K., Salles M.: On Choosing Rationally When Preferences Are Fuzzy. *Fuzzy Sets Syst.* **34** (1990) 197-212
5. Barrett C.R., Pattanaik P.K., Salles M.: Rationality and Aggregation of Preferences in an Ordinal Fuzzy Framework. *Fuzzy Sets Syst.* **49** (1992) 9-13
6. Bělohlávek R.: Fuzzy Relational Systems. Foundations and Principles, Kluwer (2002)
7. Fodor J., Roubens M.: Fuzzy Preference Modelling and Multicriteria Decision Support, Kluwer Academic Publishers, Dordrecht (1994)
8. Georgescu I.: On the Axioms of Revealed Preference in Fuzzy Consumer Theory. *J. Syst. Science Syst. Eng.* **13** (2004) 279-296
9. Georgescu I.: Revealed Preference, Congruence and Rationality. *Fund. Inf.* **65** (2005) 307-328
10. Hájek P.: Metamathematics of Fuzzy Logic. Kluwer (1998)
11. Kulshreshtha P., Shekar B.: Interrelationship among Fuzzy Preference - based Choice Function and Significance of Rationality Conditions: a Taxonomic and Intuitive Perspective. *Fuzzy Sets Syst.* **109** (2000) 429-445
12. Richter M.: Revealed Preference Theory. *Econometrica* **34** (1966) 635-645
13. Richter M.: Rational Choice. In Chipman, J.S. et al. (eds.): Preference, Utility, and Demand. New-York, Harcourt Brace Jovanovich (1971)
14. Samuelson P.A.: A Note on the Pure Theory of Consumers' Behaviour. *Economica* **5** (1938) 61-71
15. Sen A.K.: Choice Functions and Revealed Preference. *Rev. Ec. Studies* **38** (1971) 307-312
16. De Wilde Ph.: Fuzzy Utility and Equilibria. *IEEE Trans. Syst., Man and Cyb.* **34** (2004) 1774-1785
17. Wang X.: A Note on Congruous Conditions of Fuzzy Choice Functions. *Fuzzy Sets Syst.* **145** (2004) 355-358
18. Zeleny M.: Multiple Criteria Decision Making. McGraw-Hill, New York (1982)

An Argumentation-Based Approach to Multiple Criteria Decision

Leila Amgoud, Jean-Francois Bonnefon, and Henri Prade

Institut de Recherche en Informatique de Toulouse (IRIT),
118, route de Narbonne,
31062 Toulouse Cedex 4 France
{amgoud, bonnefon, prade}@irit.fr

Abstract. The paper presents a first tentative work that investigates the interest and the questions raised by the introduction of argumentation capabilities in multiple criteria decision-making. Emphasizing the positive and the negative aspects of possible choices, by means of arguments in favor or against them is valuable to the user of a decision-support system. In agreement with the symbolic character of arguments, the proposed approach remains qualitative in nature and uses a bipolar scale for the assessment of criteria. The paper formalises a multicriteria decision problem within a logical argumentation system. An illustrative example is provided. Various decision principles are considered, whose psychological validity is assessed by an experimental study.

Keywords: Argumentation; multiple-criteria decision, qualitative scales.

1 Introduction

Humans use arguments for supporting claims e.g. [5] or decisions. Indeed, they explain past choices or evaluate potential choices by means of arguments. Each potential choice has usually pros and cons of various strengths. Adopting such an approach in a decision support system would have some obvious benefits. On one hand, not only would the user be provided with a “good” choice, but also with the reasons underlying this recommendation, in a format that is easy to grasp. On the other hand, argumentation-based decision making is more akin with the way humans deliberate and finally make a choice. Indeed, the idea of basing decisions on arguments pro and cons is very old and was already somewhat formally stated by Benjamin Franklin [10] more than two hundreds years ago.

Until recently, there has been almost no attempt at formalizing this idea if we except works by Fox and Parsons [9], Fox and Das [8], Bonnet and Geffner [3] and by Amgoud and Prade [2] in decision under uncertainty. This paper focuses on multiple criteria decision making. In what follows, for each criterion, one assumes that we have a $\langle \mathcal{C}_i, \succ_i \rangle$ ordered scale which enables us to distinguish between $\mathcal{C}_i$ (giving birth to $\mathcal{C}_i$ a choice x) and $\mathcal{C}_i$ (giving birth to $\mathcal{C}_i$ a choice x). Such a scale

has a neutral point, or more generally a neutral area that separates positive and negative values. The lower bound of the scale stands for total dissatisfaction and the upper bound for total satisfaction; the closer to the upper bound the value of criterion c_i for choice x is, the stronger the value of c_i is an argument in favor of x ; the closer to the lower bound the value of criterion c_i for choice x is, the stronger the value of c_i is an argument against x .

In this paper, we propose an argumentation-based framework in which arguments provide the pros and cons of decisions are built from knowledge bases, which may be pervaded with uncertainty. Moreover, the arguments may not have equal forces and this make it possible to compare pairs of arguments. The force of an argument is evaluated in terms of three components: its certainty degree, the importance of the criterion to which it refers, and the (dis)satisfaction level of this criterion. Finally, decisions can be compared, using different principles, on the basis of the strength of their relevant arguments (pros or cons).

The paper is organized as follows. Section 2 states a general framework for argumentation-based decision, and various decision principles. This framework is then instantiated in section 3. Lastly, section 4 reports on the psychological validity of these decision principles.

2 A General Framework for Multiple Criteria Decision

Solving a decision problem amounts to defining a pre-ordering, usually a complete one, on a set $\mathcal{X}$ of possible choices (or decisions), on the basis of the different consequences of each decision. Argumentation can be used for defining such a pre-ordering. The basic idea is to construct arguments in favor of and against each decision, to evaluate such arguments, and finally to apply some principle for comparing the decisions on the basis of the arguments and their quality or strengths. Thus, an argumentation-based decision process can be decomposed into the following steps:

1. Constructing arguments in favor of / against each decision in $\mathcal{X}$.
2. Evaluating the strength of each argument.
3. Comparing decisions on the basis of their arguments.
4. Defining a pre-ordering on $\mathcal{X}$.

2.1 Basic Definitions

Formally, an argumentation-based decision framework is defined as follows:

Definition 1 (Argumentation-based decision framework).

argumentation-based decision framework $\langle \mathcal{X} \mid \mathcal{A} \succeq \langle \text{Princ} \rangle \rangle$ is a tuple

- $\mathcal{X}$, a non-empty set of decisions;
- $\mathcal{A}$, a set of arguments;
- $\succeq$, a pre-ordering on $\mathcal{A}$;
- $\langle \text{Princ} \rangle$, a set of decision principles that take as input a decision $x \in \mathcal{X}$ and return a set of arguments $\mathcal{A}_x$ in $\mathcal{A}$.

output $\langle \mathcal{X} \mid x_1 \triangleleft_{Princ} x_2 \rangle$, where $x_1, x_2 \in \mathcal{X}$ at least as preferred as x_2 by principle $Princ$

Notation: Let A, B be two arguments of $\mathcal{A}$. If $\succeq$ is a pre-order, then $A \succeq B$ means that A is at least as ‘strong’ as B .

$\succ$ and $\approx$ will denote respectively the strict ordering and the relation of equivalence associated with the preference between arguments. Hence, $A \succ B$ means that A is strictly preferred to B . $A \approx B$ means that A is preferred to B and B is preferred to A .

Different definitions of $\succeq$ or different definitions of $\triangleleft_{Princ}$ may lead to different decision frameworks which may not return the same results.

Each decision may have arguments in its favor, and arguments against it. An argument in favor of a decision represents the good consequences of that decision. In a multiple criteria context, this will represent the criteria which are positively satisfied. On the contrary, an argument against a decision may highlight the criteria which are insufficiently satisfied. Thus, in what follows, we define two functions which return for a given set of arguments and a given decision, all the arguments in favor of that decision and all the arguments against it.

Definition 2 (Arguments pros/cons). $x \in \mathcal{X}$

- $Arg_P(x)$: set of arguments in favor of x
- $Arg_C(x)$: set of arguments against x

2.2 Some Principles for Comparing Decisions

At the core of our framework is the use of a principle that allows for an argument-based comparison of decisions. Below we present some intuitive principles $Princ$, whose psychological validity is discussed in section 4. A simple principle consists in counting the arguments in favor of each decision. The idea is to prefer the decision which has more supporting arguments.

Definition 3 (Counting arguments pros: CAP). $\langle \mathcal{X} \mid \mathcal{A} \rangle \succeq_{CAP} \langle \mathcal{X} \mid \mathcal{B} \rangle$ iff $x_1, x_2 \in \mathcal{X}$, $x_1 \triangleleft_{CAP} x_2$ and $CAP, |Arg_P(x_1)| > |Arg_P(x_2)|$ and $|B|$

Likewise, one can also compare the decisions on the basis of the number of arguments against them. A decision which has less arguments against it will be preferred.

Definition 4 (Counting arguments cons: CAC). $\langle \mathcal{X} \mid \mathcal{A} \rangle \succeq_{CAC} \langle \mathcal{X} \mid \mathcal{B} \rangle$ iff $x_1, x_2 \in \mathcal{X}$, $x_1 \triangleleft_{CAC} x_2$ and $CAC, |Arg_C(x_1)| < |Arg_C(x_2)|$

Definitions 3 and 4 do not take into account the strengths of the arguments. In what follows, we propose two principles based on the preference relation between

the arguments. The first one, that we call the *Promotion focus* principle (Prom), takes into account only the supporting arguments (i.e. the arguments PRO a decision), and prefers a decision which has at least one supporting argument which is preferred to (or stronger than) any supporting argument of the other decision. Formally:

Definition 5 (Promotion focus). $\langle \mathcal{X} \mid \mathcal{A} \rangle \succeq_{\text{Prom}} \langle \mathcal{X} \mid \mathcal{B} \rangle$ iff $\exists x_1 \in \mathcal{X} \exists A \in \text{Arg}_P(x_1) \forall B \in \text{Arg}_P(x_2) A \succ B$

Note that the above relation may be found too restrictive, since when the strongest arguments in favor of x_1 and x_2 have equivalent strengths (in the sense of $\approx$), x_1 and x_2 cannot be compared. Clearly, this could be refined in various ways by counting arguments of equal strength.

The second principle, that we call the *Prevention focus* principle (Prev), considers only the arguments against decisions when comparing two decisions. With such a principle, a decision will be preferred when all its cons are weaker than at least one argument against the other decision. Formally:

Definition 6 (Prevention focus). $\langle \mathcal{X} \mid \mathcal{A} \rangle \succeq_{\text{Prev}} \langle \mathcal{X} \mid \mathcal{B} \rangle$ iff $\exists B \in \text{Arg}_C(x_2) \forall A \in \text{Arg}_C(x_1) B \succ A$

Obviously, this is but a sample of the many principles that we may consider. Human deciders may actually use more complicated principles, such as for instance the following one. First, divide the set of all (positive or negative) arguments into strong and weak ones. Then consider only the strong ones if any, and apply the Prevention focus principle. In absence of any strong argument, apply the Promotion focus principle. This combines risk-aversion in the realm of extreme consequences, with risk-tolerance in the realm of mild consequences.

3 A Specification of the General Framework

In this section, we give some definitions of what might be an argument in favor of a decision, an argument against a decision, of the strengths of arguments, and of the preference relations between arguments. We will show also that our framework capture different multiple criteria decision rules.

3.1 Basic Concepts

In what follows, $\mathcal{L}$ denotes a propositional language, $\vdash$ stands for classical inference, and $\equiv$ stands for logical equivalence. The decision maker is supposed to be equipped with three bases built from $\mathcal{L}$:

1. a base $\mathcal{K}$ gathering the available information about the world.
2. a base $\mathcal{C}$ containing the different constraints.
3. a base $\mathcal{G}$ of goals (expressed in terms of goals to be reached).

Beliefs in $\mathcal{K}$ may be more or less certain. In the multiple criteria context, this opens the possibility of having uncertainty on the (dis)satisfaction of the criteria. Such a base is supposed to be equipped with a total preordering $\geq$.

$$a \geq b \text{ iff } a \text{ is at least as certain as } b.$$

For encoding it, we use the set of integers $\{0, 1, \dots, n\}$ as a linearly ordered scale, where n stands for the highest level of certainty and '0' corresponds to the complete lack of information. This means that the base $\mathcal{K}$ is partitioned and stratified into $\mathcal{K}_1, \dots, \mathcal{K}_n$ ($\mathcal{K} = \mathcal{K}_1 \cup \dots \cup \mathcal{K}_n$) such that formulas in $\mathcal{K}_i$ have the same certainty level and are more certain than formulas in $\mathcal{K}_j$ where $j < i$. Moreover, $\mathcal{K}_0$ is not considered since it gathers formulas which are completely not certain.

Similarly, criteria in $\mathcal{C}$ may not have equal importance. The base $\mathcal{C}$ is then also partitioned and stratified into $\mathcal{C}_1, \dots, \mathcal{C}_n$ ($\mathcal{C} = \mathcal{C}_1 \cup \dots \cup \mathcal{C}_n$) such that all criteria in $\mathcal{C}_i$ have the same importance level and are more important than criteria in $\mathcal{C}_j$ where $j < i$. Moreover, $\mathcal{C}_0$ is not considered since it gathers formulas which are completely not important, and which are not at all criteria.

Each criterion can be translated into a set of consequences, which may not be equally satisfactory. Thus, the consequences are associated with the satisfaction level of the corresponding criterion. The criteria may be satisfied either in a positive way (if the satisfaction degree is higher than the neutral point of the considered scale) or in a negative way (if the satisfaction degree is lower than the neutral point of the considered scale). For instance, consider the criterion "closeness to the sea" for a house to let for vacations. If the distance is less than 1 km, the user may be fully satisfied, moderately satisfied if it's between 1 and 2 km, slightly dissatisfied if it is between 2 and 3 km, and completely dissatisfied if it is more than 3km from the sea. Thus, the set of consequences will be partitioned into two subsets: a set of "satisfactory" $\mathcal{G}^+$ and a set of "dissatisfactory" ones $\mathcal{G}^-$.

Since the goals may not be equally satisfactory, the base $\mathcal{G}^+$ (resp. $\mathcal{G}^-$) is also supposed to be stratified into $\mathcal{G}^+ = \mathcal{G}_1^+ \cup \dots \cup \mathcal{G}_n^+$ (resp. $\mathcal{G}^- = \mathcal{G}_1^- \cup \dots \cup \mathcal{G}_n^-$) where goals in $\mathcal{G}_i^+$ (resp. $\mathcal{G}_i^-$) correspond to the same level of (dis)satisfaction and are more important than goals in $\mathcal{G}_j^+$ (resp. $\mathcal{G}_j^-$) where $j < i$. Note that some $\mathcal{G}_i$'s may be empty if there is no goal corresponding to this level of importance. For the sake of simplicity, in all our examples, we only specify the strata which are not empty. In the above example, taking $n = 2$, we have $\mathcal{G}_2^+ = \{dist < 1km\}$, $\mathcal{G}_1^+ = \{1 \leq dist < 2km\}$, $\mathcal{G}_1^- = \{2 \leq dist \leq 3km\}$ and $\mathcal{G}_2^- = \{3 < dist\}$.

A goal g_i^j is associated to a criterion c_i by a propositional formula of the form $g_i^j \rightarrow c_i$ meaning just that the goal g_i^j refers to the evaluation of criterion c_i . Such formulas will be added to $\mathcal{K}_n$. More generally, one may think of goals involving several criteria, e.g. $dist \leq 1km$ or $price \leq 500$.

3.2 Arguments Pros and Cons

An argument $\langle S, x, g, c \rangle$ a decision takes the form of an $\langle S, x, g, c \rangle$. The idea is that a decision has some justification if it leads to the satisfaction of some criteria, taking into account the knowledge. Formally:

Definition 7 (Argument). An argument $\langle S, x, g, c \rangle$ is a tuple $\langle S, x, g, c \rangle$ such that $x \in \mathcal{X}$, $c \in \mathcal{C}$, $S \subseteq \mathcal{K}$, $S^x \vdash g$ and $g \rightarrow c \in \mathcal{K}_n$. S is the support of the argument, x is the conclusion, c is the goal, S^x is the support of the goal, g is the goal, c is the goal, $\mathcal{K}$ is the knowledge base, $\mathcal{X}$ is the set of decisions, $\mathcal{C}$ is the set of goals.

Let's now define the two functions which return the arguments in favor and the arguments against a decision. Intuitively, an argument is in favor of a given decision if that decision satisfies positively a criterion. In other terms, it satisfies goals in $\mathcal{G}^+$. Formally:

Definition 8 (Arguments pros). For $x \in \mathcal{X}$

$$Arg_P(x) = \{A = \langle S, x, g, c \rangle \in \mathcal{A} \mid \exists j \in \{0, 1, \dots, n\} \text{ and } g \in \mathcal{G}_j^+\}.$$

$Sat(A)$ is the satisfaction of the argument A by the decision x .

An argument is against a decision if the decision satisfies insufficiently a given criterion. In other terms, it satisfies goals in $\mathcal{G}^-$. Formally:

Definition 9 (Arguments cons). For $x \in \mathcal{X}$

$$Arg_C(x) = \{A = \langle S, x, g, c \rangle \in \mathcal{A} \mid \exists j \in \{0, 1, \dots, n\} \text{ and } g \in \mathcal{G}_j^-\}.$$

$Dis(A)$ is the dissatisfaction of the argument A by the decision x .

3.3 The Strengths of Arguments

In [1], it has been argued that arguments may have forces of various strengths. These forces allow an agent to compare different arguments in order to select the 'best' ones, and consequently to select the best decisions.

Generally, the force of an argument can rely on the beliefs from which it is constructed. In our work, the beliefs may be more or less certain. This allows us to attach a certainty level to each argument. This certainty level corresponds to the smallest number of a stratum met by the support of that argument.

Moreover, the criteria may not have equal importance also. Since a criterion may be satisfied with different grades, the corresponding goals may have (as already explained) different (dis)satisfaction degree.

Thus, the the force of an argument depends on three components: the support of the argument, the degree of the criterion, and the degree of that criterion. Formally:

Definition 10 (Force of an argument). Let $A = \langle S, x, g, c \rangle$ be an argument. The force of A is defined as $Force(A) = \langle \alpha, \beta, \lambda \rangle$ where

- $\alpha = |\{c_i \mid \alpha_i \leq \alpha, S_j \neq \emptyset\}|$, $S_j = S \cap \mathcal{K}_j$
- $\beta = |\{c_i \mid \beta_i \leq \beta, c \in \mathcal{C}_i\}|$
- $\lambda = Sat(A)$, $A \in Arg_P(x)$ $\lambda = 1$, $Dis(A)$, $A \in Arg_C(x)$ $\lambda = 0$

3.4 Preference Relations Between Arguments

An argumentation system should balance the levels of satisfaction of the criteria with their relative importance. Indeed, for instance, a criterion c_i highly satisfied by x is not a strong argument in favor of x if c_i has little importance. Conversely, a poorly satisfied criterion for x is a strong argument against x only if the criterion is really important. Moreover, in case of uncertain criteria evaluation, one may have to discount arguments based on such evaluation. This is quite similar with the situation in argument-based decision under uncertainty [2]. In other terms, the force of an argument represents to what extent the decision will satisfy the most important criteria.

This suggests the use of a combination of the certainty level, the satisfaction / dissatisfaction degree and the importance of the criterion. This requires the commensurateness of the three scales.

Definition 11 (Conjunctive combination). Let $A = \langle S, x, g, c \rangle$ and $B = \langle T, y, h, d \rangle$ be two arguments. The conjunctive combination of A and B is defined as $Force(A) \cdot \langle \alpha, \beta, \lambda \rangle \cdot Force(B) = \langle \alpha', \beta', \lambda' \rangle$ where

- $\alpha' = \alpha \cdot \beta$
- $\beta' = \beta \cdot \lambda$
- $\lambda' = \lambda \cdot \alpha$

Example 1. Let $A = \langle \{0, 1, 2, 3, 4, 5\}, x, g, c \rangle$ and $B = \langle \{0, 1, 2, 3, 4, 5\}, y, h, d \rangle$ be two arguments. The conjunctive combination of A and B is defined as $Force(A) \cdot \langle \alpha, \beta, \lambda \rangle \cdot Force(B) = \langle \alpha', \beta', \lambda' \rangle$ where

- $\alpha' = \alpha \cdot \beta$
- $\beta' = \beta \cdot \lambda$
- $\lambda' = \lambda \cdot \alpha$

However, a simple conjunctive combination is open to discussion, since it gives an equal weight to the certainty level, the satisfaction/dissatisfaction degree of the criteria and to the importance of the criteria. Indeed, one may prefer an argument that satisfies for sure an important criteria even rather poorly, than an argument which satisfies very well a non-important criterion but with a weak certainty level. This suggests the following preference relation:

Definition 12 (Semi conjunctive combination). Let $A = \langle S, x, g, c \rangle$ and $B = \langle T, y, h, d \rangle$ be two arguments. The semi conjunctive combination of A and B is defined as $Force(A) \cdot \langle \alpha, \beta, \lambda \rangle \cdot Force(B) = \langle \alpha', \beta', \lambda' \rangle$ where

- $\alpha' = \alpha \cdot \beta$
- $\beta' = \beta \cdot \lambda$
- $\lambda' = \lambda \cdot \alpha$

This definition gives priority to the certainty of the information, but is less discriminating than the previous one.

The above approach assumes the commensurateness of two or three scales, namely the certainty scale, the importance scale, and the weighting scale. This requirement is questionable in principle. If this hypothesis is not made, one can still define a relation between arguments as follows:

Definition 13 (Strict combination). $\langle \alpha \ \beta \ \lambda \rangle \prec \langle \alpha' \ \beta' \ \lambda' \rangle$ iff $Force(B) \prec Force(A)$ and

- $\alpha \geq \alpha'$
- $\alpha \prec \alpha' \implies \beta > \beta'$
- $\alpha \prec \alpha' \implies \beta \prec \beta' \implies \lambda > \lambda'$

3.5 Retrieving Classical Multiple Criteria Aggregations

In this section we assume that information in the base $\mathcal{K}$ is fully certain.

A simple approach in multiple criteria decision making amounts to evaluate each x in $\mathcal{X}$ from a set $\mathcal{C}$ of m different criteria c_i with $i = 1, \dots, m$. For each c_i , x is then evaluated by an estimate $c_i(x)$, belonging to the evaluation scale used for c_i . Let 0 denotes the neutral point of the scale, supposed here to be bipolar univariate.

When all criteria have the same level of importance, counting positive or negative arguments obviously corresponds to the respective use of the following evaluation functions for comparing decisions

$$\sum_i c'_i(x) \quad or \quad \sum_i c''_i(x)$$

where $c'_i(x) = 1$ if $c_i(x) > 0$ and $c'_i(x) = 0$ if $c_i(x) < 0$, and $c''_i(x) = 0$ if $c_i(x) > 0$ and $c''_i(x) = 1$ if $c_i(x) < 0$.

Proposition 1. *Let $\langle \mathcal{X}, \mathcal{A} \rangle \models \langle CAP \rangle$. Then, for any $x_1, x_2 \in \mathcal{X}$ and $C_1, C_n \in \mathcal{C}$, $x_1 \models_{CAP} x_2$ implies $\sum_i c'_i(x_1) \geq \sum_i c'_i(x_2)$.*

Proposition 2. *Let $\langle \mathcal{X}, \mathcal{A} \rangle \in \langle \text{CAC} \rangle$. Then, for any $x_1, x_2 \in \mathcal{X}$ and $\mathcal{C} \in \mathcal{C}_n$, $x_1 \in \mathcal{C}$ and $x_2 \notin \mathcal{C}$, $\sum_i c_i''(x_1) \leq \sum_i c_i''(x_2)$.*

When all criteria have the same level of importance, the promotion focus principle amounts to use $\max_i c'_i(x)$ with $c'_i(x) = c_i(x)$ if $c_i(x) > 0$ and $c'_i(x) = 0$ if $c_i(x) < 0$ as an evaluation function for comparing decisions.

Proposition 3. *Let $\langle \mathcal{X}, \mathcal{A} \rangle$ be a $\mathcal{C}_n$ -preorder and let $\langle \mathcal{C}_n, \mathcal{A} \rangle$ be a $\mathcal{C}_n$ -preorder. Then, for all $x_1, x_2 \in \mathcal{X}$ and $c_1, c_2 \in \mathcal{C}_n$, it holds that:*

The prevention focus principle amounts to use $\min_i c_i''(x)$ with $c_i''(x) = 0$ if $c_i(x) > 0$ and $c_i''(x) = -c_i(x)$ if $c_i(x) < 0$.

Proposition 4. *Let $\langle \mathcal{X}, \mathcal{A} \rangle$ be a pre-ordered set, $\langle \text{Prev} \rangle$ a pre-order on $\mathcal{C}$, and $\mathcal{C} = \bigcup_{n \in \mathbb{N}} \mathcal{C}_n$. Then, for $x_1, x_2 \in \mathcal{X}$ and $c_1, c_2 \in \mathcal{C}$, $c_1 \leq_{\text{Prev}} c_2$ implies*

$$\min_{c_1 \in \mathcal{C}_n} c_1''(x_1) \leq \min_{c_2 \in \mathcal{C}_n} c_2''(x_2).$$

When each criterion $c_i(x)$ is associated with a level of importance w_i ranging on the positive part of the criteria scale, the above $c'_i(x)$ is changed into $\min(c'_i(x), w_i)$ in the promotion case.

Proposition 5. $\langle \mathcal{X}, \mathcal{A} \rangle \models \langle \text{Prom} \rangle$ iff $x_1, x_2 \in \mathcal{X}$ and $x_1 \triangleleft_{\text{Prom}} x_2 \iff \max_i \min(c'_i(x_1), w_i) \geq \max_i \min(c'_i(x_2), w_i)$

Similar proposition holds for the prevention focus principle. Thus, weighted disjunctions and conjunctions [7] are retrieved.

3.6 Example: Choosing a Medical Prescription

Imagine we have a set $\mathcal{C}$ of 4 criteria for choosing a medical prescription: Availability (c_1), Reasonableness of the price (c_2), Efficiency (c_3), and Acceptability for the patient (c_4). We suppose that c_1, c_3 are more important than c_2, c_4 . Thus, $\mathcal{C} = \mathcal{C}_2 \cup \mathcal{C}_1$ with $\mathcal{C}_2 = \{c_1, c_3\}$, $\mathcal{C}_1 = \{c_2, c_4\}$.

These criteria are valued on the same qualitative bipolar univariate scale $\{-2, -1, 0, 1, 2\}$ with neutral point 0. From a cognitive psychology point of view, this corresponds to the distinction often made by humans between what is strongly positive, weakly positive, neutral, weakly negative, or strongly negative. Each criterion c_i is associated with a set of 4 goals g_i^j where $j = 2, 1, -1, -2$ denotes the fact of reaching levels 2, 1, -1, -2 respectively. This gives birth to the following goals bases:

$\mathcal{G}^+ = \mathcal{G}_2^+ \cup \mathcal{G}_1^+$ with $\mathcal{G}_2^+ = \{e(x, c_1) = 2, e(x, c_2) = 2, e(x, c_3) = 2, e(x, c_4) = 2\}$, $\mathcal{G}_1^+ = \{e(x, c_1) = 1, e(x, c_2) = 1, e(x, c_3) = 1, e(x, c_4) = 1\}$. $\mathcal{G}^- = \mathcal{G}_2^- \cup \mathcal{G}_1^-$ with $\mathcal{G}_2^- = \{e(x, c_1) = -2, e(x, c_2) = -2, e(x, c_3) = -2, e(x, c_4) = -2\}$, $\mathcal{G}_1^- = \{e(x, c_1) = -1, e(x, c_2) = -1, e(x, c_3) = -1, e(x, c_4) = -1\}$.

Let $\mathcal{X} = \{x_1, x_2\}$ be a set of two potential decisions regarding the prescription of drugs. Suppose that the three alternatives, x_1 and x_2 receive the following evaluation vectors:

- $e(x_1) = (-1, 1, 2, 0)$,
- $e(x_2) = (1, -1, 1, 1)$,

where the i th component of the vector corresponds to the value of the i th criterion. This is encoded in $\mathcal{K}$. All the information in $\mathcal{K}$ are assumed to be fully certain.

$\mathcal{K} = \{e(x_1, c_1) = -1, e(x_1, c_2) = 1, e(x_1, c_3) = 2, e(x_1, c_4) = 0, e(x_2, c_1) = 1, e(x_2, c_2) = -1, e(x_2, c_3) = 1, e(x_2, c_4) = 1, (e(x, c) = y) \rightarrow c\}$. Note that the last formula in $\mathcal{K}$ is universally quantified.

Let's now define the pros and cons each decision.

- $A_1 = \langle \{e(x_1, c_2) = 1\}, x_1, e(x_1, c_2) = 1, c_2 \rangle$
- $A_2 = \langle \{e(x_1, c_3) = 2\}, x_1, e(x_1, c_3) = 2, c_3 \rangle$
- $A_3 = \langle \{e(x_1, c_1) = -1\}, x_1, e(x_1, c_1) = -1, c_1 \rangle$
- $A_4 = \langle \{e(x_1, c_4) = 0\}, x_1, e(x_1, c_4) = 0, c_4 \rangle$
- $A_5 = \langle \{e(x_2, c_1) = 1\}, x_2, e(x_2, c_1) = 1, c_1 \rangle$
- $A_6 = \langle \{e(x_2, c_2) = -1\}, x_2, e(x_2, c_2) = -1, c_2 \rangle$
- $A_7 = \langle \{e(x_2, c_3) = 1\}, x_2, e(x_2, c_3) = 1, c_3 \rangle$
- $A_8 = \langle \{e(x_2, c_4) = 1\}, x_2, e(x_2, c_4) = 1, c_4 \rangle$

$$\begin{aligned} Arg_P(x_1) &= \{A_1, A_2\}, Arg_C(x_1) = \{A_3\}, \\ Arg_P(x_2) &= \{A_5, A_7, A_8\}, Arg_C(x_2) = \{A_6\}. \end{aligned}$$

If we consider an argumentation system in which decisions are compared w.r.t the CAP principle, then $x_2 \triangleleft x_1$. However, if a CAC principle is used, the two decisions are indifferent.

Now let's consider an argumentation system in which a conjunctive combination criterion is used to compare arguments and the Prom principle is used to compare decisions. In that case, only arguments pros are considered.

$Force(A_1) = (2, 1, 1)$, $Force(A_2) = (2, 2, 2)$, $Force(A_5) = (2, 2, 1)$, $Force(A_7) = (2, 2, 1)$, $Force(A_8) = (2, 1, 1)$. It is clear that $A_2 \succ A_5, A_7, A_8$. Thus, x_1 is preferred to x_2 .

In the case of the Prev principle, only arguments against the decisions are considered, namely A_3 and A_6 . Note that $Force(A_3) = (2, 2, 1)$ and $Force(A_6) = (2, 1, 1)$. The two decisions are then indifferent using the conjunctive combination. The leximin refinement of the minimum in the conjunctive combination rule leads to prefer A_3 to A_6 . Consequently, according to Prev principle x_2 will be preferred to x_1 .

This example shows that various Princ may lead to different decisions in case of alternatives hard to separate.

4 Psychological Validity of Argumentation-Based Decision Principles

Bonnefon, Glasspool, McCloy, and Yule [4] have conducted an experimental test of the psychological validity of the counting and Prom/Prev principles for argumentation-based decision. They presented 138 participants with 1 to 3 arguments in favor of some action, alongside with 1 to 3 arguments against the action, and recorded both the decision (take the action, not take the action, impossible to decide) and the confidence with which it was made. Since the decision situation was simplified in that sense that the choice was between taking a given action or not (plus the possibility of remaining undecided), counting arguments pro and counting arguments con predicted similar decisions (because, e.g., an argument for taking the action was also an argument against not taking it). Likewise, and for the same reason, the Prom and Prev principles predicted similar decisions.

The originality of the design was in the way arguments were tailored participant by participant so that the counting principle on the one hand and the Prom and Prev principles on the other hand made different predictions with respect to the participant's decision: During a first experimental phase, participants rated the force of 16 arguments for or against various decisions; a computer program then built online the decision problems that were to be presented in the second experimental phase (i.e., the decision phase proper). For example, the program looked for a set of 1 argument pro and 3 arguments con such that the argument pro was preferred to any of the 3 arguments con. With such a problem, a count-

ing principle would predict the participant to take the action, but a Prom/Prev principle would predict the participant not to take the action.

Overall, 828 decisions were recorded, of which 21% were correctly predicted by the counting principle, and 55% by the Prom/Prev principle. Quite strikingly, the counting principle performed significantly below chance level (33%). The 55% hit rate of the Prom/Prev principle is far more satisfactory, its main problem being its inability to predict decisions made in situations that featured only one argument pro and one argument con, of comparable forces. The measure of the confidence with which decisions were made yielded another interesting result: The decisions that matched the predictions of the Prom/Prev principles were made with higher confidence than the decisions that did not, in a statistically significant way. This last result suggests that the Prom/Prev principle has indeed some degree of psychological validity, as the decisions that conflict with its predictions come with a feeling of doubt, as if they were judged atypical to some extent.

The dataset also allowed for the test of the refined decision principle introduced at the end of section 2.2. This principle fared well regarding both hit rate and confidence attached to the decision. The overall hit rate was 64%, a significant improvement over the 55% hit rate of the Prom/Prev principles. Moreover, the confidence attached to the decisions predicted by the refined principle was much higher (with a mean difference of more than two points on a 5-point scale) than the confidence in decisions it did not predict.

5 Conclusion

Some may wonder why bother about argumentation-based decision in multiple criteria decision problems, since the aggregation functions that can be mimicked in an argumentation-based approach would remain much simpler than sophisticated aggregation functions such as a general Choquet integral. There are several reasons however, for studying argumentation-based multiple criteria decision. A first one is related to the fact that in some problems criteria are intrinsically qualitative, or even if they are numerical in nature they are qualitatively perceived (as in the above example of the criterion 'being close to the sea'), and then it is useful to develop models which are close to the way people deal with decision problems. Moreover, it is also nice to notice that the argumentation-based approach provides a unified setting where inference, or decision under uncertainty can be handled as well. Besides, the logical setting of argumentation-based decision enables to have the values of consequences of possible decisions assessed through a non trivial inference process (in contrast with the above example) from various pieces of knowledge, possibly pervaded with uncertainty, or even partly inconsistent.

The paper has sketched a general method which enables us to $\text{arg}(\text{arg})$ and $\text{arg}(\text{arg})$ preferred decision choices. We have shown that it is possible to design a logical machinery which directly manipulates arguments with their strengths and returns preferred decisions from them.

The approach can be extended in various directions. It is important to study other decision principles which involve the strengths of arguments, and to compare the corresponding decision systems to classical multiple criteria aggregation processes. These principles should be also empirically validated through experimental tests. Moreover, this study can be related to another research trend, illustrated by a companion paper [6], on the axiomatization of particular qualitative decision principles in bipolar settings. Another extension of this work consists of allowing for inconsistent knowledge or goal bases.

References

1. L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *International Journal of Automated Reasoning*, Volume 29, N2:125–169, 2002.
2. L. Amgoud and H. Prade. Using arguments for making decisions. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 10–17, 2004.
3. B. Bonet and H. Geffner. Arguing for decisions: A qualitative model of decision making. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, pages 98–105, 1996.
4. J. F. Bonnefon, D. Glasspool, R. McCloy, , and P. Yule. Qualitative decision making: Competing methods for the aggregation of arguments. *Technical report*, 2005.
5. C. I. Chesñevar, A. G. Maguitman, and R. P. Loui. Logical Models of Argument. *ACM Computing Surveys*, 32(4):337–383, December 2000.
6. D. Dubois and H. Fargier. On the qualitative comparison of sets of positive and negative affects. In *Proceedings of ECSQARU'05*, 2005.
7. D. Dubois and H. Prade. Weighted minimum and maximum operations, an addendum to 'a review of fuzzy set aggregation connectives'. *Information Sciences*, 39:205–210, 1986.
8. J. Fox and S. Das. *Safe and Sound. Artificial Intelligence in Hazardous Applications*. AAAI Press, The MIT Press, 2000.
9. J. Fox and S. Parsons. On using arguments for reasoning about actions and values. In *Proceedings of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, Stanford, 1997.
10. B. Franklin. Letter to j. b. priestley, 1772, in the complete works, j. bigelow, ed.,. *New York: Putnam*, page 522, 1887.

Algorithms for a Nonmonotonic Logic of Preferences

Souhila Kaci¹ and Leendert van der Torre²

¹ Centre de Recherche en Informatique de Lens (C.R.I.L.)—C.N.R.S.,
Rue de l'Université SP 16, 62307 Lens Cedex, France

² CWI Amsterdam and Delft University of Technology, The Netherlands

Abstract. In this paper we introduce and study a nonmonotonic logic to reason about various kinds of preferences. We introduce preference types to choose among these kinds of preferences, based on an agent interpretation. We study ways to calculate “distinguished” preference orders from preferences, and show when these distinguished preference orders are unique. We define algorithms to calculate the distinguished preference orders.

Keywords: logic of preferences, preference logic.

1 Introduction

Preferences guide human decision making from early childhood (e.g., “which ice cream flavor do you prefer?”) up to complex professional and organisational decisions (e.g., “which investment funds to choose?”). Preferences have traditionally been studied in economics and applied to decision making problems. Moreover, the logic of preference has been studied since the sixties as a branch of philosophical logic. Preferences are inherently a multi-disciplinary topic, of interest to economists, computer scientists, OR researchers, mathematicians, logicians, philosophers, and more.

Preferences are a relatively new topic to artificial intelligence and are becoming of greater interest in many areas such as knowledge representation, multi-agent systems, constraint satisfaction, decision making, and decision-theoretic planning. Recent work in AI and related fields has led to new types of preference models and new problems for applying preference structures [1]. Explicit preference modeling provides a declarative way to choose among alternatives, whether these are solutions of problems to solve, answers of data-base queries, decisions of a computational agent, plans of a robot, and so on. Preference-based systems allow finer-grained control over computation and new ways of interactivity, and therefore provide more satisfactory results and outcomes. Logics of preference are used to compactly represent and reason about preference relations.

A particularly challenging topic in preference logic is concerned with non-monotonic reasoning about preferences. A few constructs have been proposed [6, 14, 11], for example based on mechanisms developed in non-monotonic reasoning such as gravitation towards the ideal, or compactness, but there is no consensus yet in this area. Nevertheless, non-monotonic reasoning about preferences is an important issue, for example when reasoning under uncertainty. When an agent compactly communicates its preferences, another agent has to interpret it and find the most likely interpretation.

A drawback of the present state of the art in the logic of preference is that proposed logics typically formalize only preferences of one kind, formalizing for example strong preferences, defeasible preferences, non-strict preferences, *ceteris paribus* preferences (interpreted either as “all else being equal” or as “under similar circumstances”), etc. These logics formalize logical relations among one kind of preferences, but relations among distinct kinds of preferences have not been considered. Consequently, when formalizing preferences, one has to choose which kind of preference statements are used for all preferences under consideration. However, often we would like to use several kinds of preference statements at the same time.

We are interested in developing and using a logic with more than one kind of preferences, which we call a logic of preferences – in contrast to the usual reference to the logic of preference. In particular we are interested in nonmonotonic logic of preferences. To interpret the various kinds of preferences we use total pre-orders on worlds, which we call preference orders. We consider the following questions:

1. How to define a logic of preferences to reason about for example strong and weak preferences? How are they related to conditional logics?
2. How to choose among kinds of preferences when formalizing examples?
3. How to calculate “distinguished” preference orders from preferences? Are the distinguished preference orders unique?
4. How can we define algorithms to calculate the distinguished preference orders?

To define our logic of preferences, we define four kinds of strict preferences of p over q as “the best/worst p is preferred over the best/worst q ”. We define conditionals “if p , then q ” as usual as a preference of p and q over p and the absence of q .

To choose among kinds of preferences, we introduce an agent interpretation of the four kinds of preferences studied in this paper. We interpret a preference of p over q as a game between an agent arguing for p and an agent arguing for q . We distinguish locally optimistic, pessimistic, opportunistic and careful preference types.

To calculate a preference order from preferences, we start from a generalization of System Z, which is usually characterized as gravitating towards the ideal for defeasible conditionals, and also known as minimal specificity. We also define the inverse of gravitating towards the worst. In general we need to combine both kinds of mechanisms, for which we study a strict dominance of one of the mechanisms. We provide new algorithms to derive distinguished orders.

The layout of this paper is as follows. We treat each question above mentioned in a subsequent section. Section 2 introduces the logic of preferences we use in this paper. Section 3 introduces the preference types. Section 4 introduces the non-monotonic extensions to define distinguished preference orders. Section 5 introduces algorithms to calculate distinguished preference orders.

2 Logic of Preferences

The logical language extends propositional logic with four kinds of preferences. A small m stands for min and a capital M stands for max, as will be explained in the semantics below.

Definition 1 (Language). Given a set $A = \{a_1, \dots, a_n\}$ of propositional atoms, we define the set L_0 of propositional formulas and the set L of preference formulas as follows.

$$\begin{aligned} L_0 \ni p, q: & a_i \mid (p \wedge q) \mid \neg p \\ L \ni \phi, \psi: & p \succ^m q \mid p \succ^M q \mid p \succ^m q \mid p \succ^M q \mid \neg\phi \mid (\phi \wedge \psi) \end{aligned}$$

Disjunction $\vee$, material implication $\supset$ and equivalence $\leftrightarrow$ are defined as usual. Moreover, we define conditionals in terms of preferences by $p \xrightarrow{m} q =_{\text{def}} p \wedge q \succ^m p \wedge \neg q$, etc. We abbreviate formulas using the following order on logical connectives: $\neg \mid \vee, \wedge \mid \supset, \leftrightarrow$. For example, $p \vee q > r \supset s$ is interpreted as $((p \vee q) > r) \supset s$.

In the semantics of the four kinds of preferences, a preference of p over q is interpreted as a preference of $p \wedge \neg q$ over $q \wedge \neg p$. This is standard and known as von Wright's expansion principle [16].

Definition 2 (Semantics). Let A be a finite set of propositional atoms, L a propositional logic based on A , W the set of propositional interpretations of L , and $\succeq$ a total pre-order on W . We write $w \succ w'$ for $w \succeq w'$ without $w' \succeq w$, we write $\max(p, \succeq)$ for $\{w \in W \mid w \models p, \forall w' \in W : w' \models p \Rightarrow w \succeq w'\}$, and we write $\min(p, \succeq)$ for $\{w \in W \mid w \models p, \forall w' \in W : w' \models p \Rightarrow w' \succeq w\}$.

$$\begin{aligned} \models p \succ^m q & \text{ iff } \forall w \in \min(p \wedge \neg q, \succeq) \text{ and } \forall w' \in \min(\neg p \wedge q, \succeq) \text{ we have } w \succ w' \\ \models p \succ^M q & \text{ iff } \forall w \in \min(p \wedge \neg q, \succeq) \text{ and } \forall w' \in \max(\neg p \wedge q, \succeq) \text{ we have } w \succ w' \\ \models p \succ^m q & \text{ iff } \forall w \in \max(p \wedge \neg q, \succeq) \text{ and } \forall w' \in \min(\neg p \wedge q, \succeq) \text{ we have } w \succ w' \\ \models p \succ^M q & \text{ iff } \forall w \in \max(p \wedge \neg q, \succeq) \text{ and } \forall w' \in \max(\neg p \wedge q, \succeq) \text{ we have } w \succ w' \end{aligned}$$

Moreover, logical notions are defined as usual, in particular:

- $\models \{\phi_1, \dots, \phi_n\}$ iff $\models \phi_i$ for $1 \leq i \leq n$,
- $\models \phi$ iff for all $\succeq$, we have $\models \phi$,
- $S \models \phi$ iff for all $\succeq$ such that $\models S$, we have $S \models \phi$.

The $\succ^m$'s preference is the strongest one while $\succ^M$'s preference is the weakest one [15]. The following example illustrates the logic of preferences.

Example 1. We have $\models p \succ^M q \leftrightarrow (p \wedge \neg q) \vee (\neg p \wedge q) \xrightarrow{M} p$, which expresses a well-known relation between a defeasible conditional $\xrightarrow{M}$ and preferences $\succ^M$. Moreover, we have $\models p \succ^m q \supset p \succ^M q$, which expresses that strong preferences $\succ^m$ imply defeasible preferences $\succ^M$.

The following definition illustrates how a preference order – represented in a *qualitative* form by a total pre-order $\succeq$ on worlds – can also be represented by a well ordered partition of W . This is an equivalent representation, in the sense that each preference order corresponds to one ordered partition and vice versa. This equivalent representation as an ordered partition makes some definitions easier to read.

Definition 3 (Ordered partition). A sequence of sets of worlds of the form $(E_1, \dots, E_n)$ is an ordered partition of W iff $\forall i$, E_i is nonempty, $E_1 \cup \dots \cup E_n = W$ and $\forall i, j$, $E_i \cap E_j = \emptyset$ for $i \neq j$. An ordered partition of W is associated with pre-order $\succeq$ on W iff $\forall \omega, \omega' \in W$ with $\omega \in E_i, \omega' \in E_j$ we have $i \leq j$ iff $\omega \succeq \omega'$.

3 Preference Types as Agent Types

The logic of preferences now forces us to choose among the four kinds of preferences when we formalize an example in the logic. From the literature it is only known how to choose among monopolar preferences such as “I prefer p ”, or more involved “Ideally p ”, “ p is my goal”, “I desire p ”, “I intend p ”, etc. In such cases we can distinguish two notions of lifting worlds to sets of worlds.

Definition 4 (Agent types for the lifting problem). *Let S be a set ordered by a total pre-order $\succeq$. The lifting problem is the selection of an element of S . We define the following agent types for the lifting problem:*

- **Optimistic agent:** *The agent selects the elements of S which are maximal w.r.t. $\succeq$.*
- **Pessimistic agent:** *The agent selects the elements of S which are minimal w.r.t. $\succeq$.*

However, this cannot directly be used for our four kinds of preferences, due to the bipolar representation of preferences. To choose among these kinds of preferences, we introduce an agent interpretation of preferences. We interpret a preference of p over q as a game between an agent arguing for p and an agent arguing for q . Thus, the agent argues that p is better than q against a (possibly hypothetical) opponent.

Example 2. Assume an agent is looking for a flight ticket on the web, and it prefers web-service FastTicket to web-service TicketNow. If the agent is opportunistic, it is optimistic about FastTicket and pessimistic about TicketNow, but when it is careful, it is pessimistic about FastTicket, and optimistic about TicketNow. Clearly, an opportunistic agent has many preferences, whereas a careful agent has only a few preferences.

Preference types can now be defined in terms of agent types.

Definition 5 (Preference types). *Consider an agent expressing its preference of p over q . We define the following preference types:*

- **Locally optimistic:** *the agent is optimistic about p and optimistic about q .*
- **Locally pessimistic:** *the agent is pessimistic about p and pessimistic about q .*
- **Opportunistic:** *the agent is optimistic about p and pessimistic about q .*
- **Careful:** *the agent is pessimistic about p and optimistic about q .*

The following example illustrates that the preference types are a useful metaphor to distinguish among the kinds of preferences, but that their use should not be taken too far.

Example 3 (Continued). The agent types are very strong, which makes them useful in practice but which also has the consequence that one has to be careful when using them, for example when formalizing examples. This is illustrated by several properties about preference types in the logic. For example, when a careful agent prefers FastTicket to TicketNow, an opportunistic agent with the same preference order holds the same preference. Moreover, if a careful agent prefers FastTicket to TicketNow, then it follows that it cannot hold the inverse preference of TicketNow over FastTicket at the same time. An opportunistic agent, however, can hold both inverse preferences at the same time.

It seems that careful preference type is too weak. However it may be useful when all other preference types give an empty set of models [15]:

Example 4. Let j and f be two propositional variables which stand for marriage with John and Fred, respectively. Let $\mathcal{P}_{xy} = \{\top \xrightarrow{x} y j, \top \xrightarrow{x} y f, \top \xrightarrow{x} y \neg(j \wedge f)\}$ be a set of Sue's preferences about its marriage with John or Fred. $\mathcal{P}_{xy}$ induces the following set of constraints: $\{j \xrightarrow{x} y \neg j, f \xrightarrow{x} y \neg f, \neg(j \wedge f) \xrightarrow{x} y (j \wedge f)\}$. The first constraint means that Sue prefers to be married to John over not being married to him. The second constraint means that Sue prefers to be married to Fred over not being married to him and the last constraint means that Sue prefers not to be married to both. There is no pre-order satisfying any of the sets $\mathcal{P}_{MM}$, $\mathcal{P}_{mM}$ and $\mathcal{P}_{mm}$ while the following pre-order $(\{j \neg f, \neg j f\}, \{j f, \neg j \neg f\})$ satisfies $\mathcal{P}_{Mm}$.

4 Nonmonotonic Logic of Preferences

We study fragments of the logic that consist of sets of preferences only. We call such sets of preferences a preference specification.

Definition 6 (Preference Specification). A preference specification is a tuple $\langle \mathcal{P}_{MM}, \mathcal{P}_{Mm}, \mathcal{P}_{mM}, \mathcal{P}_{mm} \rangle$ where $\mathcal{P}_{xy}$ ($xy \in \{MM, Mm, mM, mm\}$) is a set of preferences of the form $\{p_i \xrightarrow{x} y q_i : i = 1, \dots, n\}$.

In this section we consider the problem of finding pre-orders $\succeq$ that satisfy each desire of a single set $\mathcal{P}_{xy}$ – i.e., models of $\mathcal{P}_{xy}$. In the following section, we consider models of two or more sets of preferences.

Definition 7 (Model of a set of preferences). Let $\mathcal{P}_{xy}$ be a set of preferences and $\succeq$ be a total pre-order. $\succeq$ is a model of $\mathcal{P}_{xy}$ iff $\succeq$ satisfies each preference $p_i \xrightarrow{x} y q_i$ in $\mathcal{P}_{xy}$.

Shoham [13] characterizes nonmonotonic reasoning as a mechanism that selects a subset of the models of a set of formulas, which we call distinguished models in this paper. Shoham calls these models “preferred models”, but we do not use this terminology as this meta-logical terminology may be confused with preferences in logical language and preference orders in semantics.

In this paper we compare total pre-orders based on the so-called specificity principle. The minimal specificity principle is gravitating towards the least specific pre-order, while the maximal specificity principle is gravitating towards the most specific pre-order. These have been used in non-monotonic logic to define the distinguished model of a set of conditionals of the kind $M \xrightarrow{M}$, sometimes called defeasible conditionals.

Definition 8 (Minimal/Maximal specificity principle). Let $\succeq$ and $\succeq'$ be two total pre-orders on a set of worlds W represented by ordered partitions $(E_1, \dots, E_n)$ and $(E'_1, \dots, E'_n)$ respectively. We say that $\succeq$ is at least as specific as $\succeq'$, written as $\succeq \sqsubseteq \succeq'$, iff $\forall \omega \in W$, if $\omega \in E_i$ and $\omega \in E'_j$ then $i \leq j$. $\succeq$ is said to be the least (resp. most) specific pre-order among a set of pre-orders $\mathcal{O}$ if there is no $\succeq'$ in $\mathcal{O}$ such that $\succeq' \sqsubseteq \succeq$, i.e., $\succeq' \sqsubseteq \succeq$ without $\succeq \sqsubseteq \succeq'$ (resp. $\succeq \sqsubseteq \succeq'$).

The following example illustrates minimal and maximal specificity.

Example 5. Consider the rule $p \overset{x}{\rightarrow} y q$. Applying the minimal specificity principle on $p \overset{M}{\rightarrow} q$ or $p \overset{m}{\rightarrow} q$ gives the following model $\succeq = (\{pq, \neg pq, \neg p\neg q\}, \{p\neg q\})$. The preferred worlds in this model are those which do not violate the rule. More precisely pq belongs to the set of preferred worlds since it satisfies the rule but $\neg pq$ and $\neg p\neg q$ are preferred too since they do not violate the rule even if they do not satisfy it. Now applying the maximal specificity principle on $p \overset{m}{\rightarrow} q$ gives the following model $\succeq' = (\{pq\}, \{\neg pq, p\neg q, \neg p\neg q\})$. We can see that the preferred worlds are only those which satisfy the rule.

Shoham defines non-monotonic consequences of a logical theory as all formulas which are true in the distinguished models of the theory. An attractive property occurs when there is only one distinguished model, because in that case it can be decided whether a formula non-monotonically follows from a logical theory by calculating the unique distinguished model, and testing whether the formula is satisfied by the distinguished model. Likewise, all non-monotonic consequences can be found by calculating the unique distinguished model and characterizing all formulas satisfied by this model.

Theorem 1. *The following table summarizes uniqueness of distinguished models.*

$\mathcal{P}_{mm}$		$\mathcal{P}_{mM}$		$\mathcal{P}_{Mm}$		$\mathcal{P}_{MM}$	
least	most	least	most	least	most	least	most
no	yes [9]	yes [5]	yes	no	no	yes [12, 3]	no

Proof. Most of the uniqueness proofs have been given in the literature, as indicated in the table. The only exception is the uniqueness of most specific model of $\mathcal{P}_{mM}$, which can be derived from the uniqueness of the least specific model of $\mathcal{P}_{mM}$. We do not give the details here – it follows from the more general Theorem 3 below. Here we give counterexamples for the uniqueness in the other cases. Let $A = \{p, q\}$ such that we have four distinct worlds.

Non-uniqueness of most specific models of $\overset{M}{>} \overset{M}{\rightarrow}$:

$\mathcal{P}_{MM}\{p \overset{M}{>} \overset{M}{\rightarrow} \neg p\}$, $\succeq = (\{pq\}, \{p\neg q, \neg pq, \neg p\neg q\})$, $\succeq' = (\{p\neg q\}, \{\neg pq, \neg p\neg q, pq\})$.

Non-uniqueness of least specific models of $\overset{m}{>} \overset{m}{\rightarrow}$:

$\mathcal{P}_{mm}\{p \overset{m}{>} \overset{m}{\rightarrow} \neg p\}$, $\succeq = (\{pq, p\neg q, \neg pq\}, \{\neg p\neg q\})$, $\succeq' = (\{pq, p\neg q, \neg p\neg q\}, \{\neg pq\})$.

Non-uniqueness of least specific models of $\overset{M}{>} \overset{m}{\rightarrow}$:

$\mathcal{P}_{Mm}\{p \overset{M}{>} \overset{m}{\rightarrow} \neg p\}$, $\succeq = (\{pq, p\neg q, \neg pq\}, \{\neg p\neg q\})$, $\succeq' = (\{pq, p\neg q, \neg p\neg q\}, \{\neg pq\})$.

Non-uniqueness of most specific models of $\overset{M}{>} \overset{m}{\rightarrow}$:

$\mathcal{P}_{Mm}\{p \overset{M}{>} \overset{m}{\rightarrow} \neg p\}$, $\succeq = (\{pq\}, \{p\neg q, \neg pq, \neg p\neg q\})$, $\succeq' = (\{p\neg q\}, \{pq, \neg pq, \neg p\neg q\})$

There are two consequences of Theorem 1 which are relevant for us now. First, as we are interested in developing algorithms for unique distinguished models, in the remainder of this paper we only focus on $\overset{M}{>} \overset{M}{\rightarrow}$, $\overset{m}{>} \overset{M}{\rightarrow}$ and $\overset{m}{>} \overset{m}{\rightarrow}$ preference types. Secondly, constraints of the form $\overset{m}{>} \overset{M}{\rightarrow}$ are in between $\overset{M}{>} \overset{M}{\rightarrow}$ and $\overset{m}{>} \overset{m}{\rightarrow}$, in the sense that there is a unique least specific model for $\overset{m}{>} \overset{M}{\rightarrow}$ and $\overset{M}{>} \overset{M}{\rightarrow}$, and there is a unique most specific model for $\overset{m}{>} \overset{M}{\rightarrow}$ and $\overset{m}{>} \overset{m}{\rightarrow}$.

5 Algorithms for Nonmonotonic Logic of Preferences

We now consider distinguished models of sets of preferences of distinct types. It directly follows from Theorem 1 that our only hope to find a unique least or most specific model of a set of preferences is that we may find a unique least specific model for preferences for constraints of both $m_{>^M}$ and $M_{>^M}$, and a unique most specific model for $m_{>^M}$ and $m_{>^m}$. In all other cases we already do not have a unique distinguished model for one of the preferences. However, it does not follow from Theorem 1 that a least specific model of a set of $m_{>^M}$ and $M_{>^M}$ together is unique, and it does not follow from the theorem that a most specific model for $m_{>^M}$ and $m_{>^m}$ together is unique! We therefore consider the two following questions in this section:

1. Is a least specific model of a set of $m_{>^M}$ and $M_{>^M}$ together unique? Is a most specific model for $m_{>^M}$ and $m_{>^m}$ together unique? If so, how can we find these unique models?
2. How can we define distinguished models that consists of all three kinds of preferences?

5.1 $\mathcal{P}_{MM}$ and $\mathcal{P}_{mM}$

The following definition derives a unique distinguished model from $\mathcal{P}_{MM}$ and $\mathcal{P}_{mM}$ together. This algorithm generalizes the algorithms given in [3, 5], in the sense that when one of the sets is empty, we get one of the original algorithms.

Definition 9. Given two sets of preferences $\mathcal{P}_{MM} = \{C_i = p_i \overset{M}{>} q_i : i = 1, \dots, n\}$ and $\mathcal{P}_{mM} = \{C'_j = p'_j \overset{m}{>} q'_j : j = 1, \dots, n'\}$, let associated constraints be sets of pairs $\mathcal{C} = \{(L(C_i), R(C_i))\} \cup \{(L(C'_j), R(C'_j))\}$, where $L(C_i) = |p_i \wedge \neg q_i|$, $R(C_i) = |\neg p_i \wedge q_i|$, $L(C'_j) = |p'_j \wedge \neg q'_j|$ and $R(C'_j) = |\neg p'_j \wedge q'_j|$ (where $|\alpha|$ is $\{s \in W \mid w \models \alpha\}$). Algorithm 1.1 computes a unique distinguished model of $\mathcal{P}_{MM} \cup \mathcal{P}_{mM}$.

Algorithm 1.1: Handling mixed preferences $M_{>^M}$ and $m_{>^M}$.

begin

$l \leftarrow 0$;

while $W \neq \emptyset$ **do**

$- l \leftarrow l + 1$;

$- E_l = \{\omega : \forall (L(C_i), R(C_i)), (L(C'_j), R(C'_j)) \in \mathcal{C}, \omega \notin R(C_i), \omega \notin R(C'_j)\}$;

if $E_l = \emptyset$ **then**

$\perp$ Stop (inconsistent constraints)

$- W = W - E_l$;

$-$ remove from $\mathcal{C}$ each $(L(C_i), R(C_i))$ such that $L(C_i) \cap E_l \neq \emptyset$;

$-$ replace each $(L(C'_j), R(C'_j))$ in $\mathcal{C}$ by $(L(C'_j) - E_l, R(C'_j))$;

$-$ remove from $\mathcal{C}$ each $(L(C'_j), R(C'_j))$ such that $L(C'_j)$ is empty;

return $(E_1, \dots, E_l)$

end

We first explain the algorithm, then we illustrate it by an example, and finally we show that the distinguished model computed is the unique least specific one. At each step of the algorithm, we look for worlds which can have the actual highest ranking in the preference order. This corresponds to the actual minimal value l . These worlds are those which do not appear in any right part of the actual set of constraints $\mathcal{C}$ i.e., they do not falsify any constraint. Once these worlds are selected, the two types of constraints have different treatments:

1. We remove constraints $(L(C_i), R(C_i))$ such that $L(C_i) \cap E_l \neq \emptyset$, because such constraints are satisfied. Worlds in $R(C_i)$ will necessarily belong to E_j with $j > l$, i.e., they are less preferred than worlds in the actual set E_l .
2. Concerning the constraints $(L(C'_j), R(C'_j))$, we reduce their left part by removing the elements of the actual set E_l . While $L(C'_j) \neq \emptyset$, such a constraint is not yet satisfied since the constraint $p'_j \succ^M q'_j$ induces a constraint stating that each $p'_j \wedge \neg q'_j$ world should be preferred to all $\neg p'_j \wedge q'_j$ worlds. A pair $(L(C'_j), R(C'_j))$ is then removed only when $L(C'_j) \subseteq E_l$.

The least specific criterion can be checked by construction. At each step l we put in E_l all worlds which do not appear in any $R(C_i)$ or $R(C'_j)$ and which are not yet put in some E_j with $j < l$. If $\omega \in E_l$, then it necessarily falsifies some constraints which are not falsified by worlds of E_j for $j < l$. If we would put some ω of E_l in E_j with $j < l$, then we get a contradiction.

Example 6. Let r, j and w be three propositional variables which stand respectively for “it rains”, “to do jogging” and “put a sport wear”. Let $\{\omega_0 : \neg r \neg j \neg w, \omega_1 : \neg r \neg j w, \omega_2 : \neg r j \neg w, \omega_3 : \neg r j w, \omega_4 : r \neg j \neg w, \omega_5 : r \neg j w, \omega_6 : r j \neg w, \omega_7 : r j w\}$. Let $\mathcal{P} = \{C_1 : r \wedge \neg j \succ^M r \wedge j, C_2 : (j \vee r) \wedge w \succ^M (j \vee r) \wedge \neg w, C_3 : \neg j \wedge \neg w \succ^M \neg j \wedge w\}$. The first constraint means that if it rains then the agent prefers to do jogging. The second constraint means that if the agent does jogging or it rains then it prefers to put a sport wear and the third constraint means that if the agent will not do jogging then it prefers to not put a sport wear.

We have $\mathcal{C} = \{(L(C_1), R(C_1)), (L(C_2), R(C_2)), (L(C_3), R(C_3))\}$, i.e., $\{(\{\omega_4, \omega_5\}, \{\omega_6, \omega_7\}), (\{\omega_3, \omega_5, \omega_7\}, \{\omega_2, \omega_4, \omega_6\}), (\{\omega_0, \omega_4\}, \{\omega_1, \omega_5\})\}$. We put in E_1 worlds which do not appear in any $R(C_i)$. Then $E_1 = \{\omega_0, \omega_3\}$. We remove $(L(C_2), R(C_2))$ and replace $(L(C_3), R(C_3))$ by $(L(C_3) - E_1, R(C_3)) = (\{\omega_4\}, \{\omega_1, \omega_5\})$. Then $\mathcal{C} = \{(\{\omega_4, \omega_5\}, \{\omega_6, \omega_7\}), (\{\omega_4\}, \{\omega_1, \omega_5\})\}$. Now $E_2 = \{\omega_2, \omega_4\}$ so both constraints in $\mathcal{C}$ are removed. Lastly $E_3 = \{\omega_1, \omega_5, \omega_6, \omega_7\}$. Finally, the computed distinguished model of $\mathcal{P}$ is $\succeq = (\{\omega_0, \omega_3\}, \{\omega_2, \omega_4\}, \{\omega_1, \omega_5, \omega_6, \omega_7\})$.

The above algorithm computes the least specific model of $\mathcal{P}_{MM} \cup \mathcal{P}_{mM}$ which is unique. To show the uniqueness property, we follow the line of the proofs given in [4, 5]. We first define the maximum of two preference orders.

Definition 10. Let $\succeq$ and $\succeq'$ be two preference orders represented by their well ordered partitions $(E_1, \dots, E_n)$ and $(E'_1, \dots, E'_{n'})$ respectively. We define the MAX operator by $\text{MAX}(\succeq, \succeq') = (E''_1, \dots, E''_{\min(n, n')})$, such that $E''_1 = E_1 \cup E'_1$ and $E''_k = (E_k \cup E'_k) - (\bigcup_{i=1, \dots, k-1} E''_i)$ for $k = 2, \dots, \min(n, n')$, and the empty sets E''_k are eliminated by renumbering the non-empty ones in sequence.

We put $\mathcal{P} = \mathcal{P}_{MM} \cup \mathcal{P}_{mM}$. Let $\mathcal{M}(\mathcal{P})$ be the set of models of $\mathcal{P}$ in the sense of Definition 7. Given Definition 10, the following lemma shows that the $\mathcal{MAX}$ operator is internal to $\mathcal{M}(\mathcal{P})$.

Lemma 1. *Let $\succeq$ and $\succeq'$ be two elements of $\mathcal{M}(\mathcal{P})$. Then,*

1. $\mathcal{MAX}(\succeq, \succeq') \in \mathcal{M}(\mathcal{P})$,
2. $\mathcal{MAX}(\succeq, \succeq')$ is less specific than $\succeq$ and $\succeq'$,
3. If $\succeq^*$ is less specific than both $\succeq$ and $\succeq'$ then it is less specific than $\mathcal{MAX}(\succeq, \succeq')$.

Proof. The proof of item 1 is given in the appendix. The proofs of item 2 and 3 can be found in [4].

We also have the following Lemma:

Lemma 2. *There exists a unique preference order in $\mathcal{M}(\mathcal{P})$ which is the least specific one, denoted by $\succeq_{spec}$, and defined by: $\succeq_{spec} = \mathcal{MAX}\{\succeq : \succeq \in \mathcal{M}(\mathcal{P})\}$.*

Proof. From point 1 of Lemma 1, $\succeq_{spec}$ belongs to $\mathcal{M}(\mathcal{P})$. Suppose now that $\succeq_{spec}$ is not unique. This means that there exists another preference order $\succeq^*$ which also belongs to $\mathcal{M}(\mathcal{P})$ and $\succeq_{spec}$ is not less specific than $\succeq^*$. Note that $\succeq_{spec}$ is the result of combining elements of $\mathcal{M}(\mathcal{P})$ using the $\mathcal{MAX}$ operator. Now supposing that $\succeq_{spec}$ is not less specific than $\succeq^*$ contradicts point 2 of Lemma 1.

We can now conclude:

Theorem 2. *Algorithm 1.1 computes the least specific model of $\mathcal{M}(\mathcal{P})$.*

Proof. Following Lemma 1 it computes a preference order which belongs to the set of the least specific models and following Lemma 2, this preference order is unique.

5.2 $\mathcal{P}_{mm}$ and $\mathcal{P}_{mM}$

Algorithm 1.2. computes a distinguished model of $\mathcal{P}_{mM} \cup \mathcal{P}_{mm}$. This algorithm is structurally similar to Algorithm 1.1., and the proof that this algorithm produces the most specific model of these preferences is analogous to the proof of Theorem 2.

Let $\mathcal{P}_{mm} = \{C_i = p_i \mathrel{m>} q_i : i = 1, \dots, n\}$ and $\mathcal{P}_{mM} = \{C'_j = p'_j \mathrel{m>} q'_j : j = 1, \dots, n'\}$. Let $\mathcal{C} = \{(L(C_i), R(C_i))\} \cup \{(L(C'_j), R(C'_j))\}$, where $L(C_i) = | p_i \wedge \neg q_i |$, $R(C_i) = | \neg p_i \wedge q_i |$, $L(C'_j) = | p'_j \wedge \neg q'_j |$ and $R(C'_j) = | \neg p'_j \wedge q'_j |$.

Example 7 (Continued). Let $\mathcal{P}_{mM} = \{\neg j \wedge \neg w \mathrel{m>} \neg j \wedge w\}$ and $\mathcal{P}_{mm} = \{\neg j \wedge w \wedge r \mathrel{m>} \neg j \wedge w \wedge \neg r\}$.

Following Algorithm 1.2, we have $\succeq_{mM,mm} = (\{\omega_0, \omega_4\}, \{\omega_5\}, \{\omega_1, \omega_2, \omega_3, \omega_6, \omega_7\})$.

Theorem 3. *Let $\mathcal{P} = \mathcal{P}_{mM} \cup \mathcal{P}_{mm}$. Then Algorithm 1.2 computes the most specific model of $\mathcal{P}$ which is unique.*

Proof (sketch). Follows the same lines as the proof of Theorem 2. It can also be derived from Theorem 2 using symmetry of the two algorithms.

Algorithm 1.2: Handling mixed preferences $\succ^M$ and $\succ^m$.

```

begin
   $l \leftarrow 0$ ;
  while ( $W \neq \emptyset$ ) do
     $l \leftarrow l + 1$ ;
     $E_l = \{\omega : \forall (L(C_i), R(C_i)), \forall (L(C'_j), R(C'_j)) \in \mathcal{C}, \omega \notin L(C_i), \omega \notin L(C'_j)\}$ ;
    if  $E_l = \emptyset$  then
      Stop (inconsistent constraints)
    else
      - Remove from  $W$  elements of  $E_l$ ;
      - Remove from  $\mathcal{C}$  constraints s.t.  $R(C_i) \cap E_l \neq \emptyset$ ;
      - Replace each  $(L(C'_j), R(C'_j))$  in  $\mathcal{C}$  by  $(L(C'_j), R(C'_j) - E_l)$ ;
      - Remove from  $\mathcal{C}$  constraints with empty  $R(C'_j)$ 
    return  $(E'_1, \dots, E'_l)$  s.t.  $\forall 1 \leq j \leq l, E'_j = E_{l-j+1}$ 
end

```

5.3 $\mathcal{P}_{MM}$, $\mathcal{P}_{mm}$ and $\mathcal{P}_{mM}$

To find a distinguished model of three kinds of preferences, we want to combine the two algorithms. It has been argued in [2, 8] that, in the context of preference modeling, the minimal specificity principle models constraints which should not be violated while the maximal specificity principle models what is really desired by the agent. In our setting, this combination of the least specific and the most specific models leads to a refinement of the first one by the latter.

Definition 11. Let $\succeq''$ be the result of combining $\succeq$ and $\succeq'$ corresponding to the least specific and the most specific models respectively. Then,

- if $\omega \succ \omega'$ then $\omega \succ'' \omega'$,
- if $\omega \simeq \omega'$ then $(\omega \succeq'' \omega' \text{ iff } \omega \succeq' \omega')$.

Example 8 (Continued from Examples 6 and 7). We have a unique least specific pre-order $\succeq_{MM, mM} = (\{\omega_0, \omega_3\}, \{\omega_2, \omega_4\}, \{\omega_1, \omega_5, \omega_6, \omega_7\})$, and a unique most specific pre-order $\succeq_{mM, mm} = (\{\omega_0, \omega_4\}, \{\omega_5\}, \{\omega_1, \omega_2, \omega_3, \omega_6, \omega_7\})$. Following the combination method of Definition 11, we get the following unique distinguished model: $(\{\omega_0\}, \{\omega_3\}, \{\omega_4\}, \{\omega_2\}, \{\omega_5\}, \{\omega_1, \omega_6, \omega_7\})$.

6 Summary

In this paper we introduce and study a logic of preferences, which we understand as a logic that formalizes reasoning about various kinds of preferences.

To define mixed logics of preference, we use total orders on worlds called the preference order. We define four kinds of strict preferences of p over q as "the best/worst p is preferred over the best/worst q ".

To choose among types of preferences, we introduce an agent interpretation of preferences. We interpret a preference of p over q as a game between an agent arguing for p and an agent arguing for q . For an ordered set S an optimistic agent selects the maximal

element of S , and a pessimistic agent selects the minimal element of S . For a preference of p over q , a locally optimistic agent is optimistic about p and optimistic about q , a locally pessimistic agent is pessimistic about p and pessimistic about q , an opportunistic agent is optimistic about p and pessimistic about q , and a careful agent is pessimistic about p and optimistic about q .

To calculate a preference order from preferences, we start from a generalization of System Z, which is usually characterized as gravitating towards the ideal. $\max$ is gravitating towards the ideal or minimal specificity, $\min$ is gravitating towards the worst or maximal specific for $\overset{M}{\succ} \overset{M}{M}$ and $\overset{m}{\succ} \overset{M}{M}$, and most specific for $\overset{m}{\succ} \overset{m}{m}$ and $\overset{m}{\succ} \overset{M}{M}$. We show that also for $\overset{M}{\succ} \overset{M}{M}$ and $\overset{m}{\succ} \overset{M}{M}$ preferences together the least specific model is unique, and we show that for $\overset{m}{\succ} \overset{m}{m}$ and $\overset{m}{\succ} \overset{M}{M}$ preferences together the most specific preference order is unique. For these cases, we have provided algorithms to compute the unique models. We also propose a way to compute a distinguished model of $\overset{M}{\succ} \overset{M}{M}$, $\overset{m}{\succ} \overset{M}{M}$ and $\overset{m}{\succ} \overset{m}{m}$ preferences together, combining the developed algorithms.

The results in this paper can be generalized to *ceteris paribus* preferences using frames [7] or Hansson functions [10]. This is subject of future research. We will also consider consequences of our framework for the discussion on bipolarity [2, 8], distinguishing between bipolarity in logic (left hand side and right hand side of constraint) and in nonmonotonic reasoning (least or most specific).

References

1. Special issue on preferences of computational intelligence. *Computational intelligence*, 20(2), 2004.
2. S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Bipolar representation and fusion of preferences in the possibilistic logic framework. In *8th International Conference on Principle of Knowledge Representation and Reasoning (KR'02)*, pages 421–432, 2002.
3. S. Benferhat, D. Dubois, and H. Prade. Representing default rules in possibilistic logic. In *Proceedings of 3rd International Conference of Principles of Knowledge Representation and Reasoning (KR'92)*, pages 673–684, 1992.
4. S. Benferhat, D. Dubois, and H. Prade. Possibilistic and standard probabilistic semantics of conditional knowledge bases. *Logic and Computation*, 9:6:873–895, 1999.
5. S. Benferhat and S. Kaci. A possibilistic logic handling of strong preferences. In *International Fuzzy Systems Association (IFSA'01)*, pages 962–967, 2001.
6. C. Boutilier. Toward a logic for qualitative decision theory. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation, (KR'94)*, pages 75–86, 1994.
7. J. Doyle and M. P. Wellman. Preferential semantics for goals. In *National Conference on Artif. Intellig. AAA'91*, pages 698–703, 1991.
8. D. Dubois, S. Kaci, and H. Prade. Bipolarity in reasoning and decision – an introduction. the case of the possibility theory framework. In *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference, IPMU'04*, pages 959–966, 2004.
9. D. Dubois, S. Kaci, and H. Prade. Ordinal and absolute representations of positive information in possibilistic logic. In *Proceedings of the International Workshop on Nonmonotonic Reasoning (NMR' 2004)*, Whistler, June, pages 140–146, 2004.
10. S.O. Hansson. What is *ceteris paribus* preference? *Journal of Philosophical Logic*, 25:307–332, 1996.

11. J. Lang, L. Van Der Torre, and E. Weydert. Utilitarian desires. *Autonomous Agents and Multi-Agent Systems*, 5:329–363, 2002.
12. J. Pearl. System z: A natural ordering of defaults with tractable applications to default reasoning. In R. Parikh. Eds, editor, *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge (TARK'90)*, pages 121–135. Morgan Kaufmann, 1990.
13. Y. Shoham. Nonmonotonic logics: Meaning and utility. In *Procs of IJCAI 1987*, pages 388–393, 1987.
14. S. Tan and J. Pearl. Qualitative decision theory. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'94)*, pages 928–933, 1994.
15. L. van der Torre and E. Weydert. Parameters for utilitarian desires in a qualitative decision theory. *Applied Intelligence*, 14:285–301, 2001.
16. G. H. von Wright. *The Logic of Preference*. University of Edinburgh Press, 1963.

Appendix

Proposition 1

Let $\succeq$ and $\succeq'$ be two elements of $\mathcal{M}(\mathcal{P})$. Then,

1. $\mathcal{MAX}(\succeq, \succeq') \in \mathcal{M}(\mathcal{P})$.

Proof

Let $\mathcal{P} = \mathcal{P}_{MM} \cup \mathcal{P}_{mM}$. Let $\succeq$ and $\succeq'$ be two elements of $\mathcal{M}(\mathcal{P})$.

Suppose that $\succeq$ and $\succeq'$ are represented by $(E_1, \dots, E_n)$ and $(E'_1, \dots, E'_h)$ respectively. Let $\succeq'' = \mathcal{MAX}(\succeq, \succeq')$. To show that $\succeq'' \in \mathcal{M}(\mathcal{P})$, we show that $\succeq''$ satisfies all constraints $p \mathrel{M \succ} q$ and $p' \mathrel{m \succ} q'$ in $\mathcal{P}$.

Let $(E''_1, \dots, E''_{\min(n,m)})$ be the well ordered partition associated to $\succeq''$. Recall that the best models of $p \wedge q$ w.r.t. $\succeq$ are defined by $\max(p \wedge q, \succeq) = \{\omega : \omega \models p \wedge q \text{ s.t. } \nexists \omega', \omega' \models p \wedge q \text{ with } \omega \in E_i, \omega' \in E_j \text{ and } j < i\}$.

Similarly the worst models of $p \wedge q$ w.r.t. $\succeq$ are defined by $\min(p \wedge q, \succeq) = \{\omega : \omega \models p \wedge q \text{ s.t. } \nexists \omega', \omega' \models p \wedge q \text{ with } \omega \in E_i, \omega' \in E_j \text{ and } j > i\}$.

Let $p \mathrel{M \succ} q$ be a constraint in $\mathcal{P}$.

Following Definition 7, $\succeq$ belongs to $\mathcal{M}(\mathcal{P})$ means that $\max(p \wedge \neg q, \succeq) \subseteq E_i$ and $\max(\neg p \wedge q, \succeq) \subseteq E_j$ with $i < j$. Also $\succeq'$ belongs to $\mathcal{M}(\mathcal{P})$ means that $\max(p \wedge \neg q, \succeq') \subseteq E'_k$ and $\max(\neg p \wedge q, \succeq') \subseteq E'_m$ with $k < m$.

Following Definition 10, $\max(p \wedge \neg q, \succeq'') \subseteq E''_{\min(i,k)}$ and $\max(\neg p \wedge q, \succeq'') \subseteq E''_{\min(j,m)}$. Now since $i < j$ and $k < m$, we have $\min(i, k) < \min(j, m)$. Hence $\succeq''$ satisfies $p \mathrel{M \succ} q$.

Similarly we show that each constraint $p' \mathrel{m \succ} q'$ in $\mathcal{P}$ is satisfied by $\succeq''$.

$\succeq$ (resp. $\succeq'$) satisfies $p' \mathrel{m \succ} q'$ means that $\min(p' \wedge \neg q', \succeq) \subseteq E_i$ (resp. $\min(p' \wedge \neg q', \succeq') \subseteq E'_k$) and $\max(\neg p' \wedge q', \succeq) \subseteq E_j$ (resp. $\max(\neg p' \wedge q', \succeq') \subseteq E'_m$) s.t. $i < j$ (resp. $k < m$). Following Definition 10, $\min(p' \wedge \neg q', \succeq'') \subseteq E''_{\min(i,k)}$ and $\max(\neg p' \wedge q', \succeq'') \subseteq E''_{\min(j,m)}$. Again since $i < j$ and $k < m$ then $\min(i, k) < \min(j, m)$. Hence $\succeq''$ satisfies $p' \mathrel{m \succ} q'$.

Expressing Preferences from Generic Rules and Examples – A Possibilistic Approach Without Aggregation Function

Didier Dubois¹, Souhila Kaci², and Henri Prade¹

¹ I.R.I.T., 118 route de Narbonne, 31062 Toulouse Cedex 4, France

² C.R.I.L., Rue de l'Université SP 16 62307 Lens Cedex, France

Abstract. This paper proposes an approach to representing preferences about multifactorial ratings. Instead of defining a scale of values and aggregation operations, we propose to express rationality conditions and other generic properties, as well as preferences between specific instances, by means of constraints restricting a complete pre-ordering among tuples of values. The derivation of a single complete pre-order is based on possibility theory, using the minimal specificity principle. Some hints for revising a given preference ordering when new constraints are required, are given. This approach looks powerful enough to capture many aggregation modes, even some violating co-monotonic independence.

Keywords: Preference aggregation, Possibility theory.

1 Introduction

A classical and popular way for expressing preferences among possible alternatives is to evaluate the choices by means of criteria, then to use some aggregation function for combining these elementary evaluations into a global one for each possible choice, and finally to rank-order the choices on the basis of the global evaluations. Another way, which does not require the commensurateness of the elementary evaluations, is to design procedures for combining the complete pre-orders associated with each criterion into a unique one, but this leads generally to impossibility or triviality results in more symbolic settings.

In this paper we try another route that assumes that preferences can be specified through explicit constraints on a complete pre-order to be determined between choices. These constraints will reflect Pareto ordering together with other specifications expressing, for instance, that a criterion is more important than another one, or stipulating some preference ordering among particular choices.

The paper is organized as follows. Section 2 states the problem and the notations. Section 3 explains the general approach proposed here for the specification of preferences, which is illustrated on different examples. Section 4 further discusses the revision of a complete pre-ordering obtained from generic constraints by constraints issued from particular examples. Section 5 illustrates the approach on an example for which it is known that the pre-order to be found does not admit a representation by a Choquet integral. Section 6 briefly surveys related works inside and outside the possibilistic framework.

2 Framework

It is assumed that objects to be rank-ordered are vectors of satisfaction levels belonging to a linearly ordered scale $S = \{s_1, \dots, s_h\}$ with $s_1 < \dots < s_h$, each vector component referring to a particular criterion. Thus, it is supposed that there exists a unique scale S on which all the criteria can be estimated (commensurateness hypothesis).

Preferences are expressed through comparisons of such vectors $u_i = \{a_1^i, \dots, a_n^i\}$ (written $a_1^i \dots a_n^i$ for short) where $a_j^i \in S$ under the form of constraints

$$a_1 \dots a_n > a'_1 \dots a'_n$$

expressing that $u = a_1 \dots a_n$ is preferred to (or is more satisfactory than) $u' = a'_1 \dots a'_n$. Some components may remain unspecified and replaced by a variable x_j if the j th component is free to take any value in the scale.

In any case, Pareto ordering is always assumed to hold. This can be written

$$\forall i, \forall x'_i, x_1 \dots x_n > x'_1 \dots x'_n \text{ if } \forall i, x_i \geq x'_i \text{ and } \exists k, x_k > x'_k.$$

Let $\mathcal{V}$ be the set of all vectors $a_1 \dots a_n$ such that $\forall j, a_j \in S$. The problem considered can be stated as follows. Given a set of constraints $\mathcal{C} = \{u^i > u'^i : i = 1, \dots, m\}$, where the u^i 's and u'^i 's are instantiated vectors whose components belong to S , find a complete pre-order $\geq$ on $\mathcal{V}$ that agrees with $\mathcal{C}$, and does not introduce stricter preference constraints than what is required by $\mathcal{C}$ and Pareto ordering. Constraints in $\mathcal{C}$ may be of different types. Namely they can be generic as the ones which encode the agreement with Pareto ordering, or refer to particular examples of preferences that the user wants to enforce. Note that some complete pre-orders such as the one induced by minimum aggregation are ruled out as soon as Pareto ordering is enforced.

Other generic constraints of particular interest include those pertaining to the expression of the relative importance of criteria. The greater importance of criterion j w.r.t. criterion k can be expressed under different forms. One way to state it is by exchanging x_j and x_k and writing

$$x_1 \dots x_j \dots x_k \dots x_n > x_1 \dots x_k \dots x_j \dots x_n \text{ when } x_j > x_k.$$

One may think of other ways of expressing that j is more important than k . For instance, one may restrict the above preferences to extreme values of S for the x_i 's such that $i \neq j$ and $j \neq k$, since weights of importance in conjunctive aggregation can be obtained in this way for a large family of operators (e.g., [7]). A more drastic way for expressing relative importance would be to use a lexicographic ordering of the vector evaluations based on a linear order of the levels of importance for the criteria. In this case, the problem of ordering the vectors would be immediately solved. Note that the first above view of relative importance, which is used in the following, is a *ceteris paribus* preference of subvector (x_j, x_k) w.r.t. (x_k, x_j) for $x_j > x_k$, where the first (resp. second) component refers to criterion j (resp. k), which expresses preferential independence. Equal importance can be expressed by stating that any two vectors where x_j and x_k are exchanged, and otherwise identical, have the same levels of satisfaction.

Another example of constraints that may be of interest pertains to the comparison of subvectors (x, y) with respect to $(x \ominus 1, y \oplus 1)$ for criteria of equal importance, where $\ominus 1$ and $\oplus 1$ denote the shifts in S to the element next to x respectively below and above it, provided that x is neither the bottom nor the top element of S . A preference such as $(x, y) > (x \ominus 1, y \oplus 1)$ is in the spirit of Pigou-Dalton transfer in social choice which enables the ordering induced by the sum (and thus Pareto ordering) on vectors of real

numbers to be refined by stating $(\dots, x_j, \dots, x_k, \dots) > (\dots, x_j - \varepsilon, \dots, x_k + \varepsilon, \dots)$ where $0 \leq \varepsilon \leq x_j - x_k$. This refinement has also an equivalent form named Lorenz dominance. See, e.g., [10].

3 General Principle of the Approach

Our aim in this section is to rank-order all possible vectors. Since the global scale depends on the constraints, we use the interval $[0, 1]$ to encode it. The scale $[0, 1]$ is richer and more refined than the scale S . Indeed S only offers a finite number of levels for discriminating alternatives. For this purpose, we use a possibility distribution π , which is a function from a set of alternatives $\mathcal{V}$ to $[0, 1]$, and provides a complete pre-order between alternatives on the basis of their possibility degrees. When the number of alternatives is large, preferences are usually expressed in a more compact way. In this paper, they are expressed through relative constraints on possibility distributions. Namely, the elementary preference between evaluation vectors, $u > u'$, will be encoded by the constraint $\pi(u) > \pi(u')$.

Generally these constraints induce partial pre-orders on the set of alternatives, so we use a *completion* principle to construct a complete pre-order which is consistent with these partial pre-orders. The chosen completion principle depends essentially on the scale considered to rank-order the alternatives. We distinguish two completion principles in possibility theory: minimal and maximal specificity principles which respectively compute the largest and smallest possibility distributions encoding complete pre-orders consistent with the partial pre-orders.

The interval $[0, 1]$ is a unipolar scale which may have two different readings: a negative and a positive reading. In the negative view, the value 1 means that nothing prevents alternatives from having such a possibility degree from being totally satisfactory while the value 0 means that the corresponding alternatives are not satisfactory at all. This is the *minimal* specificity principle since we look for the largest possibility degree. The positive view of the interval $[0, 1]$ assigns the value 1 to alternatives that are really satisfactory and the value 0 to those on which there is no information about their satisfaction level. This is the *maximal* specificity principle since we look for the smallest possibility distribution. Indeed the negative view models penalties while the positive one models rewards. We consider in this paper the negative reading of the interval $[0, 1]$ and use the minimal specificity principle to construct complete pre-orders.

The complete preorder generated by a possibility distribution may also be represented by a well ordered partition of the form $(E_1, \dots, E_k)$ s.t.:

- $E_1 \cup \dots \cup E_k = \mathcal{V}$ and $E_i \cap E_j = \emptyset$ for $i \neq j$,
- $\forall u, u' \in \mathcal{V}$, if $u \in E_i$ and $u' \in E_j$ with $i < j$ then $\pi(u) > \pi(u')$,
- $\forall u, u' \in E_i$, we have $\pi(u) = \pi(u')$.

As already said, we distinguish between several types of constraints in this framework: i) instantiated constraints pertaining to particular examples, ii) generic principles such as Pareto ordering, constraints expressing equal importance between criteria, preference of a set of criteria over another set, contextual preference of some criteria w.r.t. others, etc. From a collection of such constraints, assuming that they are consistent, a

unique possibility distribution will be derived, which is the largest possibility distribution obeying these constraints. The application of this principle known as the minimal specificity principle (e.g. [2]) is justified by the fact that otherwise, there would exist arbitrary preferences between instantiated vectors. Clearly all the elementary preference constraints can be gathered under the form,

$$\pi(u) > \max\{\pi(u') : u' \in U'\} \text{ where } U' \text{ is a subset of } \mathcal{V} \text{ and } u \notin U'. \quad (1)$$

A more general form of constraints is worth introducing. Namely,

$$\max\{\pi(u) : u \in U\} > \max\{\pi(u') : u' \in U'\}. \quad (2)$$

Such a constraint, together with the minimal specificity principle that maximizes each $\pi(u)$ as much as possible, tends to realize the constraint

$$\pi(u) > \max\{\pi(u') : u' \in U'\}$$

for a maximal possible number of u in $U \setminus U'$, leaving room for exceptions if they are required by other constraints. Thus one can state default preferences, such as, for instance for 3-component vector, the greater importance of criterion 1 over criterion 2

$$\forall x, y, z, \pi(xyz) > \pi(yxz) \text{ if } x > y$$

together with exceptions in case of specific values of the 3rd criterion, namely $\pi(xyz_0) < \pi(yxz_0)$.

Algorithm 1.1 (initially designed for handling possibilistic constraints of the form $\pi(p \wedge q) > \pi(p \wedge \neg q)$) modeling default rules “if p then q generally”), gives the least specific (which is unique) possibility distribution satisfying a set of constraints of the form (1) or (2) [1]. Let $\mathcal{C} = \{C_i : i = 1, \dots, m\}$ be a set of constraints such that each C_i is of the form (1) or (2). Let $\mathcal{L}_\mathcal{C} = \{(L(C_i), R(C_i)) : C_i \in \mathcal{C}\}$ such that if $C_i : \max\{\pi(u) : u \in U\} > \max\{\pi(u') : u' \in U'\}$ is a constraint in $\mathcal{C}$ then $L(C_i) = U$ and $R(C_i) = U'$.

Note that applying the minimal specificity principle gives the most *compact* possibility distribution satisfying the considered set of constraints [13, 1]. This can be checked by construction, noticing that at each step, the algorithm puts as many alternatives in E_k as possible.

Algorithm 1.1:

begin

$k = 0$;

while $\mathcal{V}$ is not empty **do**

$k \leftarrow k + 1$;

$E_k = \{u : \forall (L(C_i), R(C_i)) \in \mathcal{L}_\mathcal{C}, u \notin R(C_i)\}$;

if $E_k = \emptyset$ **then** Stop (inconsistent constraints);

$\mathcal{V} = \mathcal{V} - E_k$;

 Remove from $\mathcal{C}$ each $(L(C_i), R(C_i))$ s.t. $L(C_i) \cap E_k \neq \emptyset$;

return $(E_1, \dots, E_k)$

end

One obvious advantage of this constraint-based approach is that it leads to check the consistency of preference aggregation requirements. In case of inconsistency, no ordering would be found.

Example 1. Assume we have two criteria that can take values a, b or c , with $a > b > c$. Pareto ordering forces to have $\pi(xy) > \pi(x'y')$ as soon as $x > x'$ and $y \geq y'$ or $x \geq x'$ and $y > y'$ for x, y, x', y' ranging in $\{a, b, c\}$.

The application of the minimal specificity principle leads to

$$\pi(aa) > \pi(ab) = \pi(ba) > \pi(ac) = \pi(bb) = \pi(ca) > \pi(bc) = \pi(cb) > \pi(cc).$$

Note that letting $\pi(ac) = \pi(ca) > \pi(bb)$ or the converse would lead to express more constraints than what is only specified by Pareto constraints. In fact, it may look a little surprising to get $\pi(ac) = \pi(bb) = \pi(ca)$. However this is justified by the fact that the minimal specificity principle gives to each alternative the highest possible rank (i.e., possibility degree). The alternatives ac, bb and ca cannot have the highest possibility degree since following Pareto ordering, they are strictly less preferred than aa, ab and ba respectively. Indeed to ensure that we associate the highest possibility degree to these alternatives, the minimal specificity principle keeps the three pairs of evaluations at the same level, and they are ranked immediately below ab and ba . The maximal specificity principle applied to Pareto constraints only would yield the same result. ■

It is worth noticing that the minimal specificity principle doesn't enforce any preference between criteria if not explicitly provided. More precisely if there is no constraint relating some criteria then the minimal specificity principle assumes that they have an equal importance.

In the above example, there is no constraint relating the two criteria x and y . However, due to minimal specificity principle, the possibility distribution obtained from Pareto constraints satisfies the following equality:

$$\forall x, y, \pi(xy) = \pi(yx).$$

Assume now that there is another set of additional constraints, denoted $\mathcal{C}$, expressing relative importance between criteria. We suppose that these constraints are *consistent* with Pareto constraints otherwise no possibility distribution can be computed. We distinguish two approaches to deal with these constraints together with Pareto constraints. The first approach consists of first computing the possibility distribution associated to Pareto constraints following minimal specificity principle and then modifying this possibility distribution with the instantiated constraints derived from $\mathcal{C}$. The modification process performs a minimal change on the existing possibility distribution in order to obey the additional constraints. It consists in refining π (i.e., by splitting the existing layers into distinct new layers). The second approach consists of computing the possibility distribution by applying the minimal specificity principle on a single set gathering Pareto and the additional constraints.

The second approach could be dubbed "direct completion". It is the most natural one and it determines the correct solution to the solution ranking problem. This result is independent of the order of acquisition of the constraints. The first approach by successive revision steps sounds computationally simpler, and provides a partial ranking at each step. However, proceeding in this way, the order in which constraints are processed may alter the final result, and even violate the constraints that were used to generate the initial ranking. So the idea is to develop an iterative procedure where each step consists in

a simple revision step, and feasibility of the obtained ranking with respect to constraints previously used is also maintained. After providing an illustration of the two strategies on an example, an algorithm is proposed for the successive revision procedure.

Example 2. (continued)

Recall that the possibility distribution associated to Pareto constraints and following minimal specificity principle is

$$\pi(aa) > \pi(ab) = \pi(ba) > \pi(ac) = \pi(bb) = \pi(ca) > \pi(bc) = \pi(cb) > \pi(cc).$$

We assume now that the first criterion is more important, which is expressed by

$$\forall x \forall y \text{ s.t. } x > y, \pi(xy) > \pi(yx). \quad (3)$$

The following ordering enforces constraints (3) by splitting the equivalence classes in the above ordering:

$$\pi(aa) > \pi(ab) > \pi(ba) > \pi(ac) = \pi(bb) > \pi(ca) > \pi(bc) > \pi(cb) > \pi(cc).$$

Let us consider now a single set composed of Pareto constraints and the following constraints $\{ab > ba, ac > ca, bc > cb\}$ corresponding to the relative importance constraints expressed by Equation (3). Then we obtain the following more compact possibility distribution (7 layers instead of 8):

$$\pi(aa) > \pi(ab) > \pi(ac) = \pi(ba) > \pi(bb) = \pi(ca) > \pi(bc) > \pi(cb) > \pi(cc). \quad \blacksquare$$

Algorithm 1.2 gives a procedure to modify a possibility distribution by a set of constraints such that the obtained possibility distribution is the same as the one obtained from applying the minimal specificity principle on a single set composed of all the constraints. The idea of the modification process is described as follows. We consider each instantiated constraint $c_i : u > u'$ issued from additional constraints $\mathcal{C}$. Since the latter are supposed to be consistent with previous constraints, c_i cannot be falsified. It is either satisfied or u and u' belongs to the same layer in the possibility distribution. In the second case, we shift u' in the immediate next layer.

When all instantiated constraints are incorporated in the possibility distribution, it may be the case that inconsistencies occur i.e., the new possibility distribution no longer obeys the previous constraints due to the fact that some alternatives are shifted from initial layers to others. To solve inconsistencies, starting from the highest layer, we apply the shifting process and move alternatives responsible for conflicts to next layers. This procedure is formalized in Algorithm 1.2 and illustrated on Example 3.

Example 3. (continued)

Let us consider the possibility distribution obtained by applying the minimal specificity principle when considering Pareto constraints only. We have $\pi(aa) > \pi(ab) = \pi(ba) > \pi(bb) = \pi(ca) = \pi(ac) > \pi(bc) = \pi(cb) > \pi(cc)$.

Then $E_1 = \{aa\}$, $E_2 = \{ab, ba\}$, $E_3 = \{bb, ca, ac\}$, $E_4 = \{bc, cb\}$ and $E_5 = \{cc\}$.

Constraints induced by relative importance constraints are $ab > ba$, $ac > ca$ and $bc > cb$.

Let us start with the constraint $ab > ba$. $ab =_\pi ba$ so we keep ab in E_2 and put ba in E_3 . We get $E_1 = \{aa\}$, $E_2 = \{ab\}$, $E_3 = \{bb, ca, ac, ba\}$, $E_4 = \{bc, cb\}$, $E_5 = \{cc\}$. Now we have $ac =_\pi ca$ so we keep ac in E_3 and put ca in E_4 . Also $bc =_\pi cb$ so we keep bc in E_4 and put cb in E_5 . Indeed we get $E_1 = \{aa\}$, $E_2 = \{ab\}$, $E_3 = \{bb, ac, ba\}$, $E_4 = \{bc, ca\}$ and $E_5 = \{cc, cb\}$.

Algorithm 1.2:**begin**

- Let π be the possibility distribution and $\geq_\pi$ be the total pre-order associated to π ;
- Let $(E_1, \dots, E_k)$ be the well ordered partition associated to π ;
- Let $\mathcal{C}$ be the new set of relative importance constraints and $\mathcal{L}_\mathcal{C}$ be the instantiation of $\mathcal{C}$ with $\mathcal{V}$;

for each constraint $x > y$ in $\mathcal{L}_\mathcal{C}$ do**-****if $y >_\pi x$ then Stop (the new set of constraints is inconsistent with π) else**

- Let $x, y \in E_i$;

- if $x =_\pi y$ then**

- if $i < k$ then Move y from E_i to E_{i+1} ;**

- else $k \leftarrow k + 1$, Move y from E_i to E_k**

- $i \leftarrow i + 1$;

while $i \leq k$ do

- if alternatives in E_i violate a relative importance constraint then**

- if $i < k$ then**

- Move alternatives of E_i responsible of conflicts to E_{i+1}

- else $k \leftarrow k + 1$, Move alternatives of E_i responsible of conflicts to E_k**

- $i \leftarrow i + 1$;

end

Let us now run the second part of the procedure. Alternatives in E_3 violate Pareto constraints since we should have $ba > bb$. bb is the alternative which is responsible on this conflict so we move bb into E_4 . We get $E_1 = \{aa\}$, $E_2 = \{ab\}$, $E_3 = \{ac, ba\}$, $E_4 = \{bb, bc, ca\}$ and $E_5 = \{cc, cb\}$.

Now constraints of E_4 violate Pareto constraints since we should have $bb > bc$ so we move bc into E_5 . We get $E_1 = \{aa\}$, $E_2 = \{ab\}$, $E_3 = \{ac, ba\}$, $E_4 = \{bb, ca\}$ and $E_5 = \{bc, cc, cb\}$.

Constraints of E_5 violate Pareto and relative importance constraints since we should have $bc > cc$, $bc > cb$ and $cb > cc$. Following the procedure, this turns out to split E_5 into three strata containing respectively bc , cb and cc .

So the result of the modification is $E_1 = \{aa\}$, $E_2 = \{ab\}$, $E_3 = \{ac, ba\}$, $E_4 = \{bb, ca\}$, $E_5 = \{bc\}$, $E_6 = \{cb\}$ and $E_7 = \{cc\}$. ■

4 Mixing Generic Rules and Examples

In the approach, different types of constraints can be considered, namely generic ones which express general principles, and instantiated ones which come from examples of situations where decision maker's preferences are clearly stated. We show in this section how the possibility distribution obtained from generic constraints can be revised in order to obey the examples when these examples are inconsistent with the generic constraints. The result of revision may no longer satisfy the old generic constraints but it should satisfy Pareto constraints. Let $\pi = (E_1, \dots, E_k)$ be a possibility distribution

and u^1, u^2 be two alternatives. Suppose that the user requires an additional constraint on u^1 and u^2 stating that $u^1 > u^2$. There are three possible cases:

1. If $u^1 >_\pi u^2$ then π is unchanged.
2. If $u^1 =_\pi u^2$ then a minimal change takes place in such a way that u^2 remains greater than the alternatives that were below it before the revision:
 - Suppose that $u^1, u^2 \in E_i$.
 - The result of revising π is $\pi' = (E'_1, \dots, E'_{k+1})$ s.t.
 - for $j = 1, \dots, i-1$, $E'_j = E_j$,
 - $E'_i = E_i / \{u^2\}$, $E'_{i+1} = \{u^2\}$, for $j = i+2, \dots, k+1$, $E'_j = E_{j-1}$,
3. If $u^1 <_\pi u^2$ then the idea is again a minimal change in the sense that a minimal discounting of u^2 is performed in order to preserve the maximal number of ordering relations that u^2 was satisfying before revision:
 - Suppose that $u^2 \in E_i$ and $u^1 \in E_j$. We have necessarily $i < j$.
 - Let E_p (resp. E_l) be the lowest (resp. highest) stratum in π s.t. $p \geq i$ (resp. $l \leq j$) and u^2 (resp. u^1) can be put in E_p (resp. E_l) without violating Pareto constraints.
 - if $p < l$ then we cannot enforce $u^1 > u^2$ without violating Pareto constraints,
 - if $l < p$ then the result of revision is $\pi' = (E'_1, \dots, E'_k)$ s.t.
 - * remove u^1 and u^2 from E_j and E_i respectively,
 - * $E'_l = E_l \cup \{u^1\}$ and $E'_{l+1} = E_l \cup \{u^2\}$,
 - * $E'_i = E_i$ for $i \neq l, l+1$,
 - * remove the empty E'_j and renumber the non-empty ones in sequence.
 - if $l = p$ then the result of revision is $\pi' = (E'_1, \dots, E'_{k+1})$ s.t.
 - * remove u^1 and u^2 from E_j and E_i respectively,
 - * $E'_j = E_j$ for $j = 1, \dots, l-1$,
 - * $E'_l = E_l \cup \{u^1\}$, $E'_{l+1} = \{u^2\}$, $E'_j = E_{j-1}$ for $j = l+2, \dots, k+1$.

In all cases, we remove the empty E'_j and renumber the non-empty ones in sequence.

Example 4. Let us consider the following example with three criteria M, P and L which stand for *mathematics*, *physics* and *literature* respectively, and three candidates C_1, C_2 and C_3 rated on the three levels a, b and c respectively.

M and P are supposed to have an importance greater than the one of L , and the result of the global aggregation on the three criteria should be such that the candidate C_3 is preferred to C_1 and C_1 is preferred to C_2 .

Let $\pi(xyz)$ denote the level of acceptability of having x in M , y in P and z in L , where x, y and z take their value in the set $\{a, b, c\}$. The following constraints on possibility degrees encode the different preferences given above:

1. C_3 is preferred to C_1 and C_1 is preferred to C_2 is encoded by:

$$\pi(bbb) > \pi(abc) > \pi(cca).$$
2. P is more important than L is encoded by:

$$\pi(xyz) > \pi(xzy) \text{ for all } x \text{ if } y > z.$$

Table 1

	M	P	L
C_1	a	b	c
C_2	c	c	a
C_3	b	b	b

3. M is more important than L is encoded by:

$$\pi(xyz) > \pi(zyx) \text{ for all } y \text{ if } x > z.$$

4. π is increasing w.r.t. x, y and z (the greater the grades, the better the candidate).

This is Pareto constraint that is written in the following form:

$$\pi(xyz) > \pi(x'y'z') \text{ if } x \geq x', y \geq y', z \geq z' \text{ and } (x > x' \text{ or } y > y' \text{ or } z > z').$$

In this example, generic rules are the constraints given in points 2–4 and examples are given in the point 1. Let $U = \{aaa, aab, aac, aba, abb, abc, acb, acc, baa, bab, bac, bba, bbb, bbc, bca, bcb, bcc, caa, cab, cac, cba, cbb, cbc, cca, ccb, ccc\}$ be the set of all possible alternatives. Applying Algorithm 1.1 on the generic rules gives the following possibility distribution $\pi = (E_1, \dots, E_{11})$ where :

$$E_1 = \{aaa\}, E_2 = \{aab\}, E_3 = \{aac, baa, aba\}, E_4 = \{abb, bab, acb, caa\},$$

$$E_5 = \{bba, abc, bac\}, E_6 = \{acb, bbb, cab\}, E_7 = \{acc, bbc, bca, cac, cba\},$$

$$E_8 = \{bcb, cbb, cca\}, E_9 = \{bcc, cbc\}, E_{10} = \{ccb\}, E_{11} = \{ccc\}.$$

Note that since only relative importance of M and P over L is explicitly expressed then the minimal specificity principle supposes that implicitly M and P have equal importance. Indeed we can check that the complete pre-order obtained above satisfies:

$$\pi(xyz) = \pi(yxz) \text{ for all } x, y \text{ and } z.$$

Now examples are $bbb > abc > cca$. We already have $abc >_\pi cca$ but $bbb <_\pi abc$.

After applying the revision procedure described at the beginning of this section, we get the following possibility distribution $\pi' = (E'_1, \dots, E'_{13})$ where

$$E'_1 = \{aaa\}, E'_2 = \{aab\}, E'_3 = \{aac, aba, baa\}, E'_4 = \{aca, caa, abb, bab\},$$

$$E'_5 = \{bba, bac\}, E'_6 = \{acb, bbb, cab\}, E'_7 = \{abc\}, E'_8 = \{acc, bbc, bca, cac, cba\},$$

$$E'_9 = \{bcb, cbb, cca\}, E'_{10} = \{bcc, cbc\}, E'_{11} = \{ccb\}, E'_{12} = \{ccc\}.$$

Note that now if the equal importance between criteria is not explicitly stated then the result of revision may violate it. In the above example, if we say explicitly that M and P have equal importance then the example is extended to $bbb > abc = bac > cca$.

Let us now introduce an exception to the relative importance constraint given in point 3 $cba > abc$. This example means that although M is more important than L , the candidate having the highest grade in L and the lowest grade in M is preferred to the candidate having the converse grades, provided that both have grade b in P .

Applying the revision procedure described in this section gives the following possibility distribution: $\pi' = (E'_1, \dots, E'_{12})$ where

$$E'_1 = \{aaa\}, E'_2 = \{aab\}, E'_3 = \{aac, aba, baa\}, E'_4 = \{aca, caa, abb, bab\},$$

$$E'_5 = \{bba, bac\}, E'_6 = \{acb, bbb, cab, cba\}, E'_7 = \{abc\}, E'_8 = \{acc, bbc, bca, cac\},$$

$$E'_9 = \{bcb, cbb, cca\}, E'_{10} = \{bcc, cbc\}, E'_{11} = \{ccb\}, E'_{12} = \{ccc\}. \blacksquare$$

Computing a whole possibility distribution can be heavy since the number of alternatives grows exponentially with the number of criteria (i.e., variables). One way to overcome this problem is to focus on particular queries. More precisely, given two al-

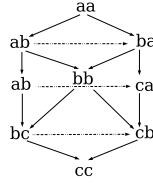


Fig. 1. Partial pre-orders induced by constraints

ternatives u^1 and u^2 , the question is to find whether u^1 is strictly preferred to u^2 , or the converse, or if they are equally preferred. Based on the partial pre-orders expressed by the set of constraints, it is possible to answer this query by finding a path from u^1 to u^2 . Fig. 1 summarizes the different partial pre-orders generated by the constraints given in Example 1. Indeed if there is a sequential path from u^1 to u^2 this means that u^1 is preferred to u^2 and if there is no sequential path between them then they are equally preferred. The complete pre-order associated to π can be obtained by such queries.

5 An Example Not Representable by a Choquet Integral

The aim of this section is to show that our framework is powerful enough to model some problems that may have no solution using numerical aggregations. Henre is an example.

Let c and p be two criteria which stand respectively for “cost” and “performance” to buy a car. A possible alternative is a couple (c, p) . The aim of the user is to choose a powerful car with a cheap price. This means that the value function is decreasing w.r.t. c and increasing w.r.t. p .

Let A, B, C and D be four cars described as follows: $A : (c = 50000, p = 100)$, $B : (c = 70000, p = 110)$, $C : (c = 50000, p = 130)$ and $D : (c = 70000, p = 160)$.

The user expresses the following preferences

$$A = (50000, 100) \geq B = (70000, 110) \text{ and } C = (50000, 130) \leq D = (70000, 160).$$

Let us now consider another set of cars: $A' : (c = 30000, p = 130)$, $B' : (c = 40000, p = 160)$, $C' : (c = 30000, p = 100)$ and $D' : (c = 40000, p = 110)$ for which the user gives the following preferences

$$A' = (30000, 130) \geq B' = (40000, 160) \text{ and } C' = (30000, 100) < D' = (40000, 110).$$

The authors of [9] have shown that this example cannot be represented by a Choquet integral since the choices given by the user are contradictory “co-monotonic” choices. Let us now show that this example can be encoded in our framework by means of a revision of a set of generic rules by a set of examples.

First we have the following set of constraints: $(x, \alpha) > (x, \beta)$ if $\alpha > \beta$, $(x, \alpha) > (y, \alpha)$ if $x < y$ and, $(x, \alpha) > (y, \beta)$ if $x < y$ and $\alpha > \beta$.

Possible alternatives are $\mathcal{V} = \{(30000, 100), (30000, 110), (30000, 130), (30000, 160), (40000, 100), (40000, 110), (40000, 130), (40000, 160), (50000, 100), (50000, 110), (50000, 130), (50000, 160), (70000, 100), (70000, 110), (70000, 130), (70000, 160)\}$.

The application of Algorithm 1.1 gives the following possibility distribution:

$$\begin{aligned} E_1 &= \{(30000, 160)\}, E_2 = \{(30000, 130), (40000, 160)\}, \\ E_3 &= \{(30000, 110), (40000, 130), (50000, 160)\}, \\ E_4 &= \{(30000, 100), (40000, 110), (50000, 130), (70000, 160)\}, \\ E_5 &= \{(40000, 100), (50000, 110), (70000, 130)\}, \\ E_6 &= \{(50000, 100), (70000, 110)\}, E_7 = \{(70000, 100)\}. \end{aligned}$$

Let us now revise this possibility distribution by the examples $A \geq B, C \leq D, A' \geq B'$ and $C' < D'$. The constraints $A \geq B, C \leq D$ and $A' \geq B'$ are satisfied in the above possibility distribution. There is no constraint stating strict comparisons between A and B (resp. C and D, A' and B') and since the Algorithm 1.1 computes the least specific possibility distribution, they are equally preferred. However we have $C' > D'$ in the above possibility distribution so we need to revise the latter in order to have $C' < D'$.

We get: $E_1 = \{(30000, 160)\}, E_2 = \{(30000, 130), (40000, 160)\},$
 $E_3 = \{(30000, 110), (40000, 130), (50000, 160)\},$
 $E_4 = \{(40000, 110), (50000, 130), (70000, 160)\},$
 $E_5 = \{(30000, 100)\}, E_6 = \{(40000, 100), (50000, 110), (70000, 130)\},$
 $E_7 = \{(50000, 100), (70000, 110)\}, E_8 = \{(70000, 100)\}.$

6 Related Works

The approach presented here relies on i) the idea of expressing generic constraints on the complete pre-order to be found, as well as instantiated ones that reflect preferences between particular examples, and on ii) the application of minimal specificity principle, in the possibilistic framework, for accomodating exceptions without introducing more strict preferences than required. It has been first suggested in [3]. This approach is related to the concern of refining Pareto ordering for rank-ordering conjoint multifactorial evaluations by obtaining qualitative counterparts of different aggregation modes [11, 8].

In the last past years there has been an important research trend in AI in preference representation using logical languages (see [6] for a comparative survey oriented toward computational tractability) for handling symbolic ways of expressing extended preferences. In particular, a powerful representation format of such preferences is provided by “CP-nets” and “TCP-nets” [4], which enable a pre-order to be built from local conditional constraints. Wilson [15] has proposed a logic of conditional preferences, which encompasses TCP-nets, and which is based on the specification of preferences on partially instantiated evaluation vectors. However, as TCP-nets, this approach mainly focuses on binary-valued criteria. Moreover, in this approach, the building of the complete pre-ordering resorts to principles different from the minimal specificity principle, by taking their inspiration from Bayesian nets algorithms.

The proposed approach, which is no longer motivated by the logical expression of preferences and which can directly handle non-binary criteria, appears to be conceptually simpler by giving priority to Pareto ordering, allowing for expressions of very general forms of relative importance constraints together with the possibility of specifying particular cases and exceptions. For instance, our approach would allow to represent preferences considered in [5], such as “if it is the same thing, I prefer the cheapest one”.

7 Conclusion

The proposed approach based on the possibility theory representation setting, relies on very simple principles of completion and revision. It concerns a large class of multicriteria decision problems. Still the approach is preliminary in various respects. Topics for further research include i) the study of the relation between the expressions of qualitative independence in the possibilistic setting [12] and the expression of importance constraints in the present framework, ii) the determination of what particular sets of constraints could capture particular aggregation functions, and iii) the comparison with the results provided by other methods on similar sets of constraints [15, 14].

References

1. S. Benferhat, D. Dubois, and H. Prade. Representing default rules in possibilistic logic. In *Proceedings of 3rd International Conference KR'92*, pages 673–684, 1992.
2. S. Benferhat, D. Dubois, and H. Prade. Possibilistic and standard probabilistic semantics of conditional knowledge bases. *Logic and Computation*, 9:6:873–895, 1999.
3. S. Benferhat, D. Dubois, and H. Prade. Towards a possibilistic logic handling of preferences. *Applied Intelligence*, 14(3):303–317, 2001.
4. C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.
5. J. Chomicki. Preference formulas in relational queries. *ACM Transactions on Databases Systems*, 1-40, 2003.
6. S. Coste-Marquis, J. Lang, P. Liberatore, and P. Marquis. Expressive power and succinctness of propositional languages for preference representation. In *Proceedings of KR'04*, pages 203–212, 2004.
7. D. Dubois, J.-L. Marichal, H. Prade, M. Roubens, and R. Sabbadin. The use of the discrete sugeno integral in decision-making: a survey. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, 9:539–561, 2001.
8. D. Dubois and H. Prade. On different ways of ordering conjoint evaluations. In *Proceedings of the 25th Linz seminar on Fuzzy Set Theory*, pages 42–46, 2004.
9. F. Modave, D. Dubois, M. Grabisch, and H. Prade. L'Intégrale de Choquet: un Outil de Représentation en Décision Multicritères. In *Rencontres francophones sur la logique floue et ses applications (LFA'97)*, pages 81–90, 1997.
10. H. Moulin. *Axioms of Cooperative Decision Making*. In Wiley, New York, 1988.
11. J. Moura-Pires and H. Prade. Specifying fuzzy constraints interactions without using aggregation operators. In *Proceedings of FUZZ-IEEE'00*, pages 228–233, 2000.
12. N. Benamor, S. Benferhat, D. Dubois, K. Mellouli, and H. Prade. A theoretical framework for possibilistic independence in weakly ordered setting. *International Journal of Uncertainty Fuzziness and Knowledge-based Systems*, 10, 2002.
13. J. Pearl. System Z: A natural ordering of defaults with tractable applications to default reasoning. In *Proceedings TARK'90*, pages 121–135, 1995.
14. R. Slowinski, S. Greco, and P. Fortemps. Multicriteria Decision Support Using Rules Representing Rough-graded Preference Relations. In *Proceedings of EUROFUSE'04*, pages 494–504, 2004.
15. N. Wilson. Extending cp-nets with stronger conditional preference statements. In *Proceedings of AAAI 2004*, pages 735–741, 2004.

On the Qualitative Comparison of Sets of Positive and Negative Affects

Didier Dubois and Hélène Fargier

IRIT, 118 route de Narbonne,
31062 Toulouse Cedex, France
{dubois, fargier}@irit.fr

Abstract. Decisions can be assessed by sets of positive and negative arguments — the problem is then to compare these sets. Studies in psychology have shown that the scale of evaluation of decisions should then be considered as bipolar. The second characteristic of the problem we are interested in is the qualitative nature of the decision process — decisions are often made on the basis of an ordinal ranking of the arguments rather than on a genuine numerical evaluation of their degrees of attractiveness or rejection. In this paper, we present and axiomatically characterize two methods based on possibilistic order of magnitude reasoning that are capable of handling positive and negative affects. They are extensions of the maximin and maximax criteria to the bipolar case. More decisive rules are also proposed, capturing both the Pareto principle and the idea of order of magnitude reasoning.

1 Introduction

Let us consider the following very simple situation where each possible decision d is assessed by a finite subset of arguments (or $\mathcal{C}(d) \subseteq X$). X is the set of all possible arguments pertaining to d : an argument is typically a criterion satisfied by d , a risk run by choosing d , a good, or a bad, consequence of d . The point is that some of them are positive, and thus attractive for the decision maker, while others are negative and should be avoided. For instance, when choosing a house, having a garden, a garage is a positive argument. Being close to an airport is a negative argument. Under this view, comparing decisions aims at comparing sets of arguments. For the sake of simplicity, we suppose, without loss of generality that each argument is intrinsically positive, negative or indifferent, but cannot be both. In this paper, we further assume that decisions should be made on the basis of an ordinal ranking of the arguments rather than on a numerical evaluation of their pros and cons. We are thus in search of a method that is both qualitative and capable of handling positive and negative affects.

Studies in psychology have shown that the scale of evaluation of decisions should often be considered as bipolar [15] (see also [16]). The simultaneous presence of positive and negative affects prevents decisions from being simple to make. In the best case, the decision maker is able to map them onto a so called

“net predisposition” expressed on a single scale. Cumulative Prospect Theory [17] proposes to compute the net predisposition as the difference between two capacity functions, the first one measuring the importance of the group of positive affects, the second one the importance of the group of negative affects. More general models, namely bi-capacities and bipolar capacities encompass more sophisticated situations, where e.g. the positive importance of a set of affects can depend on the negative ones.

The handling of qualitative information is not a new question in decision making. Among other motivations is the practical fact that the elicitation of the information required by a quantitative model is often not an easy task. Another motivation is the qualitativensness of human reasoning. The most famous decision rule of this kind is the maximin rule of Wald [18]. It only presupposes that the arguments in X can be ranked in terms of merits by means of some utility function u valued on any ordinal scale. Decisions are then ranked according to the merit of their worst arguments, following a pessimistic attitude — it captures the handling of negative affects. Purely positive decisions are sometimes separately handled in a symmetric way, namely on the basis of their best arguments. The case of ordinal ranking procedures from bipolar information has retained less attention. To the best of our knowledge, the only past work on this topic is in [4]. They propose to merge all positive affects into a degree of satisfaction (using the max rule). If high, this degree does not play any role and the decision is made on the basis of the negative affects (using Wald’s principle). If low, it is understood as a negative affect and merged with the other ones.

In the present paper, we follow a more systematic direction of research, trying to characterize a set of procedures that are at the same time ordinal and bipolar. Unsurprisingly, the reader will see that the corresponding decision rules are strongly related to possibility theory – and to their refinement by lexi-max/discrimax and/or leximin/discrimin comparison.

2 Background

The present work obviously relies on two sets of tools: on the one hand, tools for evaluating sets (basically, capacities and extensions) and on the other hand, the characterization of ordinal set-functions for the qualitative unipolar case.

2.1 Measuring the Importance of Sets

Capacity functions are designed to measure the importance of subsets A of a set X on a common, unidirectional scale. The intuition is that the larger the set, the higher its importance. Formally:

Definition 1. A capacity σ on X is a function $\sigma: 2^X \rightarrow [0, 1]$ such that $\sigma(\emptyset) = 0$, $\sigma(S) = 1$, and $\forall A, B \subseteq X, A \subseteq B \implies \sigma(A) \leq \sigma(B)$.

In our context, if d is supported by a set of positive arguments A ($\mathcal{C}(d) = A$), then this decision can be evaluated by means of $\sigma(A)$ — i.e. capacities suit the situations where all the elements of X are positive.

In the presence of positive and negative affects, the simplest idea is to assume that X contains two subsets of arguments, the good and the bad ones, respectively denoted by X^+ and X^- and that the net predisposition depends on the importance of each group. The importance of the positive one should then be measured by capacity σ^+ , while the importance of the negative one should be measured by a second one σ^- : the higher σ^+ , the more convincing the set of arguments and conversely the higher σ^- , the more deterring the arguments. Following Cumulative Prospect Theory [17], net predisposition is given by:

$$\forall A \subseteq X, CTP(A) = \sigma^+(A^+) - \sigma^-(A^-) \text{ where } A^+ = A \cap X^+, A^- = A \cap X^-$$

Variants can be built that measure the utility of A by some function of $\sigma^+(A^+)$ and $\sigma^-(A^-)$. All assume a kind of separability between X^+ and X^- . But this assumption does not always hold; for instance, the negativity of an argument may depend on positive ones – e.g. being skilled is more positive for young postulants when applying for a management position. B - σ were introduced [10, 11] so as to handle such non separable bipolar preferences: σ is defined on $Q(X) := \{(A^+, A^-) \in 2^X, A^+ \cap A^- = \emptyset\}$ and increase (resp. decrease) with the addition of elements in A^+ (resp. A^-). CTP is recovered letting $\sigma(A^+, A^-) = \sigma^+(A^+) - \sigma^-(A^-) = CTP(A)$.

Bipolar capacities [12] go one step further in the generalization. This model uses two measures, a measure of positiveness (that increases with the addition of positive arguments and the deletion of negative arguments) and a measure of negativeness (that increases with the addition of negative arguments and the deletion of positive arguments). Formally :

Definition 2. A bipolar capacity on X is a function $\sigma : Q(X) \mapsto [0, 1]^2$, $\sigma(A, B) = (a, b)$ where $a, b \in [0, 1]$ and $\sigma(\emptyset, B) = (0, b)$ $b \in [0, 1]$ $\sigma(X, \emptyset) = (1, 0)$ $\sigma(\emptyset, X) = (0, 1)$ L $\sigma(C, D) = (c, d), \sigma(E, F) = (e, f). E \subseteq C, D \subseteq F \Rightarrow c \geq e \quad f \geq d$

Bi-capacities do not suit the measure of importance of sets. Originally, they are issued from bi-cooperative games [5], where players are divided into two groups, the “pro” and the “cons”: player x is sometimes in favour, sometimes against, but cannot be both simultaneously. That is why x can appear in the first or the second argument of σ , but never simultaneously, and this is why A and B must be disjoint. When measuring the importance of subsets of $X = X^+ \cup X^-$, we had rather use $Q'(X) = 2^{X^+} \times 2^{X^-}$. The importance of a subset of X is then a function $\sigma' : X \mapsto R$ defined by $\sigma'(A) = \sigma(A \cap X^+, A \cap X^-)$, where σ is a bi-capacity on $Q'(X)$. Notice that this model captures incompatibilities that arise when positive and negative affects are conflicting.

2.2 Ordinality

As said previously, the ordinal comparison of sets was extensively used, especially in Artificial Intelligence. Comparison rules and axiomatic systems were proposed, e.g. [7, 13, 8]. Unsurprisingly, axioms for ordinal comparison of sets are defined

in a pure comparative, relational framework rather than using capacities. This is done without loss of generality, since any capacity σ leads to a weak order. Let us first recall that, for any relation $\succeq$, one can define:

- its symmetric part : $A \sim B \iff A \succeq B \text{ and } B \succeq A$
- its asymmetric part: $A \succ B \iff A \succeq B \text{ and } \text{not}(B \succeq A)$
- the incomparability relation: $A \bowtie B \iff \text{not}(A \succeq B) \text{ and } \text{not}(B \succeq A)$

$\succeq$ is said to be quasi-transitive iff $\succ$ is transitive. $\succeq$ is a weak order iff it is complete and transitive. Now:

Definition 3. A capacity σ is said to be *ordinal* iff σ is a weak order and σ satisfies the following property: $(X \succ \emptyset) \implies (A \subseteq C, D \subseteq B, A \succeq B \implies C \succeq D)$.

Contrary to numerical capacities, this framework is not limited to complete and transitive relations. The following discrimax order, that relies on a possibility distribution $\pi : X \mapsto [0, 1]$, is only quasi-transitive: $A \succeq_{Discrimax} B$ iff $\Pi(A \setminus B) \geq \Pi(B \setminus A)$, where $\Pi(V) = \max_{x \in V} \pi(x)$ (see [8]). Another example is given by a family of possibility distributions, say $\mathcal{F}$. It yields a transitive but incomplete relation : $A \succeq_{\mathcal{F}} B \iff \forall \pi \in \mathcal{F}, \Pi(A) \geq \Pi(B)$

The major part of the concepts pertaining to ordinal capacities was proposed in the context of uncertainty representations. X is then a set of states, subsets of X are events and $\succeq$ is a confidence relation, for instance a comparative probability, an acceptance relation, a qualitative possibility, etc. But these mathematical concepts make sense in other domains as well, for instance to compare sets of goods, sets of arguments, coalitions of criteria, of voters, etc.

The basic property of ordinal reasoning is Negligibility that presupposes a qualitative scale where each level is of an order of magnitude much higher than the next lower level. Disjoint subsets are compared on the basis of the order of magnitude of their evaluations. It usually comes along with a notion of Closeness.

Definition 4. A capacity σ is said to be *ordinal of magnitude N* (OM-N) iff σ is a weak order and σ satisfies the following property: $(A \bowtie B \implies A \bowtie C \cup B)$.

- NEG:** $\forall A, B, C$ negligible w.r.t. A , $A \succ B \implies A \succ C \implies A \succ B \cup C$
CLO: $\forall A, B, C$ $A \sim B \implies (A \succ C \implies A \sim C) \implies A \sim B \cup C$.

An event is close to another iff their ratings have the same order of magnitude: a set is obviously close to itself, and to any union of sets of the same order of magnitude. Axiom NEG states that, if B and C are negligible w.r.t. A , then so is also $B \cup C$. This feature is at the foundation of many uncertainty frameworks proposed in AI. For instance, kappa or possibility functions obey it, and it is used in the preferential inference approach to non-monotonic reasoning [14].

The characterizations of qualitative relations are based on the idea that the comparative capacity on sets derives from the basic relation between their elements [7, 13, 8]. In the context of complete and transitive relations, axioms NEG and CLO completely define the so-called OM-relations:

Proposition 1. T

$$\begin{array}{l} - \succeq^{OM} \\ - T \end{array} \quad \begin{array}{l} OM \\ \pi^X \end{array} : \quad OM(Y) = Max_{y \in Y} \pi(y) : A \succeq^{OM} B \iff OM(A) \geq OM(B)$$

π encodes the order of magnitude of the elements of X and obviously coincides with $\succeq^{OM}$ on singletons, i.e. $\pi(x) \geq \pi(y) \iff \{x\} \succeq \{y\}$. The proposition means that under transitivity and completeness, $A \succeq B$ iff the order of magnitude of each state in B is not higher than the one of some state in A . Other relations were proposed and characterized, that are not stricto sensu OM relations, but refine $\succeq^{OM}$ i.e. satisfy:

$$COM \forall A, B \subseteq S, A \succ^{OM} B \implies A \succ B.$$

3 The Basic Ordinal Comparison of Sets of Arguments

We are looking for qualitative decision rules capable of comparing mixed sets of positive and negative arguments on the basis of their individual importance. For the sake of simplicity, we suppose that X is divided into three subsets: X^+ is the set of positive arguments, X^- is the set of negative arguments and X^0 is the set of indifferent ones. X^0, X^+ and X^- are assumed to be disjoint. For any $A \subseteq X$, let $A^+ = A \cap X^+$ and $A^- = A \cap X^-$ be respectively the positive and negative subsets of A . The proposed model assumes that the set of positive arguments X^+ as well as the set of negative arguments X^- is valid for the whole decision set. For each d , $\mathcal{C}(d)$ is the set of arguments relevant for d , including positive and negative ones. Arguments outside $\mathcal{C}(d)$ are irrelevant for d .

Levels of importance can be attached to the elements of X . As usual, they can be described on a totally ordered scale of magnitude $L = [0_L, 1_L]$, e.g. by a function $\pi : X \mapsto L - \pi(x) = 0_L$ means that the decision maker is indifferent to argument x ; the order of magnitude 1_L is the highest level of attraction or repulsion (according to whether it applies to a positive or negative argument). π is supposed to be non trivial, i.e. at least one x receives a positive order of magnitude. By construction, $\forall x_0 \in X^0, \pi(x_0) = 0_L$, so that $OM(A \cup \{x_0\}) = OM(A)$: X_0 does not affect the decision process.

This is clearly a simpler approach than usual MCDM frameworks where each $x \in X$ is a full-fledged criterion rated on a bipolar utility scale like $L_x = [-1_x, +1_x]$. L_x contain a neutral value 0_x , and each group of criteria has a degree of importance in some other positive unipolar scale like $[0, 1]$. Our framework can be embedded into the MCDM framework where each criterion would take value in the binary scale $\{-1, 0\}$ for negative arguments and $\{0, 1\}$ for positive arguments and $\pi(x)$ is the importance of criterion x . Given a decision d , the utility of x for d is not zero only if $x \in \mathcal{C}(d)$.

Amgoud et al. [1] also compare decisions in terms of positive or negative arguments. They use a more complex scheme for evaluating the strength or arguments, whereby an argument possesses both a level of importance and a

degree of certainty, and involves criteria whose satisfaction is a matter of degree. They then compare sets of arguments with very simple optimistic or pessimistic rules, independently of the polarity of the arguments. Our evaluation setting is simpler, but our comparison schemes are more expressive, and truly bipolar.

A first approach to the ranking of decisions may assume that the order of magnitude of A is no longer a unique level like in the unipolar case, but a pair of levels $(OM(A^+), OM(A^-))$. This yields the following Pareto-like rule, which does not assume commensurateness between the evaluation of positive and negative arguments:

Definition 5. $A \succeq_{\pi} B \iff \begin{array}{ll} OM(A^+) \geq OM(B^+) & OM(A^-) \leq OM(B^-) \\ OM(V) = \max_{x \in V} \pi(x) \end{array}$

Abusing notation, we will write $\succeq$ instead of $\succeq_{\pi}$. It is easy to see that $\succeq$ is reflexive and transitive. A and B are close to each other iff both their positive and negative parts share the same order of magnitude; B is negligible w.r.t. A ($A \succ B$) in two cases: either $OM(A^+) \geq OM(B^+)$ and $OM(A^-) < OM(B^-)$, or $OM(A^+) > OM(B^+)$ and $OM(A^-) \leq OM(B^-)$. A and B are indifferent when $OM(A^+) = OM(B^+)$ and $OM(A^-) = OM(B^-)$. In other cases, there is a conflict and A is not comparable with B — $\succeq$ is partial. Maybe too partial: for instance, when $OM(A^-) > OM(A^+)$, $\succeq$ concludes that A is incomparable with $B = \emptyset$ and this even if the positiveness of A is negligible w.r.t its negativeness. In this case, one would rather say that getting A is bad and that getting nothing is preferable. Another drawback is observed when $OM(A^+) > OM(B^+)$ and $OM(A^-) = OM(B^-)$: the above definition enforces $A \succ B$, and this even if $OM(A^+)$ is very weak w.r.t $OM(A^-) = OM(B^-)$ — in the latter case, a rational decider would examine the negative arguments in details before concluding.

The above decision rule does not account for the fact that the two evaluations that are used In the following, we propose a more realistic decision rule for comparing A and B , that focuses on arguments of maximal strength $i = OM(A \cup B)$ in $A \cup B$. The minimum requirement is to obey the following very simple existential principle: A is at least as good as B iff, at level $OM(A \cup B)$ the existence of arguments in favour of B is counterbalanced by the existence of arguments in favour of A and the existence of arguments against A is cancelled by the existence of arguments against B . Let us now formalize the following accounting for commensurate dominance:

Definition 6.

$A \succeq^{Poss} B \iff \begin{array}{ll} OM(A \cup B) = OM(B^+) \implies OM(A \cup B) = OM(A^+) \\ OM(A \cup B) = OM(A^-) \implies OM(A \cup B) = OM(B^-) \end{array}$

Like $\succeq$, relation $\succeq^{Poss}$ collapses to the $\max$ rule if $X = X^+ \cup X^0$. But $\succeq^{Poss}$ weakens the basic property of $\succeq$. Indeed, $OM(A^+) \geq OM(B^+)$ and $OM(B^-) \geq OM(A^-)$ together imply $A \succeq^{Poss} B$ but the converse is not valid. The counter-intuitive behaviours previously pointed out can thus be escaped.

$\succeq^{Poss}$ is also reflexive and transitive. Notice that the range of incompleteness of $\succeq^{Poss}$ is very different from the one of $\succeq$: incomparability appears with sets A such that $OM(A^+) = OM(A^-) > 0_L$. These display an internal

contradiction: in this case, we do not know whether A is good or bad, and in particular, whether it is better than the absence of arguments — thus $A \approx \emptyset$. A non conflicting non-empty set A is either such that $OM(A^+) > OM(A^-)$ and then $A > \emptyset$, or $OM(A^-) > OM(A^+)$ and then $\emptyset > A$.

The existence of internal conflicts is a necessary condition for incomparability: $A \approx B$ if and only if $(A \approx \emptyset \text{ and } OM(A) > OM(B))$ or $(B \approx \emptyset \text{ and } OM(B) > OM(A))$. The condition is not sufficient: a pair of conflicting set that share the same order of magnitude is indifferent. Indeed, $A \sim^{Poss} B$, if $OM(A) = OM(B)$ provided that either $A > \emptyset, B > \emptyset$ or $A < \emptyset, B < \emptyset$ or yet $A \approx \emptyset, B \approx \emptyset$. Finally, five cases of strict dominance of A over B exist: $A > \emptyset > B$; $A > \emptyset$ and $OM(A) > OM(B)$; conversely, $B < \emptyset$ and $OM(A) < OM(B)$; $A \approx \emptyset$ and $OM(A) = OM(B^-) > OM(B^+)$; and conversely $B \approx \emptyset$ and $OM(A^+) = OM(B) > OM(A^-)$.

One might object that $\succeq^{Poss}$ is not decisive enough since only arguments at the highest level are taken into account. In particular, it may happen that $A \subsetneq B$ and $A \sim B$ — the usual drowning effect of possibility theory reappears here. Variants are proposed in Section 5 that overcome this difficulty. Let us turn to axiomatics justifying the above rules.

4 Axioms for Ordinal Comparison on a Bipolar Scale

As usual in axiomatic characterizations, an abstract relation $\succeq$ is considered and the natural properties that it should obey are formalized. We first need a comparative framework capable of encompassing bipolar comparisons — a kind of “comparative bipolar capacity”. The basic notion is the separation of X in good and bad arguments. The first axiom states that any argument is either positive or negative, i.e. better than nothing or worse than nothing.

Clarity of arguments $\forall x \in X, \{x\} \succeq \emptyset \text{ or } \emptyset \succeq \{x\}$

We now scale arguments, defining the sets of positive and negative arguments and a relation $\succeq_{\mathbb{X}}$ on $\mathbb{X} = X \cup \{0\}$ that should be complete and transitive:

$$x \succeq_{\mathbb{X}} y \iff \{x\} \succeq \{y\} \quad x \succeq_{\mathbb{X}} 0 \iff \{x\} \succeq \emptyset \quad 0 \succeq_{\mathbb{X}} x \iff \emptyset \succeq \{x\}$$

$$X^+ = \{x, \{x\} \succ \emptyset\} \quad X^- = \{x, \emptyset \succ \{x\}\} \quad X^0 = \{x, \emptyset \sim \{x\}\}$$

Moreover, arguments that are indifferent to the decision maker cannot affect the preference.

Status quo consistency

$$\{x\} \sim \emptyset \iff (\forall A, B : A \succeq B \iff A \cup \{x\} \succeq B \iff A \succeq B \cup \{x\})$$

Under this axiom we can forget about X_0 . Monotonicity can obviously not be obeyed as such in a bipolar scaling. Indeed, if B is a set of negative arguments, it generally happens that $A \succ A \cup B$. We rather need axioms of monotonicity — positive and negative arguments — basically, the one of bipolar capacities, expressed in a comparative way.

Positive monotonicity $\forall C, C' \subseteq X^+, \forall A, B : A \succeq B \implies C \cup A \succeq B \setminus C'$

Negative monotonicity $\forall C, C' \subseteq X^-, \forall A, B : A \succeq B \implies C \setminus A \succeq B \cup C'$

We finally assume that the bipolar scale encodes all the relevant information, saying that only the positiveness and the negativeness of A and B are to be taken into account: if A is at least as good as B on both the positive and the negative side, then A is at least as good as B . This is expressed by an axiom of unanimity.

Unanimity $\forall A, B \neq \emptyset, A^+ \succeq B^+ \text{ and } A^- \succeq B^- \implies A \succeq B$

This yields the following generalization of comparative capacities:

Definition 7. $A \succeq_{2^X} B$ iff $A \succeq_{2^X} B$ and $A \succeq_{2^X} B$ monotonic bipolar set relation
 $A \succeq_{2^X} B$ iff $A \succeq_{2^X} B$ and $A \succeq_{2^X} B$ monotonic bipolar set relation
 $A \succeq_{2^X} B$ iff $A \succeq_{2^X} B$ and $A \succeq_{2^X} B$ monotonic bipolar set relation

Both $\succeq$ and $\succeq^{Poss}$ are monotonic bipolar set relations. But the definition encompasses numerous models, not necessarily qualitative (e.g. cumulative prospect theory in its full generality). In order to focus on the family relations that are based on order of magnitude reasoning, we need two axioms of negligibility. The first one enforces this property for positive sets, the second one, for negative sets.

NEG+ $\forall A, B, C$ pairwise disjoint sets, $A \succ B$ and $A \succ C \implies A \succ B \cup C$

NEG- $\forall A, B, C$ pairwise disjoint sets, $B \succ A$ and $C \succ A \implies B \cup C \succ A$

The first axiom is significant when $B \cup C \succ C, B$, and trivial when B or C have a negative affect on each other (i.e. when $B \succ B \cup C$ or $C \succ B \cup C$). The second axiom is effective for negative affects. Its satisfaction is immediate for positive affects, and is significant in terms of negligibility when $B \cup C \leq B, C$.

Since the union of positive and negative affects can generate incomparability, closeness should be expressed carefully w.r.t positive and negative sets:

CLO $\forall A, B, C \quad A \sim B \text{ and } B \sim C \implies A \sim B \cup C$

CLO+ $\forall B, C \quad B \succeq C \text{ and } C \subseteq X^+ \implies B \sim B \cup C$

CLO- $\forall B, C \quad B \preceq C \text{ and } C \subseteq X^- \implies B \sim B \cup C$

Proposition 2. $B \succeq_{2^X} C \iff B \succeq_{2^X} C \text{ and } B \succeq_{2^X} C \iff B \succeq_{2^X} C$ $NEG+, NEG-, CLO, CLO+, CLO-$.

We propose to use the axiom of strong unanimity that states that only indifference can enforce indifference:

Strong Unanimity $\forall A, B \neq \emptyset \left\{ \begin{array}{l} A^+ \succeq B^+ \text{ and } A^- \succ B^- \implies A \succ B \\ A^+ \succ B^+ \text{ and } A^- \succeq B^- \implies A \succ B \end{array} \right.$

Strong unanimity is for instance not satisfied by $\succeq^{Poss}$ nor by Benferhat-Kaci's system but it is characteristic of $\succeq$.

Definition 8. $L \succ_{2^X} M$ iff $L \succ_{2^X} M$ and $L \succ_{2^X} M$ monotonic bipolar set relation
 $L \succ_{2^X} M$ iff $L \succ_{2^X} M$ and $L \succ_{2^X} M$ monotonic bipolar set relation
 $L \succ_{2^X} M$ iff $L \succ_{2^X} M$ and $L \succ_{2^X} M$ monotonic bipolar set relation

Theorem 1. $G \succ_{2^X} H$ iff $G \succ_{2^X} H$ and $G \succ_{2^X} H$ monotonic bipolar set relation
 $G \succ_{2^X} H$ iff $G \succ_{2^X} H$ and $G \succ_{2^X} H$ monotonic bipolar set relation
 $G \succ_{2^X} H$ iff $G \succ_{2^X} H$ and $G \succ_{2^X} H$ monotonic bipolar set relation

R . The restriction of $\succeq$ to singletons coincides obviously with $\succeq_{\mathbb{X}}$.

The possibilistic bipolar rule is characterized by an axiom of expressing a stability of the relation with respect to disjunction:

Sep $\forall A, B, C$ such that $(A \cup B) \cap C = \emptyset, A \succeq B \implies A \cup C \succeq B \cup C$

Theorem 2. T

$$\begin{aligned} & - \succeq \text{ is a total order on } \mathcal{P}(X) \text{ of cardinality } 2^X \\ & - \text{ } \text{NEG+}, \text{NEG-}, \text{CLO=}, \text{CLO+}, \text{CLO-}; \\ & - \pi : X \mapsto [0_L, 1_L] \text{ is a bijection } \succeq = \succeq^{Poss}. \end{aligned}$$

Theorem 1 says that $\succeq$ is the comparison that can be drawn from $\succeq_{\mathbb{X}}$, understood as an order of magnitude scale and applying the principles of OM reasoning and strong unanimity only. theorem 2 shows that $\succeq^{Poss}$ plays the same role in bipolar ordinal decision making as $\succeq^{OM}$ does in the unipolar case. $\succeq^{Poss}$ obviously collapses to $\succeq^{OM}$ when X^- is empty. The characterization is a little more complex, since OM reasoning should be expressed on both sides. Interestingly, an axiom of separability is needed in the bipolar case only — in a purely positive scaling, separability is indeed a consequence of CLO and NEG [7], but this is no longer true in the bipolar scaling¹.

5 Refining the Basic Order of Magnitude Comparison

$\succeq^{Poss}$ thus encodes the most natural model of bipolar order of magnitude, and no other model is possible when transitivity and separability are required. But as $\succeq^{OM}$ does, it is quite inefficient as a decision rule — it suffers from a drowning effect. In the following, we propose comparison principles that derive relations compatible with $\succeq^{Poss}$ but more decisive. This compatibility principle is expressed by a condition of refinement: $A \succ^{Poss} B \implies A \succ B$. All the relations presented here satisfy it.

Let us first study the degenerated case where all arguments share the same importance. In this case, $\succeq^{Poss}$ is equivalent to the following existential rule:

$$\begin{aligned} A \succeq^{\exists} \emptyset & \iff A- = \emptyset \\ \emptyset \succeq^{\exists} A & \iff A+ = \emptyset \\ \forall A, B \neq \emptyset : A \succeq^{\exists} B & \iff \begin{cases} B+ \neq \emptyset \Rightarrow A+ \neq \emptyset \text{ and} \\ A- \neq \emptyset \Rightarrow B- \neq \emptyset \end{cases} \end{aligned}$$

Other rules can be derived by application, to the bipolar case, of the usual principles of comparison by inclusion and by cardinality:

$$\begin{aligned} A \succeq^{\subseteq} B & \iff A+ \supseteq B+ \text{ and } A- \subseteq B- \\ A \succeq^{bicard} B & \iff |A+| \geq |B+| \text{ and } |A-| \leq |B-| \\ A \succeq^{card} B & \iff |A+| - |A-| \geq |B+| - |B-| \end{aligned}$$

¹ We could thus replace Sep by less demanding conditions, e. g.: Sep+ : $C \succeq \emptyset \succeq A \succeq B \Rightarrow A \cup C \succeq B \cup C$ and Sep- : $A \succeq B \succeq \emptyset \succeq C \Rightarrow A \cup C \succeq B \cup C$. But since $\succeq^{Poss}$ is fully separable, using SEP better highlights this important feature.

$A \succeq^\exists, \succeq^\subseteq$ and $\succeq^{bicard}$ do not assume any compensation between positive and negative arguments. $\succeq^\subseteq$ cancels arguments that appears in both A and B . $\succeq^{bicard}$ then considers that any positive (resp. negative) argument in A can be cancelled by one positive (resp. negative) argument in B . Making one step further, $\succeq^{card}$ accepts that, within A (and within B) a positive argument can be cancelled by negative one. These rules are increasingly decisive:

Proposition 3. $A \succ^\exists B \implies A \succ^\subseteq B \implies A \succ^{bicard} B \implies A \succ^{card} B$

Let us now enter the general case. The idea is to work levelwise. For instance, $\succeq^{Poss}$ simply applies $\succeq^\exists$ at level $OM(A \cup B)$.

Definition 9 (*i*-section). $F \quad i \in L:$

$$A_i = \{x \in A, \pi(x) = i\} \quad A$$

$$A_i^+ = A_i \cap X^+ \quad A_i^- = A_i \cap X^- \quad (\quad) \quad i-$$

Proposition 4. $A \succeq^{Poss} B \iff A_i \succeq^\exists B_i \quad i = OM(A \cup B)$.

The application of the inclusion based-rule to the higher discriminating level of magnitude yields the following preference relation:

Definition 10 (Discri).

$$A \sim^{discri} B \iff A = B$$

$$A \succ^{discri} B \iff \exists i \in L \quad \begin{cases} \forall j > i & A_j^+ = B_j^+ & A_j^- = B_j^- \\ A_i \succ^\subseteq B_i \end{cases}$$

i.e. $A \succ^{discri} B$ if, at the first higher discriminating level, say level i , either $B_i^+ \subseteq A_i^+$ and $A_i^- \subsetneq B_i^-$ or $A_i^- \subseteq B_i^-$ and $A_i^+ \subsetneq B_i^+$. When $X = X^+$ (resp. $X = X^-$), sets of positive (resp. negative) arguments are to be compared; unsurprisingly, it is easy to check that in this case, $\succeq^{discri}$ collapses to the discrimax (resp. discrimmin) procedure [3]. Like these procedures, $\succeq^{discri}$ is reflexive, complete, non transitive – but quasi-transitive.

$\succeq^{discri}$ cancels any argument appearing in both A and B . One could moreover accept the cancellation of any positive (resp. negative) argument in A by another positive (resp. negative) argument in B that share the same order of magnitude. This yields the following extension of the leximax and leximin procedures.

Definition 11 (BiLexi).

$$A \sim^{BiLexi} B \iff \forall i, |A_i^+| = |B_i^+|$$

$$A \succ^{BiLexi} B \iff \exists i \in L \quad \begin{cases} |A_i^-| = |B_i^-| \\ \forall j > i, |A_j^+| = |B_j^+| & |A_j^-| = |B_j^-| \\ A_i \succ^{bicard} B_i \end{cases}$$

So, the process scans levels top-down as long as A and B share the same number of arguments in both the negative and the positive sides. It stops when a difference appears. If A_i is better than B_i , i.e. contains a higher number of positive arguments and a lower number of negative ones, A is preferred to B . But if one set wins on the positive side, and the other on the negative side, a

conflict is revealed and the procedure concludes to an incomparability. It easy to show that $\succeq^{Bilexi}$ is reflexive, transitive, but not complete.

Finally, following the principles of $\succeq^{card}$ we get the following order, that also generalizes the leximax and leximin procedures:

Definition 12 (Lexi).

$$\begin{aligned} A \sim^{lexi} B &\iff \forall i, |A_i^+| - |A_i^-| = |B_i^+| - |B_i^-| \\ A \succ^{lexi} B &\iff \exists i \in L \quad \begin{cases} \forall j > i, |A_j^+| - |A_j^-| = |B_j^+| - |B_j^-| \\ |A_i^+| - |A_i^-| > |B_i^+| - |B_i^-| \end{cases} \end{aligned}$$

The latter rule is in accordance with Cumulative Prospect Theory. Indeed:

Proposition 5. $T \quad \sigma^+ \quad \sigma^-$

$$A \succeq^{lexi} B \iff \sigma^+(A^+) - \sigma^-(A^-) \geq \sigma^+(B^+) - \sigma^-(B^-)$$

The proposition is obvious using the classical encoding of the leximax procedure by a capacity, e.g. $\sigma^+(V) = \sigma^-(V) = \sum_{i \in L} |V_i| \dot{Card}(X)^i$. Interestingly, this rule is also fully in accordance with OM reasoning since it refines $\succeq$ — it is also the case with the three former relations. The four rules can be ranked from the least ($\succeq^{Poss}$) to the most decisive.

Proposition 6. $A \succ^{Poss} B \implies A \succ^{discr} B \implies A \succ^{bilexi} B \implies A \succ^{lexi} B$

It can be shown that $\succeq^{discr}$, $\succeq^{bilexi}$ and $\succeq^{lexi}$ are efficient, in the sense that they satisfy the principles of preadditivity and Pareto optimality:

$$\textbf{ADD: } \forall A, B, C \text{ such that } (A \cup B) \cap C = \emptyset : A \succeq B \iff A \cup C \succeq B \cup C$$

$$\textbf{Pareto: } A \neq B, A^+ \supseteq B^+, A^- \subseteq B^- \implies A \succ B$$

This concludes our argumentation in favour of $\succeq^{lexi}$: it cumulates the practical advantages of CPT (completeness, transitivity and representability by a function), is efficient in the sense of Pareto and is in accordance with but more decisive than OM reasoning. Following our preliminary work on the unipolar case, we think that the characterization of $\succeq^{discr}$, $\succeq^{bilexi}$ and $\succeq^{lexi}$ is not a major difficulty and we leave it for further research.

6 Conclusion

The proposed work is an extension of possibility theory to the handling of sets containing two-sorted elements considered as positive or negative. The results were couched in a terminology borrowing to argumentation and decision theories, and indeed we consider they can be relevant for both. Our framework is a qualitative counterpart to Cumulative Prospect Theory and more recent proposals using bicapacities. It is far less expressive, even if it could be extended to

elements whose positiveness and negativeness depend on the considered decision (using a duplication process of such x as x^+ and x^- and considering subsets containing one of them at most). The paper is also relevant in argumentation for the evaluation of sets of arguments in inference processes [6], and argument-based decisions [2]. The next step in our research is naturally the extension to (qualitative) bipolar criteria whose satisfaction is a matter of degree [11]. In the future, comparison between our decision rules and those adopted in the above works as well as aggregation processes in finite bipolar scales [9] is in order.

References

1. L. Amgoud, J.F. Bonnefon, and H. Prade. An argumentation-based approach to multiple criteria decision. In these proceedings.
2. L. Amgoud and H. Prade. Using arguments for making decisions: A possibilistic logic approach. In *Proceedings of UAI*, pages 10–17, 2004.
3. F.A. Behringer. On optimal decisions under complete ignorance: a new criterion stronger than both Pareto and maxmin. *Europ. J. Op. Res.*, 1:295–306, 1977.
4. S. Benferhat and S. Kaci. Representing and reasoning with prioritized preferences. Working Notes, Bipolarity Workshop, Le Fossat, France, 2005.
5. J.M. Bilbao, J.R. Fernandez, A. Jim´enez Losada, and E. Lebr´on. Bicooperative games. In J.M. Bilbao, editor, *Cooperative games on combinatorial structures*, pages 23–26. Kluwer Academic Publishers, Dordrecht, 2000.
6. C. Cayrol and M-C.Lagasquie-Schiex. Gradual handling of contradiction in argumentation frameworks. In *Proc. of IPMU’02*, pages 83–90, Annecy, France, 2002.
7. D. Dubois. Belief structures, possibility theory and decomposable confidence measures on finite sets. *Computers and Artificial Intelligence*, 5(5):403–416, 1986.
8. D. Dubois and H. Fargier. An axiomatic framework for order of magnitude confidence relations. In *Proceedings of UAI’04*, pages 138–145, 2004.
9. M. Grabisch. The Moebius transform on symmetric ordered structures and its application to capacities on finite sets. *Discrete Math.*, 28(1-3):17–34, 2004.
10. M. Grabisch and Ch. Labreuche. Bi-capacities for decision making on bipolar scales. In *EUROFUSE’02 Workshop on Information Systems*, pages 185–190, 2002.
11. M. Grabisch and Ch. Labreuche. Bi-capacities — parts I and II. *Fuzzy Sets and Systems*, 151(2):211–260, 2005.
12. S. Greco, B. Matarazzo, and R. Slowinski. Bipolar Sugeno and Choquet integrals. In *EUROFUSE’02 Workshop on Information Systems*, 2002.
13. J. Y. Halpern. Defining relative likelihood in partially-ordered structures. *J. Artif. Intell. Res. (JAIR)*, 7:1–24, 1997.
14. S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44(1-2):167–207, 1990.
15. C. E. Osgood, G.J. Suci, and P. H. Tannenbaum. *The Measurement of Meaning*. University of Illinois Press, Chicago, 1957.
16. P. Slovic, M. Finucane, E. Peters, and D.G. MacGregor. Rational actors or rational fools? implications of the affect heuristic for behavioral economics. *The Journal of Socio-Economics*, 31:329–342, 2002.
17. A. Tversky and D. Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, 5:297–323, 1992.
18. A. Wald. *Statistical Decision Functions*. Wiley, 1950.

Symmetric Argumentation Frameworks

Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis

CRIL–CNRS/Université d’Artois*,

rue de l’Université - S.P. 16,

F-62307 Lens Cedex - France

{coste, devred, marquis}@cril.univ-artois.fr

Abstract. This paper is centered on the family of Dung’s finite argumentation frameworks when the attacks relation is symmetric (and nonempty and irreflexive). We show that while this family does not contain any well-founded framework, every element of it is both coherent and relatively grounded. Then we focus on the acceptability problems for the various semantics introduced by Dung, yet generalized to sets of arguments. We show that only two distinct forms of acceptability are possible when the considered frameworks are symmetric. Those forms of acceptability are quite simple, but tractable; this contrasts with the general case for which all the forms of acceptability are intractable (except for the ones based on grounded or naive extensions).

1 Introduction

Modelling argumentation is known as a major issue of many AI problems, including de-feasible reasoning and some forms of dialogue between agents (see e.g., [1, 2, 3, 4, 5]). In a nutshell, argumentative reasoning is concerned with the interaction of arguments. A key notion for any theory of argumentation is the acceptability one: intuitively, an argument is considered acceptable if it can be argued successfully against attacking arguments. Formally, the acceptability of an argument (resp. a set of arguments taken as a whole) is characterized by the membership (resp. the containment) of it to some selected sets of arguments, referred to as extensions.

Several theories of argumentation have been proposed so far (see among others [6, 7, 8, 9, 10]). In Elvang-Gøransson *et al.*’s theory (refined and extended by several authors, including [7, 11, 12, 13, 14, 15, 16, 17, 18, 19]), one considers in the beginning a set of assumptions and some background knowledge; then an argument is a pair consisting of a statement (the conclusion of the argument) and a (often minimal) subset of assumptions (the support of the conclusion) which is consistent with the background knowledge and such that the conclusion is a logical consequence of it and the background knowledge. Several forms of interaction between arguments have been investigated, including among others the rebuttal relation (an argument rebuts a second one when the conclusion of the former is equivalent to the negation of the conclusion of the

* The authors have been partly supported by the the Région Nord/Pas-de-Calais through the IRCICA Consortium and by the European Community FEDER Program.

latter). In Dung's approach¹ [6], no assumption is made about the nature of an argument (it can be a statement supported by some assumptions like in the theory introduced by Elvang-Gøransson *et al.* but this is not mandatory). What really matters is the way arguments interact w.r.t. the attacks relation. In contrast to Elvang-Gøransson *et al.*'s theory, Dung's theory of argumentation is not concerned with the generation of arguments; arguments and the way they interact are considered as initial data of any argumentation framework. Several notions of extensions have been defined by Dung, reflecting several reasons according to which arguments can be taken together. A major feature of Dung's theory is that it encompasses many approaches to nonmonotonic reasoning and logic programming as special cases.

In this paper, we focus on the family of finite argumentation frameworks obtained by requiring the attacks relation to be symmetric; we also assume that the attacks relation is not empty (which is not so strong an assumption since the argumentation frameworks which violate it are trivial ones: no interactions between arguments exist) and that it is irreflexive; the latter assumption is also sensible since an argument which attacks itself is in some sense paradoxical and the problem of reasoning with paradoxical statements is hard by itself but mainly independent from the argumentation issue. Thus, paradoxical statements are typically not viewed as arguments (for instance, it cannot be the case that the support of a conclusion contradicts the conclusion in Elvang-Gøransson *et al.*'s approach). The symmetry requirement is also not so strong; for instance, the rebuttal relation in Elvang-Gøransson *et al.*'s theory is clearly symmetric.

Our contribution is twofold. We show that while no symmetric argumentation framework is also well-founded, every symmetric argumentation framework is both coherent and relatively grounded. Then we focus on the acceptability problems for the various semantics introduced by Dung, yet generalized to sets of arguments. We show that only two distinct forms of acceptability are possible when considering symmetric frameworks. Finally, we show that those forms of acceptability are quite simple, but tractable for symmetric frameworks, while they are intractable in the general case (except for the ones based on grounded or naive extensions).

The rest of this paper is organized as follows. In Section 2, we recall the main definitions and results pertaining to Dung's theory of argumentation. In Section 3, we focus on symmetric argumentation frameworks and present our contribution. Finally, Section 4 concludes the paper.

2 Dung's Theory of Argumentation

Let us present some basic definitions at work in Dung's theory of argumentation [6]. We restrict them to finite argumentation frameworks.

Definition 1 (finite argumentation frameworks).

A finite argumentation framework is a pair $AF = \langle A, R \rangle$ where A is a finite set of so-called arguments and R is a binary relation over A (a subset of $A \times A$), the attacks relation.

¹ Also refined and extended by several authors, including [20, 21, 22, 23, 24].

Clearly enough, the set of finite argumentation frameworks is a proper subset of the set of Dung's finitary argumentation frameworks, where every argument must be attacked by finitely many arguments.

The definition above clearly shows that a finite argumentation framework is nothing but a finite digraph.

Example 1. Let $AF = \langle A, R \rangle$ be a finite argumentation framework with $A = \{a, b, c, d, e\}$ and $R = \{(e, c), (c, e), (b, c), (c, b), (b, d), (d, b), (c, d), (d, c)\}$. AF is depicted on Figure 1. One can observe that R is a symmetric relation; clearly, this is not always the case for Dung's frameworks but this choice is motivated by the desire to take advantage of AF as a running example throughout the paper.

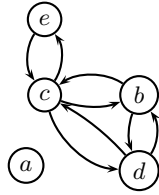


Fig. 1. Digraph for AF

A first important notion is the notion of acceptability: an argument a is acceptable w.r.t. a set of arguments whenever it is defended by the set, i.e., every argument which attacks a is attacked by an element of the set.

Definition 2 (acceptability w.r.t. a set).

Let $AF = \langle A, R \rangle$ be a finite argumentation framework. An argument $a \in A$ is acceptable w.r.t. a subset S of A if and only if for every $b \in A$ s.t. $(b, a) \in R$, there exists $c \in S$ s.t. $(c, b) \in R$. A set of arguments is acceptable w.r.t. S when each of its elements is acceptable w.r.t. S .

In the graph theory literature, a set of vertices which is acceptable w.r.t. itself is said to be *semidominant*.

A second important notion is the notion of absence of conflicts. Intuitively, two arguments should not be considered together whenever one of them attacks the other one.

Definition 3 (conflict-free sets).

Let $AF = \langle A, R \rangle$ be a finite argumentation framework. A subset S of A is conflict-free if and only if for every $a, b \in S$, we have $(a, b) \notin R$.

The conflict-free subsets of A which are maximal w.r.t. $\subseteq$ are called the **naive extensions** of AF in [3]. In the graph theory literature, such conflict-free sets are also called *independent sets*.

Requiring the absence of conflicts and the form of autonomy captured by self-acceptability leads to the notion of admissible set.

Definition 4 (admissible sets).

Let $AF = \langle A, R \rangle$ be a finite argumentation framework. A subset S of A is admissible if and only if S is conflict-free and acceptable w.r.t. S .

In the graph theory literature, a set of vertices which is both independent and semidominant is called a *semikernel*.

Example 2 (Example 1 (cont'ed)). $\{e, d\}$, $\{e, b\}$, $\{c\}$ are admissible sets given AF .

The significance of the concept of admissible sets is reflected by the fact that every extension of an argumentation framework under the standard semantics introduced by Dung (preferred, stable, complete and grounded extensions) is an admissible set, satisfying some form of optimality:

Definition 5 (extensions).

Let $AF = \langle A, R \rangle$ be a finite argumentation framework.

- A subset S of A is a preferred extension of AF if and only if it is maximal w.r.t. $\subseteq$ among the set of admissible sets for AF .
- A subset S of A is a stable extension of AF if and only if it is admissible and for every argument a from $A \setminus S$, there exists $b \in S$ s.t. $(b, a) \in R$.
- A subset S of A is a complete extension of AF if and only if it is admissible and it coincides with the set of arguments acceptable w.r.t. itself.
- A subset S of A is the grounded extension of AF if and only if it is the least element w.r.t. $\subseteq$ among the complete extensions of AF .

Example 3 (Example 1 (cont'ed)). Let $E_1 = \{a\}$, $E_2 = \{a, e, b\}$, $E_3 = \{a, c\}$ and $E_4 = \{a, d, e\}$. E_1 is the grounded extension of AF . E_2 , E_3 and E_4 are the preferred extensions of AF and the stable extensions of AF . E_1 , E_2 , E_3 and E_4 are the complete extensions of AF .

In the graph theory literature, sets S of vertices s.t. every vertex outside S is in the direct image of at least one element of S are also called *dominating sets*. Sets of vertices that are both independent and dominating are referred to as the *kernels* of the graph AF . The sets of vertices which are the *maximal semikernels* of the graph AF are the preferred extensions of AF .

Formally, complete extensions of AF can be characterized as the fixed points of its characteristic function $\mathcal{F}_{AF}$, and among them, the grounded extension of AF is the least element [6]:

Definition 6 (characteristic functions).

The characteristic function, denoted $\mathcal{F}_{AF}$, of an argumentation framework $AF = \langle A, R \rangle$ is defined as follows: $\mathcal{F}_{AF} : 2^A \rightarrow 2^A$
 $\mathcal{F}_{AF}(S) = \{a \mid a \text{ is acceptable w.r.t. } S\}$

Finally, several notions of acceptability of an argument (or more generally a set of arguments) can be defined by requiring the membership to one (credulous acceptability) or every extension (skeptical acceptability) of a specific kind. Obviously enough, credulous acceptability and skeptical acceptability w.r.t. the grounded extension coincide, since the grounded extension of an argumentation framework is unique.

Among other things, Dung has shown that every argumentation framework AF has at least one preferred extension, while it may have zero, one or many stable extensions.

The purest argumentation frameworks AF in Dung's theory are those for which all the notions of acceptability coincide. This means that AF has a unique complete extension (the grounded one), which is also stable and preferred.

Definition 7. An argumentation framework $AF = \langle A, R \rangle$ is well-founded if and only if there does not exist an infinite sequence $a_0, a_1 \dots a_n \dots$ of arguments from A , such that for each i , $(a_{i+1}, a_i) \in R$.

Proposition 1. Every well-founded argumentation framework has exactly one complete extension which is grounded, preferred and stable.

Dung has provided a sufficient condition for an argumentation framework AF to satisfy this requirement, the well-foundation of AF :

Proposition 2. Let $AF = \langle A, R \rangle$ be a finite argumentation framework. AF is well-founded if there is no cycle in the digraph $\langle A, R \rangle$.

Dung has also shown that every stable extension is preferred and every preferred extension is complete; however, none of the converse inclusions holds. When all the preferred extensions of an argumentation framework are stable ones, the framework is said to be coherent:

Definition 8 (coherent argumentation frameworks).

Let $AF = \langle A, R \rangle$ be a finite argumentation framework. AF is coherent if and only if every preferred extension of AF is also stable.

Example 4 (Example 1 (cont'd)). Every preferred extension of AF is a stable extension as well. Hence AF is coherent.

This is particularly interesting since for any coherent AF , the notion of credulous (resp. skeptical) acceptability w.r.t. the preferred arguments coincides with the notion of credulous (resp. skeptical) acceptability w.r.t. the stable arguments.

Since the grounded extension of AF is the least complete extension of it, it is included in every preferred extension of AF (hence in every stable extension of AF). This shows that the notion of acceptability w.r.t. the grounded extension is always at least as demanding as any form of credulous or skeptical acceptability w.r.t. the preferred extensions or the stable ones (except for credulous acceptability w.r.t. the stable extensions when no such extensions exist since no argument can be accepted in that case for such semantics — note that such an exception cannot be the case when AF is coherent).

Nevertheless, the grounded extension of AF is not always equal to the intersection of all its preferred extensions. Interesting argumentation frameworks are those for which this condition is satisfied:

Definition 9 (relatively grounded argumentation frameworks).

Let $AF = \langle A, R \rangle$ be a finite argumentation framework. AF is relatively grounded if and only if its grounded extension is equal to the intersection of all its preferred extensions.

Example 5 (Example 1 (cont'd)). $E_2 \cap E_3 \cap E_4 = E_1$. Hence AF is relatively grounded.

In this case, the notion of skeptical acceptability w.r.t. the preferred extensions coincides with the notion of acceptability w.r.t. the grounded extension.

3 Symmetric Argumentation Frameworks

3.1 Definitions and Properties

Let us now make precise the argumentation frameworks we are interested in.

Definition 10 (symmetric argumentation frameworks).

A symmetric argumentation framework is a finite argumentation framework $AF = \langle A, R \rangle$ where R is assumed symmetric, nonempty and irreflexive.

Example 6 (Example 1 (cont'd)). AF is a symmetric argumentation framework.

First of all, it is easy to show that no symmetric argumentation framework is among the purest ones:

Proposition 3. *No symmetric argumentation framework is well-founded.*

Proof. Since R is nonempty and symmetric, a cycle can always be found in AF . ■

Nevertheless, this does not prevent symmetric argumentation frameworks from exhibiting interesting properties. An easy result is:

Proposition 4. *Let $AF = \langle A, R \rangle$ be a symmetric argumentation framework. $S \subseteq A$ is admissible if and only if S is conflict-free.*

Proof. Since R is symmetric, every argument a of A defends itself against all the arguments which attack it, so every $a \in A$ is acceptable w.r.t a . Hence, for all $S \subseteq A$, every $a \in A$ is acceptable w.r.t. $S \cup \{a\}$. Hence, for all $S \subseteq A$, every $a \in S$ is acceptable w.r.t. S . Hence, S is admissible if S is conflict-free. ■

Thus, the preferred extensions of a symmetric $AF = \langle A, R \rangle$ are the maximal subsets of A w.r.t. $\subseteq$ among those which are conflict-free, i.e. the naive extensions of AF [3]. In particular, every conflict-free subset of A is included in a preferred extension of AF .

Another consequence is that:

Proposition 5. *Every symmetric argumentation framework is coherent.*

Proof. Every preferred extension $E \subseteq A$ is a naive extension. Hence, each argument not in E is in conflict with E . Since R is symmetric, each argument not in E is attacked by E . Hence, E is a stable extension. ■

Since every symmetric argumentation framework has a preferred extension, every symmetric argumentation framework has a stable extension, which is necessarily nonempty. Actually, this is an easy consequence of a more general result from graph theory stating that symmetric graphs are kernel perfect. This means that every induced subgraph of a symmetric graph has a kernel.

Proposition 6. *Let $AF = \langle A, R \rangle$ be a symmetric argumentation framework. Every $a \in A$ belongs to at least one preferred (or equivalently, stable or naive) extension of AF .*

Proof. Immediate, since R is irreflexive and symmetric. ■

Example 7 (Example 1 (cont'd)). $E_2 \cup E_3 \cup E_4 = A$. Hence every argument of A belongs to a preferred extension of AF .

As to the grounded extension, we can prove that:

Proposition 7. *Let $AF = \langle A, R \rangle$ be a symmetric argumentation framework. The grounded extension of AF is given by $\{a \in A \mid \nexists b \in A, (b, a) \in R\}$.*

Proof. According to Definition 6, $\mathcal{F}_{AF}(\emptyset)$ is the set of arguments of AF which are not attacked. There are two cases:

1. Either every argument of A is attacked. Then $\mathcal{F}_{AF}(\emptyset) = \emptyset$ is the least complete extension of AF (w.r.t. $\subseteq$). Hence $\emptyset$ is the grounded extension of AF .
2. Or some arguments of A are not attacked. Let $S' = \mathcal{F}_{AF}(\emptyset)$ be the set of such arguments. Since R is symmetric, if an argument is not attacked, then it does not attack any argument. Hence, there is no $a \in A \setminus S'$ s.t. a is acceptable w.r.t. S' . Hence $\mathcal{F}_{AF}^2(\emptyset) = \mathcal{F}_{AF}(S') = S'$. So, S' is the least complete extension of AF (w.r.t. $\subseteq$). Hence S' is the grounded extension of AF . ■

Subsequently, the grounded extension of AF can be computed in time linear in $|AF|$ in the worst case.

We have also shown that:

Proposition 8. *Let $AF = \langle A, R \rangle$ be a symmetric argumentation framework. $a \in A$ belongs to every preferred (or equivalently, stable or naive) extension of AF if and only if there is no $b \in A$ s.t. $(b, a) \in R$.*

Proof.

$\Leftarrow$ Immediate from Proposition 7 and the fact that the grounded extension is included into every preferred extension.

$\Rightarrow$ Let $b \in A$ such that $(b, a) \in R$. According to Proposition 6, there is a preferred extension E such that $b \in E$. But a belongs to E . Thus E is not conflict-free. So, b does not exist. ■

A direct corollary of this proposition is the following one:

Proposition 9. *Every symmetric argumentation framework is relatively grounded.*

Proof. Immediate from Propositions 7 and 8. ■

Example 8 (Example 1 (cont'd)). a is not attacked. a belongs to every preferred extension of AF and it is the unique argument of the grounded extension E_1 of AF .

As a consequence, there are at most two distinct forms of acceptability for symmetric argumentation frameworks: all the forms of skeptical acceptability coincide with the notion of acceptability w.r.t. the grounded extension; credulous acceptability w.r.t.

preferred extensions and credulous acceptability w.r.t. stable extensions coincide with credulous acceptability w.r.t. naive extensions. Nevertheless, according to Proposition 6 credulous acceptability for single arguments is not so interesting since it trivializes for symmetric argumentation frameworks.

Accordingly, one has to consider more general acceptability problems if one wants to get more than one semantics, which is expected here; indeed, skeptical acceptability is rather poor since it characterizes as acceptable only those arguments of A which are not attacked.

3.2 Acceptability Problems and Complexity Issues

This is why we turn to acceptability problems for *sets of arguments*, i.e., the question is now to determine whether or not it is reasonable to accept some arguments *together*:

Definition 11 (acceptability problems).

$\text{ACCEPTABILITY}_{I,E}$ is the following decision problem (also viewed as the language of its positive instances in the usual way):

- **Input:** A finite argumentation framework $AF = \langle A, R \rangle$ and a set of arguments $S \subseteq A$.
- **Question:** Is S included into:
 $I=\forall$: every E extension of AF ?
 $I=\exists$: at least one E extension of AF ?
 where E is either N (naive), P (preferred), S (stable), C (complete) or G (grounded).

For instance, $\text{ACCEPTABILITY}_{\forall,S}$ denotes the skeptical acceptability problem under the stable semantics. We also use the notation $\text{ACCEPTABILITY}_{\cdot,G}$ to denote the acceptability problem under the grounded semantics (obviously enough, $\text{ACCEPTABILITY}_{\cdot,G} = \text{ACCEPTABILITY}_{\forall,G} = \text{ACCEPTABILITY}_{\exists,G}$ since an argumentation framework always has a unique grounded extension).

We can easily complete previous complexity results for skeptical acceptability of single arguments [25, 26]:

Proposition 10. *The following complexity results hold:²*

- $\text{ACCEPTABILITY}_{\forall,P}$ is Π_2^P -complete.
- $\text{ACCEPTABILITY}_{\forall,S}$ is coNP-complete.
- $\text{ACCEPTABILITY}_{\forall,C} = \text{ACCEPTABILITY}_{\cdot,G}$ is in P.
- $\text{ACCEPTABILITY}_{\forall,N}$ is in P.

Proof. Clearly enough, considering sets of arguments has no impact w.r.t. skeptical acceptability whatever the underlying semantics: a set S of arguments is skeptically acceptable if and only if S is a subset of all the extensions under consideration if and

² We assume the reader acquainted with basic notions of complexity theory; see e.g., [27] otherwise.

only if every element of S is skeptically acceptable. Hence the complexity of skeptical acceptability for sets of arguments coincides with the corresponding complexity of skeptical acceptability for single arguments, as identified by Dunne and Bench-Capon (when the set of arguments is finite and the attacks relation is not empty) [26]. Now, since the grounded extension of an argumentation framework AF is the intersection of all its complete extensions, it also comes that the two languages $\text{ACCEPTABILITY}_{\forall, C}$ and $\text{ACCEPTABILITY}_{\exists, C}$ coincide. Finally, a set of arguments S is included into every naive extension of $AF = \langle A, R \rangle$ if and only if S is conflict-free and for every argument $a \in A \setminus S$ and every argument $b \in S$ if $(a, b) \in R$ then $(a, a) \in R$. This can be tested in time polynomial in $|AF| + |S|$.

The picture is not the same when credulous acceptability is considered since it can be the case that both arguments a and b are credulously acceptable (this is always the case in presence of symmetric argumentation frameworks) but that the set $\{a, b\}$ does not belong to any of the selected extensions.

Example 9 (Example 1 (cont'd)). $c \in E_3$ and $d \in E_4$. Hence each of c and d is credulously acceptable. However, it is not cautious to believe in the set of arguments $\{c, d\}$ because this set is not conflict-free.

Nevertheless, considering sets of arguments instead of arguments alone does not lead to a complexity shift:

Proposition 11. *The following complexity results hold:*

- $\text{ACCEPTABILITY}_{\exists, P} = \text{ACCEPTABILITY}_{\exists, C}$ is NP-complete.
- $\text{ACCEPTABILITY}_{\exists, S}$ is NP-complete.
- $\text{ACCEPTABILITY}_{\exists, N}$ is in P.

Proof. The equality $\text{ACCEPTABILITY}_{\exists, P} = \text{ACCEPTABILITY}_{\exists, C}$ comes easily from the fact that the preferred extensions of an argumentation framework AF are exactly the complete extensions of AF which are maximal w.r.t. $\subseteq$ (this is a straightforward consequence of the fact that every preferred extension of AF is a complete extension of AF and that every admissible set of arguments of AF (including its complete extensions) is included in a preferred extension of AF (Theorem 2 from [6])). Then the membership results come from the following nondeterministic algorithms running in time polynomial in the input size: guess $S' \subseteq A$ then check that S' is a complete (resp. stable) extension of AF and that $S \subseteq S'$. It is easy to show that the check step can be done in (deterministic) polynomial time. The hardness results are direct consequences of the fact that their restrictions to the case S contains a single argument are already NP-hard [25, 26]. Finally checking whether a set S of argument belongs to a naive extension is equivalent to checking whether S is conflict-free, which can be done easily in polynomial time. ■

One can observe that the notion of complete extension does not lead to semantics which differ from semantics obtained when some other extensions are considered (thus, skeptical acceptability w.r.t. complete extensions coincides with acceptability w.r.t. the grounded extension while credulous acceptability w.r.t. complete extensions coincides

with credulous acceptability w.r.t. preferred extensions); this explains why in Dung's work the notion of complete extension is viewed more as a link between preferred extensions and the grounded one than as a semantics *per se*.

Now, considering symmetric frameworks leads complexity to decrease in a significant way:

Proposition 12. *Let us consider the restriction of $\text{ACCEPTABILITY}_{I,E}$ when AF is symmetric. Under this requirement, one can prove that:*

- $\text{ACCEPTABILITY}_{\forall,P} = \text{ACCEPTABILITY}_{\forall,S} = \text{ACCEPTABILITY}_{\forall,C}$
 $= \text{ACCEPTABILITY}_{\forall,G} = \text{ACCEPTABILITY}_{\forall,N}$ is in $\mathbf{P}$.
- $\text{ACCEPTABILITY}_{\exists,P} = \text{ACCEPTABILITY}_{\exists,S} = \text{ACCEPTABILITY}_{\exists,C}$
 $= \text{ACCEPTABILITY}_{\exists,N}$ is in $\mathbf{P}$.

Proof. The first point is a direct consequence of Propositions 7 and 8. The equalities at the second point come from Propositions 4 and 5 and from the facts that the preferred extensions of an argumentation framework AF are exactly the complete extensions of AF which are maximal w.r.t. $\subseteq$ and that every admissible set of arguments of AF (including its complete extensions) is included in a preferred extension of AF (see the proof of Proposition 11). Tractability comes from Proposition 4: $S \subseteq A$ is included in a preferred extension of AF – or equivalently, included in a stable extension or included in a complete extension or included in a naive extension – if and only if S is conflict-free. Note that while credulous acceptability can be decided easily, the notion does not trivialize when S is not a singleton (which means that the set of positive instances is not always the set of all instances of the problem). ■

To sum up, the various semantics in Dung's theory applied to symmetric frameworks lead to consider a set of arguments as acceptable when (1) every element of it is not attacked (the skeptical acceptability) or (2) it is conflict-free (the credulous acceptability). In both cases, acceptability can be decided in an efficient way.

4 Conclusion

We have studied the properties offered by symmetric argumentation frameworks, under the (quite realistic) assumptions that the set of arguments is finite and the attacks relation is nonempty and irreflexive. Such frameworks are shown coherent and relatively grounded. This ensures that the various notions of acceptability proposed so far reduce at most to two. Extending them to sets of arguments, one obtains two notions of acceptability which are rather simple in essence but tractable; we have shown that this contrasts with the general case for which all the generalized forms of acceptability are intractable (under the usual assumptions of complexity theory), except for the ones based on grounded or naive extensions.

This work calls for several perspectives. One of them consists in investigating other preference criteria as a basis for additional semantics for argumentation frameworks. Indeed, refining preferred extensions can prove valuable whenever skeptical (resp. credulous) acceptability w.r.t. preferred extensions is considered too cautious (resp. too liberal). For instance, one can select the preferred extensions which are maximal w.r.t.

cardinality. One can also associate to every preferred set S of arguments of AF the sum (or the maximum) of the numbers of attacks against each element of S ; on this ground, one can prefer the admissible sets associated to the least numbers if one thinks that a set of arguments which is not attacked is better than a set of arguments which is massively attacked. One can also adhere to the opposite point of view and prefer in a Popperian style sets of arguments which are robust enough to survive to many attacks. A second perspective consists in investigating the acceptability issue from the complexity point of view whenever a limited amount of non symmetric attacks is allowed. Finally, it would be interesting to point out other graph-theoretic properties for argumentation frameworks which would ensure tractable inference under various semantics.

References

1. Toulmin, S.: *The Uses of Argument*. Cambridge University Press (1958)
2. Prakken, A., Vreeswijk, G.: *Logics for defeasible argumentation*. Volume 4 of *Handbook of Philosophical Logic*, Second edition. Kluwer Academic Publishers (2002) 219–318
3. Bondarenko, A., Dung, P.M., Kowalski, R., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence* **93** (1997) 63–101
4. Parsons, S., Sierra, C., Jennings, N.: Agents that reason and negotiate by arguing. *Journal of Logic and Computation* **8** (1998) 261–292
5. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and complexity of some formal inter-agent dialogues. *Journal of Logic and Computation* **13** (2003) 348–376
6. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* **77** (1995) 321–358
7. Elvang-Gøransson, M., Fox, J., Krause, P.: Dialectic reasoning with inconsistent information. In: *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*. (1993) 114–121
8. Pollock, J.: How to reason defeasibly. *Artificial Intelligence* **57** (1992) 1–42
9. Simari, G., Loui, R.: A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence* **53** (1992) 125–157
10. Vreeswijk, G.: Abstract argumentation systems. *Artificial Intelligence* **90** (1997) 225–279
11. Elvang-Gøransson, M., Fox, J., Krause, P.: Acceptability of arguments as logical uncertainty. In: *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*. (1993) 85–90
12. Elvang-Gøransson, M., Hunter, A.: Argumentative logics: Reasoning with classically inconsistent information. *Data and Knowledge Engineering* **16** (1995) 125–145
13. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. *Artificial Intelligence* **128** (2001) 203–235
14. Amgoud, L., Cayrol, C.: On the acceptability of arguments in preference-based argumentation. In: *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*. (1998) 1–7
15. Amgoud, L., Cayrol, C.: Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning* **29** (2002) 125–169
16. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence* **34** (2002) 197–215

17. Cayrol, C.: From non-monotonic syntax-based entailment to preference-based argumentation. In: Proceedings of the 3rd European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty. Volume 946 of Lecture Notes on Artificial Intelligence. (1995)
18. Cayrol, C.: On the relation between argumentation and non-monotonic coherence-based entailment. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence. (1995)
19. Dimopoulos, Y., Nebel, B., Toni, F.: On the computational complexity of assumption-based argumentation for default reasoning. *Artificial Intelligence* **141** (2002) 57–78
20. Baroni, P., Giacomin, M., G.Guida: Extending abstract argumentation systems theory. *Artificial Intelligence* **120** (2000) 251–270
21. Baroni, P., Giacomin, M.: Solving semantic problems with odd-length cycles in argumentation. In: Proceedings of the 7th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty. Volume 2711 of Lecture Notes on Artificial Intelligence. (2003) 440–451
22. Baroni, P., Giacomin, M.: A recursive approach to argumentation: motivation and perspectives. In: Proceedings of the 10th International Workshop on Non-Monotonic Reasoning. (2004) 50–58
23. Cayrol, C., Doutre, S., Lagasque-Schiex, M.C., Mengin, J.: Minimal defence: a refinement of the preferred semantics for argumentation frameworks. In: Proceedings of the 9th International Workshop on Non-Monotonic Reasoning. (2002) 408–415
24. Cayrol, C., Lagasque-Schiex, M.C.: Gradual handling of contradiction in argumentation frameworks. In: Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems. (2002) 83–90
25. Dimopoulos, Y., Torres, A.: Graph theoretical structures in logic programs and default theories. *Theoretical Computer Science* **170** (1996) 209–244
26. Dunne, P., Bench-Capon, T.: Coherence in finite argument system. *Artificial Intelligence* **141** (2002) 187–203
27. Papadimitriou, C.: Computational complexity. Addison-Wesley (1994)

Evaluating Argumentation Semantics with Respect to Skepticism Adequacy

Pietro Baroni and Massimiliano Giacomin

Università di Brescia,
Dipartimento di Elettronica per l'Automazione,
Via Branze 38, I-25123 Brescia, Italy
{baroni, giacomin}@ing.unibs.it

Abstract. Analyzing argumentation semantics with respect to the notion of skepticism is an important issue for developing general and well-founded comparisons among existing approaches. In this paper, we show that the notion of skepticism plays also a significant role in order to better understand the behavior of a specific semantics in different situations. Building on an articulated classification of argument justification states into seven distinct classes and on the definition of a weak and a strong version of skepticism relation, we define the property of skepticism adequacy of an argumentation semantics, which basically consists in requiring a lesser commitment when transforming a unidirectional attack into a mutual one. We then verify the skepticism adequacy of some literature proposals and obtain the rather surprising result that some semantics fail to satisfy this basic property.

1 Introduction

A variety of approaches to the definition of argumentation semantics are available in the literature. On the one hand, several traditional proposals, such as stable [5, 8], grounded [6] and preferred [5] semantics, are encompassed in the well-established theory of [5], based on the unifying notion of admissibility. On the other hand, some counterintuitive behaviors exhibited by any admissibility-based semantics, and in particular by preferred semantics, have been recently pointed out in [1], where we have proposed an original semantics, called CF2, able to overcome these limitations. Exploiting the ideas initially introduced in [1], a recursive schema for argumentation semantics has been subsequently identified [4] and four novel semantics based on this schema have been defined and compared in [2]. In the face of such a variety of existing proposals, comparisons between alternative semantics have been often carried out by considering specific examples where their behaviors significantly differ and pointing out which of them appears intuitively more sound. This is for instance the case of “floating arguments”, used to compare unique-status with respect to multiple-status approaches [9], or of odd-length cycles, used to compare preferred semantics with CF2 semantics in [1]. While the analysis of

single examples may provide very insightful indications about the relationships existing between different semantics, it appears that conceptual tools for analysis and comparison at a more general level are also needed. The

introduced in [3] provides a contribution in this direction. Starting from an articulated classification of the possible justification states of an argument, two versions (weak and strong) of the skepticism relation have been identified, which entail two distinct partial orders on the justification states with respect to their level of commitment. The skepticism relation turns out to be a useful tool for analysis in order to compare the behavior of different proposals, at a general level, with reference to the same argumentation framework. Some results in this direction are provided in [3]. In this paper we take a different perspective, concerning skepticism analysis at an level. In fact, another interesting question concerns the characterization of how each single semantics behaves in the light of modifications introduced in the argumentation framework. In particular, as discussed below, there are modifications of the argumentation framework which should intuitively lead to a lesser level of commitment: it is then interesting to verify whether this intuition is respected by a given semantics at a formal level in terms of the skepticism relation. The present work aims at setting up the formal framework underlying this kind of analysis and then at applying it to some significant proposals of argumentation semantics. The paper is organized as follows. In Sect. 2 the background concepts of argumentation semantics are recalled, while in Sect. 3 the skepticism relation is defined. Section 4 sets up the framework for intra-semantics analysis by introducing the property of and applies it to the cases of grounded, preferred and CF2 semantics. Finally Sect. 5 concludes the paper.

2 Reviewing Argumentation Semantics

Our work adopts as a basic reference the general theory proposed by Dung [5] which is based on the primitive notion of :

Definition 1. A $AF = \langle \mathcal{A}, \rightarrow \rangle$, $\mathcal{A}$ $\rightarrow \subseteq (\mathcal{A} \times \mathcal{A})$ $\mathcal{A}$.

The idea is that arguments are simply conceived as the elements of the set $\mathcal{A}$, whose origin is not specified, and the interaction between them is modeled by the binary relation of attack $\rightarrow$. An argumentation framework $AF = \langle \mathcal{A}, \rightarrow \rangle$ can be represented as a directed graph, called a , where nodes are the arguments and edges correspond to the elements of the attack relation $\rightarrow$. Given a node $\alpha \in \mathcal{A}$, we define $\text{parents}_{AF}(\alpha) = \{\beta \in \mathcal{A} \mid \beta \rightarrow \alpha\}$. Since we will consider properties of sets of arguments, we extend the attack relation $\rightarrow$ as follows: given an argument α and a set of arguments S , $S \rightarrow \alpha$ iff $\exists \beta \in S : \beta \rightarrow \alpha$, $\alpha \rightarrow S$ iff $\exists \beta \in S : \alpha \rightarrow \beta$. Moreover, we will use the notion of of AF to a given subset $S \subseteq \mathcal{A}$, defined as $AF \downarrow_S = \langle S, \rightarrow \cap (S \times S) \rangle$.

Defining a specific amounts to specifying the criteria for deriving from an argumentation framework a set of , each

one representing a conflict-free set of arguments deemed to be collectively acceptable. Given a generic argumentation semantics $\mathcal{S}$, the set of extensions of a given argumentation framework $AF = \langle \mathcal{A}, \rightarrow \rangle$ prescribed by $\mathcal{S}$ will be indicated as $\mathcal{E}_{\mathcal{S}}(AF)$. The $\mathcal{S}$ -acceptability of each argument is then defined on the basis of $\mathcal{E}_{\mathcal{S}}(AF)$; in particular, an argument is considered as $\mathcal{S}$ -acceptable if it belongs to all extensions. Different semantics are therefore introduced by defining different notions of extension. Those in Dung's framework are all based on the concepts of acceptability and admissibility:

Definition 2. G

$$AF = \langle \mathcal{A}, \rightarrow \rangle:$$

- $A \in S \subseteq \mathcal{A}$ conflict-free $\iff \nexists \alpha, \beta \in S \quad \alpha \rightarrow \beta$.
- $A \in S \subseteq \mathcal{A}$ acceptable $\iff S \subseteq \mathcal{A} \quad \forall \beta \in \mathcal{A}, \beta \rightarrow \alpha \implies S \rightarrow \beta$.
- $A \in S \subseteq \mathcal{A}$ admissible $\iff S \subseteq \mathcal{A}$ conflict-free and $S \subseteq \mathcal{A}$ acceptable.

Then, the two traditional proposals of argumentation semantics can be introduced, namely the grounded and preferred semantics. The grounded semantics adheres to the so-called *single extension* property, since for a given argumentation framework AF it always identifies a single extension, called *grounded extension*, which can be defined as follows [5]:

Definition 3. G

$$AF = \langle \mathcal{A}, \rightarrow \rangle, \quad \text{characteristic function } F_{AF}, \quad \text{grounded extension } GE_{AF}, \quad \text{preferred extension } PE_{AF},$$

$$F^1 = F, F^{i+1}(F^i), \quad F_{AF}(E) = \bigcup_{i \geq 1} F_{AF}^i(\emptyset), \quad E \subseteq \mathcal{A}.$$

The grounded extension gives rise to a classification of arguments into three justification states, namely *justified*, *defeated*, and *provisionally defeated*, belonging to GE_{AF} and considered as justified, *attacked* by GE_{AF} and rejected, and *undecided*, that are neither included in GE_{AF} nor attacked by it, reflecting a sort of undecided state. Preferred semantics follows, instead, a multiple-status approach, by identifying a set of *preferred extensions*:

Definition 4. G

$$AF = \langle \mathcal{A}, \rightarrow \rangle, \quad E \subseteq \mathcal{A} \text{ preferred extension of } AF \iff E \subseteq \mathcal{A} \text{ admissible and } \nexists T \subseteq \mathcal{A} \text{ admissible such that } E \subset T.$$

In the context of preferred semantics, basically three justification states for an argument can be envisaged on the basis of its membership to extensions [5]: an argument may belong to all extensions, to no extension or to some (not all) of them, roughly corresponding to the states of undefeated, defeated and provisionally defeated in grounded semantics.

Being a multiple-status approach, preferred semantics supports a finer discrimination of the so-called *preferred extensions* [9, 7], which has been traditionally considered an advantage wrt. grounded semantics. However, in [1] we have

pointed out limitations of preferred semantics when dealing with odd-length cycles, and we have introduced a semantics called *CF2* overcoming them. This proposal is based on a definition of extensions along the (SCCs) of AF, namely the equivalence classes of nodes under the relation of mutual reachability, denoted as SCCS_{AF} :

Definition 5. G $\text{AF} = \langle \mathcal{A}, \rightarrow \rangle$, $E \subseteq \mathcal{A}$
 CF2 , $E \in \mathcal{RE}(\text{AF})$,

- $E \in \mathcal{MI}_{\text{AF}}$ $|\text{SCCS}_{\text{AF}}| = 1$
- $\forall S \in \text{SCCS}_{\text{AF}} (E \cap S) \in \mathcal{RE}(\text{AF} \downarrow_{S_{\text{AF}}^{\text{UP}}(E)})$

 $\mathcal{MI}_{\text{AF}}$ AF ,
 $S \subseteq \mathcal{A}$, $S_{\text{AF}}^{\text{UP}}(E) = \{\alpha \in S \mid \nexists \beta \in E : \beta \notin S, \beta \rightarrow \alpha\}$.

Due to space limitations, an intuitive explanation of the above definition cannot be given in this paper: the reader is referred to [1, 2, 4] for details and further analysis of *CF2*. An example of its application is given in Sect. 4.2.

3 Characterizing Skepticism

A traditional example of skepticism analysis concerns the comparison between grounded and preferred semantics, based on the observation that the former is more skeptical than the latter, since the grounded extension is included in all preferred extensions. This entails that all arguments that are undefeated (defeated) according to grounded semantics are also undefeated (defeated) according to preferred semantics. On the other hand, provisionally defeated arguments according to grounded semantics can generally assume any state according to preferred semantics. From this perspective, the comparison of skepticism between semantics is based on a relationship among extensions, while the relation holding at the level of justification states is regarded as a consequence of the one holding at the level of extensions: if a semantics is less skeptical than another then it assigns to each argument a state which features a higher level of commitment with respect to that assigned by the more skeptical one. In fact, intuition confirms that the state of provisionally defeated is by nature less committed with respect to both the states of undefeated and defeated, which are at the same (highest) level of commitment¹. Following an alternative perspective, one may introduce as a primitive notion the above mentioned order of justification states wrt. their level of commitment, and define a skepticism relation between semantics accordingly: if a semantics assigns to each argument a state which features a higher level of commitment then it is less skeptical.

¹ Note that the level of commitment must be clearly distinguished from the level of confidence (or credibility): the justification states featuring the highest and the lowest level of confidence have both the highest level of commitment.

Since justification states are a function of the set of extensions, following the first perspective guarantees a higher level of generality: any skepticism relationship based on justification states can be expressed also in terms of extensions, but not vice versa. Accordingly, we will start by introducing a

$\preceq$ on sets of extensions, where $\mathcal{E}_1 \preceq \mathcal{E}_2$ indicates that the set of extensions $\mathcal{E}_1$ is more skeptical than $\mathcal{E}_2$. Any basic skepticism relation induces a corresponding skepticism relation $\leq$ between semantics: $\mathcal{S}_1 \leq \mathcal{S}_2$ iff for any argumentation framework AF , $\mathcal{E}_{\mathcal{S}_1}(AF) \preceq \mathcal{E}_{\mathcal{S}_2}(AF)$. Finally, a partial order $\preceq$ on justification states reflecting their level of commitment is in turn induced: a justification state $JS1$ is less committed than a justification state $JS2$, denoted as $JS1 \preceq JS2$, iff there are an argumentation framework $AF = \langle \mathcal{A}, \rightarrow \rangle$, an argument $\alpha \in \mathcal{A}$ and two semantics $\mathcal{S}_1, \mathcal{S}_2$ with $\mathcal{S}_1 \leq \mathcal{S}_2$, such that $JS1$ and $JS2$ are the justification states assigned to α by $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively.

In order to develop the above concepts, the first step to take is a systematic analysis of the possible justification states of an argument. In fact, as pointed out in [3], the traditional identification of three states with two levels of commitment recalled above is insufficient for an adequate characterization of skepticism.

3.1 Justification States

As a starting point, we consider the relationship between an argument α and a particular extension E ; three main situations can be envisaged, namely

- $\alpha \in E$, if $\alpha \in E$;
- $\alpha \notin E$, if $\alpha \notin E \wedge E \rightarrow \alpha$;
- $\alpha \notin E$, if $\alpha \notin E \wedge E \not\rightarrow \alpha$.

Taking into account the existence of multiple extensions, one can consider that an argument can be in any of the above three states with respect to all, some or none of the extensions. This gives rise to 27 hypothetical combinations. It is however easy to see that some of them are impossible, for instance if an argument is in a given state with respect to all extensions this clearly excludes that it is in another state with respect to any extension. Directly applying this kind of considerations, seven possible $J \times S$ emerge for an argument α with respect to a set of extensions $\mathcal{E}$:

- JS1** $\forall E \in \mathcal{E}$, α is in E ;
- JS2** $\forall E \in \mathcal{E}$, α is definitely out of E ;
- JS3** $\forall E \in \mathcal{E}$, α is provisionally out of E ;
- JS4** $\exists E \in \mathcal{E}$ such that α is definitely out of E , $\exists E \in \mathcal{E}$ such that α is provisionally out of E , and $\nexists E \in \mathcal{E}$ such that α is in E ;
- JS5** $\exists E \in \mathcal{E}$ such that α is in E , $\exists E \in \mathcal{E}$ such that α is provisionally out of E , and $\nexists E \in \mathcal{E}$ such that α is definitely out of E ;
- JS6** $\exists E \in \mathcal{E}$ such that α is in E , $\exists E \in \mathcal{E}$ such that α is definitely out of E , and $\nexists E \in \mathcal{E}$ such that α is provisionally out of E ;
- JS7** $\exists E \in \mathcal{E}$ such that α is in E , $\exists E \in \mathcal{E}$ such that α is definitely out of E , and $\exists E \in \mathcal{E}$ such that α is provisionally out of E .

It is easy to see that if the semantics enforces a unique-status approach, i.e. $|\mathcal{E}| = 1$, then only *JS1*, *JS2* and *JS3* may hold. In case of the grounded semantics, i.e. $\mathcal{E} = \{\text{GE}_{\text{AF}}\}$, they correspond to the states of undefeated, defeated and provisionally defeated, respectively.

3.2 The Weak and Strong Skepticism Relations

Using as a basis the fact that, for any argumentation framework, the grounded extension is included in all preferred extensions, one may consider a generalization to the case of two multiple-status semantics prescribing that the extensions of S_1 satisfy some constraint of inclusion in those of S_2 . A direct way of achieving this generalization is given by the following basic skepticism relation $\preceq_W$:

Definition 6. G

$$\mathcal{E}_1 \preceq_W \mathcal{E}_2, \mathcal{E}_1 \preceq_W \mathcal{E}_2$$

$$\forall E_2 \in \mathcal{E}_2 \exists E_1 \in \mathcal{E}_1 : E_1 \subseteq E_2$$

T

$$\preceq_W.$$

In the following, we will refer to $\preceq_W$ and $\leq_W$ as

Relation $\leq_W$ is in a sense unidirectional, since it only constrains the extensions of S_2 , while $\mathcal{E}_{S_1}(\text{AF})$ may contain additional extensions unrelated to those of S_2 . One may wonder whether a more symmetric relationship is more appropriate, where it is also required that any extension of S_1 is included in one extension of S_2 . To this purpose, we introduce the following definition:

Definition 7. G

$$\mathcal{E}_1 \preceq_S \mathcal{E}_2, \mathcal{E}_1 \preceq_S \mathcal{E}_2$$

$$\forall E_2 \in \mathcal{E}_2 \exists E_1 \in \mathcal{E}_1 : E_1 \subseteq E_2,$$

$$\forall E_1 \in \mathcal{E}_1 \exists E_2 \in \mathcal{E}_2 : E_1 \subseteq E_2$$

T

$$\preceq_S.$$

In the following, we will refer to $\preceq_S$ and $\leq_S$ as

As shown in [3], the weak skepticism relation $\leq_W$ gives rise to the partial order of justification states whose Hasse diagram is shown in Fig. 1(a), which will be denoted as $\preceq_W$ in the following, while the partial order $\preceq_S$ induced by the strong skepticism relation $\leq_S$ is represented in Fig. 1(b). Basically, arcs connect

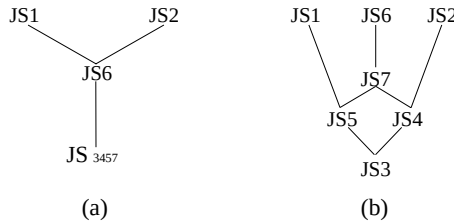


Fig. 1. The $\preceq_W$ and $\preceq_S$ semi-lattices of justification states

pairs of comparable states, and lower states are less committed than higher ones. Considering for instance Fig. 1(a), where JS_{3457} denotes the disjunction of the states listed in the subscript, the minimally committed state is JS_{3457} , while $JS1$ and $JS2$ are maximally committed. Then, given two semantics $\mathcal{S}_1 \leq_W \mathcal{S}_2$ if an argument α is in $JS6$ according to $\mathcal{S}_1$ then its justification state according to $\mathcal{S}_2$ is $JS1$, $JS2$, or $JS6$ itself. It is proved in [3] that both $\leq_W$ and $\leq_S$ are preorders, i.e. they are reflexive and transitive. As a further useful property, note that if $\mathcal{E}_1 = \{E_1\}$, which is always the case when the first semantics $\mathcal{S}_1$ is a unique-status approach, both $\leq_W$ and $\leq_S$ are equivalent to $\forall E_2 \in \mathcal{E}_2 \ E_1 \subseteq E_2$. In particular, if $\mathcal{S}_1$ and $\mathcal{S}_2$ are the grounded and the preferred semantics respectively, then the traditional relation between grounded and preferred semantics is recovered.

4 Analyzing Semantics Behavior

Having defined two alternative versions of the skepticism relation, let us investigate how it can support $\dots$ analysis by introducing the notion of skepticism adequacy for an argumentation semantics.

4.1 Defining Skepticism Adequacy

We aim at defining the skepticism adequacy of an argumentation semantics, referring to its behavior with respect to modifications of the argumentation framework whose expected impact on the level of commitment at a semantic level can be easily characterized from an intuitive point of view. To this purpose, let us consider the very simple argumentation framework presented in Fig. 2(a) consisting of two nodes α and β , where α attacks β but not vice versa. This is a situation where the status assignment of any argumentation semantics corresponds to the maximum level of commitment: it is universally accepted that α should be definitely justified and β definitely rejected. Now if we consider the argumentation framework of Fig. 2(b) where an attack from β to α has been added, we obtain a situation where clearly a lesser level of commitment is appropriate: given the mutual attack between the two arguments, neither of them can be assigned a definitely committed status and both should be rather assigned a status of the kind “provisionally defeated”, in absence of any reason for preferring either of them. The ability to discriminate between these situations is a fundamental requirement, which all the semantics previously mentioned satisfy.

Extending this reasoning, consider a couple of nodes α and β in a generic argumentation framework AF such that $\alpha \rightarrow \beta$ while $\beta \not\rightarrow \alpha$. Consider now an argumentation framework AF' obtained from AF by simply adding an attack relation from β to α while leaving all the rest unchanged. It seems reasonable

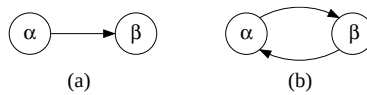


Fig. 2. A chain of two nodes and its simple variant

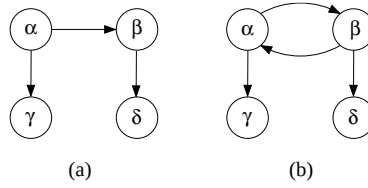


Fig. 3. Propagation of less committed states

to expect that the status assignment of the arguments in AF' does not feature a higher level of commitment with respect to AF . In fact, converting a unidirectional attack into a mutual one can only make the states of the involved nodes less committed (of course they can remain the same if they are strictly determined by other arguments, independently of the attack relations between α and β). In turn, having α or β in a less committed state may only give rise to other less committed states in the nodes they attack: intuitively, the more undecided is the state of an attacker, the more undecided should be the state of the attacked node, and, in turn, of the nodes attacked by the latter and so on. For example, consider the argumentation frameworks of Fig. 3 where the nodes γ and δ attacked respectively by α and β have been added. In the case represented in Fig. 3(a), γ is definitely rejected (as attacked by the undefeated node α) while δ is definitely accepted (in virtue of the [7] as its only defeater β is definitely rejected). In the argumentation framework of Fig. 3(b) both γ and δ should inherit a less committed state from their attackers, after the introduction of the mutual attack between α and β .

On these grounds, we define the property of skepticism adequacy of a semantics $\mathcal{S}$ with respect to a given basic skepticism relation $\preceq$:

Definition 8. G $\preceq$, $\mathcal{S} \preceq$
 $AF = \langle \mathcal{A}, \rightarrow \rangle$, $\alpha, \beta \in \mathcal{A} : \alpha \neq \beta$
 $\beta \wedge \alpha \rightarrow \beta$, $\mathcal{E}_{\mathcal{S}}(AF^{\rightleftharpoons(\beta, \alpha)}) \preceq \mathcal{E}_{\mathcal{S}}(AF)$, $AF^{\rightleftharpoons(\beta, \alpha)} = \langle \mathcal{A}, \rightarrow \cup \{(\beta, \alpha)\} \rangle$.

Skepticism adequacy appears to be an intuitive requirement: the analysis in the following subsection shows however that not all semantics satisfy it.

4.2 Verifying Skepticism Adequacy

As already mentioned, $\preceq_W$ and $\preceq_S$ are equivalent in the case of a unique-status approach, therefore, considering grounded semantics, we have just to prove that the grounded extension of an argumentation framework AF contains the grounded extension of $AF^{\rightleftharpoons(\beta, \alpha)}$. The skepticism adequacy of grounded semantics is demonstrated in Proposition 1, which requires a preliminary lemma.

Lemma 1. L $AF = \langle \mathcal{A}, \rightarrow \rangle$
 $\alpha, \beta \in \mathcal{A}$ $\alpha \rightarrow \beta$. G A^* A
 $A^* \subseteq A$ A AF , $F_{AF^{\rightleftharpoons(\beta, \alpha)}}(A^*) \subseteq F_{AF}(A)$.

P. Considering a generic $\gamma \in F_{AF^{\rightleftharpoons(\beta, \alpha)}}(A^*)$, we have to prove that $\gamma \in F_{AF}(A)$, i.e. that γ is acceptable with respect to A in AF . To this purpose, let

us consider a generic argument $\delta \in \text{parents}_{\text{AF}}(\gamma)$, and let us prove that $A \rightarrow \delta$ in AF. By definition of $\text{AF}^{\rightleftharpoons(\beta, \alpha)}$, it is easy to see that $\delta \in \text{parents}_{\text{AF}^{\rightleftharpoons(\beta, \alpha)}}(\gamma)$, and since $\gamma \in F_{\text{AF}^{\rightleftharpoons(\beta, \alpha)}}(A^*)$ it must be the case that $A^* \rightarrow \delta$ holds in $\text{AF}^{\rightleftharpoons(\beta, \alpha)}$. Since $A^* \subseteq A$, we also have that $A \rightarrow \delta$ in $\text{AF}^{\rightleftharpoons(\beta, \alpha)}$. Now, if this condition holds also in AF, then the claim is proved. Otherwise, by definition of $\text{AF}^{\rightleftharpoons(\beta, \alpha)}$ it must be the case that $\alpha = \delta$, $\beta \in A$ and $\delta \rightarrow \beta$ in AF. As a consequence, the hypothesis of admissibility of A entails that, also in this case, $A \rightarrow \delta$ in AF. $\square$

Proposition 1. G

$$\begin{array}{lll} \alpha, \beta \in \mathcal{A} & \alpha \rightarrow \beta, & \text{AF} = \langle \mathcal{A}, \rightarrow \rangle \\ & & \text{GE}_{\text{AF}^{\rightleftharpoons(\beta, \alpha)}} \subseteq \text{GE}_{\text{AF}}. \end{array}$$

P. Taking into account the definition of grounded extension, it is sufficient to prove that $\forall i \geq 1 \ F_{\text{AF}^{\rightleftharpoons(\beta, \alpha)}}^i(\emptyset) \subseteq F_{\text{AF}}^i(\emptyset)$. This can be easily proved by induction on i , taking into account Lemma 1 and the fact that $\forall i \geq 1 \ F_{\text{AF}}^i(\emptyset)$ is admissible [5]. In particular, in the basis case Lemma 1 can be applied with $A^* = A = \emptyset$ to prove that $F_{\text{AF}^{\rightleftharpoons(\beta, \alpha)}}(\emptyset) \subseteq F_{\text{AF}}(\emptyset)$, while in the induction step it can be applied with $A^* = F_{\text{AF}^{\rightleftharpoons(\beta, \alpha)}}^i(\emptyset)$ and $A = F_{\text{AF}}^i(\emptyset)$, where $A^* \subseteq A$ is inductively assumed, to prove that $F_{\text{AF}^{\rightleftharpoons(\beta, \alpha)}}^{i+1}(\emptyset) \subseteq F_{\text{AF}}^{i+1}(\emptyset)$. $\square$

In the case of multiple-status approaches, the two relations are not equivalent. As a simple example, consider again Fig. 2: it turns out that both for preferred and CF2 semantics AF admits as unique extension $\{\alpha\}$, while $\text{AF}^{\rightleftharpoons(\beta, \alpha)}$ admits $\{\alpha\}$ and $\{\beta\}$ as extensions. This clearly entails that, while $\preceq_W$ is satisfied, $\preceq_S$ is not, therefore preferred and CF2 semantics are not adequate with respect to the strong basic skepticism relation. Actually, this is due to the fact that, as pointed out in [3], $\preceq_S$ represents a very strong requirement for skepticism comparability. In fact, in multiple-status approaches less committed justification states typically arise from the presence of additional extensions, which however gives rise to incomparability according to $\preceq_S$. Therefore, in the context of multiple-status approaches, only $\preceq_W$ -adequacy is significant.

In order to verify the $\preceq_W$ -adequacy of preferred and CF2 semantics, let us consider the example shown in Fig. 4. As to preferred semantics, it turns out that $\mathcal{PE}_{\text{AF}} = \{\emptyset\}$ and $\mathcal{PE}_{\text{AF}^{\rightleftharpoons(\gamma, \delta)}} = \{\{\alpha, \delta\}\}$, therefore preferred semantics is not $\preceq_W$ -adequate. While somewhat surprising, this counterintuitive behavior has a counterpart at the level of justification states. In fact, according to preferred semantics all arguments in AF are provisionally defeated while in $\text{AF}^{\rightleftharpoons(\gamma, \delta)}$ two of them, namely α and δ , are undefeated. Other counterintuitive behaviors of preferred semantics when dealing with odd-length cycles have been analyzed in [1, 4, 2]. Turning to CF2 semantics, AF and $\text{AF}^{\rightleftharpoons(\beta, \alpha)}$ admit the same set of extensions, namely $\{\{\alpha, \delta\}, \{\beta, \delta\}, \{\gamma\}\}$. In fact, AF consists of two SCCs, i.e. $S_1 = \{\alpha, \beta, \gamma\}$ and $S_2 = \{\delta\}$. According to Definition 5, $(E \cap S_1)$ can be obtained by applying recursively the definition of $\mathcal{RE}(\text{AF})$ on $\text{AF} \downarrow_{S_1}$. Since $|\text{SCCS}_{\text{AF} \downarrow_{S_1}}| = 1$, the maximal conflict-free sets of $\text{AF} \downarrow_{S_1}$, i.e. $\{\alpha\}, \{\beta\}, \{\gamma\}$ are selected. Then, for each selection, $\mathcal{RE}(\text{AF} \downarrow_{S_2 \cup_{\text{AF}}^{\text{UP}}(E)})$ is evaluated. It coincides with $\{\delta\}$ except in the case selection $\{\gamma\}$ is considered, where $S_2 \cup_{\text{AF}}^{\text{UP}}(E) = \emptyset$ since $\gamma \rightarrow \delta$. On the other hand, $\text{AF}^{\rightleftharpoons(\beta, \alpha)}$ consists of a single SCC, therefore its maximal conflict-free sets are directly considered as extensions yielding the same results as above.

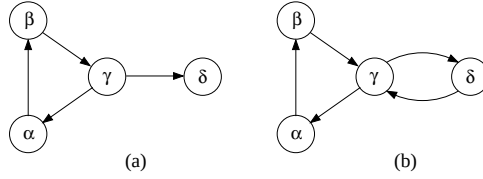


Fig. 4. A problematic example for preferred semantics

Therefore, in this case the condition of $\preceq_W$ -adequacy is satisfied and in both argumentation frameworks no argument is justified.

It is possible to prove that $\preceq_W$ -adequacy holds in general for CF2 semantics. This follows from the (actually stronger) result in Proposition 2, which requires two preliminary lemmas whose proofs are omitted due to space limitations.

Lemma 2. $F \quad AF, \mathcal{RE}(AF) \subseteq \mathcal{MI}_{AF}.$

Lemma 3. $L \quad AF = \langle \mathcal{A}, \rightarrow \rangle$
 $SCC \quad \Theta \subseteq \text{SCCS}_{AF}. \quad T \quad \bigcup_{S \in \Theta} S \quad \hat{S},$
 $E \subseteq \mathcal{A},$

$$(E \cap \hat{S}) \in \mathcal{RE}(AF \downarrow_{\hat{S}_{AF}^{UP}(E)}) \quad \forall S \in \Theta \quad (E \cap S) \in \mathcal{RE}(AF \downarrow_{S_{AF}^{UP}(E)})$$

Proposition 2. $G \quad AF = \langle \mathcal{A}, \rightarrow \rangle$
 $\alpha, \beta \in \mathcal{A} \quad \alpha \neq \beta \wedge \alpha \rightarrow \beta, \mathcal{RE}(AF) \subseteq \mathcal{RE}(AF^{\rightleftharpoons(\beta, \alpha)}).$

P . First, let us prove the claim in the case that $|\text{SCCS}_{AF^{\rightleftharpoons(\beta, \alpha)}}| = 1$. By Definition 5, we have in this case that $\mathcal{RE}(AF^{\rightleftharpoons(\beta, \alpha)}) = \mathcal{MI}_{AF^{\rightleftharpoons(\beta, \alpha)}}$. Since, by Lemma 2, $\mathcal{RE}(AF) \subseteq \mathcal{MI}_{AF}$, it is sufficient to prove that $\mathcal{MI}_{AF} = \mathcal{MI}_{AF^{\rightleftharpoons(\beta, \alpha)}}$. This directly follows from the fact that AF and $AF^{\rightleftharpoons(\beta, \alpha)}$ admit exactly the same conflict-free sets, since the addition of the edge (β, α) to AF does not generate additional conflicts in $AF^{\rightleftharpoons(\beta, \alpha)}$, due to the presence of (α, β) in AF .

Note, in particular, that the claim necessarily holds when AF consists of exactly two nodes, namely α and β .

The proof now proceeds by induction on the number of nodes, assuming inductively that the Proposition holds for any argumentation framework having a strictly lesser number of nodes than AF (in particular, strictly included in $\mathcal{A}$):

$$\forall AF' = \langle \mathcal{A}', \rightarrow' \rangle : \mathcal{A}' \subsetneq \mathcal{A} \wedge \alpha \in \text{parents}_{AF'}(\beta), \mathcal{RE}(AF') \subseteq \mathcal{RE}(AF'^{\rightleftharpoons(\beta, \alpha)}) \quad (1)$$

Of course, we have to consider only the case that $|\text{SCCS}_{AF^{\rightleftharpoons(\beta, \alpha)}}| > 1$, since the other case is already covered by the first part of the proof. Let $S_\alpha, S_\beta \in \text{SCCS}_{AF}$ be the SCCs of AF including α and β , respectively (notice that it may be the case that $S_\alpha = S_\beta$). In $AF^{\rightleftharpoons(\beta, \alpha)}$, all the nodes in S_α and S_β become mutually reachable with the addition of (β, α) , therefore there must be a strongly connected component $\hat{S} \in \text{SCCS}_{AF^{\rightleftharpoons(\beta, \alpha)}}$ such that $S_\alpha, S_\beta \subseteq \hat{S}$. Moreover, any path in AF is preserved in $AF^{\rightleftharpoons(\beta, \alpha)}$, and any new path includes the additional arc (β, α) : therefore, any SCC of AF either is merged into $\hat{S}$ or is preserved

unchanged in $\text{AF}^{\rightleftharpoons(\beta, \alpha)}$. As a consequence, the set SCCS_{AF} can be partitioned into two non-empty subsets Θ (including the SCCs merged into $\hat{S}$) and Ψ , related to the SCCs of $\text{AF}^{\rightleftharpoons(\beta, \alpha)}$ as follows:

$$\text{SCCS}_{\text{AF}^{\rightleftharpoons(\beta, \alpha)}} = \{\hat{S}\} \cup \Psi, \text{ where } \hat{S} = \bigcup_{S \in \Theta} S \quad (2)$$

and

$$S_\alpha \subseteq \hat{S}, S_\beta \subseteq \hat{S}, \hat{S} \subsetneq \mathcal{A} . \quad (3)$$

The fact that $\hat{S}$ is a strict subset of $\mathcal{A}$ follows from $|\text{SCCS}_{\text{AF}^{\rightleftharpoons(\beta, \alpha)}}| > 1$. Now, let us consider a generic extension $E \in \mathcal{RE}(\text{AF})$. According to Definition 5, we have that

$$\forall S \in \text{SCCS}_{\text{AF}} (E \cap S) \in \mathcal{RE}(\text{AF} \downarrow_{S_{\text{AF}}^{\text{UP}}(E)}) . \quad (4)$$

In order to simplify the notation, let us denote $\text{AF}^{\rightleftharpoons(\beta, \alpha)}$ as AF^* : we have to prove that $E \in \mathcal{RE}(\text{AF}^*)$, which according to Definition 5 holds iff

$$\forall S \in \text{SCCS}_{\text{AF}^*} (E \cap S) \in \mathcal{RE}(\text{AF}^* \downarrow_{S_{\text{AF}^*}^{\text{UP}}(E)}) .$$

Let us consider first a generic strongly connected component $S \in \Psi$. Since, according to (2) and (3), $\alpha \notin S$ and $\beta \notin S$, we obviously have that $S_{\text{AF}}^{\text{UP}}(E) = S_{\text{AF}^*}^{\text{UP}}(E)$ and $\text{AF} \downarrow_{S_{\text{AF}}^{\text{UP}}(E)} = \text{AF}^* \downarrow_{S_{\text{AF}^*}^{\text{UP}}(E)}$. By substitution in (4), this yields $(E \cap S) \in \mathcal{RE}(\text{AF}^* \downarrow_{S_{\text{AF}^*}^{\text{UP}}(E)})$, therefore only the analogous condition for $\hat{S}$ remains to be verified.

On the basis of (2), $\hat{S} = \bigcup_{S \in \Theta} S$, and according to (4) we have in particular that $\forall S \in \Theta (E \cap S) \in \mathcal{RE}(\text{AF} \downarrow_{S_{\text{AF}}^{\text{UP}}(E)})$. As a consequence, the application of Lemma 3 to Θ yields

$$(E \cap \hat{S}) \in \mathcal{RE}(\text{AF} \downarrow_{\hat{S}_{\text{AF}}^{\text{UP}}(E)}) \quad (5)$$

where, taking into account that $\alpha, \beta \in \hat{S}$, we have that

$$\hat{S}_{\text{AF}}^{\text{UP}}(E) = \hat{S}_{\text{AF}^*}^{\text{UP}}(E) . \quad (6)$$

In order to get to the desired conclusion, we consider two cases for α and β . In the first case, $\alpha \notin \hat{S}_{\text{AF}}^{\text{UP}}(E)$ or $\beta \notin \hat{S}_{\text{AF}}^{\text{UP}}(E)$. Since the additional edge (β, α) does not belong to $\text{AF}^* \downarrow_{\hat{S}_{\text{AF}}^{\text{UP}}(E)}$, we have that $\text{AF} \downarrow_{\hat{S}_{\text{AF}}^{\text{UP}}(E)} = \text{AF}^* \downarrow_{\hat{S}_{\text{AF}}^{\text{UP}}(E)}$, which according to (6) is in turn equal to $\text{AF}^* \downarrow_{\hat{S}_{\text{AF}^*}^{\text{UP}}(E)}$. As a consequence, in this case the conclusion directly follows by substitution in (5).

Let us now turn to the other case, namely $\alpha \in \hat{S}_{\text{AF}}^{\text{UP}}(E)$ and $\beta \in \hat{S}_{\text{AF}}^{\text{UP}}(E)$, and let us consider the argumentation framework $\text{AF} \downarrow_{\hat{S}_{\text{AF}}^{\text{UP}}(E)}$, which obviously includes the edge (α, β) . Since $\hat{S}_{\text{AF}}^{\text{UP}}(E) \subsetneq \mathcal{A}$ by (3), the induction hypothesis (1) can be applied with $\text{AF}' = \text{AF} \downarrow_{\hat{S}_{\text{AF}}^{\text{UP}}(E)}$, yielding $\mathcal{RE}(\text{AF} \downarrow_{\hat{S}_{\text{AF}}^{\text{UP}}(E)}) \subseteq \mathcal{RE}((\text{AF} \downarrow_{\hat{S}_{\text{AF}}^{\text{UP}}(E)})^{\rightleftharpoons(\beta, \alpha)})$. Taking into account (5), it turns out that $(E \cap \hat{S}) \in \mathcal{RE}((\text{AF} \downarrow_{\hat{S}_{\text{AF}}^{\text{UP}}(E)})^{\rightleftharpoons(\beta, \alpha)})$. It is easy to see that $(\text{AF} \downarrow_{\hat{S}_{\text{AF}}^{\text{UP}}(E)})^{\rightleftharpoons(\beta, \alpha)} = \text{AF}^* \downarrow_{\hat{S}_{\text{AF}}^{\text{UP}}(E)}$, yielding $(E \cap \hat{S}) \in \mathcal{RE}(\text{AF}^* \downarrow_{\hat{S}_{\text{AF}}^{\text{UP}}(E)})$. Substituting from (6), we finally get the desired conclusion that $(E \cap \hat{S}) \in \mathcal{RE}(\text{AF}^* \downarrow_{\hat{S}_{\text{AF}^*}^{\text{UP}}(E)})$. $\square$

5 Conclusions

Building on the skepticism relations introduced in [3], we have defined the notion of skepticism adequacy of a given argumentation semantics. Only the weak version of this notion is appropriate in the context of multiple-status approaches, while the weak and strong relations coincide in the case of unique-status approaches. As to the latter context, grounded semantics turns out to be adequate, as to the former, the recently introduced CF2 semantics satisfies skepticism adequacy while preferred semantics does not. While problems of preferred semantics when dealing with specific examples have been discussed in [1, 4, 2], this result concerns a more abstract property and confirms that CF2 represents an interesting alternative to overcome these limitations.

Acknowledgements. We thank the referees for their helpful comments.

References

1. Baroni, P., Giacomin, M.: Solving semantic problems with odd-length cycles in argumentation. In: Proc. of ECSQARU 2003, Aalborg, Denmark, LNAI 2711, Springer-Verlag (2003) 440–451
2. Baroni, P., Giacomin, M.: A recursive approach to argumentation: motivation and perspectives. In: Proc. of the 10th International Workshop on Non-Monotonic Reasoning (NMR 2004), Whistler BC, Canada (2004) 50–58
3. Baroni, P., Giacomin, M., Giovanni, G.: Towards a formalization of skepticism in extension-based argumentation semantics. In: Proc. 4th Workshop on Computational Models of Natural Argument (CMNA 2004), Valencia, Spain (2004) 47–52
4. Baroni, P., Giacomin, M.: A general recursive schema for argumentation semantics. In: Proc. of ECAI 2004, Valencia, Spain (2004) 783–787
5. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming, and n-person games. *Artificial Intelligence* **77** (1995) 321–357
6. Pollock, J.L.: How to reason defeasibly. *Artificial Intelligence* **57** (1992) 1–42
7. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In Gabbay, D., Guenther, F., eds.: *Handbook of Philosophical Logic*. Kluwer, Dordrecht (2001)
8. Reiter, R.: A logic for default reasoning. *Artificial Intelligence* **13** (1980) 81–132
9. Schlechta, K.: Directly sceptical inheritance cannot capture the intersection of extensions. *Journal of Logic and Computation* **3** (1993) 455–467

Logic of Dementia Guidelines in a Probabilistic Argumentation Framework

Helena Lindgren and Patrik Eklund

Department of Computing Science,
University of Umeå SE-90187 Umeå, Sweden

Abstract. In order to give full support for differential diagnosis of dementia in medical practice, one single clinical guideline is not sufficient. A synthesis guideline has been formalized using core features from selected clinical guidelines for the purpose of providing decision support for clinicians in clinical practice. This guideline is sufficient for typical cases in the domain, but in order to give support in atypical cases additional clinical guidelines are needed which are pervaded with more uncertainty. In order to investigate the applicability of a probabilistic formalism language for the formalization of these guidelines, a case study was made using the qualitative probabilistic reasoning approach developed in [1]. The case study is placed in context of a foundational view of transformations between logics. The clinical decision-making motivation and utility for this transformation will be given together with some formal indications concerning this transformation.

Keywords: argumentation, dementia diagnosis, knowledge representation.

1 Introduction

Dementia is a medical domain, which gains an increasing attention because of the growing elderly population. The number of people suffering from such cognitive diseases as dementia is growing, which puts large strain on health care. Currently, efforts are made to improve dementia care in Sweden by educating the personnel and supporting teams in dementia care. A decision-support system with the scope of cognitive diseases is being developed for the purpose of supporting clinicians in their diagnostic reasoning and decision making concerning interventions [2]. The system should also disseminate clinical guidelines and support a continuing medical education in the users.

The domain knowledge residing in the clinical guidelines can be formalized in different ways. The language used in the guidelines are different in that some use sets of features as sufficient evidence for a diagnosis, while other use a language pervaded with more uncertainty, and therefore require more interpretation. Some guidelines use both. We chose to use the most common guideline in clinical practice in northern Sweden as the base in our system; the chapter concerning cognitive disorders in the fourth edition of *D*

(DSM-IV) developed by the American Psychiatric Association [3]. As will be shown, this guideline will not be sufficient for diagnosis of the different dementia types in clinical practice. Therefore a language is needed for the formalization of the guidelines, which express different degrees of certainty, and which can be used to present the evidence in a lucid way to the user. In the process of evaluating logical languages, this paper will show how the argumentation logic framework (here denoted $\mathcal{L}^{QP}$) developed in [1, 4] may be used for the purpose. In the framework the consequence relation $\vdash_{QP}$ defined in [1] is used to reason about changes in probabilities.

Traditionally, a logic $\mathcal{L}_\Sigma$ over a signature $\Sigma = (S, \Omega)$, where S is the set of sorts and Ω is the set of operators producing terms, consists of a set L_Σ (or L , if the underlying signature is clear from context) of formulas and a relation $\models \subseteq \text{Alg}(\Sigma) \times L$, where $\text{Alg}(\Sigma)$ is the set of all algebras over the signature Σ . We frequently write $\Phi \models \varphi$, $\Phi \subseteq L$, to mean that for all $A \in \text{Alg}(\Sigma)$ we have $A \models \varphi$, i.e. $(A, \varphi) \in \models$ whenever $A \models \Phi$. In this situation, satisfaction transforms to being $\models \subseteq PL \times L$, where P is the powerset functor. In this setting, $\models$ is called a *satisfaction* relation.

Logic calculus involving a set of inference rules establishes a relation $\vdash \subseteq PL \times L$, where again we write $\Phi \vdash \varphi$ instead of $(\Phi, \varphi) \in \vdash$. Traditional soundness and completeness thus means $\models = \vdash$. From a computational point of view we are always interested in the purely syntactic part, i.e. in $\mathcal{L} = (L, \vdash)$.

A *($\vdash$)-theory* for $(L, \vdash)$ is any set $\Phi \subseteq L$ of formulas such that $p \in \Phi$ whenever $\Phi \vdash p$, i.e. Φ is the set of all formulas derivable from Φ using the proof derivation $\vdash$.

A *propositional logic* $\mathcal{L}^\pi = (L^\pi, \models_\pi)$ can be viewed as a situation in form of a one-sorted signature where Ω consists of constants, $\neg$ as a unary operator, and $\wedge$ as a binary operator, with disjunction $\vee$ and implication $\rightarrow$ as the usual shorthand forms based on $\neg$ and $\wedge$.

Note that we may interpret formulas in Φ to be *truth* formulas. Thus we could equivalently say (p, true) is in Φ whenever p is in Φ . Similarly, we would have (q, false) in Φ , whenever $\neg q$ is in Φ . We then make truth values in the semantic domain more visible. This is useful when we extend to many-valuedness.

In argumentation logic, which in some sense extends propositional logic, however, with the binary operator $\rightarrow$ not acting as a usual material operator, but rather producing formulas based on terms over its signature. Well-formed formulas are traditionally all terms built upon the signature, and in the case of argumentation logic, includes also expressions $a \rightarrow b$ where a and b are terms (propositions) over the signature.

The argumentation logic $\mathcal{L}^{QP} = (L^{QP}, \vdash_{QP})$ comes equipped with a logic calculus but not strictly with a satisfaction relation, even if semantic domains are introduced. Further, as will be evident, $\vdash_{QP}$ is not a subset of $PL^{QP} \times L^{QP}$. Nevertheless, in our case study on dementia diagnosis, we will be interested in the transformation from $\mathcal{L}^\pi$ to $\mathcal{L}^{QP}$. A complete formal description of this transformation, however, is outside the scope of this paper. The purpose of the transformation is to allow different support to diagnostic reasoning, depending

on the complexity of the patient's case at hand. Why this is a desirable property of a decision support system for the domain, will be evident in the description of the domain knowledge in subsequent sections. Because of space limits, the guidelines presented here will be limited to rules which are used in the example. A complete description can be found in [5].

2 Argumentation Logic

Semantic domains in the framework are given by

$$\begin{aligned} S_\iota &= \{++, +-, +, 0, -, --, ?, \} \\ S_\sigma &= \{0, \Downarrow, \downarrow, \leftrightarrow, \uparrow, \Uparrow, 1, \updownarrow, \iota\} \end{aligned}$$

where S_ι and S_σ represent two dictionaries of signs defined in the framework which give information about changes in probabilities when arguments are introduced and combined. Roughly, $+$, $-$, $\downarrow$ and $\uparrow$ are signs which indicate a possible increase or a decrease of the probability where the amount may not be known and $++$, $--$, $\Uparrow$, $\Downarrow$, 0 and 1 are signs which indicate a state of or a change to certainly true or false. In cases where the direction of the change is not known the signs $\updownarrow$ and ι are used and when it is not known whether there is a change the sign $?$ is used. $\leftrightarrow$ is used in the case where it is known that there is no change. A subset of signs will be of use in our example. For a more detailed distinction, see [1].

Various extensions of two-valued propositional logic become available. Given a semantic domain D (set of truth values), we may aim at introducing a many-valued propositional logic $\mathcal{L}^{MV\pi} = (L^{MV\pi}, \vdash)$, where $L^{MV\pi}$ consists of formulas (p, s) with $p \in L^\pi$ and $s \in D$. Connectives in $\mathcal{L}^{MV\pi}$ have thus been introduced, and we expect the embedding of $\mathcal{L}^\pi$ into $\mathcal{L}^{MV\pi}$ to respect some homomorphic properties. It is further notable that many-valued extensions of propositional logic can be related to adaptive and knowledge acquisition frameworks [6].

Let now L^{QP} be some many-valued extension of L^π with respect to S_σ . C in $\mathcal{L}^{QP}$ are $(i : l : s)$, where i is its name (or index), l is a well-formed formula, i.e. $l \in L^{QP}$, and $s \in S_\iota$. A set of such clauses is called a C (of clauses). We write I_Δ for the name (or index) set related to a database Δ , i.e. $I_\Delta = \{i \mid (i : l : s) \in \Delta\}$.

A conditional uncertainty over L^{QP} , or L for short, is a mapping

$$\tau_L^{cond} : L \times L \times L \rightarrow [0, 1]$$

where we write $\tau_L^{cond}(a \mid b, X)$ instead of $\tau_L^{cond}(a, b, X)$. Clearly, τ_L^{cond} should fulfill suitable properties ([1]). For describing conditional uncertainty we actually do not need to fix our semantic view concerning τ_L^{cond} neither in form of probabilities or as possibilities, or as something else.

Semantics of clauses could be defined e.g. as $(i : a \rightarrow b : s)$ being true if and only if

$$\tau_L^{cond}(b \mid a, X) \geq \tau_L^{cond}(b \mid \neg a, X)$$

for all terms X over the signature for which $(i : X \rightarrow b : s)$ is true for any $s \in S_L$. See [1] for more detail.

3 Argumentation Logic Calculus

Let Δ be a database of clauses. An argument a (for a well-formed formula p) is a triple (p, G, s) , where $p \in L^{QP}$, $G \subseteq I_\Delta$, and $s \in S_L$. The set G represents the set of supporting clauses for the proposition, or support , or basis , or ground , p .

Note that, for a given database Δ we are mainly interested in the set of arguments $A_p = \{(p, G, s) \mid \Delta \vdash_{QP} (p, G, s)\}$ concerning some fixed proposition p , derivable from the database Δ .

The consequence relation $\vdash_{QP}$ is used to build new arguments from old in a database Δ . In the building process when the rules are used, signs are handled and combined, in order to reach a value of validity of a proposition. Every distinct argument with the sign s concerning p has to be taken into account and combined in an aggregation process. A number of different arguments for a certain claim have to be mapped into a single measure, which is a process called flattening . The flattening function flat^A maps a set of arguments A_p for a proposition p to an overall measure of validity v in the proposition, i.e. $\text{flat}^A : A_p \mapsto (p, v)$ where v is some combination of signs in S_L .

Before the flattening function can be used to obtain the overall measure of confidence in a claim, arguments have to be derived from the database. A set of axioms , inference rules and inference rules are defined for the argumentation consequence relation $\vdash_{QP}$. The rules are used to handle conjunctions, implications and negations in the arguments obtained from a database in order to create chains of arguments pointing to a certain claim. The following inference rules are denoted Ax , $\wedge I$, and $\rightarrow E$, respectively.

$$\begin{array}{c}
 \frac{(i : p : s) \in \Delta}{\Delta \vdash_{QP} (p, \{i\}, s)} \quad (Ax) \\
 \\
 \frac{\Delta \vdash_{QP} (p, G, s), \Delta \vdash_{QP} (p', G', s')}{\Delta \vdash_{QP} (p \wedge p', G \cup G', \mathbf{conj}_{\text{intro}}(s, s'))} \quad (\wedge I) \\
 \\
 \frac{\Delta \vdash_{QP} (p, G, s), \Delta \vdash_{QP} (p \rightarrow p', G', s')}{\Delta \vdash_{QP} (p', G \cup G', \mathbf{imp}_{\text{elim}}(s, s'))} \quad (\rightarrow E)
 \end{array}$$

The introduction axiom Ax is used to derive arguments from a database. The axiom $\wedge I$ shows how two arguments derived from a database concerning two different claims can be synthesized into one claim by using a combination function $\mathbf{conj}_{\text{intro}}^A$ to compute the support value and introducing a conjunction in its antecedent. The elimination axiom $\wedge E$ shows how the support for a claim p' , deduced from p , can be generated by using the grounds for both claims and by computing the support value using another combination function $\mathbf{imp}_{\text{elim}}^A$, and thus eliminating the implication connective. These computations are local.

In [1] these axioms are denoted $\mathcal{A}$ and $\mathcal{B}$ rules to distinguish them from $\mathcal{C}$ rules which lead the inferences in the opposite direction. It should be noted that in this framework $\rightarrow$ does not represent material implication, but is seen as a constraint on the conditional probabilities. It provides information about how probabilities or beliefs will change if the formula is activated in a context, but not necessarily to which extent. In [1], additional inference rules are defined, which are not used in our case study.

4 Dementia Diagnosis

Clinical guidelines have been developed in the domain of cognitive diseases for the purpose of research or clinical use. Some of the guidelines have been evaluated regarding specificity (amount of correct diagnoses), sensitivity (amount of detected cases out of detectable cases) and inter-rater reliability. A combination of guidelines is suggested where a guideline with high sensitivity is used initially followed by a guideline with high specificity. The chapter concerning cognitive diseases in the clinical guideline DSM-IV [3] was chosen as base in the decision-support system because it has been reported to have high sensitivity, was recommended to be used for the diagnosis of dementia and Alzheimer's disease, and was perceived by experts as the most usable in clinical practice. In order to evaluate the utility of the knowledge in the guideline in the context of clinical practice of dementia diagnosis, a formalization of the content of the guideline was made within a model of clinical reasoning in diagnosing dementia [2]. In this paper we will focus on on the part of the process in which a differential diagnosis is made among possible causes for a state of dementia. In the DSM-IV two types of dementia are specified, vascular dementia (VaD) and Alzheimer's disease (AD). These are complemented with a general category

in which a number of conditions are listed as examples without samples of criteria. Before diagnosing someone as having Alzheimer's disease, other medical conditions have to be considered as potential causes of the cognitive deficit and be ruled out. The chapter concerning cognitive diseases in DSM-IV was found insufficient in that it lacks diagnostic criteria for certain types of dementia. Thus, for the differential diagnosis of dementia, it is necessary to integrate consensus criteria for the less common diagnoses Lewy body type of dementia (DLB) [7] and frontotemporal degenerative dementia (FTD) [8] in the reasoning procedure in order to accomplish a full investigation and differential diagnosis in the domain.

4.1 Extending Dementia Diagnosis Using Consensus Criteria

The process of establishing differential diagnosis can be viewed as a separate guideline. Let $\Phi_{\mathcal{L}^\pi}^{DSM-IV}$ be a guideline for establishing the type of dementia based on the chapter concerning cognitive diseases in the clinical guideline DSM-IV. The guideline will consist of a set of rules formulated in propositional logic, which correspond to sets of features necessarily present or absent in a patient in order to establish the type of dementia. Let also $\Phi_{\mathcal{L}^\pi}^{consDLB}$ and $\Phi_{\mathcal{L}^\pi}^{consFTD}$ be

guidelines based on consensus criteria for establishing the diagnoses DLB and FTD respectively and $\Phi_{\mathcal{L}^\pi}^{core}$ be the synthesis guideline of the clinical guidelines including the DSM-IV. Does

$$\Phi_{\mathcal{L}^\pi}^{core} = \Phi_{\mathcal{L}^\pi}^{DSM-IV} \cup \Phi_{\mathcal{L}^\pi}^{consDLB} \cup \Phi_{\mathcal{L}^\pi}^{ConsFTD}$$

improve utility and reliability?

There are three core features specified for a dementia of Lewy body type (DLB) namely fluctuating cognition, gait disturbance similar to Parkinsonism (extrapyramidal sign) and visual hallucinations. The core features for FTD are typical behavioral symptoms indicating a disturbance of functions associated with the frontotemporal regions of cortex. The consensus criteria for DLB and FTD contain, part from the core features defined in the corresponding guidelines, supportive and exclusive features that may support a diagnostic process. The intended function of these are not representable in propositional logic and are excluded from the guidelines at this point. In some interpretations of the consensus criteria for DLB there are levels of firmness of the diagnosis defined, depending on the number of core features present in a patient, ie. or . This is also not represented in the guideline $\Phi_{\mathcal{L}^\pi}^{consDLB}$.

A synthesis guideline $\Phi_{\mathcal{L}^\pi}^{core}$ of the clinical guidelines can now be created that represents in propositional logic the differential diagnostic procedure when the core features in the specified clinical guidelines are considered.

$$\begin{aligned} \Phi_{\mathcal{L}^\pi}^{core} = \{ & \\ & Dementia \wedge GradualOnset \wedge Progressive \wedge \neg VaD \wedge \\ & \quad \neg DementiaDuetoGeneralMedicalCondition \\ & \quad \rightarrow AD \\ & Dementia \wedge focalSigns \rightarrow VaD \\ & Dementia \wedge VascularSignsInXray \rightarrow VaD \\ & Dementia \wedge GeneralMedicalCondition \\ & \quad \rightarrow DementiaDuetoGeneralMedicalCondition \\ & Parkinson's \rightarrow GeneralMedicalCondition \\ & HeadTrauma \rightarrow GeneralMedicalCondition \\ & Dementia \wedge JudgementDeficit \wedge gradualOnset \wedge \\ & \quad Progressive \wedge SocialSkillDeficit \wedge ADLdeficit \wedge \\ & \quad EmotionalBlunting \wedge \neg SevereAmnesia \wedge \\ & \quad \neg SpatialDesorientation \wedge \neg OtherNeurologicalSymptoms \\ & \quad \rightarrow FTD \\ & Dementia \wedge FluctuatingCognition \wedge Extrapyramidal \wedge \\ & \quad VisualHallucinations \rightarrow DLB \\ & \} \end{aligned}$$

The guideline now contains sets of necessary features which are required to be present in a patient in order to diagnose a certain cognitive disorder, formulated as rules in propositional logic. By using the guideline a mayor part of cases of different dementia types can be detected since the underlying clinical guidelines have relatively good overall sensibility. The cases where one single diagnosis can be matched to evidence found in a patient by using the guideline, we chose to call $\dots$ cases in order to distinguish these from $\dots$ cases when more evidence is required to reach a conclusion concerning diagnosis. The integrated clinical guidelines are known to be sensitive in detecting pathology per se, but not as useful when differentiating diagnoses in complicated cases or detecting multi-diagnosis. Therefore the guideline $\Phi_{\mathcal{L}^\pi}^{core}$ needs to be further extended with clinical guidelines with higher specificity in order to provide support in atypical cases, which will represent the next phase within the differential diagnostic step of the clinical reasoning process.

4.2 Dementia Diagnosis in Atypical Cases - Representing Uncertainty

The clinical guidelines with a higher documented specificity are sometimes considered as less useful in clinical practice since these tend to have been evolved for research purposes and contain more specifics concerning each diagnosis, which makes them appear as less practical in clinical environments. By integrating these guidelines into a decision support system, they may contribute to clinical practice in a more direct way, supporting diagnosis in the atypical cases where specificity is beneficial. The clinical guidelines which are interesting in the dementia context are the NINCDS-AIRENS for vascular dementia and NINCDS-ARDRA for Alzheimer's disease, and the parts of the consensus guidelines for DLB and FTD which were not used in diagnosing typical cases, concerning supportive and contradictory features, levels of reliability in diagnoses, etc. A review of guidelines can be found in [9]. To make the synthesis guideline practical, we chose to distinguish the guideline for typical cases from atypical, therefore we create a new guideline $\Phi_{\mathcal{L}}^{atyp}$, where $\mathcal{L}$ can be any logical framework suitable for the purpose of handling ambiguous and incomplete information. We will in this article consider the probabilistic argumentation framework defined by Parsons and colleagues. In order to allow a comparison of the clinical guidelines, the guideline $\Phi_{\mathcal{L}^\pi}^{core}$ will be translated into $\Phi_{\mathcal{L}^{QP}}^{core}$ using the dictionaries defined as semantic domains in the framework.

We now need to create $\Phi_{\mathcal{L}^{QP}}^{atyp}$, and $\Phi_{\mathcal{L}^{QP}}^{core}$, and compare with existing $\Phi_{\mathcal{L}^\pi}^{core}$. When the requirements for a specific diagnosis in $\Phi_{\mathcal{L}^\pi}^{core}$ is met, the diagnosis can be set according to the underlying clinical guidelines. Therefore, the presence of evidence specified in these rules will generate confidence in the diagnosis which is as close to certainty as the dictionary allows. Consequently, all the rules will be labelled with ++, except for the rules r4 and r5 that is added, which explicitly rule out AD in the presence of other diagnoses. The following subset of rules will be used in the example given below.

$$\begin{aligned}
\Phi_{\mathcal{LQP}}^{core} \supset \{ & \\
& (r1 : (Dementia \wedge FocalSigns) \rightarrow VaD : ++) \\
& (r2 : (Dementia \wedge VascularSignsOnXray) \rightarrow VaD : ++) \\
& (r3 : (Dementia \wedge GradualOnset \wedge Progressive) \wedge (DLB, \Downarrow) \\
& \quad \wedge (VaD, \Downarrow)) \rightarrow AD : ++) \\
& (r4 : (DLB, \Uparrow) \rightarrow AD : --) \\
& (r5 : (VaD, \Uparrow) \rightarrow AD : --) \\
& (r6 : (Dementia \wedge FluctuatingCognition \wedge VisualHallucinations \\
& \quad \wedge Extrapyramidal) \rightarrow DLB : ++) \\
& \}
\end{aligned}$$

The clinical guidelines considered in this section are pervaded with uncertainty in that different levels of reliability of diagnoses are defined, such as *probable* and *possible*. In addition, sets of *core* features and *supportive* features as well as *atypical* features are specified. The presence of a supportive feature is not necessary for diagnosis but

it is *probable*. Since the guidelines do not specify to what extent each supportive feature supports a certain diagnosis, it is suitable to consider all of them as if detected in a patient, their presence increases the probability of the patient having the diagnosis the features support. In the probabilistic argumentation framework, this increase or decrease is registered, although the exact value of the increase or decrease is not known. Following the notions of the argumentation framework, the influence of a supportive feature on a diagnosis is integrated in the knowledge base as the tuple (i: feature $\rightarrow$ diagnosis: +), and consequently, information about a contradictory feature will be represented as the tuple (i: feature $\rightarrow$ diagnosis: -). The third element of the tuple is an element from a dictionary, in this case the dictionary $S_i = \{++, +-, +, 0, -, +-, --, ?\}$.

Other sets of features are defined such as if the set is present a *probable* diagnosis can be set, or a *possible* diagnosis. The number of features differ in these sets, as well as the dignity of a certain feature, depending on which disease is in focus. The diagnostic evidence required for diagnosis specified in these clinical guidelines, are more restrictive in diagnosis than the evidence required in DSM-IV and $\Phi_{\mathcal{LQP}}^{core}$. For example in DSM-IV one feature of those specified for diagnosing VaD is enough for diagnosis, while in the NINCDS-AIRENS the same feature only supports a possible VaD. Consequently, the guidelines $\Phi_{\mathcal{LQP}}^{core}$ and $\Phi_{\mathcal{LQP}}^{atyp}$ provide different support for the same diagnosis, considering the same evidence. Therefore the distinction between sources of knowledge will be kept in order to provide the context of a hypothesis to a physician who uses the support system.

Since the probabilistic argumentation language QP does not have means to distinguish between sets of features supporting a possible diagnosis and supportive features, both types of rules will be labelled with + in the following example. Sets of features supporting a probable diagnosis are labelled with ++, meaning almost certainty in the framework, since the only stronger evidence de-

fined in the clinical guidelines is biopsy, which is not usable knowledge in clinical practice. Consequently, a Δ_{core} inferred by $\Phi_{\mathcal{L}^{QP}}^{atyp}$ will be valued as reliable as diagnoses suggested by the guideline $\Phi_{\mathcal{L}^{QP}}^{core}$.

We will consider three of the dementia diagnoses in the following example and limit the medical domain knowledge to a subset of supporting and contradictory features and diagnostic rules. The following set of rules defines the guideline $\Phi_{\mathcal{L}^{QP}}^{atyp}$.

$$\begin{aligned}
 \Phi_{\mathcal{L}^{QP}}^{atyp} = \{ & \\
 (r1 : (Dementia \wedge FocalSigns \wedge VascularSignsOnXray) \rightarrow VaD : ++) & \\
 (r2 : (Dementia \wedge FocalSigns) \rightarrow VaD : +) & \\
 (r3 : (Dementia \wedge VascularSignsOnXray) \rightarrow VaD : +) & \\
 (r4 : (Dementia \wedge GradualOnset \wedge Progressive \wedge (DLB, \Downarrow) \wedge & \\
 (VaD, \Downarrow)) \rightarrow AD : ++) & \\
 (r5 : (Dementia \wedge GradualOnset \wedge Progressive) \rightarrow AD : +) & \\
 (r6 : (Dementia \wedge FluctuatingCognition \wedge Extrapyramidal) & \\
 \rightarrow DLB : ++) & \\
 (r7 : (Dementia \wedge FluctuatingCognition \wedge VisualHallucinations & \\
 \rightarrow DLB : ++) & \\
 (r8 : (Dementia \wedge Extrapyramidal \wedge VisualHallucinations & \\
 \rightarrow DLB : ++) & \\
 (r9 : (Dementia \wedge FluctuatingCognition) \rightarrow DLB : +) & \\
 (r10 : (Dementia \wedge VisualHallucinations) \rightarrow DLB : +) & \\
 (r11 : (Dementia \wedge Extrapyramidal) \rightarrow DLB : +) & \\
 (r12 : (Dementia \wedge FluctuatingCognition) \rightarrow VaD : +) & \\
 (r13 : (Dementia \wedge progressive) \rightarrow VaD : -) & \\
 (r14 : (Dementia \wedge FocalSigns) \rightarrow DLB : -) & \\
 \} &
 \end{aligned}$$

Consider a database Δ_{core} containing the guideline $\Phi_{\mathcal{L}^{QP}}^{core}$ and another database Δ_{atyp} containing the guideline $\Phi_{\mathcal{L}^{QP}}^{atyp}$. Consider further the dictionaries $S_\iota = \{++, +-, +, 0, -, -+, --, ?\}$, $S_\sigma = \{0, \Downarrow, \downarrow, \leftrightarrow, \uparrow, \uparrow, 1, \uparrow, \imath\}$, corresponding combination, elimination and flattening functions, and the patient Olle presenting the evidence dementia, focal neurological signs, fluctuating cognition, gradual onset, progressive course, extrapyramidal signs and visual hallucinations.

In the clinical decision process the investigation has proceeded to the third step, which is to determine the type of dementia. The evidence concerning the patient is integrated into the databases, formulated as the following facts

- (f1: Dementia: $\uparrow$)
- (f2: FocalSigns: $\uparrow$)

- (f3: GradualOnset: $\uparrow$)
- (f4: Progressive: $\uparrow$)
- (f5: FluctuatingCognition: $\uparrow$)
- (f6: VisualHallucinations: $\uparrow$)
- (f7: Extrapyramidal: $\uparrow$)

The arrow $\uparrow$ represents that the certainty of the evidence changes to 1 if it is not 1 already. From the database arguments can be formed, in a process of finding the most reliable suggestion of a dementia diagnosis in Olle's case. Initially, the evidence is considered in the context of the guideline $\Phi_{\mathcal{LQP}}^{core}$.

- $\Delta_{core} \vdash_{QP}$ (DLB: {r6,f1,f5-f7}: $\uparrow$)
- $\Delta_{core} \vdash_{QP}$ (VaD: {r1,f1,f2}: $\uparrow$)
- $\Delta_{core} \vdash_{QP}$ (AD: {r4,r6,f1,f5-7}: $\downarrow$)
- $\Delta_{core} \vdash_{QP}$ (AD: {r5,r1,f1,f2}: $\downarrow$)

The guideline yields maximum support for VaD and DLB, while AD is supported only in the absence of alternative explanations. Since the result of two confirmed diagnoses can be considered unsatisfactory in the perspective of the limited evidence, further reasoning is needed in order to decide which diagnosis is the most likely, or whether there is coexistence of diseases.

In the context of the guideline $\Phi_{\mathcal{LQP}}^{atyp}$, the same evidence generates the following arguments

- $\Delta_{atyp} \vdash_{QP}$ (DLB, {r6-r8,f1,f5-f7}, $\uparrow$)
- $\Delta_{atyp} \vdash_{QP}$ (DLB, {r14, f2}, $\downarrow$)
- $\Delta_{atyp} \vdash_{QP}$ (VaD, {r2, f1, f2}, $\uparrow$)
- $\Delta_{atyp} \vdash_{QP}$ (VaD, {r12, f1, f5}, $\uparrow$)
- $\Delta_{atyp} \vdash_{QP}$ (VaD, {r13, f1, f4}, $\downarrow$)
- $\Delta_{atyp} \vdash_{QP}$ (AD, {r5, f1, f2}, $\uparrow$)

In order to compute the overall measure of confidence in each hypothetical diagnosis the flattening function defined in Table 1 is used, which produces the following result

- flat : $A^{atyp} \mapsto$ (DLB, $\uparrow$)
- flat : $A^{atyp} \mapsto$ (VaD, $\uparrow$)
- flat : $A^{atyp} \mapsto$ (AD, $\uparrow$)

DLB is the diagnosis with the highest support in this context. The supportive and contradictory evidence contribute to the outcome only when no argument supported with the highest level of support is present, since the value dominates the computations. The contribution of the guideline $\Phi_{\mathcal{LQP}}^{atyp}$ in the case of VaD in the example, is the valuation of the presence of both supportive and contradictory features as ambiguous and stating that the change in probability is unknown based on the facts. Consequently, the level of support for the hypothesis VaD has been reconsidered from being confirmed within the context of $\Phi_{\mathcal{LQP}}^{core}$ to unknown in the context of $\Phi_{\mathcal{LQP}}^{atyp}$.

Table 1. Flattening function [1]

	1	↑	↑	↔	↓	↓	0	↓	z
1	1	1	1	1	1		1	1	
↑	1	↑	↑	↑	↑		↑	↑	
↑	1	↑	↑	↑	↓	↓	↑	↑	
↔	1	↑	↑	↔	↓	↓	↑	↔	
↓	1	↑	↑	↓	↓	↓	↑	↓	
↓			↓	↓	↓	↓	↓	↓	
0		0	0	0	0	0	0	0	
↑	1	↑	↑	↑	↑	↑	↑	↑	
z	1	↑	↑	↑	↑	↑	↑	↑	z

5 Conclusions

The clinical diagnostic reasoning process contains mainly inferences which are evidential, i.e., moves from evidence towards detecting causes as described in previous section. The rules in the probabilistic argumentation system QPR are supposed to be causally directed, with the diagnosis determining expected evidence. If the inference connective on the other hand is seen as a causal connection between the amount of belief in hypotheses based on evidence, the evidence manifested in a patient causes an increase in the reliability of a particular hypothesis. Reasoning about beliefs in this sense, is then possible within the framework as shown in the example. If the same example would be reformulated with the connective pointing in opposite direction, as true causal connections, the evidential implication revision rule defined in [1] may be used. Other approaches to argumentation, such as in [10, 11, 12], should also be considered. In fact, this has been observed in [5] including further examples of rule bases. Generally, semantics of possibilities stems from questions on combining logic with probability. Questions concerning the logic of causality are far from trivial as can be seen e.g. from the foundational viewpoints presented in [12]. On programming level, degrees of justification of a belief must always be considered. Some general methodologies thereof can be found in [11].

The probabilistic argumentation framework allows the distinction between hypotheses that are considered certain and hypotheses that are supported with less certainty, which is a useful property for diagnostic support. Still, the probabilistic setting lacks means to distinguish between features and sets of features supporting diagnoses in a reasoning process. In addition, the framework gives no support in the presence of both supportive and contradictory evidence for a certain diagnosis. Therefore, the possibility to use different dictionaries with signs corresponding to the vocabulary in clinical guidelines will be investigated.

The result of the inferences using the evidential rule [1] would not contribute much to the reasoning because all inferences would yield an increased support for each diagnosis, but without distinction. This view is correct, in the perspective of probabilities of occurrences governing the change in the support for hypotheses.

The clinical guidelines are based on statistical evidence, evidence which has been interpreted by domain experts into knowledge guiding evidential reasoning. As can be seen in the example, the interpretation can vary, depending on views of how to treat atypical cases among other things.

In future work we will further develop the foundational understanding of the argumentation logic used, and in particular concerning techniques to move from one logic to the other. Semantic descriptions obviously also need to be further specified for respective logics. The given example shows the possibility to provide decision support at critical points in a diagnostic process, where a subset of clinical guidelines is sufficient for supporting diagnosis in typical patient's cases, and where additional support and knowledge are provided in atypical cases. A synthesis of different guidelines is needed for accomplishing the task of diagnosing cognitive disorders, while the ambiguities between guidelines can be handled if the guideline context is kept. In this way the physician is given means to value and compare the outcome of the different guidelines in the atypical cases and a base on which decisions can be made.

References

1. S. Parsons. A Proof Theoretic Approach to Qualitative Probabilistic Reasoning. *International Journal of Approximate Reasoning*, **19** (1998), 265-297.
2. H. Lindgren, P. Eklund, S. Eriksson. Clinical Decision Support System in Dementia Care. In *Proc. of MIE2002: Health Data in the Information Society*, IOS Press, (2002), 568-576.
3. American Psychiatric Association. *Diagnostic and Statistical Manual of Mental Disorders, Fourth Edition, Text Revision (DSM-IV-TR®)*. American Psychiatric Association, 1994.
4. S. Parsons. On Precise and Correct Qualitative Probabilistic Inference. *International Journal of Approximate Reasoning*, **35** (2004), 111-135.
5. H. Lindgren. Managing Knowledge in the Development of a Decision-Support System for the Investigation of Dementia. UMNAD 01/05, Department of Computing Science, University of Umeå, Sweden, 2005.
6. P. Eklund, F. Klawonn. Neural Fuzzy Logic Programming. *IEEE Trans. Neural Networks*, **3** No 5 (1992), 815-818.
7. I.G. McKeith, D. Galasko, K. Kosaka et al. Consensus guidelines for the clinical and pathologic diagnosis of dementia with Lewy bodies (DLB): report of the Consortium on DLB international workshop. *Neurology*, **54** (1996), 1050-1058.
8. D. Neary, J.S. Snowden, L. Gustafson, U. Passant, D. Stuss, S. Black, et al. Frontotemporal Lobar Degeneration - A Consensus on Clinical Diagnostic Criteria. *Neurology*, **51** (1998), 1546-1554.
9. J. O'Brien, D. Ames, A. Burns (Eds). *Dementia*, Arnold, 2000.
10. J. Fox, S. Parsons, Arguing about beliefs and actions. In A. Hunter and S. Parsons (Eds), *Applications of uncertainty formalisms*, LNAI 1455, Springer Verlag, 1998.
11. J. L. Pollock. Defeasible reasoning with variable degrees of justification. *Artificial Intelligence*, **133** (2001), 233-282.
12. J. Kohlas. Probabilistic argumentation systems: A new way to combine logic with probability. *Journal of Applied Logic*, **1** (2003), 225-253.

Argument-Based Expansion Operators in Possibilistic Defeasible Logic Programming: Characterization and Logical Properties

Carlos I. Chesñevar¹, Guillermo R. Simari², Lluís Godo³, and Teresa Alsinet¹

¹ Departament de Computer Science – Universitat de Lleida,
C/Jaume II, 69 – 25001 Lleida, Spain
`{cic, tracy}@eps.udl.es`

² Department of Computer Science and Engineering – Universidad Nacional del Sur,
Alem 1253, (8000) Bahía Blanca, Argentina
`grs@cs.uns.edu.ar`

³ Artificial Intelligence Research Institute (IIIA-CSIC),
Campus UAB - 08193 Bellaterra, Barcelona, Spain
`godo@iiia.csic.es`

Abstract. Possibilistic Defeasible Logic Programming (P-DeLP) is a logic programming language which combines features from argumentation theory and logic programming, incorporating as well the treatment of possibilistic uncertainty and fuzzy knowledge at object-language level. Defeasible argumentation in general and P-DeLP in particular provide a way of modelling non-monotonic inference. From a logical viewpoint, capturing defeasible inference relationships for modelling argument and warrant is particularly important, as well as the study of their logical properties. This paper analyzes two non-monotonic operators for P-DeLP which model the expansion of a given program $\mathcal{P}$ by adding new weighed facts associated with argument conclusions and warranted literals, resp. Different logical properties for the proposed expansion operators are studied and contrasted with a traditional SLD-based Horn logic. We will show that this analysis provides useful comparison criteria that can be extended and applied to other argumentation frameworks.

Keywords: argumentation, logic programming, uncertainty, non-monotonic inference.

1 Introduction and Motivations

Possibilistic Defeasible Logic Programming (P-DeLP) [1] is a logic programming language which combines features from argumentation theory and logic programming, incorporating as well the treatment of possibilistic uncertainty and fuzzy knowledge at object-language level. These knowledge representation features are formalized on the basis of PGL [2, 3], a possibilistic logic based on Gödel fuzzy logic. In PGL formulas are built over fuzzy propositional variables and the certainty degree of formulas is expressed with a necessity measure. In a

logic programming setting, the proof method for PGL is based on a complete calculus for determining the maximum degree of possibilistic entailment of a fuzzy goal. The top-down proof procedure of P-DeLP has already been integrated in a number of real-world applications such as intelligent web search [4] and natural language processing [5], among others.

Formalizing argument-based reasoning by means of suitable inference operators offers a useful tool. On the one hand, from a theoretical viewpoint logical properties of defeasible argumentation can be easier studied with such operators at hand. On the other hand, actual implementations of argumentation systems could benefit from such logical properties for more efficient computation in the context of real-world applications. This paper analyzes two non-monotonic

for P-DeLP, intended for modelling the effect of expanding a given program by introducing new facts, associated with argument conclusions and warranted literals, respectively. Their associated logical properties are studied and contrasted with a traditional SLD-based Horn logic. We contend that this analysis provides useful comparison criteria that can be extended and applied to other argumentation frameworks. As we will show in this paper, expansion operators in an argumentative framework like P-DeLP provide an interesting counterpart to traditional consequence operators in logic programming [6]. Our approach differs from such consequence operators as we want to analyze the role of argument conclusions and warranted literals when represented as new weighed facts in the context of object-level program clauses.

For the sake of simplicity we will restrict our analysis to the fragment of P-DeLP built over classical propositions, hence based on possibilistic logic [7] and not on PGL itself (which involves fuzzy propositions). The rest of the paper is structured as follows: first in Section 2 we outline some fundamentals of (non-monotonic) inference relationships. Section 3 summarizes the P-DeLP framework. In Section 4 we characterize two expansion operators for capturing the effect of expanding a P-DeLP program by adding argument conclusions and warranted literals, as well as their emerging logical properties. Finally, in Section 5 we discuss related work the most important conclusions that have been obtained.

2 Non-monotonic Inference Relationships: Fundamentals

In classical logic, inference rules allow us to determine whether a given wff γ follows via “ $\vdash$ ” from a set Γ of wffs, where “ $\vdash$ ” is a (satisfying idempotence, and monotonicity). As non-monotonic and defeasible logics evolved into a valid alternative to formalize commonsense reasoning a similar concept was needed to capture the notion of logical consequence without demanding some of these requirements (*e.g.* monotonicity). This led to the definition of a more generic notion of inference in terms of

Given a set Γ of wffs in an arbitrary logical language $\mathcal{L}$, we write $\Gamma \vdash \gamma$ to denote an inference relationship “ $\vdash$ ”, where γ is a (non-monotonic) consequence of Γ . We define an $C_{\vdash}$ associated with an inference relationship, with $C_{\vdash}(\Gamma) = \{\gamma \mid \Gamma \vdash \gamma\}$. Given an inference relationship “ $\vdash$ ” and a set Γ of

sentences, the following are called $\dots$ (or $\dots$) properties associated with the inference operator $C_{\sim}(F)$:

1. **Inclusion (IN)**: $\Gamma \subseteq C(\Gamma)$
2. **Idempotence (ID)**: $C(\Gamma) = C(C(\Gamma))$
3. **Cut (CT)**: $\Gamma \subseteq \Phi \subseteq C(\Gamma)$ implies $C(\Phi) \subseteq C(\Gamma)$
4. **Cautious monotonicity (CM)**: $\Gamma \subseteq \Phi \subseteq C(\Gamma)$ implies $C(\Gamma) \subseteq C(\Phi)$.
5. **Cumulativity (CU)**: $\gamma \in C(\Gamma)$ implies $\phi \in C(\Gamma \cup \{\gamma\})$ iff $\phi \in C(\Gamma)$, for any wffs $\gamma, \phi \in \mathcal{L}$.
6. **Monotonicity (MO)**: $\Gamma \subseteq \Phi$ implies $C(\Gamma) \subseteq C(\Phi)$

These properties are called $\dots$, since they can be applied to any language $\mathcal{L}$, and are abstractly defined for an arbitrary inference relationship “ $\sim$ ”. Nevertheless, other properties which link a classical inference operator Th with an arbitrary inference relationship can be stated. Next we summarize the most important non-pure properties (for an in-depth discussion, see [8]).

1. **Supraclassicality**: $Th(A) \subseteq C(A)$
2. **Left logical equivalence (LL)**: $Th(A) = Th(B)$ implies $C(A) = C(B)$
3. **Right weakening (RW)**: If $x \supset y \in Th(A)$ and $x \in C(A)$ then $y \in C(A)$.¹
4. **Conjunction of conclusions (CC)**: If $x \in C(A)$ and $y \in C(A)$ then $x \wedge y \in C(A)$.
5. **Subclassical cumulativity (SC)**: If $A \subseteq B \subseteq Th(A)$ then $C(A) = C(B)$.
6. **Left absorption (LA)**: $Th(C(\Gamma)) = C(\Gamma)$.
7. **Right absorption (RA)**: $C(Th(\Gamma)) = C(\Gamma)$.
8. **Rationality of negation (RN)**: if $A \sim z$ then either $A \cup \{x\} \sim z$ or $A \cup \{\sim x\} \sim z$.
9. **Disjunctive rationality (DR)**: if $A \cup \{x \vee y\} \sim z$ then $A \cup \{x\} \sim z$ or $A \cup \{y\} \sim z$.
10. **Rational monotonicity (RM)**: if $A \sim z$ then either $A \cup \{x\} \sim z$ or $A \sim \sim x$.

3 The P-DeLP Programming Language: Fundamentals

The classical fragment of P-DeLP language $\mathcal{L}$ is defined from a set of ground atoms (propositional variables) $\{p, q, \dots\}$ together with the connectives $\{\sim, \wedge, \leftarrow\}$. The symbol $\sim$ stands for $\neg$. A $\dots$ $L \in \mathcal{L}$ is a ground (fuzzy) atom q or a negated ground (fuzzy) atom $\sim q$, where q is a ground (fuzzy) propositional variable. A $\dots$ in $\mathcal{L}$ is a formula of the form $Q \leftarrow L_1 \wedge \dots \wedge L_n$, where $Q, L_1, \dots, L_n$ are literals in $\mathcal{L}$. When $n = 0$, the formula $Q \leftarrow$ is called a $\dots$ and simply written as Q . The term $\dots$ will be used to refer to any literal $Q \in \mathcal{L}$.² In the following, capital and lower case letters will denote literals and atoms in $\mathcal{L}$, resp.

Definition 1 (P-DeLP formulas). *The set $Wffs(\mathcal{L})$ of wffs in $\mathcal{L}$ are facts, rules and goals built over the literals of $\mathcal{L}$. A certainty-weighted clause in $\mathcal{L}$, or simply weighted clause, is a pair of the form (φ, α) , where $\varphi \in Wffs(\mathcal{L})$ and $\alpha \in [0, 1]$ expresses a lower bound for the certainty of φ in terms of a necessity measure.*

¹ It should be noted that “ $\supset$ ” stands for material implication, to be distinguished from the symbol “ $\leftarrow$ ” used in a logic programming setting.

² Note that a conjunction of literals is not a valid goal.

The original P-DeLP language [1] is based on Possibilistic Gödel Logic or PGL [2], which is able to model both uncertainty and fuzziness and allows for a partial matching mechanism between fuzzy propositional variables. As mentioned before, in this paper, for simplicity and space reasons we will restrict ourselves to fragment of P-DeLP built on non-fuzzy propositions, and hence based on the necessity-valued classical propositional Possibilistic logic [7]. As a consequence, possibilistic models are defined by possibility distributions on the set of classical interpretations³ and the proof method for our P-DeLP formulas, written $\vdash$, is defined by derivation based on the following generalized modus ponens rule (GMP).

Generalized modus ponens (GMP):

$$\frac{(L_0 \leftarrow L_1 \wedge \dots \wedge L_k, \gamma) \quad (L_1, \beta_1), \dots, (L_k, \beta_k)}{(L_0, \min(\gamma, \beta_1, \dots, \beta_k))}$$

which is a particular instance of the well-known possibilistic resolution rule, and which provides the - fragment of P-DeLP with a complete calculus for determining the maximum degree of possibilistic entailment for weighted literals.

From now on, and if not stated otherwise, we will simply use P-DeLP to actually refer to the - fragment of the original P-DeLP.

3.1 Argumentation in P-DeLP

In P-DeLP we distinguish between and clauses. A clause (φ, α) will be referred as certain if $\alpha = 1$ and uncertain, otherwise. Moreover, a set of clauses Γ will be deemed as , denoted $\Gamma \vdash \perp$, if $\Gamma \vdash (q, \alpha)$ and $\Gamma \vdash (\sim q, \beta)$, with $\alpha > 0$ and $\beta > 0$, for some atom q in $\mathcal{L}$ ⁴. A P-DeLP program is a set of weighted rules and facts in $\mathcal{L}$ in which we distinguish certain from uncertain information. As additional requirement, certain knowledge is required to be non-contradictory. Formally:

Definition 2 (Program). *A P-DeLP program $\mathcal{P}$ (or just program $\mathcal{P}$) is a pair (Π, Δ) , where Π is a non-contradictory finite set of certain clauses, and Δ is a finite set of uncertain clauses. If $\mathcal{P} = (\Pi, \Delta)$ is a program, we will also write $\mathcal{P}^\Pi$ (resp. $\mathcal{P}^\Delta$) to identify the set of certain (resp. uncertain) clauses in $\mathcal{P}$.*

The following notion of argument, very similar to [9, 10], is an extension of that in argumentation systems by considering the necessity degree which the argument supports a conclusion.

Definition 3 (Argument. Subargument). *Given a program $\mathcal{P} = (\Pi, \Delta)$, a set $A \subseteq \Delta$ of uncertain clauses is an argument for a goal Q with necessity degree $\alpha > 0$,*

³ Although the connective $\leftarrow$ in logic programming is different from the material implication, e.g. $p \leftarrow q$ is not the same as $\sim q \leftarrow \sim p$, regarding the possibilistic semantics we assume here they share the same set interpretations.

⁴ Notice that this notion of contradiction corresponds to the case when the inconsistency degree of Γ is strictly positive as defined in possibilistic logic.

denoted $\langle \mathcal{A}, Q, \alpha \rangle$, iff: (1) $\Pi \cup \mathcal{A} \vdash (Q, \alpha)$; (2) $\Pi \cup \mathcal{A}$ is non contradictory; and (3) There is no $\mathcal{A}_1 \subset \mathcal{A}$ such that $\Pi \cup \mathcal{A}_1 \vdash (Q, \beta)$, $\beta > 0$.

Let $\langle \mathcal{A}, Q, \alpha \rangle$ and $\langle \mathcal{S}, R, \beta \rangle$ be two arguments. We will say that $\langle \mathcal{S}, R, \beta \rangle$ is a sub-argument of $\langle \mathcal{A}, Q, \alpha \rangle$ iff $\mathcal{S} \subseteq \mathcal{A}$. Notice that the goal R may be a subgoal associated with the goal Q in the argument $\mathcal{A}$.

Note that from the definition of argument, it follows that on the basis of a P-DeLP program $\mathcal{P}$ there may exist different arguments $\langle \mathcal{A}_1, Q, \alpha_1 \rangle$, $\langle \mathcal{A}_2, Q, \alpha_2 \rangle$, $\dots$, $\langle \mathcal{A}_k, Q, \alpha_k \rangle$ supporting a given goal Q , with (possibly) different necessity degrees $\alpha_1, \alpha_2, \dots, \alpha_k$. Given a program $\mathcal{P} = (\Pi, \Delta)$, we will write $\mathcal{P} \vdash_{\Delta} \langle \mathcal{A}, Q, \alpha \rangle$ to denote that the argument $\langle \mathcal{A}, Q, \alpha \rangle$ can be obtained from $\mathcal{P}$. Actually, the set $Arg(\mathcal{P}) = \{ \langle \mathcal{A}, Q, \alpha \rangle \mid \mathcal{P} \vdash_{\Delta} \langle \mathcal{A}, Q, \alpha \rangle \}$ of arguments with respect to a program $\mathcal{P}$ can be built by means of the following complete set of procedural rules:

1 Building arguments from facts (INTF)

If $(Q, 1) \in \Pi$, then $\langle \emptyset, Q, 1 \rangle \in Arg(\mathcal{P})$

If $(Q, \alpha) \in \Delta$ then $\{ \langle (Q, \alpha) \rangle, Q, \alpha \rangle \in Arg(\mathcal{P})$

2 Building Arguments by GMP (MPA):

If $\langle \mathcal{A}_1, L_1, \alpha_1 \rangle \dots \langle \mathcal{A}_k, L_k, \alpha_k \rangle \in Arg(\mathcal{P})$

and $\Pi \cup \{ (L_0 \leftarrow L_1 \wedge L_2 \wedge \dots \wedge L_k, \gamma) \} \cup \bigcup_{i=1}^k \mathcal{A}_i \not\vdash \perp$

and $(L_0 \leftarrow L_1 \wedge L_2 \wedge \dots \wedge L_k, \gamma) \in \Delta$

then $\langle \bigcup_{i=1}^k \mathcal{A}_i \cup \{ (L_0 \leftarrow L_1 \wedge L_2 \wedge \dots \wedge L_k, \gamma) \}, L_0, \beta \rangle \in Arg(\mathcal{P})$,

with $\beta = \min(\alpha_1, \dots, \alpha_k, \gamma)$.

3 Extending Arguments (EAR):

If $\langle \mathcal{A}, P, \alpha \rangle \in Arg(\mathcal{P})$, and $\Pi \cup \{ (P, \alpha) \} \vdash (Q, \alpha)$ then $\langle \mathcal{A}, Q, \alpha \rangle \in Arg(\mathcal{P})$.

3.2 Computing Warrant in P-DeLP

As in most argumentation formalisms (see e.g. [11, 12]), in P-DeLP it can be the case that there exist conflicting arguments. This is formalized next by the notions of counterargument and defeat.

Definition 4 (Counterargument). Let $\mathcal{P}$ be a program, and let $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ and $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ be two arguments wrt $\mathcal{P}$. We will say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ counterargues $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ iff there exists a subargument (called disagreement subargument) $\langle \mathcal{S}, Q, \beta \rangle$ of $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ such that $\Pi \cup \{ (Q_1, \alpha_1), (Q, \beta) \}$ is contradictory.

Defeat among arguments involves a comparison on conflicting arguments, defined on the basis of necessity measures associated with arguments.

Definition 5 (Preference criterion $\succeq$). Let $\mathcal{P}$ be a P-DeLP program, and let $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ be a counterargument for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$. We will say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is preferred over $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ (denoted $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle \succeq \langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$) iff $\alpha_1 \geq \alpha_2$. If it is the case that $\alpha_1 > \alpha_2$, then we will say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is strictly preferred over $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$, denoted $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle \succ \langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$. Otherwise, if $\alpha_1 = \alpha_2$ we will say that both arguments are equi-preferred, denoted $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle \approx \langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$.

Definition 6 (Defeat). Let $\mathcal{P}$ be a program, and let $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ and $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ be two arguments in $\mathcal{P}$. We will say that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ defeats $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ (or equivalently $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ is a defeater for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$) iff (1) Argument $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ counterargues argument $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$ with disagreement subargument $\langle \mathcal{A}, Q, \alpha \rangle$; and (2)

Either it holds that $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle \succ \langle \mathcal{A}, Q, \alpha \rangle$, in which case $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ will be called a proper defeater for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$, or $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle \approx \langle \mathcal{A}, Q, \alpha \rangle$, in which case $\langle \mathcal{A}_1, Q_1, \alpha_1 \rangle$ will be called a blocking defeater for $\langle \mathcal{A}_2, Q_2, \alpha_2 \rangle$.

As in most argumentation systems [12, 11], P-DeLP relies on an exhaustive dialectical analysis which allows to determine if a given argument is undefeated (or *admissible*) wrt a program $\mathcal{P}$. An argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ is a sequence $[\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle, \langle \mathcal{A}_1, Q_1, \alpha_1 \rangle, \dots, \langle \mathcal{A}_n, Q_n, \alpha_n \rangle, \dots]$ that can be thought of as an exchange of arguments between two parties, a *proponent* (evenly-indexed arguments) and an *opponent* (oddly-indexed arguments). In order to avoid *infinite* reasoning, argumentation theory imposes additional constraints on such an argument exchange to be considered rationally acceptable wrt a P-DeLP program $\mathcal{P}$, namely:

1. **Non-contradiction:** given an argumentation line λ , the set of arguments of the proponent (resp. opponent) should be *non-contradictory* wrt $\mathcal{P}$. Non-contradiction for a set of arguments is defined as follows: a set $S = \bigcup_{i=1}^n \{\langle \mathcal{A}_i, Q_i, \alpha_i \rangle\}$ is *contradictory* wrt $\mathcal{P}$ iff $\Pi \cup \bigcup_{i=1}^n \mathcal{A}_i$ is contradictory.
2. **No circular argumentation:** no argument $\langle \mathcal{A}_j, Q_j, \alpha_j \rangle$ in λ is a sub-argument of an argument $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ in λ , $i < j$.
3. **Progressive argumentation:** every blocking defeater $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ in λ is defeated by a proper defeater $\langle \mathcal{A}_{i+1}, Q_{i+1}, \alpha_{i+1} \rangle$ in λ .

An argumentation line satisfying the above restrictions is called *admissible*, and can be proven to be finite. Given a program $\mathcal{P}$ and an argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$, the set of all acceptable argumentation lines starting in $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ accounts for a whole dialectical analysis for $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ (i.e. all possible dialogues rooted in $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$), formalized as a *dialectical tree*, denoted $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$. Nodes in a dialectical tree $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$ can be marked as *U-nodes* and *D-nodes* (U-nodes and D-nodes, resp.). A dialectical tree will be marked as an AND-OR tree: all leaves in $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$ will be marked U-nodes (as they have no defeaters), and every inner node is to be marked as D-*node* iff it has at least one U-node as a child, and as U-*node* otherwise. An argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ is ultimately accepted as valid (or *warranted*) iff the root of $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$ is labelled as U-*node*.

Definition 7 (Warrant). *Given a program $\mathcal{P}$, and a goal Q , we will say that Q is warranted wrt $\mathcal{P}$ with a necessity degree α iff there exists a warranted argument $\langle \mathcal{A}, Q, \alpha \rangle$. We will write $\mathcal{P} \vdash_w \langle \mathcal{A}, Q, \alpha \rangle$ to denote that $\langle \mathcal{A}, Q, \alpha \rangle$ is a warranted argument on the basis of $\mathcal{P}$.*

4 Logical Properties of Argument and Warrant in P-DeLP

Our aim is to study the behavior of P-DeLP programs in the context of non-monotonic inference relationships. In order to do this, we will define different inference operators associated with arguments and with warranted goals. As stated in Section 1, we refer to such operators as *inference operators* in order to stress the fact that their output is associated with the effect of expanding a given program $\mathcal{P}$ given as an input by adding new weighed facts. Formally:

Definition 8 (Expansion operators C_+ , C_Δ and C_w). Let $\mathcal{P}$ be a P-DeLP program. We define the operators C_+ , C_Δ and C_w associated with $\mathcal{P}$ as follows:

$$C_+(\mathcal{P}) = \mathcal{P} \cup \{ (Q, 1) \mid \mathcal{P} \vdash (Q, 1) \}$$

$$C_\Delta(\mathcal{P}) = \mathcal{P} \cup \{ (Q, \alpha) \mid \mathcal{P} \vdash_\Delta \langle \mathcal{A}, Q, \alpha \rangle, \text{ for some argument } \mathcal{A} \\ \text{for a goal } Q \text{ with necessity degree } \alpha \}$$

$$C_w(\mathcal{P}) = \mathcal{P} \cup \{ (Q, \alpha) \mid \mathcal{P} \vdash_w \langle \mathcal{A}, Q, \alpha \rangle, \text{ for some argument } \mathcal{A} \\ \text{for a goal } Q \text{ with necessity degree } \alpha \}$$

Operator C_+ computes the expansion of $\mathcal{P}$ by adding new certain facts $(Q, 1)$ whenever such facts can be derived in $\mathcal{P}$ via $\vdash$.⁵ Operator C_Δ computes the expansion of $\mathcal{P}$ with new facts corresponding to defeasible knowledge derivable as argument conclusions. $C_\Delta(\mathcal{P})$ incorporates a new uncertain fact (Q, α) whenever there exists an argument $\langle \mathcal{A}, Q, \alpha \rangle$ in $\mathcal{P}$. Notice that C_Δ may contain contradictory knowledge (*i.e.* it may be the case that two arguments $\langle \mathcal{A}_1, Q, \alpha \rangle$ and $\langle \mathcal{A}_2, \sim Q, \beta \rangle$ could be inferred from a given program $\mathcal{P}$).⁶ Finally, operator C_w computes a subset of C_Δ , namely the expansion of $\mathcal{P}$ including all new facts which correspond to conclusions of warranted arguments in $\mathcal{P}$.

Proposition 1. *Let $\mathcal{P}$ be a P-DeLP program. Then, $C_+(\mathcal{P})$, $C_\Delta(\mathcal{P})$ and $C_w(\mathcal{P})$ are also P-DeLP programs. Moreover, $C_+(\mathcal{P}) \subseteq C_\Delta(\mathcal{P})$ and $C_w(\mathcal{P}) \subseteq C_\Delta(\mathcal{P})$.*

Proof. Given a P-DeLP program $\mathcal{P}$, we want to determine that $C_+(\mathcal{P})$, $C_\Delta(\mathcal{P})$ and $C_w(\mathcal{P})$ are also programs. From Def. 8, it is clear that all operators return syntactically valid programs as their output. From Def. 2, it remains to check that the strict knowledge $C_+(\mathcal{P})^\Pi$ (analogously for $C_\Delta(\mathcal{P})$ and $C_w(\mathcal{P})$) is not a contradictory set of P-DeLP clauses. Let us suppose that $C_+(\mathcal{P})^\Pi$ is contradictory. By definition of C_+ , this is only possible if $\mathcal{P}^\Pi$ is contradictory, which cannot be the case, as $\mathcal{P}$ is a P-DeLP program (absurd). Consequently, $C_+(\mathcal{P})$ is a P-DeLP program. The same line of reasoning applies for $C_\Delta(\mathcal{P})$ and $C_w(\mathcal{P})$. The inclusion relationship $C_+(\mathcal{P}) \subseteq C_\Delta(\mathcal{P})$ holds as it can be shown that $\mathcal{P} \vdash (Q, 1)$ iff $\mathcal{P} \vdash_\Delta \langle \emptyset, Q, 1 \rangle$. Since every warranted argument is an argument wrt $\mathcal{P}$, a similar analysis applies to conclude that $C_w(\mathcal{P}) \subseteq C_\Delta(\mathcal{P})$.

4.1 Logical Properties for C_Δ

Proposition 2. *Let $\mathcal{P}$ be a P-DeLP program. Then, $C_\Delta(C_\Delta(\mathcal{P})) = C_\Delta(\mathcal{P})$.*

Proof. Inclusion holds trivially, as $\mathcal{P} \subseteq C_\Delta(\mathcal{P}) \subseteq C_\Delta(C_\Delta(\mathcal{P}))$ by definition of C_Δ . Proof for idempotence is not included for space reasons.

Monotonicity does not hold for C_Δ , as expected. As a counterexample consider the program $\mathcal{P} = \{ (q, 1), (\leftarrow, 0.9) \}$. Then $(p, 0.9) \in C_\Delta(\mathcal{P})$, as there is

⁵ Operator C_+ defines in fact a consequence relationship, as it satisfies idempotence, cut and monotonicity. It can be seen as the SLD Horn resolution counterpart in the context of P-DeLP restricted to certain clauses.

⁶ For a given goal Q , we write $\sim Q$ as an abbreviation to denote “ $\sim q$ ” if $Q \equiv q$ and “ $\neg q$ ” if $Q \equiv \neg q$.

an argument $\langle \mathcal{A}, p, 0.9 \rangle$ on the basis of $\mathcal{P}$ for concluding $(p, 0.9)$, with $\mathcal{A} = \{(\leftarrow, 0.9)\}$. However, $(p, 0.9) \notin C_\Delta(\mathcal{P} \cup \{(\sim p, 1)\})$ (as no argument for $(p, 0.9)$ could exist, as condition 2 in Def. 3 would be violated). Semi-monotonicity is an interesting property for analyzing non-monotonic consequence relationships. It is satisfied if all defeasible consequences from a given theory are preserved when the theory is augmented with new information.

Proposition 3. *T* $C_\Delta(\mathcal{P}_1) \subseteq C_\Delta(\mathcal{P}_1 \cup \mathcal{P}_2)$, *i.e.* $\mathcal{P}_2^H = \emptyset$.

P . Follows directly from the structure of the inference rules for $\vdash_\Delta$. Suppose $\mathcal{P}_1 \vdash_\Delta \langle \mathcal{A}, Q, \alpha \rangle$, and consider a program $\mathcal{P}_2$ such that $\mathcal{P}_2^H = \emptyset$. Clearly, $\langle \mathcal{A}, Q, \alpha \rangle$ can be derived from $\mathcal{P}_1 \cup \mathcal{P}_2$ by applying the same sequence of steps as in $\mathcal{P}_1 \vdash_\Delta \langle \mathcal{A}, Q, \alpha \rangle$, since all preconditions in inference rules are defined wrt $\mathcal{P}_1^H$, the set of strict knowledge in $\mathcal{P}_1$, and by hypothesis, $(\mathcal{P}_1 \cup \mathcal{P}_2)^H = \mathcal{P}_1^H$.

Proposition 4. *T* $C_\Delta(\Gamma) \subseteq C_\Delta(\Gamma \cup \{\gamma\})$, *i.e.* $\gamma \in C_\Delta(\Gamma)$.
 $\phi \in C_\Delta(\Gamma \cup \{\gamma\}) \implies \phi \in C_\Delta(\Gamma)$.

P . (Sketch) Without loss of generality, we can assume $\gamma = (Q, \alpha)$ is not in Γ (otherwise the proof is straightforward). By hypothesis, $(Q, \alpha) \in C_\Delta(\Gamma)$ and there is a sequence $s_1^Q, s_2^Q, \dots, s_t^Q$ of application of inference rules in $\{\text{INTF}, \text{MPA}, \text{EAR}\}$ such that $\Gamma \vdash_\Delta \langle \mathcal{A}_1, Q, \alpha \rangle$. Let us assume now that $(R, \beta) \in C_\Delta(\Gamma \cup \{(Q, \alpha)\})$. This means that there is a sequence $r_1, r_2, \dots, r_n$ of application of inference rules as before such that $\Gamma \vdash_\Delta \langle \mathcal{A}_2, R, \beta \rangle$. Suppose now that $\langle \mathcal{A}_2, R, \beta \rangle$ does not include $\langle \mathcal{A}_1, Q, \alpha \rangle$ as a subargument. This happens iff from the structure of inference rules for $\vdash_\Delta$, (Q, α) will not be required as intermediate step in the proof of (R, β) iff $(R, \beta) \in C_\Delta(\Gamma)$. Suppose now that $\langle \mathcal{A}_2, R, \beta \rangle$ does include $\langle \mathcal{A}_1, Q, \alpha \rangle$ as a subargument. This happens iff in the sequence $r_1, r_2, \dots, r_n$ we have that $r_{i+k} = s_i^Q$, for $i = 1 \dots t$, for some $1 \leq k \leq n$. But from the initial hypothesis this sequence can be built from Γ alone. Hence $\Gamma \vdash_\Delta \langle \mathcal{A}_2, R, \beta \rangle$ or equivalently $(R, \beta) \in C_\Delta(\Gamma)$.

Note that the property of right weakening cannot be considered (in a strict sense) in P-DeLP, since the underlying logic does not allow the application of the deduction theorem. Therefore, wffs of the form $(\leftarrow, \alpha)$ cannot be derived. However, an alternative approach can be intended, introducing a new property in which right weakening is restricted to Horn-like clauses:

Proposition 5. *T* $C_\Delta(\mathcal{P}) \subseteq C_\Delta(\mathcal{P} \cup \{(H \leftarrow Y, 1)\})$, *i.e.* $C_\Delta(\mathcal{P}) \subseteq C_\Delta(\mathcal{P})$, $(H \leftarrow Y, 1) \in C_\Delta(\mathcal{P})$, *i.e.* $(Y, \alpha) \in C_\Delta(\mathcal{P})$.
 $(X \leftarrow Y, 1) \in C_\Delta(\mathcal{P}) \implies (X, \alpha) \in C_\Delta(\mathcal{P})$.

P . Supraclassicality follows from Prop. 1. For the case of right weakening, let us suppose $(Y, \alpha) \in C_\Delta(\mathcal{P})$, *i.e.* $\mathcal{P} \vdash_\Delta \langle \mathcal{A}_1, Y, \alpha \rangle$, for some argument $\langle \mathcal{A}_1, Y, \alpha \rangle$. If $(X \leftarrow Y, 1) \in C_\Delta(\mathcal{P})$, then necessarily $(X \leftarrow Y, 1) \in \mathcal{P}^H$ (by def. of C_Δ). From $(X \leftarrow Y, 1) \in \mathcal{P}$ and $\mathcal{P} \vdash_\Delta \langle \mathcal{A}_1, Y, \alpha \rangle$, by applying inference rule EAR we get $\langle \mathcal{A}_1, X, \alpha \rangle$.

Proposition 6. *T* $C_\Delta(\mathcal{P}_1) \subseteq C_\Delta(\mathcal{P}_1 \cup \mathcal{P}_2)$, *i.e.* $\mathcal{P}_1 \subseteq \mathcal{P}_2 \implies C_\Delta(\mathcal{P}_1) = C_\Delta(\mathcal{P}_2)$.

Most non-pure logical properties for C_Δ do not hold. In particular, C_Δ does not satisfy the properties of (LL) left-logical equivalence; (CC) conjunction of conclusions; (LA) left absorption; (RA) right absorption; (RN) rational negation; (RM) rational monotonicity; (DR) disjunctive rationality, as shown next.

- LL: Given two programs $\mathcal{P}_1$ and $\mathcal{P}_2$, $C_\perp(\mathcal{P}_1) = C_\perp(\mathcal{P}_2)$ does not imply $C_\Delta(\mathcal{P}_1) = C_\Delta(\mathcal{P}_2)$. Consider $\mathcal{P}_1 = \{ (y \leftarrow , 1) \}$ and $\mathcal{P}_2 = \mathcal{P}_1 \cup \{ (x \leftarrow y, 0.9) \}$.
- CC: Arguments supporting conjunctions of conclusions cannot be expressed in P-DeLP language, as goals are restricted to literals.
- LA: Consider the program $\mathcal{P} = \{(Q, \alpha)\}$, where Q is a literal, $\alpha < 1$. Then $C_\perp(C_\Delta(\mathcal{P})) = C_\perp(\{(Q, \alpha)\}) = \emptyset \neq C_\Delta(\mathcal{P})$.
- RA: Consider the same counterexample given for LA. Analogously, $C_\Delta(C_\perp(\mathcal{P})) = C_\Delta(\emptyset) = \emptyset \neq C_\Delta(\mathcal{P})$.
- RN: Consider $\mathcal{P}_1 = \{ (\sim p \leftarrow x, 1), (\sim p \leftarrow \sim x, 1), (r \leftarrow , 1), (z \leftarrow p, 1), (p \leftarrow r, 0.9) \}$. Then $\mathcal{P}_1 \vdash_\Delta \langle \mathcal{A}_1, z, 0.9 \rangle$, with $\mathcal{A}_1 = \{ (p \leftarrow r, 0.9) \}$. However, $\mathcal{P}_1 \cup \{ (x \leftarrow , 1) \} \not\vdash_\Delta \langle \mathcal{A}_1, z, 0.9 \rangle$, and $\mathcal{P}_1 \cup \{ (\sim x \leftarrow , 1) \} \not\vdash_\Delta \langle \mathcal{A}_1, z, 0.9 \rangle$.
- RM: Consider the same counterexample as given for RN. Then $\mathcal{P}_1 \vdash_\Delta \langle \mathcal{A}_1, z, 0.9 \rangle$, but it is not the case that $\mathcal{P}_1 \cup \{ (x \leftarrow , 1) \} \vdash_\Delta \langle \mathcal{A}_1, z, 0.9 \rangle$ nor $\mathcal{P}_1 \vdash_\Delta (\sim x \leftarrow , 1)$.
- DR: Clearly, C_Δ does not satisfy property (e), as disjunctions cannot be expressed as wffs in the P-DeLP object language.

4.2 Logical Properties for C_w

Next we will analyze some relevant logical properties for C_w . Notice that by definition C_w satisfies inclusion.

Proposition 7. *If $\mathcal{P} \subseteq \mathcal{Q}$, then $C_w(\mathcal{P}) \subseteq C_w(\mathcal{Q})$.*

Monotonicity does not hold, as can be seen from the counterexample used for monotonicity in C_Δ ; in that case, $(q, 0.9) \in C_w(\mathcal{P})$, but $(q, 0.9) \notin C_w(\mathcal{P} \cup \{(\sim p, 1)\})$. Semi-monotonicity does not hold either for C_w , as adding new de-feasible clauses cannot invalidate already derivable arguments, but it can enable new ones that were not present before, thus modifying the dialectical relationships among arguments. Arguments that were warranted may therefore no longer keep that epistemic status. Consider a variant of the previous counterexample: let $\mathcal{P} = \{ (q, 1), (\sim p \leftarrow , 0.9) \}$. Then $(p, 0.9) \in C_w(\mathcal{P})$, as there is an argument $\langle \mathcal{A}, p, 0.9 \rangle$ on the basis of $\mathcal{P}$. However, $(p, 0.9) \notin C_w(\mathcal{P} \cup \{(\sim p, 0.95)\})$, as $\langle \mathcal{A}, p, 0.9 \rangle$ is defeated by $\langle \mathcal{B}, \sim p, 0.95 \rangle$, with $\mathcal{B} = \{(\sim p, 0.95)\}$. There are no more arguments to consider, and hence $\langle \mathcal{A}, p, 0.9 \rangle$ is not warranted. From our current analysis cumulativity and idempotence seem to hold for the C_w operator: we have not found any counterexample showing that these two properties do not hold, and we are currently studying the formulation of a proof. In comparison with C_Δ such a formal analysis is much more complex, as dialectical trees are not structures that can be recursively defined (notice that subtrees of dialectical trees are not dialectical trees). The reason for this is given by the different dialectical constraints that have to be taken into account (see previous discussion on acceptability in argumentation lines in Section 3).

Property	C_Δ	C_w	Comments	Property	C_Δ	C_w	Comments
Inclusion	o	o	Prop. 2 and 7.	Conj. concl.	×	×	
Idempotence	o	o?	Prop. 2 & Conj. 2.	Subclass. cumm.	o	o	Prop. 6 and 9.
Cumulativity	o	o?	Prop. 4 & Conj. 1.	Left absorption	×	×	
Monotonicity	×	×		Right absorption	×	×	
Semi-monotonicity	o	×	Prop. 3 and 7.	Rational Negation	×	×	
(Horn) Supraclass.	o	o	Prop. 5 and 8	Disj. Rationality	×	×	
Left-logical equiv.	×	×		Rational Monoton.	×	×	
Horn Right Weak.	o	o	Prop. 5 and 8				

Fig. 1. Logical properties in P-DeLP: summary

Conjecture 1. $T \vdash C_w(P_1) = C_w(P_2)$ *i.e.* $\mathcal{P}_1 \subseteq \mathcal{P}_2 \subseteq C_w(\mathcal{P}_1)$

Conjecture 2. $T \vdash C_w(C_w(\mathcal{P})) = C_w(\mathcal{P})$ *i.e.* $C_w(\mathcal{P}) = C_w(C_w(\mathcal{P}))$.

Proposition 8. $T \vdash C_w(H) \subseteq C_\perp(H)$ *i.e.* $C_\perp(\mathcal{P}) \subseteq C_w(\mathcal{P})$, $(H) \vdash (Y, \alpha) \in C_\perp(\mathcal{P})$ *i.e.* $(Y, \alpha) \in C_w(\mathcal{P})$, $(X \leftarrow Y, 1) \in C_\perp(\mathcal{P})$, $(X, \alpha) \in C_w(\mathcal{P})$.

P. (Sketch) Supraclassicality follows from Prop. 1. For the case of right weakening, let us suppose $(Y, \alpha) \in C_w(\mathcal{P})$, *i.e.* $\mathcal{P} \vdash_w \langle \mathcal{A}_1, Y, \alpha \rangle$, for some argument $\langle \mathcal{A}_1, Y, \alpha \rangle$. If $(X \leftarrow Y, 1) \in C_\perp(\mathcal{P})$, then necessarily $(X \leftarrow Y, 1) \in \mathcal{P}$ (by def. of $C_\perp$). By Prop. 5, if $\mathcal{P} \vdash_\Delta \langle \mathcal{A}_1, X, \alpha \rangle$. Clearly argument $\langle \mathcal{A}_1, X, \alpha \rangle$ and $\langle \mathcal{A}_1, Y, \alpha \rangle$ have the same set of associated defeaters. Hence if $\langle \mathcal{A}_1, Y, \alpha \rangle$ is warranted, then $\langle \mathcal{A}_1, X, \alpha \rangle$ also is.

Proposition 9. $T \vdash C_\Delta(P_1) \subseteq C_\perp(P_1)$ *i.e.* $\mathcal{P}_1 \subseteq \mathcal{P}_2 \subseteq C_\perp(\mathcal{P}_1)$, $C_w(\mathcal{P}_1) = C_w(\mathcal{P}_2)$.

P. Not included for space reasons.

As for C_Δ , most non-pure logical properties for C_w do not hold. In particular, C_w does not satisfy the properties of LL, CC, LA, RA, RN, RM and DR. In all cases this is based on the existence of counterexamples following the same line of reasoning as for C_Δ .

4.3 Discussion

Figure 1 summarizes the logical properties discussed before. When analyzing argumentative inference under the operator C_Δ , idempotence shows us that adding argument conclusions as new facts to a given program does not add any new inference capabilities. Cumulativity shows us that any argument obtained from a program $\mathcal{P}$ can be kept as an intermediate proof (lemma) to be used in building more complex arguments. (Horn) supraclassicality indicates that every conclusion that follows via traditional SLD inference (involving only certain clauses) can be considered as a special form of argument (namely, an empty argument),

whereas Horn right weakening tells us that certain rules in P-DeLP preserve the usual semantics for Horn rules (the existence of a certain rule $X \leftarrow Y$ causes that every argument concluding Y is also an argument for X). Computing warrant also can be better understood in the light of the logical properties for C_w . From Horn supraclassicality it follows that every conclusion obtained from certain clauses is a particular case of warranted literal, whereas Horn right weakening indicates that non-defeasible rules behave as such in the meta-level (a strong rule $(Y \leftarrow X, 1)$ ensures that every warranted argument $\mathcal{A}$ for (X, α) allow us to ensure that (Y, α) is also warranted. Cumulativity for C_w is specially interesting, as we will further discuss in the next Section.

5 Related Work. Conclusions

Research in logical properties for defeasible argumentation can be traced back to Benferhat *et al.* [9, 10] and Vreeswijk [13]. In the context of his abstract argumentation systems, Vreeswijk showed that many logical properties for non-monotonic inference relationships turned out to be counter-intuitive for argument-based systems. Benferhat *et al.* [9] were the first who studied argumentative inference in uncertain and inconsistent knowledge bases. They defined an argumentative consequence relationship $\vdash_{\mathcal{A}}$ taking into account the existence of arguments favoring a given conclusion against the absence of arguments in favor of its contrary. In contrast, the $\vdash_{\Delta}$ relationship proposed in this paper takes into account possible argument derivable from the program. In [9, 10] the authors also extend the argumentative relation $\vdash_{\mathcal{A}}$ to prioritized knowledge bases, assessing weights to conclusions on the basis of the $\vdash_{\pi}$ -entailment relationship from possibilistic logic [7]. A direct comparison to our $\vdash_w$ relationship is not easy since we are using a logic programming framework and not general propositional logic, but roughly speaking while $\vdash_{\pi}$ takes into account the inconsistency degree associated with the whole knowledge base, our logic programming frame allows us to perform a dialectical analysis restricted only to conflicting arguments related with the query being solved. More recently there have been generic approaches connecting defeasible reasoning and possibilistic logic (e.g.[14]). Preference-based approaches to argumentation have been also developed, many of them oriented towards formalizing conflicting desires in multiagent systems [15, 16]. Part of our current work involves studying the formalization of expansion operators for such contexts.

In [11] some examples are informally presented to show that argumentation systems should assign facts a special status, and therefore should be cumulative. In the particular case of cumulativity and idempotence we have conjectured that they hold in the context of P-DeLP. Should such conjectures be true, this would provide an interesting result in comparison with [11], as it would mean that warranted conclusions (Q, α) in a given program $\mathcal{P}$ could be introduced as new uncertain facts, speeding up computation of new future queries. However, such facts would not have the same epistemic status as discussed in [11], where warranted conclusions are analyzed as certain facts for the study of cumulativity.

We have shown that P-DeLP provides a useful framework for making a formal analysis of logical properties in defeasible argumentation under uncertainty. We contend that a formal analysis of defeasible consequence is mandatory to get an in-depth understanding of the behavior of argumentation frameworks. Expansion operators provide a natural tool for characterizing that behavior, as well as useful criteria when developing and implementing new argumentation frameworks or assessing their expressive power.

Acknowledgements. We want to thank anonymous reviewers for their useful comments. This work was supported by Spanish Projects TIC2003-00950, TIN2004-07933-C03-01/03, TIN2004-07933-C03-03, by Ramón y Cajal Program (MCyT, Spain) and by CONICET (Argentina).

References

1. Chesñevar, C.I., Simari, G., Alsinet, T., Godo, L.: A Logic Programming Framework for Possibilistic Argumentation with Vague Knowledge. In: Proc. Intl. Conf. in Uncertainty in Artificial Intelligence (UAI 2004). Banff, Canada. (2004) 76–84
2. Alsinet, T., Godo, L.: A complete calculus for possibilistic logic programming with fuzzy propositional variables. In: Proc. of the UAI-2000 Conference. (2000) 1–10
3. Alsinet, T., Godo, L.: A proof procedure for possibilistic logic programming with fuzzy constants. In: Proc. of the ECSQARU-2001 Conference. (2001) 760–771
4. Chesñevar, C., Maguitman, A., Simari, G.: A first approach to argument-based recommender systems based on defeasible logic programming. In: Proc. 10th Intl. Workshop on Non-Monotonic Reasoning. Whistler, Canada. (2004) 109–117
5. Chesñevar, C., Maguitman, A.: An Argumentative Approach to Assessing Natural Language Usage based on the Web Corpus. In: Proc. of the ECAI-2004 Conference. Valencia, Spain. (2004) 581–585
6. Lifschitz, V.: Foundations of logic programming. In: Principles of Knowledge Representation. CSLI Publications (1996) 69–127
7. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In D.Gabbay, C.Hogger, J.Robinson, eds.: Handbook of Logic in Art. Int. and Logic Prog. (Nonmonotonic Reasoning and Uncertain Reasoning). Oxford Univ. Press (1994) 439–513
8. Makinson, D.: General patterns in nonmonotonic reasoning. In D.Gabbay, C.Hogger, J.Robinson, eds.: Handbook of Logic in Art. Int. and Logic Prog. Volume Nonmonotonic and Uncertain Reasoning. Oxford University Press (1994) 35–110
9. Benferhat, S., Dubois, D., Prade, H.: Argumentative inference in uncertain and inconsistent knowledge bases. In: Proc. of UAI. (1993) 411–419
10. Benferhat, S., Dubois, D., Prade, H.: Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study. part ii: The prioritized case. In Orłowska, E., ed.: Logic at work. Volume 24. Physica-Verlag, Heidelberg (1998) 473–511
11. Prakken, H., Vreeswijk, G.: Logical Systems for Defeasible Argumentation. In Gabbay, D., F.Guenther, eds.: Handbook of Phil. Logic. Kluwer (2002) 219–318
12. Chesñevar, C., Maguitman, A., Loui, R.: Logical Models of Argument. ACM Computing Surveys **32** (2000) 337–383
13. Vreeswijk, G.A.: Studies in Defeasible Argumentation. PhD thesis, Vrije University, Amsterdam (Holanda) (1993)

14. Benferhat, S., Dubois, D., Prade, H.: The possibilistic handling of irrelevance in exception-tolerant reasoning. *Annals of Math. and AI* **35** (2002) 29–61
15. Amgoud, L.: A formal framework for handling conflicting desires. In: *Proc. of the ECSQARU-2003 Conference*. (2003) 552–563
16. Amgoud, L., Cayrol, C.: Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning* **29** (2002) 125–169

Gradual Valuation for Bipolar Argumentation Frameworks

C. Cayrol and M.C. Lagasquie-Schiex

IRIT, Université Paul Sabatier, Toulouse

Abstract. In this paper, we extend the abstract argumentation framework proposed by [1] in order to take into account two kinds of interaction between arguments: a *positive* interaction (an argument can help, support another argument) and a *negative* interaction (an argument can attack another argument). In this new abstract argumentation framework, called a *bipolar argumentation framework*, we propose a *gradual interaction-based valuation* process. With this process, the value of each argument A only depends on the value of the arguments which are directly interacting with A in the argumentation system.

1 Introduction

A rational agent can express claims and judgements, aiming at reaching a decision, a conclusion, or informing, convincing, negotiating with other agents. Pertinent information may be insufficient or contrastedly there may be too much relevant but partially incoherent information. And, in case of multi-agent interaction, conflicts of interest are inevitable. So, agents can be assisted by argumentation.

Argumentation has been applied in various domains and applications such as plausible inference from inconsistent knowledge bases, decision making and negotiation (see [1, 2, 3, 4, 5, 6, 7, 8]). For example, recent works on negotiation [9, 4, 10, 11, 5, 12, 13, 14] have argued that argumentation can play a key role in finding a compromise. Indeed, an offer supported by a “good argument” has a better chance to be accepted by another agent. Argumentation may also lead an agent to change its goals and finally may constrain an agent to respond in a particular way.

In all these disparate cases, an argumentation process follows five steps. The first step is the definition of the *arguments*: the notion of argument commonly refers to the concepts of explanation, proof, justification; arguments aim to justify beliefs, or decisions; they can take the form of a piece of text or discourse, by which one tries to convince the reader that a given claim is true, or they can be seen as a logical proof of a claim¹. The second step is the definition of the different *interactions between arguments*: arguments formed from a knowledge base cannot be considered independently; indeed most of the arguments are in interaction. The third step consists in *valuing the arguments*: the basic idea behind this valuation process is to give a *weight* for each argument; the different weights make it

¹ Formally, arguments are built around an underlying representation language. Different basic forms of arguments can be encountered, depending on the language and on the rules for constructing arguments.

possible to compare the arguments; different criteria can be used in order to define the weights. In the fourth step, one *selects the most acceptable arguments*: it is necessary to define the status of arguments on the basis of all the ways in which they interact, and of the available valuation process. Then the last step permits to *conclude the argumentation*: the status of arguments in turn determines the status of conclusions; beliefs, goals or decisions in favour of which one of the best arguments exists are really justified.

In this paper, we are concerned by the valuation step. On this point, there exist many works. Some of them propose valuations built using the interactions between arguments (see [15, 16, 17, 18])². A gradual interaction-based valuation can be used in order to reflect the way in which arguments are weakened by other arguments. This notion of graduality refers to the set of the values (richer is this set, more gradual is the valuation) and its introduction in the valuation step permits to work with more than the two classical values: accepted and not accepted.

In most existing argumentation frameworks, only one kind of interaction is considered between arguments, represented by the so-called defeat relation. However, recent studies on argumentation [6, 8, 31] have shown that another kind of interaction may exist between the arguments. Indeed, an argument can defeat another argument, but it can also support another one. This suggests a notion of bipolarity, *i.e.* the existence of two independent kinds of information which have a diametrically opposed nature and which represent repellent forces. The notion of bipolarity appears in many domains and is essential in order to represent realistic knowledge (see discussions in [32, 33, 34, 35]). For instance, in [35], two kinds of preferences are distinguished: *positive* preferences (what the agent really wants) and *negative* preferences (what the agent rejects). This distinction between positive and negative preferences is supported by studies in cognitive psychology which have shown that these two types of preferences are independent and processed separately in the mind. However, bipolarity is not always related to the notion of preference.

Our purpose is to reconsider the gradual valuation of arguments in a bipolar framework. Such a valuation will reflect the way in which arguments are enforced or weakened by other arguments. Since we are concerned with the valuation step, we present our work in an abstract argumentation framework. So, first, we extend the basic abstract framework proposed by Dung, in order to handle both defeat and support relations between arguments. Section 2 recalls Dung's framework and presents our "bipolar extension". In Section 3, existing gradual valuation procedures are presented. Then, we will describe new gradual valuation procedures for the bipolar argumentation framework (see Section 4). Section 5 is devoted to some concluding remarks.

2 Argumentation Frameworks

2.1 Background

In [1], Dung has proposed an abstract framework for argumentation (it will be extended in Section 2.2.2) in which he focuses only on the definition of the status of arguments:

² Note that there also exist valuations which are defined without taking into account the interactions between arguments (see [19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 3, 30]).

an *argumentation framework* is a pair $\langle \mathcal{A}, \mathcal{R} \rangle$ of a set $\mathcal{A}$ of arguments and a binary relation $\mathcal{R}$ on $\mathcal{A}$ called a *defeat relation*. $A_i \mathcal{R} A_j$ means that A_i *defeats* A_j . The set $\{A_i \in \mathcal{A} \mid A_i \mathcal{R} A\}$ is denoted by $\mathcal{R}^-(A)$ and the set $\{A_i \in \mathcal{A} \mid A \mathcal{R} A_i\}$ is denoted by $\mathcal{R}^+(A)$. An argumentation framework may be represented by a directed graph whose nodes are arguments and edges represent the defeat relation.

Definition 1. Let $\mathcal{G}$ be the graph associated with the abstract argumentation framework $\langle \mathcal{A}, \mathcal{R} \rangle$. A leaf of $\mathcal{G}$ is an argument $A \in \mathcal{A}$ such that $\mathcal{R}^-(A) = \emptyset$. A path from A to B is a sequence of arguments $\mathcal{P} = A_1 - \dots - A_n$ such that $A = A_1$, $A_1 \mathcal{R} A_2$, $\dots$, $A_{n-1} \mathcal{R} A_n$, $A_n = B$. $\mathcal{P}(A, B)$ denotes the set of the paths from A to B .

Definition 2. An argument A_i defends A_j against B iff $B \mathcal{R} A_j$ and $A_i \mathcal{R} B$. The direct defeaters of A are the elements of $\mathcal{R}^-(A)$. The direct defenders of A are the direct defeaters of the elements of $\mathcal{R}^-(A)$.

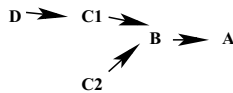
Dung focuses on the selection step of an argumentation process. So, he proposed different notions of acceptability using some characteristic properties:

Definition 3. A subset S of $\mathcal{A}$ is conflict-free iff there exist no A_i, A_j in S such that $A_i \mathcal{R} A_j$. A subset S of $\mathcal{A}$ defends collectively an argument A_i iff for each argument B , if $B \mathcal{R} A_i$ there exists C in S such that $C \mathcal{R} B$.

Then several *semantics for acceptability* have been defined by [1]. For example:

Definition 4. A subset S of $\mathcal{A}$ is an *admissible set* iff S is conflict-free and S defends collectively all its elements. A subset S of $\mathcal{A}$ is a *preferred extension* iff S is maximal for set inclusion among the admissible sets of $\mathcal{A}$.

Note that acceptability could be used for an interaction-based valuation of arguments. In that case, the value of an argument depends on its *acceptability*, i.e. its membership of some sets (acceptable sets or extensions). We obtain a binary valuation (an argument is accepted or not accepted). The previous notions are illustrated on the following argumentation system:



in which A is defended by C_1 and C_2 , and there is only one preferred extension $\{D, C_2, A\}$. So, with respect to the preferred semantics, we have a binary valuation: D, C_2, A are accepted and C_1 and B are not accepted.

2.2 Abstract Bipolar Argumentation Framework

2.2.1 Bipolarity

As already said, due for instance to the presence of inconsistency in knowledge bases, arguments may be conflicting. The defeat relation captures these conflicts.

However, most logical theories of argumentation assume that an argument A_1 which defeats a counter-argument A_3 against an argument A_2 supports A_2 , but this notion of support does not have to be formalized in a really different way than the notion of defeat. It is the case of the basic argumentation framework recalled in Section 2.1, in

which only one kind of interaction is explicitly represented by the *defeat* relation and the support of an argument A by another argument B can be represented only if B defends A (so, support and defeat are *dependent* notions). It is a parsimonious strategy, but it is not a correct description of the process of argumentation. Let us take an example:

We want to begin a hike. We prefer a sunny weather, then a sunny and cloudy one, then a cloudy but not rainy weather, in this order. We will cancel the hike only if the weather is rainy. But clouds could be a sign of rain. We look at the sky early in the morning. It is cloudy. The following exchange of informal arguments occurs:

1. Today we have time, we begin a hike.
2. The weather will be cloudy, clouds are sign of rain, we would have better to cancel the hike.
3. These clouds are early patches of mist, the day will be sunny, without clouds, so the weather will be not cloudy.
4. Clouds will not grow, so the weather will be cloudy, but not rainy.

In this exchange, we have the following path of conflicts between arguments: argument 4 defeats argument 3 which defeats argument 2 which defeats argument 1. So, with Dung's framework, argument 3 is a defender of argument 1, and argument 4 is a defeater of argument 1. Nevertheless, arguments 3 and 4 support the hike project. So, the idea of a chain of arguments and counter-arguments in which we just have to count the links and take the even one as defeaters and the odd ones as supports is an oversimplification. So, the notion of defence proposed by [1] and recalled in Section 2.1 is not sufficient to represent support.

In order to represent realistic examples in an argumentation process, we need a more powerful tool than the abstract argumentation framework proposed by Dung. In particular, we are interested in modelling situations where two *independent* kinds of interactions are available: a positive and a negative one (see for example [6] in the medical domain). So, following [6, 8], we propose a new argumentation framework: an abstract bipolar argumentation framework. We extend the basic argumentation framework of [1] by introducing a new kind of interaction between arguments: the *support* relation which represents the support, the help brought by some arguments to other arguments³. This new relation is assumed to be totally independent of the defeat relation. So, we have a bipolar representation of the interactions between arguments.

2.2.2 Formal Definition

Definition 5. An abstract bipolar argumentation framework $\langle \mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup} \rangle$ consists of a set $\mathcal{A}$ of arguments, a binary relation $\mathcal{R}_{def}$ on $\mathcal{A}$ called a defeat relation and another binary relation $\mathcal{R}_{sup}$ on $\mathcal{A}$ called a support relation.

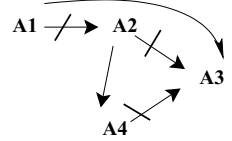
Here, we are not interested in the structure of the arguments and we consider arbitrary defeat and support relations. The only assumption is that $\mathcal{R}_{def}$ and $\mathcal{R}_{sup}$ are independent of each other: i.e. $\mathcal{R}_{sup}$ is not defined in terms of $\mathcal{R}_{def}$ (and *vice-versa*).

³ If the support relation is removed, we retrieve Dung's framework.

Consider A_i and $A_j \in \mathcal{A}$, $A_i \mathcal{R}_{\text{def}} A_j$ (resp. $A_i \mathcal{R}_{\text{sup}} A_j$) means that A_i defeats A_j (resp. A_i supports A_j). Consider $A \in \mathcal{A}$, $A \mathcal{R}_{\text{def}} B$ is represented by $A \not\rightarrow B$ and $A \mathcal{R}_{\text{sup}} B$ is represented by $A \rightarrow B$. The set $\{A_i \in \mathcal{A} | A_i \mathcal{R}_{\text{def}} A\}$ is denoted by $\mathcal{R}_{\text{def}}^-(A)$ and the set $\{A_i \in \mathcal{A} | A_i \mathcal{R}_{\text{sup}} A\}$ is denoted by $\mathcal{R}_{\text{sup}}^+(A)$. In the same way, we define $\mathcal{R}_{\text{sup}}^-(A)$ and $\mathcal{R}_{\text{def}}^+(A)$. $\langle \mathcal{A}, \mathcal{R}_{\text{def}}, \mathcal{R}_{\text{sup}} \rangle$ defines a directed graph $\mathcal{G}$ called the *bipolar graph* which is assumed acyclic.

Example 1.

The framework $\langle \mathcal{A} = \{A_1, A_2, A_3, A_4\}, \mathcal{R}_{\text{def}} = \{(A_2, A_3), (A_4, A_3), (A_1, A_2)\}, \mathcal{R}_{\text{sup}} = \{(A_2, A_4), (A_1, A_3)\} \rangle$ defines the following graph $\mathcal{G}$ with the root A_3 :



Definition 6. Let $\mathcal{G}$ be the bipolar graph associated with the abstract bipolar argumentation framework $\langle \mathcal{A}, \mathcal{R}_{\text{def}}, \mathcal{R}_{\text{sup}} \rangle$. A leaf of $\mathcal{G}$ is an argument $A \in \mathcal{A}$ such that $\mathcal{R}_{\text{def}}^-(A) = \emptyset$ and $\mathcal{R}_{\text{sup}}^-(A) = \emptyset$. A path from A to B is a sequence of arguments $\mathcal{P} = A_1 - \dots - A_n$ such that $A = A_1$, $A_1 \mathcal{R}_1 A_2$, $\dots$, $A_{n-1} \mathcal{R}_{n-1} A_n$, $A_n = B$, and $\forall i = 1, \dots, n-1$, $\mathcal{R}_i = \mathcal{R}_{\text{def}}$ or $\mathcal{R}_{\text{sup}}$. $\mathcal{P}(A, B)$ denotes the set of the paths from A to B .

Definition 7. Consider $A \in \mathcal{A}$. The direct defeaters of A are the elements of $\mathcal{R}_{\text{def}}^-(A)$. The direct supporters of A are the elements of $\mathcal{R}_{\text{sup}}^-(A)$.

2.2.3 Instantiating the Bipolar Abstract Framework

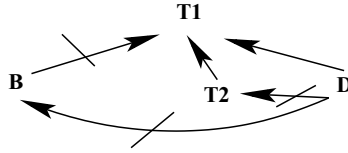
In a bipolar argumentation framework, the support relation carries positive information while the defeat relation carries negative information, and positive and negative information are represented in the same structure (the bipolar graph). It is a particularity of the argumentation context (in many other domains, positive and negative information are represented in two distinct frameworks, and sometimes they do not have the same nature). There exist many different formal definitions for these relations. With a classical logic, using the explanatory arguments (a pair $\langle \text{Explanation}, \text{Conclusion} \rangle$ where Explanation implies Conclusion, see [21, 23, 29, 19]), we give here the most useful definitions:

Definition 8. Let A_1 and A_2 be two explanatory arguments ($A_1 = \langle H_1, h_1 \rangle$ and $A_2 = \langle H_2, h_2 \rangle$). A_1 assumption-attacks (resp. assumption-supports) A_2 iff $\exists \phi \in H_2$ such that $\phi = \neg h_1$ (resp. $\phi = h_1$ and $H_1 \cup H_2$ is consistent). A_1 rebuts (resp. conclusion-supports) A_2 iff $\neg h_2 = h_1$ (resp. $h_2 = h_1$ and $H_1 \cup H_2$ is consistent).

Example 2. 3 agents (Tom, Ben and Dan) discuss about a hike:

- T_1 : Today we have time, we begin a hike.
- B : No, the weather is cloudy, clouds are sign of rain, it is more cautious to cancel the hike.
- T_2 : These clouds are early patches of mist, the day will be sunny without cloud, so we can begin the hike.
- D : No, these clouds are not early patches of mist. So, the day will not be sunny but cloudy. However, it will not rain, so we can begin the hike.

So, this argumentation system can be represented by:



3 The Existing Interaction-Based Valuation in Argumentation Frameworks

Within Dung's framework, several approaches have been proposed for valuing the arguments (see for example [28, 3, 26, 15, 16, 36, 37]). In some of them, the value of an argument depends on its interactions with the other arguments; in other ones, it depends on an intrinsic strength of the argument which can be defined for example by an explicit preference. In this paper, we are only concerned by the interaction-based valuations.

3.1 Interaction-Based Valuations with the Defeat Interaction

Let us consider the case where the set of arguments is equipped with only a binary defeat relation (see Section 2.1). The purpose is to provide a valuation of an argument which reflects the way in which this argument is defeated by other arguments. Different kinds of approaches can be distinguished. We present here only a local interaction-based valuation which will be extended in Section 4. Local approaches compute the value of an argument A from the values of the arguments which are directly related to A in the interaction graph. The first works about this kind of valuation have been realized by [15] on labelling processes, and [16] in a logical framework for deductive arguments. Then, in [36], a generic local approach is described which generalizes [16, 15]'s approaches:

Definition 9. Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation framework in Dung's sense. We assume that there exists a completely ordered set $\mathcal{V}$ with a minimum element (V_{Min}) and a maximum element (V_{Max}). Consider $A \in \mathcal{A}$ with $\{B_1, \dots, B_n\}$ denoting the set of direct defeaters of A . A local gradual valuation on $\langle \mathcal{A}, \mathcal{R} \rangle$ is a function $v : \mathcal{A} \rightarrow \mathcal{V}$ such that $v(A) = g(h(v(B_1), \dots, v(B_n)))$ with the function $h : \mathcal{V}^* \rightarrow \mathcal{H}^4$ valuing the quality of the defeat on A , and the function $g : \mathcal{H} \rightarrow \mathcal{V}$ with $g(x)$ decreasing on x .

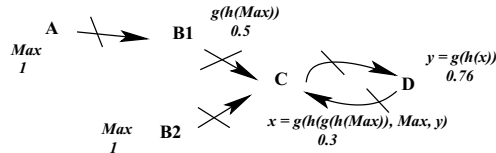
Example 3. During a discussion between reporters about the publication of an information I concerning the person X , the following arguments are presented:

- D : I is an important information, we must publish it.
- C : I concerns the person X , X is a private person and we cannot publish an information about a private person without her agreement, and X does not agree with the publication.
- B_1 : X is a minister, so X is a public person, not a private person.
- A : X has resigned, so X is no more a minister.

⁴ $\mathcal{V}^*$ denotes the set of the finite sequences of elements of $\mathcal{V}$ including the empty sequence. $\mathcal{H}$ is an ordered set.

- B_2 : I concerns a problem of public health, so even if I is a private information, we do not need the authorization of X for publishing it.

On the following figure, we show the interaction graph and the generic value of each argument (and in *italic font*, the value corresponding to [16]'s valuation⁵):



Note that we must compute fixpoints in order to value the elements of the cycles.

3.2 Interaction-Based Valuation with Defeat and Support Interactions

There exist at least two proposals for handling bipolarity on the interactions between arguments: Karacapilidis & Papadias [6] and Verheij [8].

- Karacapilidis & Papadias [6] propose an argumentation web-tool for decision making in a medical domain. This argumentation system, named HERMES, permits the expression and the weighting of arguments, verifies the coherence of preferences between arguments and values the arguments. The basic elements of this system are: *issues* (questions whose answer is open for discussion⁶), *positions* which express the support for, or the opposition to a solution, to another position, or to a constraint (a position gives an information for the discussion) and *constraints* which express a preference between two positions (so, it is a comparison tool on the set of positions). HERMES can label the solutions and the positions by the status “active” or “inactive”. At the end of the discussion, the “active” positions (resp. “inactive”) are accepted (resp. rejected). An “active” solution is a recommended choice among the other solutions concerning a same issue. Different labellings are proposed in HERMES. They are recursive: the label of an element e depends on the labels of the elements which are linked to e in the discussion graph. In HERMES, the discussion graph is acyclic, the value of a position p depends only on the *active* positions which are linked to p , and the value of a position is always binary, even when preference constraints are taken into account.

Labelling 1: A position is active if and only if there is neither support, nor defeat on this position, or if it is supported by an active position.

Labelling 2: A position is active if and only if it is not defeated by an active position.

Note that the labelling 1 and 2 do not permit to take into account in the same time the supports and the defeats on a position.

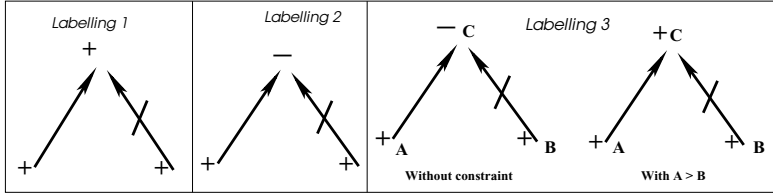
Labelling 3: A position is active if and only if there is neither support, nor defeat on this position, or if its score is strictly positive. The score of a position p is

⁵ With $h(x_1, \dots, x_n) = \sum(x_1, \dots, x_n)$, $h() = 0$, and $g(x) = \frac{1}{1+x}$.

⁶ For example, if the patient Y has the pathology X , what is the appropriate treatment? An issue is a set of *solutions*. Examples of solutions are surgical operation or use of medicines.

defined by: $\Sigma_i w(p_i) - \Sigma_j w(p_j)$ with p_i the *active* positions which support p and p_j the *active* positions which defeat p . Each position has the same initial weight and taking into account the preferences between positions modifies the relative weights of the positions.

Example 4. An active (resp. inactive) position will be denoted by + (resp. -).



- DEFLOG argumentation system proposed by Verheij [8] enables to express a support or a defeat between sentences in the language, with a new sentence using specific connectors (one for each kind of interaction). Examples of sentences (with $\nrightarrow$ for the defeat relation and $\rightarrow$ for the support relation) are: $A, B, (A \rightarrow B), (A \nrightarrow B), (C \rightarrow (A \rightarrow B)), (D \nrightarrow (A \rightarrow B))$.

Definition 10. Let S be a set of sentences. S supports a sentence H iff $H \in S$ or H is deduced from S with a sequence of supports.

S defeats a sentence H iff $\exists$ a sentence G such that S supports G and $(G \nrightarrow H)$.

S is conflict-free iff $\nexists$ any sentence H such that S supports and defeats H .

DEFLOG enables to define the dialectical interpretations of a given set of sentences.

Definition 11. Let S be a set of sentences, and (J, D) a partition of S .

(J, D) interprets S iff J is conflict-free and defeats the sentences of D .

If (J, D) interprets S , $(\text{Supp}(J), \text{Att}(J))$ is a dialectical interpretation (also called extension) of S with $\text{Supp}(J)$ (resp. $\text{Att}(J)$) denoting the set of the sentences supported (resp. defeated) by J .

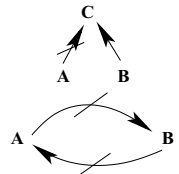
The sentences of $\text{Supp}(J)$ are the justified statements and those of $\text{Att}(J)$ are the defeated statements.

Note that the defeat relation and the support relation are explicitly expressed in the sentences. So, one can have an extension $(\text{Supp}(J), \text{Att}(J))$ of a set S such that some supported sentences by J do not belong to S . In some particular cases, the DEFLOG extensions correspond to the stable extensions of [1].

Example 5.

Let $S = \{A, B, C, B \rightarrow C, A \nrightarrow C\}$. There is no extension.

Let $S = \{A, B, A \nrightarrow B, B \nrightarrow A\}$. There are two extensions: $(\{A, A \nrightarrow B, B \nrightarrow A\}, \{B\})$, and $(\{B, A \nrightarrow B, B \nrightarrow A\}, \{A\})$.



4 Our Proposal

Our proposal is motivated by the following remarks. First, we want a valuation process which is not a selection process (as in DEFLOG). Then, we would like to define a valuation process on a rich set of values (in HERMES and DEFLOG, there are only two possible values). And the value of an argument must take into account the whole information concerning this argument (it is not the case in HERMES in which the value of an argument only depends on the *active* positions).

So, in the context of a bipolar argumentation framework as defined in Section 2.2.1, the valuation follows the same principles that have already been described in [36] completed with new principles corresponding to the “support” information. Here, we propose a *local valuation* in which the value of an argument only depends on the values of the direct defeaters or supporters of this argument.

There are 3 underlying principles. **P1**: The valuation of an argument depends on the values of its direct defeaters and of its direct supporters. **P2**: If the quality of the support (resp. defeat) increases then the value of the argument increases (resp. decreases). **P3**: If the quantity of the supports (resp. defeats) increases then the quality of the support (resp. defeat) increases.

4.1 Definition

In the respect of the previous principles, we assume that there exists a completely ordered set $\mathcal{V}$ with a minimum element (V_{Min}) and a maximum element (V_{Max}) and we propose the following formal definition for a local gradual valuation.

Definition 12. Let $\langle \mathcal{A}, \mathcal{R}_{\text{def}}, \mathcal{R}_{\text{sup}} \rangle$ be a bipolar argumentation framework. Let $A \in \mathcal{A}$ with $\mathcal{R}_{\text{def}}^-(A) = \{B_1, \dots, B_n\}$ and $\mathcal{R}_{\text{sup}}^-(A) = \{C_1, \dots, C_p\}$. A local gradual valuation on $\langle \mathcal{A}, \mathcal{R}_{\text{def}}, \mathcal{R}_{\text{sup}} \rangle$ is a function $v : \mathcal{A} \rightarrow \mathcal{V}$ such that $v(A) = g(h_{\text{sup}}(v(C_1), \dots, v(C_p)), h_{\text{def}}(v(B_1), \dots, v(B_n)))$ with the function h_{def} (resp. h_{sup}): $\mathcal{V}^* \rightarrow \mathcal{H}_{\text{def}}$ (resp. $\mathcal{V}^* \rightarrow \mathcal{H}_{\text{sup}}$)⁷ valuing the quality of the defeat (resp. support) on A , and the function $g : \mathcal{H}_{\text{sup}} \times \mathcal{H}_{\text{def}} \rightarrow \mathcal{V}$ with $g(x, y)$ increasing on x and decreasing on y .

The function h , $h = h_{\text{def}}$ or h_{sup} , must satisfy: if $x_i \geq x'_i$ then $h(x_1, \dots, x_i, \dots, x_n) \geq h(x_1, \dots, x'_i, \dots, x_n)$ **(1)**, $h(x_1, \dots, x_n, x_{n+1}) \geq h(x_1, \dots, x_n)$ **(2)**, $h() = \alpha \leq h(x_1, \dots, x_n)$, for all $x_1, \dots, x_n$ **(3)**, $h(x_1, \dots, x_n) \leq \beta$, for all $x_1, \dots, x_n$ **(4)**.

Note that Definition 12 produces a generic local gradual valuation. There exist several instances for this generic valuation:

- One of them is defined with $\mathcal{H}_{\text{def}} = \mathcal{H}_{\text{sup}} = \mathcal{V} = [-1, 1]$ interval of the real line, $h_{\text{def}}(x_1, \dots, x_n) = h_{\text{sup}}(x_1, \dots, x_n) = \max(x_1, \dots, x_n)$, and $g(x, y) = \frac{x-y}{2}$ (so, we have $\alpha = -1$, $\beta = 1$ and $g(\alpha, \alpha) = 0$).

⁷ $\mathcal{V}^*$ denotes the set of the finite sequences of elements of $\mathcal{V}$, including the empty sequence. $\mathcal{H}_{\text{def}}$ and $\mathcal{H}_{\text{sup}}$ are ordered sets.

⁸ So, α is the minimal value for a defeat (resp. a support) – i.e. there is no defeat (resp. no support) –.

⁹ So, β is the maximal value for a defeat (resp. a support) – i.e. for example, if there is an infinity of direct defeaters (resp. supporters) –.

- Example 2 (continued).** *With the first (resp. second) instance, $v(T_1) = \frac{1}{4}$ (resp. $\frac{9}{44}$).*

¹⁰ Note that $h_{\text{def}}(x_1, \dots, x_n, x_{n+1}) \geq h_{\text{def}}(x_1, \dots, x_n)$ because $\frac{x_{n+1}+1}{2} \geq 0$ when $x_{n+1} \in [-1, 1]$ (and the same for h_{sup}). We have also $h_{\text{def}}() = h_{\text{sup}}() = \alpha$, α being the minimal value of $[0, \infty[$, and β being the maximal value of $[0, \infty[$. We can verify also that $g(\alpha, \beta) = g(0, \infty) = -1$ and that $g(\beta, \alpha) = g(\infty, 0) = 1$ (1 and -1 being respectively V_{Min} and V_{Max}).

5 Conclusion

In this paper, we have proposed an extension of [1]’s abstract argumentation framework, in order to take into account two kinds of interaction between arguments: a support and a conflict. This extension is called a “bipolar abstract argumentation framework”. In this bipolar context, there already exist at least two possibilities for valuing arguments: HERMES system [6] and DEFLOG system [8]. However, both systems have some drawbacks: no graduality (only two possible values with HERMES, DEFLOG), some parts of the interacting arguments are not taken into account for the computation of the value (*cf.* HERMES), a new language is necessary (*cf.* DEFLOG), and DEFLOG directly proposes a selection process and not a valuation process.

So, we have defined a gradual interaction-based valuation. With this valuation, the value of an argument *A* only depends on the value of the arguments which directly interact with *A*. So, this valuation is a local one and the proposed definition gives a generic valuation whose two instances are studied in the paper.

In the future, we are interested in proposing a global interaction-based valuation and in comparing it to the local one. Another issue concerns the use of these valuations in order to define or to refine the acceptability of an argument or a set of arguments. And, at last, we plan to apply this bipolar framework and the joined processes (valuation and selection) on decision making problems.

References

1. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* **77** (1995) 321–357
2. Fox, J., Parsons, S.: On using arguments for reasoning about values and actions. In: *Proc. of AAAI-Symposium on qualitative preferences in deliberation and practical reasoning*. (1997) 55–63
3. Parsons, S.: Normative argumentation and qualitative probability. In: *Proc. of the 1th ECSQARU-FAPR, LNAI 1244*. (1997) 466–480
4. Amgoud, L., Maudet, N., Parsons, S.: Arguments, Dialogue and Negociation. In: *Proc. of the 14th ECAI*. (2000) 338–342
5. Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. *Journal of Logic and Computation* **8** (1998) 261–292
6. Karacapilidis, N., Papadias, D.: Computer supported argumentation and collaborative decision making: the HERMES system. *Information systems* **26** (2001) 259–277
7. Gordon, T., Karacapilidis, N.: The zeno argumentation framework. In: *Proc. of the 6th International Conference on Artificial Intelligence and Law*. (1997) 10–18
8. Verheij, B.: On the existence and multiplicity of extension in dialectical argumentation. In: *Proc. of the 9th NMR*. (2002) 416–425
9. Amgoud, L., Maudet, N., Parsons, S.: Modelling dialogues using argumentation. In: *Proc. of the 4th ICMAS*. (2000) 31–38
10. Amgoud, L., Prade, H.: Reaching agreement through argumentation: A possibilistic approach. In: *Proc. of the 9th KR*. (2004)
11. Kraus, S., Sycara, K., Evenchik, A.: Reaching agreements through argumentation: a logical model and implementation. Volume 104. *Journal of Artificial Intelligence* (1998)

12. Rahwan, I., Ramchurn, S.D., Jennings, N.R., McBurney, P., Parsons, S., Sonenberg, L.: Argumentation-based negotiation. *Knowledge engineering review* (2004)
13. Ramchurn, S.D., Jennings, N., Sierra, C.: Persuasive negotiation for autonomous agents: a rhetorical approach. In: CMNA. (2003)
14. Rahwan, I., Sonenberg, L., Dignum, F.: Towards interest-based negotiation. In: AA-MAS'2003. (2003)
15. Jakobovits, H., Vermeir, D.: Robust semantics for argumentation frameworks. *Journal of logic and computation* **9**(2) (1999) 215–261
16. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. *Artificial Intelligence* **128** (1–2) (2001) 203–235
17. Pollock, J.L.: Defeasible reasoning with variable degrees of justification. *Artificial Intelligence* **133** (2001) 233–282
18. Hunter, A.: Making argumentation more believable. In: Proc. of AAAI-04. (2004)
19. Amgoud, L., Cayrol, C.: Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning* **29** (2002) 125–169
20. Bench-Capon, T.: Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation* **13** (2003) 429–448
21. Simari, G., Loui, R.: A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence* **53** (1992) 125–157
22. Geffner, H., Pearl, J.: Conditional entailment : bridging two approaches to default reasoning. *Artificial Intelligence* **53** (1992) 209–244
23. Elvang-Goransson, M., Fox, J., Krause, P.: Dialectic reasoning with inconsistent information. In: Proc. of the 9th UAI. (1993) 114–121
24. Benferhat, S., Dubois, D., Prade, H.: Argumentative inference in uncertain and inconsistent knowledge bases. In: Proc. of the 9th UAI. (1993) 411–419
25. Dung, P., Son, T.C.: An argument-based approach to reasoning with specificity. *Artificial Intelligence* **133** (2001) 35–85
26. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics* **7** (1997) 25–75
27. Kowalski, R.A., Toni, F.: Abstract argumentation. *Artificial Intelligence and Law* **4** (1996) 275–296
28. Krause, P., Ambler, S., Elvang, M., Fox, J.: A logic of argumentation for reasoning under uncertainty. *Computational Intelligence* **11** (1) (1995) 113–131
29. Kohlas, J., Haenni, R., Berzati, D.: Probabilistic argumentation systems and abduction. In: Proc. of the 8th NMR - Uncertainty Frameworks subworkshop. (2000) 391–398
30. Pollock, J.L.: How to reason defeasibly. *Artificial Intelligence* **57** (1992) 1–42
31. Amgoud, L., Cayrol, C., Lagasque-Schiex, M.C.: On the bipolarity in argumentation frameworks. In: Proc. of the 10th NMR, Uncertainty Framework subworkshop. (2004) 1–9
32. Boutilier, C.: Towards a logic for qualitative decision theory. In: Proc. of the 4th KR. (1994) 75–86
33. Tan, S.W., Pearl, J.: Specification and evaluation of preferences under uncertainty. In: Proc. of the 4th KR. (1994) 530–539
34. Lang, J., Van der Torre, L., Weydert, E.: Utilitarian desires. *Journal of Autonomous Agents and Multi-Agents Systems* **5**(3) (2002) 329–363
35. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Bipolar representation and fusion of preferences in the possibilistic logic framework. In: Proc. of the 8th KR. (2002) 158–169
36. Cayrol, C., Lagasque-Schiex, M.C.: Gradual handling of contradiction in argumentation frameworks. In: *Intelligent Systems for Information Processing: From representation to Applications*. Elsevier (2003) 179–190
37. Amgoud, L.: Contribution à l'intégration des préférences dans le raisonnement argumentatif. PhD thesis, Université Paul Sabatier, Toulouse (1999)

On the Acceptability of Arguments in Bipolar Argumentation Frameworks

C. Cayrol and M.C. Lagasquie-Schiex

IRIT, Université Paul Sabatier, Toulouse

Abstract. In this paper, we extend the basic abstract argumentation framework proposed by Dung, by taking into account two independent kinds of interaction between arguments: a defeat relation and a support relation. In that new framework, called a bipolar argumentation framework, we focus on the concept of acceptability and propose new semantics defined from characteristic properties that a set of arguments must satisfy in order to be an output of the argumentation process. We generalize the well-known stable and preferred semantics by enforcing the coherence requirement for an acceptable set of arguments.

1 Introduction

A rational agent can express claims and judgements, aiming at reaching a decision, a conclusion, or informing, convincing, negotiating with other agents. Pertinent information may be insufficient or contrastedly there may be too much relevant but partially incoherent information. And, in case of multi-agent interaction, conflicts of interest are inevitable. So, agents can be assisted by argumentation, a process based on three steps: the exchange of arguments, the valuation of interacting arguments, and the definition of the most acceptable of these arguments.

Argumentation has been applied in various domains and applications such as plausible inference from inconsistent knowledge bases, decision making and negotiation (see [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]).

In most existing argumentation frameworks, only one kind of interaction, the conflict, is considered between arguments. It is represented by the so-called defeat relation. For example, an argument can take the form of a pair (set of premises, conclusion), where the set of premises entails the conclusion according to some logical inference schema. Then, the defeat interaction occurs between arguments in favour of and arguments against a given proposition.

However, recent studies on argumentation [5, 7, 11] have shown that another kind of interaction may exist between the arguments. Indeed, an argument can support another argument. It is the case for instance if an agent gives an argument which confirms a premise used by an argument provided by another agent.

Both relations, defeat and support, are assumed to be independent (*i.e.*, the support relation is not defined in terms of the defeat relation, and *vice-versa*).

So, we obtain a bipolar representation of the interactions between arguments. Bipolarity refers to the existence of two independent kinds of information which have a diametrically opposed nature and which represent repellent forces.

The notion of bipolarity is essential in order to represent realistic knowledge (see discussions in [12, 13, 14, 15]). For instance, in [15], two kinds of preferences are distinguished: *positive* preferences (what the agent really wants) and *negative* preferences (what the agent rejects). This distinction between positive and negative preferences is supported by studies in cognitive psychology which have shown that these two types of preferences are independent and processed separately in the mind. However, bipolarity is not always related to the notion of preference.

Abstracting from the structure of the arguments, our purpose is to revisit the concept of acceptability on the basis of the two kinds of interaction. So, we propose to extend the basic abstract argumentation framework proposed by Dung [1], by taking into account two independent relations: a defeat relation and a support relation. This new framework will be called an abstract bipolar argumentation framework. Then, following Dung's methodology, we establish characteristic properties that a set of arguments must satisfy in order to be an output of the argumentation process. Such a set of arguments must be in some sense coherent and must enable to win a dispute.

The paper is organized as follows: Section 2 briefly presents the argumentation process as well as the abstract framework proposed in [1]. Section 3 describes the bipolar extension of Dung's framework. In Section 4, we propose new semantics for acceptability. Section 5 is devoted to some concluding remarks and further issues. Proofs are omitted for lack of space.

2 Background on Argumentation Systems

An argumentation process follows three steps:

1. *the exchange of arguments*: the notion of argument commonly refers to the concepts of explanation, justification, or proof. Arguments aim to justify beliefs or decisions. They can take the form of a piece of text or discourse, by which one tries to convince the reader that a given claim is true, or they can be seen as a logical proof of a claim. Formally, arguments are built around an underlying representation language. Different basic forms of arguments can be encountered, depending on the language and on the rules for constructing arguments. Moreover, arguments formed from a knowledge base cannot be considered independently. Indeed most of the arguments are in interaction: arguments may be conflicting or on the contrary, arguments may support other arguments.
2. *the valuation of interacting arguments*: the basic idea is to assign a *weight* to each argument, in order to make comparisons between arguments. This weight represents the way in which an argument is enforced, or weakened by other arguments [16, 11].

3. *selecting the most acceptable arguments*: it is necessary to define the status of arguments on the basis of all the ways in which they interact. As an output of the argumentation system, the best arguments must be identified. Beliefs, goals or decisions in favour of which such arguments exist are really justified. Contrastedly, beliefs, goals and decisions relying upon rejected arguments will be discarded. Most often, acceptability is collective in the sense that sets of arguments are proved acceptable if they satisfy particular properties. Different kinds of properties define different semantics for acceptability.

In [1], Dung has proposed an abstract framework for argumentation in which he focuses only on the definition of the status of arguments. For that purpose, he supposes that a set of arguments is given, as well as the different conflicts between them.

We briefly recall that abstract framework:

- An *argumentation framework* is a pair $\langle \mathcal{A}, \mathcal{R}_{def} \rangle$ of a set $\mathcal{A}$ of arguments and a binary relation $\mathcal{R}_{def}$ on $\mathcal{A}$ called a *defeat relation*. $A_i \mathcal{R}_{def} A_j$ means that A_i *defeats* A_j (or A_j is *defeated* by A_i). An argumentation framework may be represented by a directed graph, called the *interaction graph*, whose nodes are arguments and edges represent the defeat relation.
- The notion of *defence* is defined from the notion of defeat by: an argument A_i *defends* A_j against B iff $B \mathcal{R}_{def} A_j$ and $A_i \mathcal{R}_{def} B$.

In Dung's framework, the *acceptability of an argument* depends on its membership of some sets, called acceptable sets or extensions. These extensions are characterised by particular properties. It is a collective acceptability. The main characteristic properties are:

- *Conflict-free*: a subset S of $\mathcal{A}$ is conflict-free iff there exist no A_i, A_j in S such that $A_i \mathcal{R}_{def} A_j$.
- *Defends collectively*: a subset S of $\mathcal{A}$ defends collectively an argument A_i iff for each argument B , if $B \mathcal{R}_{def} A_i$ there exists C in S such that $C \mathcal{R}_{def} B$.

Then several *semantics for acceptability* have been defined in [1]:

Let $\langle \mathcal{A}, \mathcal{R}_{def} \rangle$ be an argumentation framework.

- *Admissible*: a subset S of $\mathcal{A}$ is an admissible set iff S is conflict-free and S defends collectively all its elements.
- *Preferred*: a subset S of $\mathcal{A}$ is a preferred extension of $\langle \mathcal{A}, \mathcal{R}_{def} \rangle$ iff S is maximal for the set inclusion among the admissible sets of $\mathcal{A}$.
- *Stable*: a subset S of $\mathcal{A}$ is a stable extension of $\langle \mathcal{A}, \mathcal{R}_{def} \rangle$ iff S is conflict-free and S defeats each argument which does not belong to S .
- *Grounded*: a subset S of $\mathcal{A}$ is the grounded extension of $\langle \mathcal{A}, \mathcal{R}_{def} \rangle$ iff S is the least fixed point of the characteristic function of $\langle \mathcal{A}, \mathcal{R}_{def} \rangle$ ($F: 2^{\langle \mathcal{A}, \mathcal{R}_{def} \rangle} \rightarrow 2^{\langle \mathcal{A}, \mathcal{R}_{def} \rangle}$ with $F(S) = \{A \text{ such that } S \text{ defends collectively } A\}$).

The previous notions are illustrated on the following example.

Example 1. *A murder has been performed and the suspects are Liz, Mary and Peter. The following pieces of information have been gathered:*

The type of murder suggests us that the killer is a female. The killer is certainly small. Liz is tall and Mary and Peter are small. The killer has long hair and uses a lipstick. A witness claims that he saw the killer who was tall. Moreover, we are told that the witness is short-sighted, so he is no more reliable.

We use the following propositional symbols: s (the killer is small), f (the killer is a female), m (the killer is Mary), l (the killer has long hair and uses a lipstick), w (the witness is reliable), b (the witness is short-sighted).

So the following arguments can be formed:

- A_1 in favour of m , with premises $\{s, f, (s \wedge f) \rightarrow m\}$
- A_2 in favour of $\neg s$, with premises $\{w, w \rightarrow \neg s\}$
- A_3 in favour of $\neg w$, with premises $\{b, b \rightarrow \neg w\}$
- A_4 in favour of f , with premises $\{l, l \rightarrow f\}$

A_3 defeats A_2 which defeats A_1 . So A_3 defends A_1 against A_2 : $A_3 \not\rightarrow A_2 \rightarrow A_1$
Note that a defeat edge is represented by a crossed arrow on the interaction graph.

3 Bipolar Argumentation Frameworks

3.1 Bipolarity and Interaction

As already said, due for instance to the presence of inconsistency in knowledge bases, arguments may be conflicting. These conflicts are captured by the defeat relation in an argumentation system, and may be considered as negative interactions. Then, the concept of defence has been introduced in order to reinstate some of the defeated arguments, namely those whose defeaters are in turn defeated. So, most logical theories of argumentation assume that if an argument A_3 defends an argument A_1 against an argument A_2 , then A_3 is a kind of support for A_1 . The fact that A_3 defends A_1 may be considered as a positive interaction.

In the basic abstract argumentation framework recalled in Section 2, only negative interaction is explicitly represented by the *defeat* relation, and positive interaction is represented through the notion of defence. So, support and defeat are *dependent* notions. It is a parsimonious strategy, but it is not a correct description of the process of argumentation in realistic examples.

Example 1 (continued). *Consider the argument A_4 in favour of f , with premises $\{l, l \rightarrow f\}$. A_4 confirms the premise f of A_1 . So, A_4 strengthens A_1 . Contrastedly, A_3 defends A_1 against A_2 means that A_3 weakens the attack on A_1 brought by A_2 . So, on one side, A_1 gets a support and on the other side A_1 suffers a weakened defeat.*

So, we need a more powerful tool than the abstract argumentation framework proposed by Dung, in order to formalize situations where two *independent* kinds of interactions are available: a negative (which modelizes the conflicts) and a positive one (which is not a simple defence). Following [5, 7], we propose a new framework: an abstract bipolar argumentation framework.

3.2 An Abstract Bipolar Argumentation Framework

An abstract bipolar argumentation framework is an extension of the basic abstract argumentation framework introduced by [1] in which we use a new kind of interaction between arguments represented by the *support*¹ relation². This new relation is assumed to be totally independent of the defeat relation (*i.e.* it is not defined using the defeat relation). So, we have a bipolar representation of the interactions between arguments.

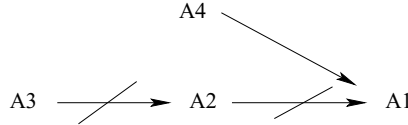
Definition 1 (Abstract bipolar argumentation framework). *An abstract bipolar argumentation framework $\langle \mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup} \rangle$ consists of a set $\mathcal{A}$ of arguments, a binary relation $\mathcal{R}_{def}$ on $\mathcal{A}$ called a defeat relation and another binary relation $\mathcal{R}_{sup}$ on $\mathcal{A}$ called a support relation.*

Consider A_i and $A_j \in \mathcal{A}$, $A_i \mathcal{R}_{def} A_j$ (resp. $A_i \mathcal{R}_{sup} A_j$) means that A_i defeats A_j (resp. A_i supports A_j).

A bipolar argumentation framework can still be represented by a directed graph, with two kinds of edges, one for the defeat relation and another one for the support relation.

Notations: Consider $A, B \in \mathcal{A}$, $A \mathcal{R}_{def} B$ is represented by $A \not\rightarrow B$ and $A \mathcal{R}_{sup} B$ is represented by $A \rightarrow B$. $\langle \mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup} \rangle$ defines a directed graph $\mathcal{G}_b$ called the *bipolar interaction graph*.

Example 1 (continued). A_4 supports A_1 and the bipolar graph is:



In the following, we abstract from the structure of the arguments and we consider arbitrary independent relations $\mathcal{R}_{def}$ and $\mathcal{R}_{sup}$. Moreover, we assume that the bipolar interaction graph is acyclic.

Definition 2 (Graphical representation of a bipolar argumentation framework). *Let $\mathcal{G}_b$ be the bipolar interaction graph associated with the abstract bipolar argumentation framework $\langle \mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup} \rangle$, we define:*

A leaf of $\mathcal{G}_b$ is an argument $A \in \mathcal{A}$ such that no argument defeats A and no argument supports A .

A path from A to B is a sequence of arguments $\mathcal{P} = A_1 - \dots - A_n$ such that $A = A_1$, $A_1 \mathcal{R}_1 A_2$, $\dots$, $A_{n-1} \mathcal{R}_{n-1} A_n$, $A_n = B$, and $\forall i = 1, \dots, n-1$, $\mathcal{R}_i = \mathcal{R}_{def}$ or $\mathcal{R}_{sup}$.

In order to define acceptability in bipolar abstract argumentation frameworks, we propose to follow Dung's methodology and to use the notion of defence which

¹ Note that the term "support" refers to a relation between 2 arguments and not a relation between premises and conclusion, as in Toulmin [17].

² If the support relation is removed, we retrieve Dung's framework.

enables to capture reinstatement. So, first, we generalize the key concept of defeat between two arguments, by combining a sequence of supports with a direct defeat:

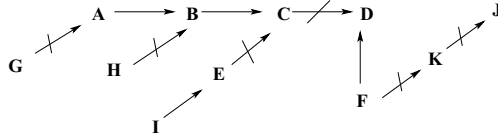
Definition 3 (Supported and indirect defeat).

A supported defeat for an argument B is a sequence³ $A_1 \mathcal{R}_1 \dots \mathcal{R}_{n-1} A_n$, $n \geq 3$, with $A_n = B$, such that $\forall i = 1 \dots n-2$, $\mathcal{R}_i = \mathcal{R}_{sup}$ and $\mathcal{R}_{n-1} = \mathcal{R}_{def}$.

An indirect defeat for an argument B is a sequence $A_1 \mathcal{R}_1 \dots \mathcal{R}_{n-1} A_n$, $n \geq 3$, with $A_n = B$, such that $\forall i = 2 \dots n-1$, $\mathcal{R}_i = \mathcal{R}_{sup}$ and $\mathcal{R}_1 = \mathcal{R}_{def}$.

The above definitions are illustrated on the following example:

Example 2. The following graph represents a bipolar argumentation system.



The paths in the graph $A-B-C-D$ and $E-C$ correspond to supported defeats. The path $G-A-B-C$ corresponds to an indirect defeat.

Taking into account sequences of supports and defeats leads to the following definitions applying to sets of arguments:

Definition 4 (Set-defeat and set-support). Let $S \subseteq \mathcal{A}$, let $A \in \mathcal{A}$.

S set-defeats A iff there exists a supported defeat or an indirect defeat for A from an element of S .

S set-supports A iff there exists a sequence of the form $A_1 \mathcal{R}_1 \dots \mathcal{R}_{n-1} A_n$, $n \geq 2$, such that $\forall i = 1 \dots n-1$, $\mathcal{R}_i = \mathcal{R}_{sup}$ with $A_n = A$ and $A_1 \in S$.

The notation “set-defeat” and “set-support” means that the defeat and the support relations apply to sets of arguments.

Example 2(continued). The set $\{A, H\}$ set-defeats D and B and set-supports B .

Using the notion of set-defeat, we are able to propose a definition for collective defence⁴:

Definition 5 (Defence by a set of arguments). Let $S \subseteq \mathcal{A}$. Let $A \in \mathcal{A}$. S defends collectively A iff $\forall B \in \mathcal{A}$, if $\{B\}$ set-defeats A then $\exists C \in S$ such that $\{C\}$ set-defeats B .

Example 2 (continued). The sets $\{G, H, I\}$ and $\{G, H, E\}$ defend D and the set $\{G, I\}$ does not defend D .

³ By extension, a sequence reduced to two arguments ($A \mathcal{R}_{def} B$) will be also called a supported defeat for B . It is also called a direct defeat on B .

⁴ We keep Dung’s original definition, but with the relation *set-defeat* instead of *defeat*.

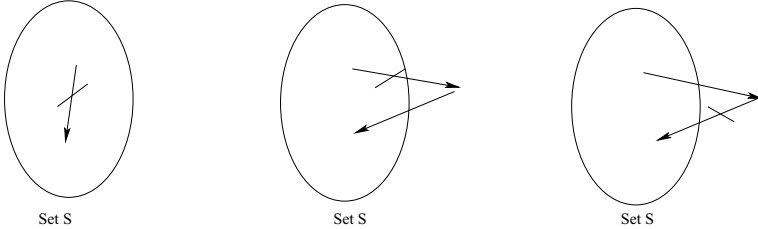
4 Acceptability in a Bipolar Framework

In Dung's framework, the *acceptability* of an argument depends on its membership of some sets, called acceptable sets or extensions. These extensions are characterised by particular properties. It is a collective acceptability. Following Dung's methodology, we establish characteristic properties that a set of arguments must satisfy in order to be an output of the argumentation process, in a bipolar framework. We recall that such a set of arguments must be in some sense coherent and must enable to win a dispute. Maximality for set-inclusion is also often required. In the following, we first investigate the notion of coherence. Then, we propose new semantics for acceptability in bipolar argumentation frameworks.

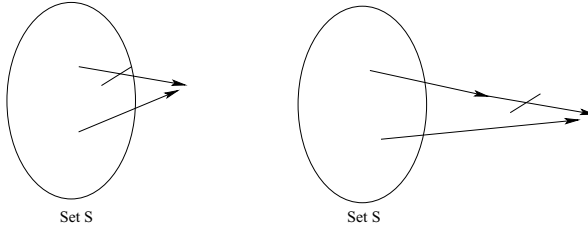
4.1 Managing the Conflicts

In the basic argumentation framework, whatever the considered semantics, selected acceptable sets of arguments are constrained to be coherent in the sense that they must be conflict-free. In a bipolar argumentation framework, the concept of coherence can be extended:

- reusing the notion of conflict-free set enforces a kind of *internal* coherence: we do not accept a set S of arguments which set-defeats one of its elements.



- taking into account the support relation leads to define a kind of *external* coherence: we do not accept a set S of arguments which set-defeats *and* set-supports the same argument.



Consider $\langle \mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup} \rangle$ a bipolar argumentation framework.

Definition 6 (Conflict-free set). Let $S \subseteq \mathcal{A}$. S is conflict-free iff $\nexists A, B \in S$ such that $\{A\}$ set-defeats⁵ B .

⁵ In the sense of Definition 4.

Example 2 (continued). *The set $\{H, B\}$ is not conflict-free (in the sense of Dung). The set $\{H, C\}$ is not conflict-free since C suffers an indirect defeat from H . The set $\{B, D\}$ is not conflict-free since D suffers a supported defeat from B . Contrastedly, $\{A, H\}$ and $\{B, F\}$ are conflict-free.*

External coherence is taken into account by the following definition⁶:

Definition 7 (Safe set). *Let $S \subseteq \mathcal{A}$. S is safe iff $\nexists B \in \mathcal{A}$ such that S set-defeats⁷ B and either S set-supports B , or $B \in S$.*

Example 2 (continued). *The set $\{A, H\}$ is not safe since A supports B and H defeats B . The set $\{B, F\}$ is not safe since D suffers a supported defeat from B and F supports D . Contrastedly, $\{G, I, H\}$ is safe.*

Note that the notion of safe set is powerful enough to encompass the notion of conflict-free. We have:

Property 1. *Let $S \subseteq \mathcal{A}$.*

If S is safe, then S is conflict-free.

If S is conflict-free and closed for $\mathcal{R}_{sup}$ then S is safe.

Example 2 (continued). *The set $\{G, H, I, E\}$ is conflict-free and closed for $\mathcal{R}_{sup}$. So it is safe.*

4.2 Extensions

From the previous notions of coherence, and extending the propositions of [1], we can propose different new semantics for the acceptability.

Definition 8 (Stable extension). *Let $\langle \mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup} \rangle$. Let $S \subseteq \mathcal{A}$. S is a stable extension of $\langle \mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup} \rangle$ iff S is conflict-free and $\forall A \notin S$, S set-defeats A .*

In this paper, we only consider acyclic bipolar frameworks, in the sense that the associated interaction graph is acyclic. In Dung's basic framework, it has been proved that, in the case of an acyclic defeat graph, there is always a unique stable extension, which is also the unique preferred extension, and the grounded extension. So, Definition 8 ensures the existence of a unique stable extension in an acyclic bipolar argumentation framework⁸.

However, the unique stable extension is not always safe.

⁶ This definition is inspired by [7] and by the definition of a controversial argument proposed in [1].

⁷ Always in the sense of Definition 4.

⁸ We instantiate the basic Dung's framework with the relation set-defeats and the obtained graph is still acyclic.

Example 3. Consider the argumentation system defined by $\mathcal{A} = \{A, B, H\}$, $H \mathcal{R}_{def} B$ and $A \mathcal{R}_{sup} B$. The set $\{A, H\}$ is the unique stable extension, and it is not safe.

So, an acyclic bipolar argumentation framework may have no safe stable extension. Indeed, the following properties enable to characterize stable extensions.

Property 2. Let S be a stable extension. If S is safe, then S is closed for $\mathcal{R}_{sup}$.

As a consequence of the two previous properties, we have:

Consequence 1. Let S be a stable extension of $\langle \mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup} \rangle$. Then S safe iff S is closed for $\mathcal{R}_{sup}$.

Now, we are interested in the preferred semantics, which is based on the concept of defence. As in the basic case, we first investigate the concept of admissibility, and then extensions will be defined as maximal (for set-inclusion) admissible sets of arguments.

Three different definitions for admissibility can be given, from the most general one to the most specific one. First, a direct translation of Dung's definition gives the definition of d-admissibility⁹.

Definition 9 (d-admissible set). Let $S \subseteq \mathcal{A}$. S is d-admissible iff S is conflict-free and defends all its elements.

Taking into account external coherence leads to s-admissibility¹⁰.

Definition 10 (s-admissible set). Let $S \subseteq \mathcal{A}$. S is s-admissible iff S is safe and defends all its elements.

Finally, external coherence can be strengthened by requiring that an admissible set is closed for $\mathcal{R}_{sup}$. So, we obtain the definition of c-admissibility¹¹.

Definition 11 (c-admissible set). Let $S \subseteq \mathcal{A}$. S is c-admissible iff S is conflict-free, closed for $\mathcal{R}_{sup}$ and defends all its elements.

From all the previous results, it follows that each c-admissible set is s-admissible, and each s-admissible set is d-admissible.

Definition 12 (Preferred extension). A set $S \subseteq \mathcal{A}$ is a d-preferred (resp. s-preferred, c-preferred) extension iff S is maximal (for set-inclusion) among the d-admissible (resp. s-admissible, c-admissible) subsets of $\mathcal{A}$.

Example 3 (continued). The set $\{A, H\}$ is the unique d-preferred extension. There are two s-preferred extensions $\{A\}$ and $\{H\}$. And there is only one c-preferred extension $\{H\}$.

⁹ “d” means “in the sense of Dung”.

¹⁰ “s” means “safe”.

¹¹ “c” means “closed for $\mathcal{R}_{sup}$ ”.

Example 2 (continued). *The set $\{G, H, E, F\}$ is s-admissible, but not c-admissible. The set $\{G, H, I, E, F, D, J\}$ is the unique c-preferred extension.*

One of the main issues with regard to extensions concerns their existence. As said above for stable extensions, the definition of d-admissibility ensures the existence of a unique d-preferred extension in an acyclic bipolar argumentation framework. It is also the unique stable extension.

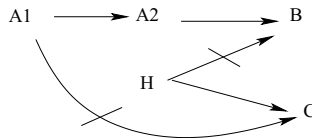
The existence of s-preferred (resp. c-preferred) extensions is guaranteed since the empty set is c-admissible. Moreover, the following results enable to characterize s-preferred and c-preferred extensions.

Property 3. *Let S be the unique stable extension of $\langle \mathcal{A}, \mathcal{R}_{def}, \mathcal{R}_{sup} \rangle$.*

1. *The s-preferred extensions and the c-preferred extensions are subsets of S .*
2. *Each s-preferred extension which is closed for $\mathcal{R}_{sup}$ is also a c-preferred extension.*
3. *If S is safe, then S is the unique c-preferred extension and also the unique s-preferred extension.*
4. *If $\mathcal{A}$ is finite, each c-preferred extension is included in a s-preferred extension.*
5. *If S is not safe, the s-preferred extensions are the subsets of S which are maximal (for set-inclusion) s-admissible.*
6. *If S is not safe, and $\mathcal{A}$ is finite, there is only one c-preferred extension.*

Example 3 (continued). *$\{H\}$ is the only s-preferred extension which is also closed for $\mathcal{R}_{sup}$. So, $\{H\}$ is the unique c-preferred extension.*

Example 4. *Consider the following argumentation system:*



$\{A_1, A_2, H\}$ is the only d-preferred extension. $\{A_1, A_2\}$ and $\{H\}$ are the only two s-preferred extensions. None of them is closed for $\mathcal{R}_{sup}$. $\emptyset$ is the unique c-preferred extension.

If we add an isolated argument A_3 (for which no interaction exists with the other arguments of the system), then we obtain: $\{A_1, A_2, A_3, H\}$ is the only d-preferred extension. $\{A_1, A_2, A_3\}$ and $\{H, A_3\}$ are the only two s-preferred extensions. None of them is closed, and $\{A_3\}$ is the unique c-preferred extension.

The above discussion enables to draw the following conclusions. In the particular case of an acyclic finite bipolar argumentation framework, two semantics present nice features: the stable semantics and the c-preferred semantics. If we are interested in internal coherence only, we will have to determine the unique stable extension, which is also the unique d-preferred extension and the grounded semantics. If we are interested in a more constrained concept of coherence, we will choose the c-preferred semantics, for which there also exists only one c-preferred extension.

5 Conclusion

In this paper, we have presented a study of the concept of acceptability in bipolar argumentation frameworks.

Following Dung's approach, we have abstracted from the structure of the arguments and proposed a general framework for argumentation, where two kinds of interaction can occur: arguments can be conflicting, or arguments can support other arguments. We have emphasized the idea that support and defeat can be independent relations, contrastedly with classical argumentation frameworks where support between arguments is captured by a notion of defence. Taking into account both relations has enabled us to propose new types of interaction between arguments: a sequence of supports, a direct defeat, an indirect defeat and a supported defeat.

Then, we have considered collective acceptability in our new abstract bipolar argumentation framework. Collective means that we have established characteristic properties that a set of arguments must satisfy in order to be an output of the argumentation process. It is natural to require a kind of coherence, and a kind of maximality as done in classical frameworks. Other properties, such as admissibility, take into account different levels of conflict, namely the existence of defeaters against defeaters.

First, we have investigated the notion of coherence and we have brought to light two kinds of coherence: internal vs external coherence. Internal coherence ensures that the considered set of arguments is conflict-free. External coherence is captured by the notion of safe set and ensures that the considered set cannot simultaneously defeat and support a same argument.

Then, combining a coherence requirement with the classical notion of admissibility, we have proposed new semantics for acceptability of sets of arguments. In particular, we have generalized the well-known stable and preferred semantics. Interesting properties have been obtained in the case of an acyclic bipolar argumentation framework. One of them states the existence of extensions. Moreover, one of the new semantics guarantees the existence of a new extension, which is a nice feature, regarding computational issues.

Future works will be mainly devoted to:

- a thorough study of the new semantics, including computational issues.
- the investigation of new characteristic properties such as for instance a generalization of "being closed for $\mathcal{R}_{sup}$ ". We are interested in sets S of arguments which are closed for $\mathcal{R}_{sup}$ and which contain any argument supporting an argument of S . Our idea is to define a meta argumentation system over such sets of arguments.

References

1. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence* **77** (1995) 321–357

2. Krause, P., Ambler, S., Elvang, M., Fox, J.: A logic of argumentation for reasoning under uncertainty. *Computational Intelligence* **11** (1) (1995) 113–131
3. Amgoud, L., Maudet, N., Parsons, S.: Arguments, Dialogue and Negotiation. In Horn, W., ed.: *Proc. of the 14th ECAI (European Conference of Artificial Intelligence)*, Berlin, Germany (2000) 338–342
4. Parsons, S., Sierra, C., Jennings, N.R.: Agents that reason and negotiate by arguing. *Journal of Logic and Computation* **8** (1998) 261–292
5. Karacapilidis, N., Papadias, D.: Computer supported argumentation and collaborative decision making: the HERMES system. *Information systems* **26** (2001) 259–277
6. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence* **34** (2002) 197–216
7. Verheij, B.: On the existence and multiplicity of extension in dialectical argumentation. In Benferhat, S., Giunchiglia, E., eds.: *Proceedings of the 9th International Workshop on Non-Monotonic Reasoning (NMR'2002)*. (2002) 416–425
8. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In Gabbay, D., Guenther, F., eds.: *Handbook of Philosophical Logic. Volume 4*. Kluwer Academic (2002) 218–319
9. Bench-Capon, T.: Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation* **13** (2003) 429–448
10. Chesñevar, C.I., Maguitman, A.G., Loui, R.P.: Logical models of argument. *ACM Computing surveys* **32** (2000) 337–383
11. Amgoud, L., Cayrol, C., Lagasque-Schiex, M.C.: On the bipolarity in argumentation frameworks. In Delgrande, J., Schaub, T., eds.: *Proc. of the 10th NMR workshop (Non Monotonic Reasoning), Uncertainty Framework subworkshop*, Whistler, BC, Canada (2004) 1–9
12. Boutilier, C.: Towards a logic for qualitative decision theory. In: *Proc. of the 4th KR, Bonn, Germany* (1994) 75–86
13. Tan, S.W., Pearl, J.: Specification and evaluation of preferences under uncertainty. In: *Proc. of the 4th KR, Bonn, Germany* (1994) 530–539
14. Lang, J., Van der Torre, L., Weydert, E.: Utilitarian desires. *Journal of Autonomous Agents and Multi-Agents Systems* **5**(3) (2002) 329–363
15. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Bipolar representation and fusion of preferences in the possibilistic logic framework. In: *Proceedings of the eighth International Conference on Principle of Knowledge Representation and Reasoning (KR'02)*. (2002) 158–169
16. Cayrol, C., Lagasque-Schiex, M.C.: Gradual handling of contradiction in argumentation frameworks. In Bouchon-Meunier, B., L.Foulloy, Yager, R., eds.: *Intelligent Systems for Information Processing: From representation to Applications*. Elsevier (2003) 179–190
17. Toulmin, S.: *The Uses of Arguments*. Cambridge University Press, Mass. (1958)

A Modal Logic for Reasoning with Contradictory Beliefs Which Takes into Account the Number and the Reliability of the Sources

Laurence Cholvy

ONERA Centre de Toulouse,
2 av Ed. Belin, 31055 Toulouse, France
cholvy@cert.fr

Abstract. This paper addresses the problem of merging beliefs provided by several sources which can be contradictory.

Among the different methods for managing contradictions, this paper focuses on the one which takes into account the number of sources that support a piece of information and their reliability degrees as well. More precisely, this paper presents a modal logic, extending *KD* logic, for reasoning with merged beliefs accordingly. It also shows that this logic unifies two different logics that have been defined in the past.

1 Introduction

This paper addresses the problem of merging beliefs provided by several information sources which share a common language for expressing information but which can be contradictory. That problem has been studied for many years in the Databases community and in the Artificial Intelligence one as well.

Most of the works that have addressed this problem have focused on characterizing the information source that results from merging different information sources. For doing so, some of them assume an order of priority among the sources, which may be expressed in a qualitative or a quantitative setting and which may represent the reliability degrees of the sources [1] [2], [21], [17], [22]. Some others assume that the information sources are equally reliable (i.e, or not-prioritized). In such a case two main kinds of merging methods are defined, respectively called arbitration and majority [16], [12], [18], [9], [19], [10], [11].

As said previously, these works aim at characterizing the information source that results from merging different information sources. This is achieved by giving a specific algorithm which defines the merging operator or by listing a set of postulates that the merging operators must satisfy. In any case, merging operators are meta-level constructs and, for a given set of information sources, they compute their merging.

Very few works have focused on characterizing the logic for reasoning with merged information i.e, have studied the logic for deducing, given the information that are believed by some sources, the information that are believed by the merged

information source. The only works we know that address this problem are our own works and Liao's ones. [4], [5], [8], [7], [13], [15], [14]. The advantage of such works is that characterizing the reasoning with merged beliefs, i.e, defining a logic for reasoning with merged beliefs helps to define what should be answers to queries when they are addressed to the information sources. More precisely, any automated theorem-prover in this logic (tableau method, meta-programming method etc) provides the specification of a query-evaluator i.e, defines a method for computing the answers to queries when they are addressed to the information sources.

In [4], [5] we have defined several versions of a logic for reasoning with merged beliefs assuming that the information sources are ordered according to their relative reliability. In [8], [7] we have defined a logic for reasoning with beliefs obtained by merging, according to a majority vote, beliefs provided by several sources that are equally reliable. The present work generalises these works and defines a logic for reasoning with beliefs obtained by merging beliefs provided by several sources by taking into account the number of sources that support a piece of information and their reliability degrees as well.

This paper is organised as follows. Section 2 presents a propositional logic for reasoning with merged beliefs which takes into account the number and the reliability of the sources. It is proved that, in the case when the information are atomic, the axiomatic is sound and complete versus the semantics, for some interesting formulas. Section 3 illustrates this logic on an example. Section 4 shows that this work generalises our previous works. Finally, section 5 is devoted to a discussion.

2 A Logic for Reasoning with Merged Beliefs Which Takes into Account the Number and the Reliability Degrees of the Sources

In this section, we define a logic, denoted F , for reasoning with merged beliefs which takes into account the number and the reliability degrees of the sources. F is an extension of KD modal logic [3].

We adopt a numerical point of view by modelling the reliability degrees of the information sources by integers.

2.1 Preliminaries

The semantics of F logic is based on multi-sets of worlds. So, we recall here some definitions about multi-sets.

Definition 1. A multi-set is a set where redundant occurrences are accepted. Let $MS_1 = [S_1, \dots, S_n]$ and $MS_2 = [S_{n+1}, \dots, S_m]$ be two multi-sets (no matter what the S_i 's are). The union of two multi-sets is defined by: $MS_1 \sqcup MS_2 = [S_1, \dots, S_m]$. The membership relation is defined by: $S \in^i MS$ iff there are exactly i occurrences of S in the multi-set MS . Notice that, in the limit case, $S \in^0 MS$ iff there is no occurrence of S in MS , i.e. $S \notin MS$.

Let L be a propositional language.

The different types of information sources we consider are defined as follows:

Definition 2. The information sources to be merged, called *information sources*, are finite and consistent sets of literals of L . If db is a primitive information source and if α is an integer, then $\alpha.db$ is an information source. It is called a *weighted information source*. If db_1 and db_2 are two information sources which are not primitive then $(db_1 * db_2)$ is an information source. It is called a *merged information source*.

For instance, if we face three primitive information sources db_1 , db_2 and db_3 then $3.db_1$ is a weighted information source, $(2.db_1 * 1.db_2)$, $(1.db_1 * 2.db_2) * 2.db_3$ and $(1.db_1 * (2.db_2 * 3.db_3))$ are merged information sources.

2.2 The Language

The language of logic F , denoted L_F , is obtained from L by adding modal operators of the form B_{db}^i and B_{db} , where i is an integer and db denotes a (primitive, weighted or merged) information source. We expect that the formula $B_{db}^i l$ means that there are exactly i occurrences of the literal l in the information source db . And we expect that the formula $B_{db} F$ means that the information source db believes F .

Informally speaking, we introduce the modalities B_{db}^i for being able to count the occurrences of a literal in an information source. The idea is that, when merging two information sources, the number of occurrences of a literal is the sum of the numbers of its occurrences in the two information sources respectively. Then we want that a literal is believed by an information source if the number of occurrences of that literal is strictly greater than the number of occurrences of its negation. The formal definition of L_F is the following:

Definition 3. If ϕ is a formula of L and if B_{db}^i and B_{db} are modal operators, then $B_{db}^i \phi$ and $B_{db} \phi$ are formulas of L_F . If ϕ_1 and ϕ_2 are formulas of L_F then, $\neg \phi_1$, $\phi_1 \wedge \phi_2$ are formulas of L_F . $\phi_1 \vee \phi_2$ and $\phi_1 \rightarrow \phi_2$ are defined from the previous ones as usually.

One can notice that modal operators only govern formulas without modal operators. For instance, assume that db_1 and db_2 are the two information sources to be merged, then the modal operators of L_F are: $B_{db_1}^i$, B_{db_1} , $B_{db_2}^i$, B_{db_2} , $B_{\alpha_1.db_1}^i$, $B_{\alpha_1.db_1}$, $B_{\alpha_2.db_2}^i$, $B_{\alpha_2.db_2}$, $B_{\alpha_1.db_1 * \alpha_2.db_2}^i$, $B_{\alpha_1.db_1 * \alpha_2.db_2}$, $B_{\alpha_2.db_2 * \alpha_1.db_1}^i$ and $B_{\alpha_2.db_2 * \alpha_1.db_1}$ (for any α_1, α_2, i). We expect that, for instance: $B_{db_1}^1 a$ means that db_1 contains one occurrence of a ; $B_{3.db_1}^3 a$ means that the weighted source $3.db_1$ contains three occurrences of a ; $B_{db_2}^0 \neg a$ means that db_2 contains no occurrence of $\neg a$; $B_{2.db_2}^0 \neg a$ means that the weighted source $2.db_2$ contains no occurrence of $\neg a$; $B_{3.db_1 * 2.db_2}^3 a$ means that the information source obtained by merging $3.db_1$ and $2.db_2$ contains three occurrences of a ; finally, we expect that $B_{3.db_1 * 2.db_2} a$ means that the information source obtained by merging $3.db_1$ and $2.db_2$, believes a .

2.3 Semantics

The semantics of F is a Kripke-type one. Models are defined by:

Definition 4. Models of F . A model of F is a tuple $\langle W, val, R, B \rangle$ such that: W is a set of worlds; val is a valuation function¹ which associates any proposition of L with a set of worlds of W ; R is a set of functions denoted f_{db} , where db is an information source (primitive, weighted or merged). Each function f_{db} associates any world of W with a multi-set of sets of worlds of W ; B is a set of functions denoted g_{db} , where db is an information source (primitive, weighted or merged). Each function g_{db} associates any world of W with a set of worlds of W .

This tuple is constrained by three conditions given below, but before, we need to give the following definition:

Definition 5. Let w and w' be two W worlds. The distance $d(w, w')$ between w and w' is defined by the number of propositional letters p such that $w \in val(p)$ and $w' \notin val(p)$ (this distance is usually called Hamming distance). Let $MS = [S_1 \dots S_n]$ be a multi-set of sets of worlds. Then the distance $dsum(w, MS)$ between a world w and MS is defined by: $dsum(w, MS) = \sum_{i=1}^n \min_{w' \in S_i} d(w, w')$. Finally, any multi-set of sets of worlds MS is associated with a pre-order $\leq_{MS}$ of W defined by: $w \leq_{MS} w'$ iff $dsum(w, MS) \leq dsum(w', MS)$.

Definition 4 (continued). Models of F .

The previous tuple $\langle W, val, R, B \rangle$ has to satisfy the following three conditions:

(C1) If db is a primitive information source and if α is an integer, then:

$\forall w \in W \quad f_{\alpha.db}(w) = f_{db}(w) \sqcup \dots (\alpha \text{ times}) \dots \sqcup f_{db}(w)$

(C2) If db_1 and db_2 are two information sources which are not primitive, then: $\forall w \in W \quad f_{db_1 * db_2}(w) = f_{db_1}(w) \sqcup f_{db_2}(w)$

(C3) If db is an information source, then $\forall w \in W \quad g_{db}(w) = \min_{\leq_{f_{db}(w)}} W$

The constraint (C1) reflects the fact that the occurrences of a literal in the weighted information source $\alpha.db$ are obtained by duplicating α times, the occurrences of this literal in db . So, it will be the case that the number of occurrences of a literal in $\alpha.db$ is α times the number of its occurrences in db . The constraint (C2) reflects the fact that the occurrences of a literal in the merged information source $db_1 * db_2$ are the union of its occurrences in db_1 and of its occurrences in db_2 . So, it will be the case that the number of occurrences of a literal in $db_1 * db_2$ is the sum of the number of its occurrences in db_1 and the number of its occurrences in db_2 . The constraint (C3) expresses that the models of the information source db are the minimal W worlds according to the pre-order $\leq_{f_{db}(w)}$.

Definition 6. Satisfaction of formulas. Let $M = \langle W, val, R, B \rangle$ be a model of F and let $w \in W$. Let p be a propositional letter of L . Let F , ϕ_1 and ϕ_2 be formulas of L_F and let db be any information source (primitive, weighted or merged). Then,

¹ It satisfies: $val(P) \neq \emptyset$ iff P is a satisfiable propositional formula, $val(\neg P) = W \setminus val(P)$, $val(P \wedge Q) = val(P) \cap val(Q)$.

$M, w \models_F p$	iff	$w \in \text{val}(p)$
$M, w \models_F \neg\phi_1$	iff	$M, w \not\models_F \phi_1$
$M, w \models_F \phi_1 \wedge \phi_2$	iff	$M, w \models_F \phi_1$ and $M, w \models_F \phi_2$
$M, w \models_F B_{db}^i \phi$	iff	$\text{val}(\phi) \in^i f_{db}(w)$
$M, w \models_F B_{db} \phi$	iff	$g_{db}(w) \subseteq \text{val}(\phi)$

Definition 7. Valid formulas in F . Let ϕ be a formula of L_F . ϕ is a valid formula in F (denoted $\models_F \phi$) iff for any model of F $M = \langle W, \text{val}, R, B \rangle$, $\forall w \in W$, $M, w \models_F \phi$.

2.4 Proof Theory

In the following axioms ϕ, ϕ_1 and ϕ_2 denote formulas of L , $l, l_1, \dots, l_n$ denote literals of L and i, j, k denote integers.

The axiom schemata of F are:

(A₀) Axiom schemata of propositional logic

(A₁) $B_{db} \neg\phi \rightarrow \neg B_{db} \phi$

(A₂) $B_{db} \phi_1 \wedge B_{db} (\phi_1 \rightarrow \phi_2) \rightarrow B_{db} \phi_2$

(A₃) $B_{db}^i l \leftrightarrow B_{db}^i \neg\neg l$

(A₄) $B_{db}^i l \rightarrow \neg B_{db}^j l$ if $i \neq j$

(A₅) $B_{db}^i l \wedge B_{db}^j \neg l \rightarrow B_{db} l$ if $i > j$

(A₆) $B_{db}^i l \wedge B_{db}^i \neg l \rightarrow \neg B_{db} l$

(A₇) $B_{db} (l_1 \vee \dots \vee l_n) \rightarrow B_{db} l_1 \vee \dots \vee B_{db} l_n$ with $\forall i \in \{1 \dots n\} \forall j \in \{1 \dots n\} l_i \neq \neg l_j$

(A₈) $B_{db}^i l \leftrightarrow B_{\alpha.db}^k l$ if $k = \alpha.i$

(A₉) $B_{db_1}^i l \wedge B_{db_2}^j l \rightarrow B_{db_1 * db_2}^k l$ if $k = i + j$

The inference rules are :

(MP) If $\vdash_F \phi_1$ and $\vdash_F (\phi_1 \rightarrow \phi_2)$ then $\vdash_F \phi_2$

(Nec) $\vdash_F \phi$ then $\vdash_F B_{db} \phi$ for any modality B_{db} .

$\vdash_F \phi$ denotes as usual, theorems of F , i.e formulas that are instances of axiom schemata or that can be deduced by using axiom schemata and inference rules.

Let us comment these axioms.

(A₀), (A₁), (A₂) express that modalities B_{db} are belief modalities.

(A₃) keeps the equivalence between l and $\neg\neg l$ in B_{db}^i modalities.

(A₄) says that the number of occurrences of a literal in an information source is unique.

(A₅), (A₆) express the majority aspect of the underlying merging operator. First, a literal l is believed by a source db if the number of its occurrences is strictly greater then the number of the occurrences of its negation. If the number of the occurrences of l is equal to the number of occurrences of its negation, then that literal and its negation are not believed by the information source.

(A₇) expresses that if an information source believes a disjunction of literals, which is not a tautology, then it believes at least one of its literals.

Excluding disjunctions which are tautologies is necessary since due to inference rule (Nec), any tautology is believed. So, for instance $a \vee \neg a$ is believed (due to Nec)

even though neither a nor $\neg a$ is believed. (A_7) prevents the case when a database believes, for instance $a \vee b$ and does not believe a nor b . This comes to restrict the information sources we consider to sets of literals.

(A₈) expresses the facts that the number of occurrences of a literal in the weighted information source $\alpha.db$ is α times the number of its occurrences in db .

(A₉) expresses the facts that the number of occurrences of a literal in the merged information source $db_1 * db_2$ is the sum of the its occurrences in db_1 and the number of its occurrences in db_2 .

2.5 Soundness and Completeness for Some Interesting Formulas

Definition 8. Let $db_1 \dots db_n$ be the primitive information sources (i.e, finite and consistent sets of literals of L) to be merged. We define the formula ψ by:

$$\psi = \bigwedge_{i=1}^n \left(\bigwedge_{l \in db_i} B_{db_i}^1 l \wedge \bigwedge_{l \notin db_i} B_{db_i}^0 l \right)$$

ψ lists the information we have about the content of the given primitive sources to be merged. More precisely, it expresses that each literal it contains has one and only one occurrence in it, and that each literal it does not contain has no occurrence in it.

The following result proves that the model theory and the proof theory previously presented are equivalent for formulas of the form $\psi \rightarrow B_{db}\phi$, where db is any information source.

Proposition 1. Let ψ be the formula defined by definition 8. Let ϕ be a formula of L and db an information source (primitive, weighted or merged). Then we have:

$$\begin{aligned} \models_F \psi \rightarrow B_{db}\phi &\iff \vdash_F \psi \rightarrow B_{db}\phi \text{ and} \\ \models_F \psi \rightarrow \neg B_{db}\phi &\iff \vdash_F \psi \rightarrow \neg B_{db}\phi \end{aligned}$$

Proposition 2. Let ψ be the formula defined by definition 8. Let ϕ be a formula of L and db an information source (primitive, weighted or merged). Then:

$$\not\models_F \psi \rightarrow B_{db}\phi \iff \vdash_F \psi \rightarrow \neg B_{db}\phi$$

2.6 Properties

Commutativity and Associativity of Merging. The following two propositions show that the merging operator underlying F -logic is commutative and associative.

Proposition 3. $\vdash_F \psi \rightarrow (B_{\alpha_1.db_1 * \alpha_2.db_2}\phi \leftrightarrow B_{\alpha_2.db_2 * \alpha_1.db_1}\phi)$

Proposition 4. $\vdash_F \psi \rightarrow (B_{(\alpha_1.db_1 * \alpha_2.db_2) * \alpha_3.db_3}\phi \leftrightarrow B_{\alpha_1.db_1 * (\alpha_2.db_2 * \alpha_3.db_3)}\phi)$

Due to these results, parenthesis will be omitted when designating a merged information source. So for instance, we will denote $2.db_1 * 1.db_2 * 2.db_3$ instead of $((2.db_1 * 1.db_2) * 2.db_3)$

Relation with a Weighted Majority Merging Operator. In [12], Konieczny and Pino-Pérez introduced a majority merging operator as follows:²

Let $db_1 \dots db_n$ be n information sources to be merged. A majority merging operator, denoted Δ_Σ , is defined such that the models of the information source which is obtained from merging $db_1 \dots db_n$ with this operator, are semantically characterized by:

$$Mod(\Delta_\Sigma([db_1, \dots, db_n])) = \min_{\substack{\Sigma \\ [db_1 \dots db_n]}} (W)$$

where W denotes the set of all the interpretations of the language L (the propositional language used to describe the contents of the informations sources). $\leq_{[db_1 \dots db_n]}^\Sigma$ is a total pre-order on W defined by:

$$w \leq_{[db_1 \dots db_n]}^\Sigma w' \text{ iff } d_\Sigma(w, [db_1 \dots db_n]) \leq d_\Sigma(w', [db_1 \dots db_n])$$

with

$$d_\Sigma(w, [db_1 \dots db_n]) = \sum_{i=1}^n \min_{w' \in Mod(db_i)} d(w, w')$$

where $Mod(db_i)$ is the set models of db_i and $d(w, w')$ is the Hamming distance.

In other words, when merging $db_1 \dots db_n$ with the operator Δ_Σ , the result is semantically characterized by the interpretations which are minimal according to the pre-order $\leq_{[db_1, \dots, db_n]}^\Sigma$.

Now, let us consider that the knowledge-bases $db_1 \dots db_n$ are associated with weights $\alpha_1 \dots \alpha_n$. We can extend the previous definitions and define a new merging operator, $\Delta_\Sigma^{\alpha_1 \dots \alpha_n}$, such that the models of the information source which is obtained from merging $db_1 \dots db_n$ with this operator, is semantically characterized by:

$$Mod(\Delta_\Sigma^{\alpha_1 \dots \alpha_n}([db_1, \dots, db_n])) = \min_{\substack{\Sigma, \alpha_1 \dots \alpha_n \\ [db_1 \dots db_n]}} (W)$$

where $\leq_{[db_1 \dots db_n]}^{\Sigma, \alpha_1 \dots \alpha_n}$ is a total pre-order on W defined by:

$$w \leq_{[db_1 \dots db_n]}^{\Sigma, \alpha_1 \dots \alpha_n} w' \text{ iff } d_{\Sigma, \alpha_1 \dots \alpha_n}(w, [db_1 \dots db_n]) \leq d_{\Sigma, \alpha_1 \dots \alpha_n}(w', [db_1 \dots db_n])$$

with

$$d_{\Sigma, \alpha_1 \dots \alpha_n}(w, [db_1 \dots db_n]) = \sum_{i=1}^n \alpha_i \cdot \min_{w' \in Mod(db_i)} d(w, w')$$

In other words, when merging $db_1 \dots db_n$ with the operator $\Delta_\Sigma^{\alpha_1 \dots \alpha_n}$, the result is semantically characterized by the interpretations which are minimal according to the pre-order $\leq_{[db_1, \dots, db_n]}^{\Sigma, \alpha_1 \dots \alpha_n}$.

² One will notice that we slightly change the presentations of these definitions to remain coherent with what has already been presented.

$\Delta_{\Sigma}^{\alpha_1 \dots \alpha_n}$ is a weighted majority merging operator. Notice that listing the postulates this merging operator satisfies has not yet been done.

We can prove the following result:

Proposition 5. Let $db_1 \dots db_n$ be n finite and consistent sets of literals to be merged. Let $\alpha_1 \dots \alpha_n$ be integers and ϕ be a formula of L . With the notations previously introduced, we have:

$$\vdash \psi \rightarrow B_{\alpha_1.db_1 * \dots * \alpha_n.db_n} \phi \iff \Delta_{\Sigma}^{\alpha_1 \dots \alpha_n}([db_1 \dots db_n]) \models \phi$$

3 Example

Let us give here some examples of proofs in F logic. We consider three information sources: $db_1 = \{a, b\}$, $db_2 = \{a, \neg c\}$, $db_3 = \{\neg a, c\}$.

By definition 8, ψ is: $B_{db_1}^1 a \wedge B_{db_1}^1 b \wedge B_{db_1}^0 c \wedge B_{db_1}^0 \neg c \wedge B_{db_1}^0 \neg a \wedge B_{db_1}^0 b \wedge B_{db_2}^1 a \wedge B_{db_2}^1 \neg c \wedge B_{db_2}^0 b \wedge B_{db_2}^0 \neg b \wedge B_{db_2}^0 \neg a \wedge B_{db_2}^0 c \wedge B_{db_3}^1 \neg a \wedge B_{db_3}^1 c \wedge B_{db_3}^0 b \wedge B_{db_3}^0 \neg b \wedge B_{db_3}^0 a \wedge B_{db_3}^0 \neg c$

Here are some theorems of F we can derive:

- (α) $\vdash \psi \rightarrow B_{3.db_1}^3 a$ (by (A_8))
- (β) $\vdash \psi \rightarrow B_{2.db_2}^2 a$ (by (A_8))
- (γ) $\vdash \psi \rightarrow B_{1.db_3}^0 a$ (by (A_8))
- (δ) $\vdash \psi \rightarrow B_{3.db_1 * 2.db_2}^5 a$ (by (α), (β) and (A_9))
- (ζ) $\vdash \psi \rightarrow B_{3.db_1 * 2.db_2 * 1.db_3}^5 a$ (by (δ), (γ) and (A_9))
- (α') $\vdash \psi \rightarrow B_{3.db_1}^0 \neg a$ (by (A_8))
- (β') $\vdash \psi \rightarrow B_{2.db_2}^0 \neg a$ (by (A_8))
- (γ') $\vdash \psi \rightarrow B_{1.db_3}^1 \neg a$ (by (A_8))
- (δ') $\vdash \psi \rightarrow B_{3.db_1 * 2.db_2}^0 \neg a$ (by (α'), (β') and (A_9))
- (ζ') $\vdash \psi \rightarrow B_{3.db_1 * 2.db_2 * 1.db_3}^1 \neg a$ (by (δ'), (γ') and (A_9))

Thus, finally, from (ζ), (ζ') and (A_5), we can prove:

$$(\iota) \vdash \psi \rightarrow B_{3.db_1 * 3.db_2 * 1.db_3} a$$

This theorem means that a is believed by the information source obtained by merging db_1 , db_2 and db_3 , when respective weights are 3, 2 and 1.

In the same way, we prove:

$$(\eta) \vdash \psi \rightarrow B_{3.db_1 * 2.db_2 * 1.db_3} b \quad \text{and}$$

$$(\nu) \vdash \psi \rightarrow B_{3.db_1 * 2.db_2 * 1.db_3} \neg c$$

Thus, from (ι), (η), (ν), (A_0) and (A_2) we prove:

$$\vdash \psi \rightarrow B_{3.db_1 * 2.db_2 * 1.db_3} (a \wedge b \wedge \neg c)$$

This theorem means that $(a \wedge b \wedge \neg c)$ is believed by the information source obtained by merging db_1 , db_2 and db_3 , when assuming that their respective degrees of reliability are 3, 2 and 1.

Notice that, by a different proof, we can also prove:

$$\vdash \psi \rightarrow B_{2.db_2 * 1.db_3 * 3.db_1} (a \wedge b \wedge \neg c)$$

Similarly we can prove :

$$\vdash \psi \rightarrow B_{1.db_1 * 2.db_2 * 3.db_3} (b \wedge c) \quad \text{and}$$

$$\vdash \psi \rightarrow \neg B_{(1.db_1 * 2.db_2) * 3.db_3} a \wedge \neg B_{(1.db_1 * 2.db_2) * 3.db_3} \neg a$$

This means that $(b \wedge c)$ is believed by the information source obtained by merging db_1 , db_2 and db_3 , when assuming that the respective weights are 1, 2 and 3. But neither a nor $\neg a$ is believed.

An application of this formal example is, for instance, multi-sensor data fusion. Consider three sensors which observe a flying object. According to the first sensor, $(\text{red} \vee \text{blue}) \wedge \text{fast}$ (600 / 100). According to the second sensor, $(\text{red} \vee \text{blue}) \wedge \neg \text{fast}$ (100 / 600). According to the third sensor, $\neg(\text{red} \vee \text{blue}) \wedge \text{fast}$ (100 / 600). Finally, according to the last sensor $\neg(\text{red} \vee \text{blue}) \wedge \neg \text{fast}$ (600 / 100).

The previous proofs allow us to conclude that, if the respective degrees of reliability of the sensors are 3, 2 and 1, then we can conclude that $\frac{600}{T} \notin \mathbb{N}$. If the respective degrees of reliability of the sensors are 1, 2 and 3, then we can conclude that $\frac{600}{T} \notin \mathbb{N}$, but we cannot decide whether it is a plane or not.

4 Related Works

In the past, [4], [5], we have defined a logic for reasoning with merged information by taking into account the reliability (total) order between the information sources. The modalities of this logic were of the form: $B_{db_1 > db_2 > \dots > db_n}$ where $db_1 \dots db_n$ are primitive sources. $B_{db_1 > db_2 > \dots > db_n} l$ intended to mean that the information source, obtained by merging the primitive sources $db_1, db_2, \dots, db_n$, assuming that db_1 is more reliable than $db_2, \dots, db_{n-1}$ is more reliable than db_n , believes l .

Detailing the semantics and the axiomatics of this logic is out of scope of this paper. Let us just illustrate it on an example. We consider as before: $db_1 = \{a, b\}$, $db_2 = \{a, \neg c\}$, $db_3 = \{\neg a, c\}$. In this logic, we can prove, for instance: $\vdash \psi \rightarrow B_{db_1 > db_3 > db_2}(a \wedge b \wedge c)$. This means that, if we consider that db_1 is more reliable than db_2 , itself more reliable than db_3 , then, the information source obtained by merging them believes that a and b and c . We can also prove, for instance: $\vdash \psi \rightarrow B_{db_2 > db_1 > db_3}(a \wedge b \wedge \neg c)$. This means that, if we consider that db_2 is more reliable than db_1 , itself more reliable than db_3 , then, the information source obtained by merging them believes that a and b and $\neg c$.

We can prove that:

$$\vdash \psi \rightarrow B_{db_1 > db_2 > \dots > db_n} \phi \iff \vdash_F \psi \rightarrow B_{2^{n-1}.db_1 * 2^{n-2}.db_2 * \dots * 2^0.db_n} \phi$$

This ensures that reasoning with total orders between informations sources, as it is done in this logic, can similarly be made in F -logic. More precisely, it shows that in F -logic, considering that the reliability degrees of the primitive sources $db_1, db_2, \dots, db_n$ are $2^{n-1}, 2^{n-2}, \dots, 2^0$, is enough to simulate a total order of reliability between the sources.

Similarly, we can formally prove that F logic allows one to reason with merged data obtained by Konieczny and Pino-Pérez’s majority operator described in section 2.6. Indeed, if $db_1 \dots db_n$ are n finite and consistent sets of literals to be

merged and if α is an arbitrary integer and ϕ a formula of L . With the notations previously introduced, we have:

$$\vdash \psi \rightarrow B_{\alpha.db_1*\alpha.db_2*...\alpha.db_n}\phi \iff \Delta_{\Sigma}([db_1...db_n]) \models \phi$$

In other words, the information source whose beliefs are characterized by theorems $\vdash \psi \rightarrow B_{\alpha.db_1*\alpha.db_2*...\alpha.db_n}\phi$, is equivalent to $\Delta_{\Sigma}([db_1...db_n])$. This proves that F logic can be used for modelling a majority merging operator: assuming in F that the weights of the different information sources are identical is enough.

5 Discussion

First of all, let us say that the work presented here has been motivated by an application in Intelligence (See [6]). Indeed, taking into account the number of the sources that emit information and their respective reliability is a requirement which is explicitly mentionned in some NATO standard about information evaluation. More precisely, that standard, [20], explicitly specifies that

- information sources (humain or not) should be given a reliability degree so that

$$r_1, r_2, \dots, r_n$$
;
- pieces of information should be associated with a credibility degree so that,

$$c_1, c_2, \dots, c_n$$
; (...)

In this present work, we gave a numerical definition of reliability degrees and suggest a numerical use of them. The method induced by this choice is very simple but is, obviously, sensitive to compensation. We could then extend this work to the case of an ordinal method of fusion. Here is one way of doing it.

First, the language must be changed. We suggest to replace modalities B_{db}^i (where i is an integer) by modalities B_{db}^i , where i is now a vector of integers. For instance, the intuitive meaning of $B_{db}^{[3221]}a$ is that, in the information source db obtained by merging several sources, a has been emitted by four sources, respectively evaluated 3, 2, 2 and 1.

Axiom **(A₅)** should be replaced by: **(A₅)** $B_{db}^i l \wedge B_{db}^j \neg l \rightarrow B_{db} l$ if $j << i$, where $<<$ is a well-chosen pre-order between vectors of integers which extends the pre-order: $[1] << [2] << [3] << \dots$

There are many possibilities to choose $<<$. For instance, we could define $<<$ as follows. Let V_1 and V_2 two vectors of integers whose lengths are respectively n_1 and n_2 . Assume that $n_2 < n_1$. Let V'_1 be the vector obtained by restricting V_1 to its n_2^{th} first integers. Then, $V_2 << V_1$ iff $V_2 <_{leximin} V'_1$ (i.e $\exists k V_2(k) < V_1(k)$ and $\forall i < k V_2(k) = V_1(k)$)

In this case, an instance of **(A₅)** is : $B_{db}^{[32]}a \wedge B_{db}^{[311]}\neg a \rightarrow B_{db}a$, which represents the fact that, if in db the proposition a has been emitted by a source

evaluated 3 and a source evaluated 2 and if $\neg a$ has been emitted by a source evaluated 3 and two sources evaluated 1, then we consider that db believes a .

Axiom $(\mathbf{A}_8)$ should be replaced by: $(\mathbf{A}_8) B_{db}^i l \leftrightarrow B_{\alpha.db}^k l$ if k is the vector $[\alpha]$ if $i = 1$; and $\square$ if $i = 0$.

For instance, two instances of $(\mathbf{A}_8)$ are $B_{db}^1 a \leftrightarrow B_{3.db}^{[3]} a$ and $B_{db}^0 a \leftrightarrow B_{3.db}^{\square} a$

Axiom $(\mathbf{A}_9)$ should be replaced by: $(\mathbf{A}_9) B_{db_1}^i l \wedge B_{db_2}^j l \rightarrow B_{db_1 * db_2}^k l$ where k is the vector obtained by concatenating the two vectors i and j .

For instance, an instance of $(\mathbf{A}_9)$ is $B_{db_1}^{[32]} a \wedge B_{db_2}^{[31]} a \rightarrow B_{db_1 * db_2}^{[3231]} a$.

In order to illustrate these modified axioms, let us take again the example of section 3. We have now:

- $(\alpha) \vdash \psi \rightarrow B_{3.db_1}^{[3]} a$ (by (A_8))
- $(\beta) \vdash \psi \rightarrow B_{2.db_2}^{[2]} a$ (by (A_8))
- $(\gamma) \vdash \psi \rightarrow B_{1.db_3}^{\square} a$ (by (A_8))
- $(\delta) \vdash \psi \rightarrow B_{3.db_1 * 2.db_2}^{[32]} a$ (by (α) , (β) and (A_9))
- $(\zeta) \vdash \psi \rightarrow B_{3.db_1 * 2.db_2 * 1.db_3}^{[32]} a$ (by (δ) , (γ) and (A_9))
- $(\alpha') \vdash \psi \rightarrow B_{3.db_1}^{\square} \neg a$ (by (A_8))
- $(\beta') \vdash \psi \rightarrow B_{2.db_2}^{\square} \neg a$ (by (A_8))
- $(\gamma') \vdash \psi \rightarrow B_{1.db_3}^{[1]} \neg a$ (by (A_8))
- $(\delta') \vdash \psi \rightarrow B_{3.db_1 * 2.db_2}^{\square} \neg a$ (by (α') , (β') and (A_9))
- $(\zeta') \vdash \psi \rightarrow B_{3.db_1 * 2.db_2 * 1.db_3}^{[1]} \neg a$ (by (δ') , (γ') and (A_9))

Thus, finally, from (ζ) , (ζ') and (A_5) , if the chosen pre-order is the leximin, we can prove:

- $(\iota) \vdash \psi \rightarrow B_{3.db_1 * 3.db_2 * 1.db_3} a$

We plan to carry on this work by completing the extension of the logic according to the previous ideas.

Acknowledgments

This work has been supported by ONERA (grant 1042801F)

References

1. C. Baral, S. Kraus, J. Minker, and V.S. Subrahmanian. Combining multiple knowledge bases. *IEEE Trans. on Knowledge and Data Engineering*, 3(2):208–220, 1991.
2. C. Baral, S. Kraus, J. Minker, and V.S. Subrahmanian. Combining knowledge bases consisting of first order theories. *Computational Intelligence*, 8(1):45–71, 1992.
3. B. F. Chellas. *Modal logic, an introduction*. Cambridge University Press, 1980.
4. L. Cholvy. Proving theorems in a multi-sources environment. In *Proceedings of IJCAI*, pages 66–71, 1993.
5. L. Cholvy. Reasoning about merged information. In *Handbook of defeasible reasoning and uncertainty management*, volume 1. Kluwer Academic Publishers, 1998.

6. L. Cholvy. Information evaluation in fusion: a case study. In *Proceedings of Information Processing and Management of Uncertainty Conference (IPMU)*, Perugia, July 2004.
7. L. Cholvy and Ch. Garion. Answering queries addressed to several databases according to a majority merging approach. *Journal of Intelligent Information Systems*, 22(2), 2004.
8. L. Cholvy and Ch. Garion. A logic to reason an contradictory beliefs with a majority approach. In *Proceedings of the IJCAI'01 Workshop: Inconsistencies in Data and Knowledge*, Seattle, august 2001.
9. S. Konieczny and R. Pino-Pérez. Merging with integrity constraints. In *Proc. of ESCQARU'99*, 1999.
10. S. Konieczny and R. Pino-Pérez. Merging information under constraints: a logical framework. *Journal of Logic and Computation*, 12(5):773–808, 2002.
11. S. Konieczny and R. Pino-Pérez. Propositional belief base merging or how to merge beliefs/goals from several sources and some links with social choice theory. *European Journal of Operational Research*, 160:785–802, 2005.
12. S. Konieczny and R. Pino-Pérez. On the logic of merging. In *Proc. of KR'98*, Trento, 1998.
13. C. Liau. A conservative approach to distributed belief fusion. In *Proceedings of 3rd International Conference on Information Fusion (FUSION)*, 2000.
14. C.J. Liau. Belief fusion and revision: an overview based on epistemic logic semantics. *Journal of Applied Non-Classical Logics*, 14(3):247–274, 2004.
15. C.J. Liau. A modal logic framework for multi-agent belief fusion. *ACM Transactions on Computational Logic*, 6(1):124–174, 2005.
16. Paolo Liberatore and Marco Schaerf. Arbitration: A commutative operator for belief revision. In *Proceedings of the Second World Conference on the Fundamentals of Artificial Intelligence (WOFAI'95)*, pages 217–228. Angkor Press, 1995.
17. J.. Lin. Integration of weighted knowledge bases. *Artificial Intelligence*, 83:363–378, 1996.
18. J. Lin and A.O. Mendelzon. Merging databases under constraints. *International Journal of Cooperative Information Systems*, 7(1):55–76, 1998.
19. J. Lin and A.O. Mendelzon. Knowledge base merging by majority. In R. Pareschi and B. Fronhoefer, editors, *Dynamic Worlds: From the Frame Problem to Knowledge Management*. Kluwer Academic, 1999.
20. North Atlantic Treaty Organization (NATO). Annex to stanag 2022 (edition 8). Technical report, Information Handling Services, DODSID, Issue DW9705.
21. S. Pradhan, J. Minker, and V.S. Subrahmanian. Combining databases with prioritized information. *Journal of Intelligent Information Systems*, 4:231–260, 1995.
22. S.Benferhat, D. Dubois, J. Lang, H. Prade, A. Saffiotti, and P. Smets. A general approach for inconsistency handling and merging information in prioritized knowledge bases. In *Proc. of KR'98*, Trento, 1998.

A Possibilistic Inconsistency Handling in Answer Set Programming

Pascal Nicolas, Laurent Garcia, and Igor Stéphan

LERIA, University of Angers, France

{pascal.nicolas, laurent.garcia, igor.stephan}@univ-angers.fr

Abstract. Both in classical logic and in Answer Set Programming, inconsistency is characterized by non existence of a model. Whereas every formula is a theorem for inconsistent set of formulas, an inconsistent program has no answer. Even if these two results seem opposite, they share the same drawback: the knowledge base is useless since one can not draw valid conclusions from it. Possibilistic logic is a logic of uncertainty able to deal with inconsistency in classical logic. By putting on every formula a degree of certainty, it defines a way to compute, with regard to these degrees, a consistent subset of formulas that can be then used in a classical inference process. In this work, we address the treatment of inconsistency in Answer Set Programming by a possibilistic approach that takes into account the non monotonic aspect of the framework.

1 Introduction

Answer Set Programming (ASP) [1] is an appropriate formalism to represent various problems issued from Artificial Intelligence and arising when available knowledge is incomplete as in non monotonic reasoning, planning, diagnosis. . . In ASP, knowledge is encoded by logical rules and solutions are obtained as models. Each model is a minimal set of atoms containing some facts and deductions obtained by applying by default some rules. So, conclusions rely on present and absent data, they form a coherent set of hypotheses and represent a rational view on the world described by the rules. Thus, in whole generality there is not a unique set of conclusions but maybe many ones or none. When there is no answer set, the program is said to be inconsistent and it is not possible to reason with it. But, as far as we know, there is no work in ASP that deals with inconsistent programs.

Possibilistic logic [2] is issued from Zadeh's possibility theory [3]. It offers a framework for representation of states of partial ignorance owing to the use of a dual pair of possibility and necessity measures. Possibility theory may be quantitative or qualitative [4] according to the range of these measures which may be the real interval $[0, 1]$, or a finite linearly ordered scale as well. Possibilistic logic provides a sound and complete machinery for handling qualitative uncertainty with respect to a semantics expressed by means of possibility distributions which rank-order the possible interpretations. In other words, it deals with uncertainty by means of classical 2-valued interpretations that can be more or less certain.

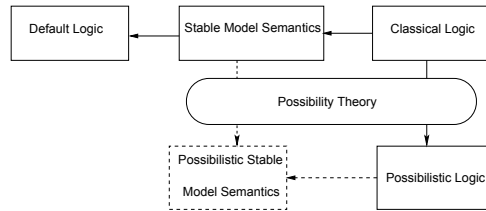


Fig. 1. Possibilistic Stable Model Semantics

In [5], we have defined *Possibilistic Stable Model Semantics* that is a new framework dealing with uncertainty in ASP. It has been developed to handle normal logic programs in which each rule is given with a certainty degree and it is based on the introduction into ASP of possibility theory concepts. Figure 1 positions our work within the linked formalisms.

In classical logic, inconsistency is characterized by the fact that every formula is a theorem. Conversely, in ASP inconsistency is characterized by the non existence of any answer. Even, if these two situations seem opposite, they share the same drawback: the knowledge base is useless since one can not draw valid conclusions from it. For our present work, the important point is that possibilistic logic can be used to handle the problem of inconsistency in classical logic bases. The certainty degrees attached to the formulas are seen as a rank-ordering of the classical base. This rank-ordering is then used to extract a consistent subbase of the initial inconsistent base. Thus it seems natural to us to developed a similar methodology with possibilistic normal logic programs to handle inconsistency in ASP. The idea is to extract, as it is done in possibilistic logic for classical bases, a consistent subprogram from which it is possible to compute some answers.

Next section 2 recalls some theoretical backgrounds about ASP and presents possibilistic stable model semantics [5]. In section 3 we expose our methodology to deal with inconsistency in ASP by a possibilistic approach. We show an equivalence result between our approach and this of possibilistic logic and we illustrate the use of our proposal in the context of combinatorial problems encoded in ASP. We conclude, in section 4, by briefly positioning our work relatively to others in the same area.

2 Theoretical Backgrounds

ASP is concerned by different kinds of logic programs and different semantics. In our work we deal with *normal logic programs*, interpreted by *stable model semantics* [6]. We consider given a non empty set of atoms $\mathcal{X}$ that determines the language of the programs. A *normal logic program* is a set of rules of the form: $c \leftarrow a_1, \dots, a_k, \text{not } b_1, \dots, \text{not } b_l$. where $k \geq 0, l \geq 0, \{a_1, \dots, a_k, b_1, \dots, b_l, c\} \subseteq \mathcal{X}$. A term like *not b* is called a *default negation*. The intuitive meaning of such a rule is: "if you have all the a_i 's and no b_j 's

then you can conclude c ". For such a rule r we use the following notations¹: the positive prerequisites of r , $body^+(r) = \{a_1, \dots, a_n\}$; the negative prerequisites of r , $body^-(r) = \{b_1, \dots, b_m\}$; the conclusion of r , $head(r) = c$ and the positive projection of r , $r^+ = head(r) \leftarrow body^+(r)$. If a program P does not contain any default negation (ie: $body^-(P) = \emptyset$), then P is a *definite logic program* and it has one minimal Herbrand model $Cn(P)$ (see [7]). The *reduct* P^X of a program P wrt. an atom set X is the definite logic program defined by $P^X = \{r^+ \mid r \in P, body^-(r) \cap X = \emptyset\}$ and it is the core of the definition of a *stable model* that is an atom set satisfying $S = Cn(P^S)$.

Note that a program may have one or many stable models or not at all. In this last case we say that the program is *inconsistent*, otherwise it is *consistent*. Since we restrict our attention to normal logic programs we are not concerned by possibly inconsistent stable models. These ones may appear when we consider *Answer Set Semantics* for *extended logic programs* [1] in which literals (and not only atoms) are used in the rules.

Example 1. $P_1 = \{a \leftarrow ., c \leftarrow a, not\ b., d \leftarrow not\ c, not\ f.\}$ has one stable model $\{a, c\}$, $P_2 = \{a \leftarrow not\ b., b \leftarrow not\ a., c \leftarrow a., c \leftarrow b.\}$ has two stable models $\{a, c\}$ and $\{b, c\}$ and $P_3 = \{a \leftarrow not\ a.\}$ is inconsistent.

In this paper, we work with programs and their models using a "rule-based" approach that needs the following materials. Let A be an atom set, r be a rule and P be a program (definite or normal). We say that r is *applicable* in A if $body^+(r) \subseteq A$ and we denote $App(P, A)$ the subset of P of its applicable rules in A . A satisfies r (or r is satisfied by A), denoted $A \models r$, if when r is applicable in A , then $head(r) \in A$. P is *grounded* if it can be ordered as the sequence $\langle r_1, \dots, r_n \rangle$ such that $\forall i, 1 \leq i \leq n, r_i \in App(P, head(\{r_1, \dots, r_{i-1}\}))$.

In [5] we have extended stable model semantics in order to take into account some certainty degrees on rules. To achieve this goal, we consider given a finite set of atoms $\mathcal{X}$ and a finite, totally ordered set of necessity values $\mathcal{N} \subseteq]0, 1]$. Then, a *possibilistic atom* is a pair $p = (x, \alpha) \in \mathcal{X} \times \mathcal{N}$. We denote by $p^* = x$ the classical projection of p and by $n(p) = \alpha$ its necessity degree. These notations can also be extended to a *possibilistic atom set* (p.a.s.) A that is a set of possibilistic atoms in which every atom x occurs at most one time. A *possibilistic definite logic program* (p.d.l.p.) is a set of *possibilistic rules* of the form:

$$(c \leftarrow a_1, \dots, a_k., \alpha)$$

where $k \geq 0, \{a_1, \dots, a_k, c\} \subseteq \mathcal{X}, \alpha \in \mathcal{N}$. The *classical projection* of a possibilistic r rule is $r^* = c \leftarrow a_1, \dots, a_k.$ $n(r) = \alpha$ is a necessity degree representing the certainty level of the information described by r . If R is a set of possibilistic rules, then $R^* = \{r^* \mid r \in R\}$ is the program obtained from R by forgetting all the necessity values.

From a possibilistic definite logic program P , we can determine, as it is done in possibilistic logic, a least specific possibility distribution defined on all the sets in $2^{\mathcal{X}}$ and characterized as follows.

¹ These functions are extended to a rule set as usual.

Proposition 1. *Let P be a p.d.l.p. then $\pi_P : 2^{\mathcal{X}} \rightarrow [0, 1]$ defined by $\forall A \in 2^{\mathcal{X}}$*

1. *if $A \not\subseteq \text{head}(\text{App}(P^*, A))$ then $\pi_P(A) = 0$*
2. *if $\text{App}(P^*, A)$ is not grounded then $\pi_P(A) = 0$*
3. *otherwise*
 - *if $\forall r \in P, A \models r^*$ then $\pi_P(A) = 1$*
 - *else $0 \leq \pi_P(A) < 1$ and $\pi_P(A) = 1 - \max_{r \in P} \{n(r) \mid A \not\models r^*\}$*

is the least specific possibility distribution.

By this way, we are able to rank every atom set with respect to its ability to be a model of P . If an atom set satisfies the condition in one of the two first items in proposition 1, then it is absolutely not possible for it to be a model of P . Otherwise, its possibility to be a model is related to the certainty degree of rules that it falsifies. Obviously, if A is the model of P , then its possibility is full and vice-versa (ie: $\pi_P(A) = 1 \iff A = \text{Cn}(P)$).

Now, we can give the definition of inference that is the evaluation of the necessity degree of each atom of the universe.

Definition 1. *Let P be a p.d.l.p. and π_P the least specific possibility distribution compatible with P , we define the two dual possibility and necessity measures:*

$$\Pi_P(x) = \max_{A \in 2^{\mathcal{X}}} \{\pi_P(A) \mid x \in A\} \quad \text{and} \quad N_P(x) = 1 - \max_{A \in 2^{\mathcal{X}}} \{\pi_P(A) \mid x \notin A\}$$

$\Pi_P(x)$ gives the level of consistency of x with respect to the p.d.l.p. P and $N_P(x)$ evaluates the level at which x is inferred from the p.d.l.p. P . This is closely related to the definitions of possibilistic logic. For instance, whenever an atom x belongs to the model of the classical program its possibility is total. Furthermore, the necessity measure allows us to introduce the following definition of a *possibilistic model* of a p.d.l.p.

Definition 2. *Let P be a possibilistic definite logic program, then the set*

$$\Pi\mathcal{M}(P) = \{(x, N_P(x)) \mid x \in \mathcal{X}, N_P(x) > 0\}$$

is its possibilistic model.

Proposition 2. *Let P be a p.d.l.p. then $:(\Pi\mathcal{M}(P))^*$ is the model of P^* .*

Example 2. Let us take $P = \{(a \leftarrow \cdot, 0.8), (b \leftarrow a., 0.6), (d \leftarrow a., 0.5), (d \leftarrow c., 0.9)\}$. The least specific possibility distribution induced by P is null for every atom set included in $\{a, b, c, d\}$ except for $\pi_P(\emptyset) = 0.2$, $\pi_P(\{a\}) = 0.4$, $\pi_P(\{a, b\}) = 0.5$, $\pi_P(\{a, d\}) = 0.4$ and $\pi_P(\{a, b, d\}) = 1$ (the model). So, we can compute the possibility of each atom: $\Pi_P(a) = \Pi_P(b) = \Pi_P(d) = 1$ and $\Pi_P(c) = 0$ and its certainty in term of necessity degree: $N_P(a) = 0.8$, $N_P(b) = 0.6$, $N_P(c) = 0$, $N_P(d) = 0.5$. Thus, $\Pi\mathcal{M}(P) = \{(a, 0.8), (b, 0.6), (d, 0.5)\}$ is the possibilistic model of P .

Now, we allow default negation in programs and we summarize the notion of *possibilistic stable model*. It extends the stable model semantics by taking into account the necessity degree in the rules of a given *possibilistic normal logic program* (p.n.l.p.). Such a program is a finite set of rules of the form:

$$(c \leftarrow a_1, \dots, a_k, \text{not } b_1, \dots, \text{not } b_l, \alpha) \quad k \geq 0, l \geq 0$$

for which we just have to precise that $\{b_1, \dots, b_l\} \subseteq \mathcal{X}$, all the rest being the same as for a p.d.l.p. (see the beginning of this section). As for normal logic programs, we need to define what is the reduction of a program.

Definition 3. *Let P be a p.n.l.p. and A be an atom set. The possibilistic reduct of P wrt. A is the p.d.l.p. $P^A = \{(r^{*+}, n(r)) \mid r \in P, \text{body}^-(r) \cap A = \emptyset\}$.*

By this way, the definition of a possibilistic stable model becomes natural.

Definition 4. *Let P be a p.n.l.p. and S a p.a.s., S is a possibilistic stable model of P if and only if $S = \Pi\mathcal{M}(P^{S^*})$.*

By analogy with normal logic programs we say that a p.n.l.p. P is *consistent* if P has at least one possibilistic stable model. Otherwise P is said to be *inconsistent*. Let us note that there is a one to one mapping between the stable models of P^* and the p.s.m. of P . In particular, if P is inconsistent then P^* is also inconsistent and if S is a p.s.m. of P then S^* is a stable model of P^* .

Example 3. $\{(c \leftarrow a, \text{not } d., 0.6), (d \leftarrow a, \text{not } c., 1), (a., 0.8), (e \leftarrow d., 0.3)\}$ has two possibilistic stable models: $\{(a, 0.8), (c, 0.6)\}$ and $\{(a, 0.8), (d, 0.8), (e, 0.3)\}$.

Now, we examine the semantics given to this framework by the definition of a possibility distribution induced by the necessity values associated to each normal rules. This distribution $\tilde{\pi}$ over all the atom sets (ie : over $2^{\mathcal{X}}$) has to reflect the ability of every atom set to be a stable model of P^* and that is formalized in the proposition 3.

Definition 5. *Let P be a p.n.l.p. and A be an atom set, then $\tilde{\pi}_P$ is the possibility distribution defined by $\tilde{\pi}_P : 2^{\mathcal{X}} \rightarrow [0, 1]$ such that $\forall A \in 2^{\mathcal{X}}, \tilde{\pi}_P(A) = \pi_{P^A}(A)$.*

Proposition 3. *Let P be a p.n.l.p. and $A \in 2^{\mathcal{X}}$ be an atom set, then $\tilde{\pi}_P(A) = 1 \iff A$ is a stable model of P^* .*

Thus, if there is no atom set A such that $\tilde{\pi}_P(A) = 1$ then P is inconsistent. This ends the presentation of possibilistic stable model semantics that we have introduced in previous works to manage uncertainty in ASP and that we use here to deal with inconsistency.

3 Inconsistency Handling in ASP

One feature of possibilistic logic is its ability to manage inconsistency of a formula set. It proposes a way to restore the consistency of a formula set by deleting

some less certain (or preferred) formulas, those with a low certainty degree. We present here an analogous idea in order to deal with inconsistent normal logic programs. The basic idea is to consider that every rule in the given program has a certainty degree. All rules are ranked by strata with respect to these degrees and the consistency restoring process has to keep the greatest number of most preferred strata.

3.1 Formal Definitions

A possibilistic base is a set Σ of pairs constituted with a classical formula and a weight that is a necessity degree. Σ is said to be *consistent* (resp. *inconsistent*) if its classical support, obtained by forgetting the weights, is classically consistent (resp. inconsistent). It is interesting to note that possibilistic logic addresses the problem of inconsistency by selecting a consistent subbase with respect to the necessity values of the formulas. α -cut (resp. strict α -cut) of Σ , denoted by $\Sigma_{\geq \alpha}$ (resp. by $\Sigma_{> \alpha}$), is the set of classical formulas in Σ having a certainty degree greater than (resp. strictly greater than) α . The inconsistency degree of Σ is $Inc(\Sigma) = \max\{\alpha : \Sigma_{\geq \alpha} \text{ is inconsistent}\}$. $Inc(\Sigma) = 0$ means that Σ is consistent. If Σ has no model, then, by discarding formulas which necessity degree is lower than the inconsistency degree, it defines an α -cut $\Sigma_{> Inc(\Sigma)}$ that is consistent. It is clear that this cut may eliminate some formulas that are not involved in the inconsistency. Nevertheless, $Inc(\Sigma)$ defines a plausibility level under which information is no more pertinent. So, it is justified to eliminate all the formulas representing this piece of knowledge. Let us mention that the inconsistency degree can be computed by means of the least specific possibility distribution of Σ .

The following presentation of our work is inspired by these general principles issued from possibilistic logic.

Definition 6. *Let P be a p.n.l.p.*

- *the strict α -cut of P is the subprogram $P_{> \alpha} = \{r \in P \mid n(r) > \alpha\}$*
- *the consistency cut degree of P is*

$$ConsCutDeg(P) = \begin{cases} 0 & \text{if } P \text{ is consistent} \\ 1 & \text{if } \forall \alpha \in \mathcal{N}, P_{> \alpha} \text{ is inconsistent} \\ \min_{\alpha \in \mathcal{N}} \{P_{> \alpha} \text{ is consistent}\} & \text{otherwise} \end{cases}$$

The consistency cut degree of a p.n.l.p. P defines the minimum level of certainty for which a strict α -cut of P is consistent. When P is inconsistent $P_{> ConsCutDeg(P)}$ is the consistent subprogram of P that we want to compute. Let us note that, because of the non monotonicity of the framework it does not ensure that a higher cut is necessarily consistent. And also, it is not necessarily the greatest (in number of rules) consistent subprogram of P . Here, our approach to restore the consistency of a p.n.l.p. is to delete the minimum number of less certain rules.

Example 4. Let $P = \left\{ (c \leftarrow \cdot, 1), (f \leftarrow \text{not } e, \text{not } f.0.9), (e \leftarrow \text{not } b., 0.8), \right. \\ \left. (a \leftarrow \text{not } a, \text{not } b, 0.7), (d \leftarrow c, \text{not } d., 0.6), (b \leftarrow c., 0.5) \right\}$
 $\text{ConsCutDeg}(P) = 0.7$ since $P(= P_{>0}), P_{>0.5}$ and $P_{>0.6}$ are inconsistent and $P_{>0.7}$ is consistent. Let us remark that $P_{>0.8}$ is inconsistent. This last point illustrates a notable difference between classical logic and stable model semantics. In classical logic, every subset of a consistent set of formulas is itself consistent. But, a subset of a consistent normal logic program is not necessarily consistent and this is due to the non monotonic nature of the formalism.

Definition 7. Let P be a p.n.l.p., its inconsistency degree is

$$\text{InconsDeg}(P) = 1 - \max_{A \in 2^{\mathcal{X}}} \{\tilde{\pi}_P(A)\}$$

This inconsistency degree can be used to characterize an inconsistent p.n.l.p. and to define a cut of an inconsistent p.n.l.p. that is still a superset of the consistent subprogram that we want to obtain.

Proposition 4. Let P be a p.n.l.p., then

- P is inconsistent $\iff \text{InconsDeg}(P) > 0$
- $\text{InconsDeg}(P) \leq \text{ConsCutDeg}(P)$.

We define our methodology of consistency restoration for a p.n.l.p. by means of the next function *cut* that computes the greatest (wrt. the certainty level of rules) consistent subprogram of P .

Definition 8. Let *cut* be the function defined on a p.n.l.p by

$$\begin{cases} \text{cut}(P) = P & \text{if } \text{InconsDeg}(P) = 0 \\ \text{cut}(P) = \text{cut}(P_{>\text{InconsDeg}(P)}) & \text{otherwise} \end{cases}$$

Proposition 5. Let P be a p.n.l.p. then $\text{cut}(P) = P_{>\text{ConsCutDeg}(P)}$.

Example 5. Let us come back to our program P in example 4 for which we have $\text{InconsDeg}(P) = 0.7$. The first call to *cut* is enough to compute the maximal consistent subprogram of P : $\text{cut}(P) = \{(c, 1), (f \leftarrow \text{not } e, \text{not } f.0.9), (e \leftarrow \text{not } b., 0.8)\}$ such that $\text{cut}(P)^*$ has one stable model $\{c, e\}$.

3.2 Relations with Possibilistic Logic

In this section, we focus our attention on possibilistic normal logic programs encoding classical possibilistic bases. Let $\mathcal{A}$ be an atom set from which a classical propositional base is built. Recall that every propositional base Σ can be encoded in a clause set. So, without loss of generality, we consider here only clause sets. On its turn, such a clause set Σ can be translated in a normal logic program $P(\Sigma)$ as following (a similar process is exposed in [8]). First, the translation of a clause $cl = (\neg a_1 \vee \dots \vee \neg a_n \vee b_1 \vee \dots \vee b_m)$ in a rule is $P(cl) = f \leftarrow a_1, \dots, a_n, b'_1, \dots, b'_m$. The encoding of a base Σ is

$$P(\Sigma) = \{P(cl) \mid cl \in \Sigma\} \cup \{x \leftarrow \text{not } x', x' \leftarrow \text{not } x. \mid x \in \mathcal{A}\} \cup \{bug \leftarrow f, \text{not } bug.\}$$

and the intuition behind this translation stands on the following remarks.

- x' is a new atom encoding the negative literal $\neg x$
- Rules $x \leftarrow \text{not } x'$. and $x' \leftarrow \text{not } x$. allow to generate all possible classical propositional interpretations by doing an exclusive choice between x and $\neg x$ for each atom x in $\mathcal{A}$.
- The goal of each rule $P(cl)$ is to conclude f (a new symbol for *false*) if the choice of atoms (x and $\neg x$) corresponds to an interpretation that does not satisfy the clause cl . By this way, if there exists a stable model not containing f , then it corresponds to an interpretation of Σ (since every clause is satisfied).
- The goal of special rule $bug \leftarrow f, \text{not } bug.$, where *bug* is a new symbol, is to discard every stable model containing f . Since *bug* appears in the head and in the negative body of this rule and nowhere else, if a stable model exists then it may not contain f .

By this way there is a one to one correspondence between the propositional models of Σ and the stable models of $P(\Sigma)$. But, as stated in [9] there is no modular mapping from program to set of clauses, only a modular transformation from set of clauses to program exists. So, in a way, ASP has better knowledge representation capabilities than propositional logic and it is interesting to study how it can be extended to the possibilistic case in particular when there is an inconsistency. To reach our goal, we first extend the transformation P to a new transformation PP for the possibilistic case in a natural way. If $(cl, \alpha) \in \Sigma$, then its encoding keep the same necessity degree α in $PP(\Sigma)$. A necessity value equal to 1 is assigned to all the other rules (the "technical" ones).

Definition 9. Let $\Sigma = \{(cl_i, \alpha_i), i = 1, \dots, n\}$ be a possibilistic base (in CNF), its encoding in a p.n.l.p. is:

$$PP(\Sigma) = \{(P(cl_i), \alpha_i) \mid (cl_i, \alpha_i) \in \Sigma\} \cup \{(x \leftarrow \text{not } x', 1), (x' \leftarrow \text{not } x., 1) \mid x \in \mathcal{A}\} \cup \{(bug \leftarrow f, \text{not } bug., 1)\}$$

In the sequel we use $\mathcal{X} = \cup_{a \in \mathcal{A}} \{a, a'\} \cup \{f, bug\}$ to make the correspondence between the language of the propositional base and the one of its translation.

Definition 10. $X \subseteq \mathcal{X}$ is a pseudo interpretation if

$$\forall a \in \mathcal{A}, (a \in X \vee a' \in X) \wedge (a \notin X \vee a' \notin X) \wedge bug \notin X \wedge f \notin X$$

The interesting point for p.n.l.p. encoding a possibilistic logic base is that, in this case, we are able to restore the consistency of a p.n.l.p. in only one step as it can be summarized in the figure 2.

In the following, we will say that a pseudo interpretation X corresponds to a classical interpretation ω if by translating each atom $a' \in X$ in literal $\neg a$,

possibilistic logic base		possibilistic normal logic program
inconsistent base Σ	$\implies$	inconsistent program $PP(\Sigma)$
$\Downarrow$		$\Downarrow$
consistent subbase $\Sigma_{>\alpha}$	$\iff$	consistent subprogram $PP(\Sigma)_{>\alpha}$
$\Downarrow$		$\Downarrow$
propositional model	$\iff$	stable model
α is the inconsistency degree of Σ and $PP(\Sigma)$		

Fig. 2. Relation between possibilistic logic and possibilistic stable model semantics

we obtain the interpretation² ω . By this way, every stable model of $PP(\Sigma)^*$ is a pseudo interpretation corresponding to a classical model for Σ and conversely.

Proposition 6. *Let Σ be a possibilistic base and $P = PP(\Sigma)$ its encoding in a p.n.l.p., $\forall X \subseteq \mathcal{X}$ we have*

$$X \text{ is not a pseudo interpretation and } \tilde{\pi}_P(X) = 0$$

or

$$X \text{ is a pseudo interpretation and } \tilde{\pi}_P(X) = \pi_\Sigma(\omega)$$

where ω is the interpretation that corresponds to X

Proposition 7. *Let Σ be a possibilistic base, then*

- $Inc(\Sigma) = InconsDeg(PP(\Sigma))$.
- if $Inc(\Sigma) = \alpha$, $PP(\Sigma_{>\alpha}) = (PP(\Sigma))_{>\alpha}$
- $InconsDeg(PP(\Sigma)) = 0 \implies (PP(\Sigma))^*$ has at least one stable model S that corresponds to a propositional model of Σ
- $InconsDeg(PP(\Sigma)) = \alpha > 0 \implies (PP(\Sigma))_{>\alpha}^*$ has at least one stable model S that corresponds to a propositional model of $\Sigma_{>\alpha}$.

These results establish that our methodology exposed in figure 2 is valid. There is a total equivalence between the management of classical bases with possibilistic logic and the management of the corresponding p.n.l.p

Example 6. Let $\Sigma = \left\{ (-e, 0.9), (b \vee c, 0.8), (\neg b \vee e, 0.7), (\neg a \vee b, 0.7), \right. \\ \left. (-d, 0.5), (a, 0.5), (\neg b \vee d, 0.3) \right\}$ be a possibilistic base. Its encoding as a p.n.l.p. is

$$PP(\Sigma) = \left\{ (f \leftarrow e., 0.9), (f \leftarrow b', c', 0.8), (f \leftarrow b, e', 0.7), (f \leftarrow a, b', 0.7), \right. \\ \left. (f \leftarrow d., 0.5), (f \leftarrow a', 0.5), (f \leftarrow b, d', 0.3), \right. \\ \left. \cup \{(x \leftarrow not\ x', 1), (x' \leftarrow not\ x., 1) \mid x \in \{a, b, c, d, e\}\} \right. \\ \left. \cup \{(bug \leftarrow f, not\ bug., 1)\} \right\}$$

² A pseudo interpretation leads necessary to an interpretation since it contains one occurrence of each atom (ie a or its negation) and no occurrence of f nor bug .

Then, we have $InconsDeg(PP(\Sigma)) = 0.5$ that corresponds to $Inc(\Sigma) = 0.5$ and the preferred consistent subprogram of $PP(\Sigma)$ is

$$\begin{aligned} PP(\Sigma)_{>0.5} = & \{(f \leftarrow e., 0.9), (f \leftarrow b', c', 0.8), (f \leftarrow b, e', 0.7), (f \leftarrow a, b', 0.7)\} \\ & \cup \{(x \leftarrow not\ x', 1), (x' \leftarrow not\ x., 1) \mid x \in \{a, b, c, d, e\}\} \\ & \cup \{(bug \leftarrow f, not\ bug., 1)\} \end{aligned}$$

So, we obtain $PP(\Sigma)_{>0.5} = PP(\Sigma_{>0.5})$ and $(PP(\Sigma)_{>0.5})^*$ has two stable models: $\{a', b', c, d, e'\}$ and $\{a', b', c, d', e'\}$. They correspond to the two propositional models: $\{\neg a, \neg b, c, d, \neg e\}$ and $\{\neg a, \neg b, c, \neg d, \neg e\}$ of $(\Sigma_{>0.5})^*$ the consistent subbase obtained in possibilistic logic.

3.3 Constraint Relaxation

One application domain for ASP is the encoding of combinatorial problems in such a way that, given a problem A , the stable models of a program $P(A)$ are the solutions of A . Designing $P(A)$ consists in writing three kinds of rules:

- *data rules* describing the particular data of the given instance,
- *guess rules* able to generate all the search space,
- *check rules*, or *constraints*, eliminating the points in the search space that are not solutions.

By this way, when A has no solution, the corresponding program $P(A)$ is inconsistent. In this case it may be interesting to relax some constraints in order to obtain an approximate solution of A . But which constraint has to be relaxed ? In a real case problem (ex: a timetabling problem), it is usual to have different kinds of constraints. Some of them are impossible to circumvent (ex: each teacher can not give two courses at the same time), but some others are only desirable (ex: do not place a course after 6PM). We see that all constraints can be ranked by level of importance (preference) and so our framework can encode A in a p.n.l.p $PP(A)$. If $PP(A)$ is inconsistent, then by means of inconsistency degree our function *cut* can be used to relax some less important constraints. Then, the resulting subprogram has a stable model that represents an approximate solution of the initial problem A . We illustrate this proposal by the following example of a 2-coloration of a graph.

Example 7. Let us consider A , the problem of coloring, by *red* or *green* the undirected graph $G = (\{v1, v2, v3\}, \{(v1, v2), (v2, v3), (v3, v1)\})$. Its encoding is

$$P(A) = \left\{ \begin{array}{l} \text{data rules: } v(1) \leftarrow . \quad v(2) \leftarrow . \quad v(3) \leftarrow . \\ \quad \quad \quad e(1, 2) \leftarrow . \quad e(2, 3) \leftarrow . \quad e(3, 1) \leftarrow . \\ \text{guess rules: } red(X) \leftarrow v(X), not\ green(X). \\ \quad \quad \quad green(X) \leftarrow v(X), not\ red(X). \\ \text{check rules: } bug \leftarrow e(X, Y), red(X), red(Y), not\ bug. \\ \quad \quad \quad bug \leftarrow e(X, Y), green(X), green(Y), not\ bug. \end{array} \right\}$$

**Fig. 3.** Constraint relaxation

But, $P(A)$ is inconsistent since it is obvious that it is impossible to color G with only two colors in such a way that two connected vertices have different colors. In such a problem, edges are the constraints of the graph. So let us suppose that these constraints can be ranked, by means of an importance degree on every edge as it is illustrated in the first graph of figure 3. The corresponding possibilistic normal logic program³ that encodes this additional information is:

$$PP(A) = \left\{ \begin{array}{l} (v(1) \leftarrow ., 1), \quad (v(2) \leftarrow ., 1), \quad (v(3) \leftarrow ., 1), \\ (e(1, 2) \leftarrow .1), \quad (e(2, 3) \leftarrow ., 0.7), \quad (e(3, 1) \leftarrow ., 0.9), \\ (red(X) \leftarrow v(X), not \ green(X), 1), \\ (green(X) \leftarrow v(X), not \ red(X), 1), \\ (bug \leftarrow e(X, Y), red(X), red(Y), not \ bug., 1), \\ (bug \leftarrow e(X, Y), green(X), green(Y), not \ bug., 1) \end{array} \right\}$$

Then, $InconsDeg(PP(A)) = 0.7$ and $cut(PP(A)) = PP(A)_{>0.7}$ is a consistent p.n.l.p. This subprogram $PP(A)$ encodes a relaxation of the initial problem A in which we eliminated the less important constraint as illustrated in the second graph of figure 3. Finally, the stable models, $\{red(1), green(2), green(3)\}$ and $\{red(1), green(2), green(3)\}$, of $cut(PP(A))^*$ encode some approximate solutions of the initial problem A .

Our proposal deals with over-constrained logic programs for which other works exist as Hierarchical Constraint Logic Programming [10]. This approach addresses the problem in a different way from ours, by a hierarchy of degrees and some error and comparator functions to choose between different solutions (see [11] for a survey on over-constrained systems).

4 Conclusion

In this work, we have proposed a methodology to restore the consistency of a normal logic program. Our proposal is underpinned by possibilistic stable model semantics that allows to rank the rules of a program by order of certainty or importance. We have defined a *cut* function that returns a consistent subprogram of the initial inconsistent one. We have shown that our approach is equivalent to

³ As usual in ASP, rules with variables are a shortcut for a set of instantiated rules for which each certainty degree is that of the rule with variables from which it comes.

that in possibilistic logic and illustrated how it can be used to relax a program encoding a combinatorial problem. This is useful in order to find an approach solution when the initial given problem has no solution.

There are many families of methods to handle inconsistency in stratified knowledge bases. Our work is part of the ones that restore consistency by selecting one or several consistent subbases. In this family, our approach is a cautious one that deletes all knowledge under a level of inconsistency. A different way is to keep a maximal number of data in every stratum. For instance, in [12] the knowledge is given by a stratified formula set $T = T_1 \cup \dots \cup T_n$ where the most important formulas are in T_1 . The *preferred subtheory* of T is $S = S_1 \cup \dots \cup S_n$ iff $\forall k, 1 \leq k \leq n, S_1 \cup \dots \cup S_k$ is consistent and maximal. So, the strategy to extract a consistent subbase from an inconsistent one is, from the most important stratum to the less important one, to compute for each stratum a subset of formulas consistent with the union of the previous ones. The next example illustrates that this strategy may give a different result than our one if we apply it to normal logic programs.

Example 8. Let us consider the inconsistent program $P = P_1 \cup P_2 \cup P_3 \cup P_4$ with $P_1 = \{b \leftarrow \text{not } a.\}$, $P_2 = \{a \leftarrow \text{not } a.\}$, $P_3 = \{a \leftarrow \text{not } b.\}$ and $P_4 = \{b \leftarrow \text{not } b.\}$. The preferred subtheory approach of [12] leads to the consistent subprogram $S = P_1 \cup \emptyset \cup P_3 \cup P_4 = \{b \leftarrow \text{not } a., a \leftarrow \text{not } b., b \leftarrow \text{not } b.\}$ that has a unique stable model $\{b\}$. On our side, we can represent the different strata of P by means of the p.n.l.p. $PP = \{(b \leftarrow \text{not } a., 1), (a \leftarrow \text{not } a., 0.8), (a \leftarrow \text{not } b., 0.6), (b \leftarrow \text{not } b., 0.4)\}$. Then, we find $InconsDeg(PP) = 0.4$ and so $cut(PP) = PP_{>0.4} = \{(b \leftarrow \text{not } a., 1), (a \leftarrow \text{not } a., 0.8), (a \leftarrow \text{not } b., 0.6)\}$ that is consistent and such that $cut(PP)^*$ has a unique stable model $\{a\}$.

For an inconsistent logic base Σ dealt with a possibilistic approach, the consistent subbase $\Sigma_{>Inc(\Sigma)}$ is always a subset of the preferred subtheories of Σ . Whereas the example 8 shows that it is not always the case for the normal logic programs. This difference comes from the non monotonic nature of stable model semantics. In future works, we envisage to apply in ASP other strategies for consistency restoring. Particularly, it would be interesting to study how to keep all rules not directly involved in the inconsistency.

References

1. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. *New Generation Computing* **9(3-4)** (1991) 363–385
2. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In Gabbay, D., Hogger, C., Robinson, J., eds.: *Handbook of Logic in Artificial Intelligence and Logic Programming*. Volume 3. Oxford University Press (1995) 439–513
3. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. In: *Fuzzy Sets and Systems*. Volume 1. (1978) 3–28
4. Dubois, D., Prade, H.: Possibility theory: qualitative and quantitative aspects. In Smets, P., ed.: *Handbook of Defeasible Reasoning and Uncertainty Management Systems*. Volume 1. Kluwer Academic Press (1998) 169–226

5. Nicolas, P., Garcia, L., Stéphan, I.: Possibilistic stable models. In: International Joint Conference on Artificial Intelligence, Edinburgh, Scotland (2005)
6. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In Kowalski, R.A., Bowen, K., eds.: International Conference on Logic Programming, The MIT Press (1988) 1070–1080
7. Lloyd, J.: Foundations of Logic Programming. 2nd edn. Symbolic Computation. Springer (1987)
8. Simons, P.: Extending and implementing the stable model semantics. Research Report A58, Helsinki University of Technology, Department of Computer Science and Engineering, Laboratory for Theoretical Computer Science, Espoo, Finland (2000) Doctoral dissertation.
9. Niemelä, I.: Logic programs with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and Artificial Intelligence* **25** (1999) 241–273
10. Wilson, M., Borning, A.: Hierarchical constraint logic programming. *Journal of Logic Programming* **16** (1993) 277–318
11. Jampel, M., Freuder, E.C., Maher, M.J., eds.: Over-Constrained Systems. In Jampel, M., Freuder, E.C., Maher, M.J., eds.: Over-Constrained Systems. Volume 1106 of Lecture Notes in Computer Science., Springer (1996)
12. Brewka, G.: Preferred subtheories: An extended logical framework for default reasoning. In: International Joint Conference on Artificial Intelligence. (1989) 1043–1048

Measuring the Quality of Uncertain Information Using Possibilistic Logic

Anthony Hunter¹ and Weiru Liu²

¹ Department of Computer Science, University College London,
Gower Street, London WC1E 6BT, UK

² School of Computer Science, Queen's University Belfast,
Belfast, Co Antrim BT7 1NN, UK

Abstract. In previous papers, we have presented a framework for merging structured information in XML involving uncertainty in the form of probabilities, degrees of beliefs and necessity measures [HL04, HL05a, HL05b]. In this paper, we focus on the quality of uncertain information before merging. We first provide two definitions for measuring information quality of individually inconsistent possibilistic XML documents, and they complement the commonly used concept of inconsistency degree. These definitions enable us to identify if an XML document is of *good* or *lower* quality when it is inconsistent, as well as enable us to differentiate between documents that have the same degree of inconsistency. We then propose a more general method to measure the quality of an inconsistent possibilistic XML document in terms of a pair of coherence measures.

1 Introduction

With the increasing use of XML for representing information on the Web, the need for modelling uncertainty in the information has emerged. A probabilistic approach is taken in [NJ02] which provides an XML structure to model and reason with probabilistic values attached to different levels of tags in a single XML document. The final probability of the value of a specific tag is calculated as multiple conditional probabilities on its ancestors' tags. In another approach, [KKA05] probability values are also attached to tags, but require that the probabilities of a set of values associated with a single tag must sum to 1.0, a condition that was not required in [NJ02]. A simple merging method is provided to integrate two probabilistic XML trees in [KKA05], whilst [NJ02] did not consider multiple XML documents. Both approaches are strongly rooted in relational databases and many operators, including queries are extensions of operations for probabilistic relational databases.

In contrast, the method of modelling, reasoning, and merging XML documents with uncertain information in our research ([HL04, HL05a, HL05b]) concerns information within the logical fusion framework [HS04]. We use probability theory, Dempster-Shafer theory, and possibility theory to model different types of uncertainty, as well as provide integration and aggregation mechanisms to merge multiple XML documents.

However, none of the research above has considered assessing the quality of uncertain information modelled in an XML document. In this paper, we focus on XML

documents where uncertainties are modelled by necessity measures and attempt to assess the quality of uncertain information when inconsistency occurs. We will proceed as follows: (Sec.2) we present formal definitions for possibilistic information in structured reports (a form of XML document). (Sec.3) we propose two definitions to identify a *good* quality structured report from a *lower* quality structured report when they both have the same degree of inconsistency. We also discuss how coherence measures can be used to measure the quality of an inconsistent structured report when it does not fall into either *good* or *lower* quality categories. Sect. 4 concludes the paper.

2 Preliminaries

We now provide basic definitions for structured reports, for possibility theory, and for representing uncertain information in terms of necessity measures in structured reports.

2.1 Structured Reports

We use XML to represent structured reports. So each structured report is an XML document, but not vice versa. If φ is a tagname (i.e an element name), and ϕ is a textentry, then $\langle \varphi \rangle \phi \langle / \varphi \rangle$ is a structured report. If φ is a tagname (i.e an element name), ϕ is a textentry, θ is an attribute name, and κ is an attribute value, then $\langle \varphi \ \theta = \kappa \rangle \phi \langle / \varphi \rangle$ is a structured report. If φ is an tagname and $\sigma_1, \dots, \sigma_n$ are structured report, then $\langle \varphi \rangle \sigma_1 \dots \sigma_n \langle / \varphi \rangle$ is a structured report.

Each structured report is isomorphic with a ground term of classical logic. This isomorphism is defined inductively as follows: (1) If $\langle \varphi \rangle \phi \langle / \varphi \rangle$ is a structured report, where ϕ is a textentry, then $\varphi(\phi)$ is a term that is isomorphic; (2) If $\langle \varphi \ \theta = \kappa \rangle \phi \langle / \varphi \rangle$ is a structured report, where ϕ is a textentry, then $\varphi(\phi, \kappa)$ is a term that is isomorphic; and (3) If $\langle \varphi \rangle \phi_1 \dots \phi_n \langle / \varphi \rangle$ is a structured report, and ϕ'_1 is a term that is isomorphic with ϕ_1 , ..., and ϕ'_n is a term that is isomorphic with ϕ_n , then $\varphi(\phi'_1, \dots, \phi'_n)$ is a term that is isomorphic.

2.2 Possibility Theory

Let Ω be a frame of discernment containing all the distinctive and exhaustive solutions to a question. A possibility measure and a necessity measure in possibility theory [DP88, SDK95, BDP97], denoted Π and N respectively, are functions from $\wp(\Omega)$ to $[0, 1]$ such that $\Pi(\wp(\Omega)) = 1$, $\Pi(\emptyset) = 0$, and $N(A) = 1 - \Pi(\bar{A})$.

$\Pi(A)$, the degree of possibility assigned to A , estimates to what extent the true event is possibly in A , and $N(A)$, the degree of necessity assigned to A , evaluates to what extent the true event is believed to be in A .

Both the possibility measure and necessity measure can be derived from a more elementary assignment, $\pi : \Omega \rightarrow [0, 1]$, which is referred to as a **possibility distribution**. The relationship between Π and π is

$$\Pi(A) = \max(\{\pi(\phi) | \phi \in A\})$$

which satisfies $\Pi(A \cup B) = \max(\Pi(A), \Pi(B))$. The usual condition associated with π is there exists ϕ_0 such that $\pi(\phi_0) = 1$, and in which case π is said to be *normal*.

```

<possibility>
  <ness value = "0.5">
    <nessitem>8°C</nessitem>
    <nessitem>10°C</nessitem>
  </ness>
  <ness value = "0.8">
    <nessitem>12°C</nessitem>
  </ness>
</possibility>

```

Fig. 1. A possibility-valid component (a PVC)

2.3 Representing Uncertain Information in Structured Reports

We extend the definitions for structured reports to represent uncertainty.

Definition 1. *The structured report $\langle \text{possibility} \rangle \sigma_1, \dots, \sigma_n \langle \text{possibility} \rangle$ is called a possibility-valid component (a PVC) iff for each $\sigma_i \in \{\sigma_1, \dots, \sigma_n\}$, σ_i is of the form*

$$\langle \text{ness value} = \kappa \rangle \sigma_1^i, \dots, \sigma_m^i \langle \text{ness} \rangle$$

and for each $\sigma_j^i \in \{\sigma_1^i, \dots, \sigma_m^i\}$, σ_j^i is of the form $\langle \text{nessitem} \rangle \phi \langle \text{nessitem} \rangle$ and $\kappa \in [0, 1]$, and ϕ is a textentry.

In possibility theory, both a degree of possibility (from Π) and a degree of necessity (from N) can be assigned to subsets of a set of possible values. In possibilistic logic, a weighted formula (p, a) implies that the weight a attached to formula p is interpreted as a lower bound on the degree of necessity $N(p)$ (with $N(p)$ being seen as a degree of belief on p) [BDP97, BDKP00]. In the context of this paper, a weight κ_i attached to a subset $\{\phi_1^i, \dots, \phi_l^i\}$ is equally interpreted as a lower bound on the degree of necessity of $\{\phi_1^i, \dots, \phi_l^i\}$. This also explains why we use tagname “ness” instead of “poss”.

The textentries in a PVC are elements of a pre-defined set containing mutually exclusive and exhaustive values for the related tagname. A structured report involving uncertain information with necessity measures should satisfy the following constraints.

Definition 2. *Let $\langle \text{possibility} \rangle \sigma_1, \dots, \sigma_n \langle \text{possibility} \rangle$ be a PVC, and let $\sigma_i \in \{\sigma_1, \dots, \sigma_n\}$ be of the form $\langle \text{ness value} = \kappa_i \rangle \sigma_i^1, \dots, \sigma_i^p \langle \text{ness} \rangle$, and let σ_i^k be of the form $\langle \text{nessitem} \rangle \phi_i^k \langle \text{nessitem} \rangle$ for $1 \leq k \leq p$. This component adheres to the necessity measure constraint in possibility theory iff the following conditions hold: (1) $\kappa_i \in [0, 1]$ (2) for all i, j , if $1 \leq i \leq n$ and $1 \leq j \leq n$ and $i \neq j$, then $\{\phi_i^1, \dots, \phi_i^p\} \neq \{\phi_j^1, \dots, \phi_j^q\}$.*

In contrast to situations in possibilistic logic where a possibilistic knowledge base can have both (p, a_1) and (p, a_2) where $a_1 \neq a_2$ are two degrees of necessity (each of which can be seen as a degree of belief) on the same logical sentence. In this case, (p, a_1) subsumes (p, a_2) when $a_1 > a_2$. Definition 2 restricts XML representation to the case where for each subset, there is only one degree of necessity associated with it in structured reports. This will reduce unnecessary XML segments in structured reports.

2.4 From Necessity Measures to Possibility Distributions

A PVC usually specifies a partial necessity measure. Here we recover the possibility distribution associated with this necessity measure using the minimum specificity principle. Let a PVC be $\langle \text{possibility} \rangle \sigma_1, \dots, \sigma_p \langle / \text{possibility} \rangle$ s.t. $\sigma_i \in \{\sigma_1, \dots, \sigma_p\}$ is of the form $\langle \text{ness value} = \kappa_i \rangle \psi_i \langle / \text{ness} \rangle$ and ψ_i is of the form

$$\langle \text{nessitem} \rangle \phi_{i_1} \langle / \text{nessitem} \rangle \dots \langle \text{nessitem} \rangle \phi_{i_x} \langle / \text{nessitem} \rangle$$

We denote the frame associated with a PVC as $\Omega = \{\phi_1, \dots, \phi_n\}$, and also let $\psi_i = \{\phi_{i_1}, \dots, \phi_{i_x}\}$ in order to make the subsequent description simpler. In this way, a PVC can be viewed as consisting of a finite set of weighted subsets of Ω , $\{(\psi_i, \kappa_i), i = 1, \dots, p\}$, where κ_i is interpreted as a lower bound on the degree of necessity $N(\psi_i)$. This representation is consistent with notations in [DP87a] and analogous with notations in possibilistic knowledge bases using possibilistic logic, where uncertain knowledge is represented as a set of weighted formulae, $\{(p_i, a_i), i = 1, \dots, n\}$. A subset ψ_i and formula p_i are thought to be equivalent if p_i is defined as $p_i = \bigvee q_j$, where q_j stands for “ $\phi_j \in \psi_i$ is true”. Therefore, when one of the elements in ψ_i is definitely true, formula p_i is definitely true as well.

Given a PVC, there is normally a family of possibility distributions associated with it and each of the distributions satisfying the condition $1 - \max\{\pi(\phi) | \phi \in \bar{\psi}_i\} \geq \kappa_i$. A common method to select one of the compatible possibility distributions is to use the **minimum specificity principle** [DP87a]. The minimum specificity principle allocates the greatest possibility degrees in agreement with the constraints $N(\psi_i) \geq \kappa_i$. This possibility distribution always exists [DP87a, BDP97] and is characterized as

$$\forall \phi \in \Omega, \pi(\phi) = \begin{cases} \min\{1 - \kappa_i | \phi \notin \psi_i\} & \text{when } \exists \psi_i \text{ such that } \phi \notin \psi_i \\ = 1 - \max\{\kappa_i | \phi \notin \psi_i\} & \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

Definition 3. Let a PVC be $\langle \text{possibility} \rangle \sigma_1, \dots, \sigma_p \langle / \text{possibility} \rangle$ where (1) $\sigma_i \in \{\sigma_1, \dots, \sigma_p\}$ is in the form $\langle \text{ness value} = \kappa_i \rangle \psi_i \langle / \text{ness} \rangle$; and (2) ψ_i is of the form $\langle \text{nessitem} \rangle \phi_{i_1} \langle / \text{nessitem} \rangle \dots \langle \text{nessitem} \rangle \phi_{i_x} \langle / \text{nessitem} \rangle$ and the set of weighted subsets is $\{(\psi_i, \kappa_i), i = 1, \dots, p\}$. Let the possibility distribution obtained using the minimum specificity principle be $\pi : \Omega \rightarrow [0, 1]$, where for each $\phi \in \Omega$, $\pi(\phi) = 1 - \nu$ and

$$\nu = \begin{cases} \max\{\kappa_1, \kappa_2, \dots, \kappa_t\} & \phi \notin \psi_j, j = 1, 2, \dots, t \text{ (where } p \geq t > 0) \\ 0 & \text{otherwise} \end{cases}$$

Example 1. The possibility distributions π_1 and π_2 below are obtained from the left and right PVCs in Figure 2 respectively using Eq (1).

$$\begin{aligned} \pi_1(\phi_1) &= 0.7, \pi_1(\phi_2) = 0.7, \pi_1(\phi_3) = 0.8, \pi_1(\phi_4) = 0.7 \\ \pi_2(\phi_1) &= 0.7, \pi_2(\phi_2) = 1, \pi_2(\phi_3) = 0.8, \pi_2(\phi_4) = 0.7 \end{aligned}$$

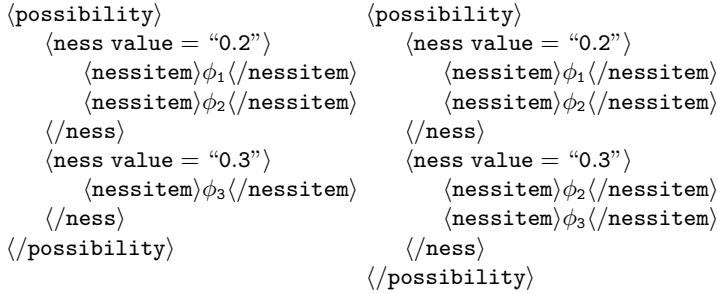


Fig. 2. Possibility-valid components (PVCs) ($\Omega = \{\phi_1, \phi_2, \phi_3, \phi_4\}$)

3 Quality of Uncertain Information with Inconsistency

3.1 Inconsistency Degree

A possibility distribution is not normal if $\forall \phi, \pi(\phi) < 1$. The value $1 - \max_{\phi \in \Omega} \pi(\phi)$ is called **the degree of inconsistency** of the original PVC and is denoted as $\text{Inc}(K)$ where K is the knowledge associated with the possibility distribution of the PVC. For instance, in Example 1, the PVC on the left is inconsistent since $\forall \phi, \pi(\phi) < 1$, whilst the right one is consistent, because $1 - \max_{\phi \in \Omega} (\pi_2(\phi)) = 0$.

Proposition 1. *Let $\{(\psi_i, a_i), i = 1, \dots, p\}$ be weighted subsets of Ω and specified in a PVC with respect to frame of discernment Ω . This PVC is **consistent** iff $\cap_i \psi_i \neq \emptyset$, otherwise the PVC is **inconsistent**.*

Example 2. Consider the two PVCs in Figure 3. The possibility distributions from them using Equation (1) are

$$\begin{aligned}
\pi_1(\phi_1) &= 0.7, \pi_1(\phi_2) = 0.7, \pi_1(\phi_3) = 0.7, \pi_1(\phi_4) = 0.7, \pi_1(\phi_5) = 0.7, \pi_1(\phi_6) = 0.7 \\
\pi_2(\phi_1) &= 0.7, \pi_2(\phi_2) = 0.7, \pi_2(\phi_3) = 0.7, \pi_2(\phi_4) = 0.7, \pi_2(\phi_5) = 0.7, \pi_2(\phi_6) = 0.7
\end{aligned}$$

The degrees of inconsistencies of the two PVCs are the same, $1 - \max_{\phi \in \Omega} (\pi_1(\phi)) = 0.3$ and $1 - \max_{\phi \in \Omega} (\pi_2(\phi)) = 0.3$. However, if we examine the structure of the weighted subsets ψ_i^1 and ψ_j^2 in detail, we will find that the right-hand side PVC is more coherent than the left one, since there is a significant overlap among the subsets ψ_j^2 in this PVC. While any two subsets in the first PVC have no common elements. This observation leads to the definitions below that further differentiates between *good* and *lower* qualities of an inconsistent PVC.

Definition 4. *Let $\langle \text{possibility} \rangle \sigma_1, \dots, \sigma_p \langle / \text{possibility} \rangle$ be PVC where (1) $\sigma_i \in \{\sigma_1, \dots, \sigma_p\}$ is in the form $\langle \text{ness value} = \kappa_i \rangle \psi_i \langle / \text{ness} \rangle$; (2) ψ_i is of the form $\langle \text{nessitem} \rangle \phi_{i_1} \langle / \text{nessitem} \rangle \dots \langle \text{nessitem} \rangle \phi_{i_x} \langle / \text{nessitem} \rangle$ and the corresponding set of weighted subsets be $\{(\psi_i, \kappa_i), i = 1, \dots, p\}$. This PVC is said to be **inconsistent with good quality**, if there exists a ψ_j , called a **separable element**, such that*

$\langle \text{possibility} \rangle$	$\langle \text{possibility} \rangle$
$\langle \text{ness value} = "0.2" \rangle$	$\langle \text{ness value} = "0.2" \rangle$
$\langle \text{nessitem} \rangle \phi_1 \langle / \text{nessitem} \rangle$	$\langle \text{nessitem} \rangle \phi_1 \langle / \text{nessitem} \rangle$
$\langle \text{nessitem} \rangle \phi_2 \langle / \text{nessitem} \rangle$	$\langle \text{nessitem} \rangle \phi_2 \langle / \text{nessitem} \rangle$
$\langle / \text{ness} \rangle$	$\langle / \text{ness} \rangle$
$\langle \text{ness value} = "0.3" \rangle$	$\langle \text{ness value} = "0.3" \rangle$
$\langle \text{nessitem} \rangle \phi_3 \langle / \text{nessitem} \rangle$	$\langle \text{nessitem} \rangle \phi_2 \langle / \text{nessitem} \rangle$
$\langle \text{nessitem} \rangle \phi_4 \langle / \text{nessitem} \rangle$	$\langle \text{nessitem} \rangle \phi_3 \langle / \text{nessitem} \rangle$
$\langle / \text{ness} \rangle$	$\langle / \text{ness} \rangle$
$\langle \text{ness value} = "0.2" \rangle$	$\langle \text{ness value} = "0.3" \rangle$
$\langle \text{nessitem} \rangle \phi_5 \langle / \text{nessitem} \rangle$	$\langle \text{nessitem} \rangle \phi_2 \langle / \text{nessitem} \rangle$
	$\langle \text{nessitem} \rangle \phi_4 \langle / \text{nessitem} \rangle$
$\langle / \text{ness} \rangle$	$\langle / \text{ness} \rangle$
$\langle \text{ness value} = "0.3" \rangle$	$\langle \text{ness value} = "0.3" \rangle$
$\langle \text{nessitem} \rangle \phi_6 \langle / \text{nessitem} \rangle$	$\langle \text{nessitem} \rangle \phi_4 \langle / \text{nessitem} \rangle$
	$\langle \text{nessitem} \rangle \phi_5 \langle / \text{nessitem} \rangle$
$\langle / \text{ness} \rangle$	$\langle / \text{ness} \rangle$
$\langle / \text{possibility} \rangle$	$\langle / \text{possibility} \rangle$

Fig. 3. PVCs ($\Omega = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6\}$)

$$\left(\bigcap_{i=1, i \neq j}^p \psi_i \right) \neq \emptyset \text{ and } \bigcap_{i=1}^p \psi_i = \emptyset \quad (2)$$

Given a PVC, there can be several separable elements ψ_j satisfying this definition. This definition identifies those PVCs each of which would have a normal possibility distribution recovered from it when the identified subset ψ_j is deleted from the PVC. As a consequence, we provide an addition normalization rule that is best suited for this type of PVCs. We assign the maximum degree of possibility to the elements that have appeared in all but one subset in a PVC which also have the highest possibility value prior to normalization.

$$\pi_{n_4}(\phi) = \begin{cases} 1 & \phi \in \left(\bigcap_{i=1}^p \psi_i \right), \psi_i \neq \psi_j, \psi_j \text{ is a separable element in Def. 4} \\ & \text{s.t. if } \exists \phi_l \in \left(\bigcap_{i=1}^p \psi_i \right), \psi_i \neq \psi_l, \text{ is a separable element} \\ & \text{in Def. 4 then } \pi(\phi) > \pi(\phi_l) \\ \pi(\phi) & \text{otherwise} \end{cases} \quad (3)$$

When there are several elements $\phi_i, \dots, \phi_j$ satisfying Eq (3) and they all have the same degree of possibility distribution, e.g., $\pi(\phi_i) = \pi(\phi_j)$, then we arbitrarily choose one of them to normalize.

This rule harnesses the 2nd of the three commonly used normalization rules as reviewed in [BDP97]:

$$\pi_{n_1}(\phi) = \frac{\pi(\phi)}{\max\{\pi(\phi_i)\}} \quad (4)$$

$$\pi_{n_2}(\phi) = \begin{cases} 1 & \text{if } \pi(\phi) = \max\{\pi(\phi_i)\} \\ \pi(\phi) & \text{otherwise} \end{cases} \quad (5)$$

$$\pi_{n_3}(\phi) = \pi(\phi) + (1 - \max\{\pi(\phi_i)\}) \quad (6)$$

As we can see, no matter which rule among these three we choose to apply, the normalized possibility distributions for the two PVCs in Fig. 3 are both reduced to a uniform distribution, e.g., for every $\phi \in \Omega$, $\pi(\phi) = 1$. However, using the new normalization rule, the right-hand side PVC in Fig. 3 has a normalized possibility distribution $\pi'_2(\phi_1) = 0.7, \pi'_2(\phi_2) = 1.0, \pi'_2(\phi_3) = 0.7, \pi'_2(\phi_4) = 0.7, \pi'_2(\phi_5) = 0.7, \pi'_2(\phi_6) = 0.7$, which assigns 1 to element ϕ_2 only. This rule produces a better normalized possibility distribution than all the other three rules.

A separable element ψ_j can be disjoint with the rest of the weighted subsets completely or it can share common elements with some weighted subsets. This leads to the following definition.

Definition 5. Let K be a PVC with a set of weighted subsets $S = \{(\psi_i, \kappa_i), i = 1, \dots, p\}$. ψ is called an **isolated separable element** if the following condition holds

$$\forall (\psi_i, \kappa_i) \in S, \psi_i \cap \psi = \emptyset \text{ when } \psi_i \neq \psi.$$

Lemma 1. Let K be a PVC which is inconsistent with good quality, if K has an isolated separable element ψ , then ψ is the only separable element.

Proposition 2. Let K be a PVC which is inconsistent with good quality and it has an isolated separable element ψ_t where $\kappa_t \geq \kappa_i$ for all other weighted subsets (ψ_i, κ_i) for $i = 1, \dots, p, i \neq t$, then

$$\text{Inc}(K) = \max(\kappa_i | i \neq t)$$

Definition 6. Let $\langle \text{possibility} \rangle \sigma_1, \dots, \sigma_p \langle / \text{possibility} \rangle$ be a PVC where (1) $\sigma_i \in \{\sigma_1, \dots, \sigma_p\}$ is in the form $\langle \text{ness value} = \kappa_i \rangle \psi_i \langle / \text{ness} \rangle$; and (2) ψ_i is of the form $\langle \text{ness item} \rangle \phi_{i_1} \langle / \text{ness item} \rangle \dots \langle \text{ness item} \rangle \phi_{i_x} \langle / \text{ness item} \rangle$ and the corresponding set of weighted subsets be $\{(\psi_i, \kappa_i), i = 1, \dots, p\}$. This PVC is said to be **inconsistent with lower quality**, if for every pair $(\psi_i, \psi_j), \psi_i \cap \psi_j = \emptyset$, when $\psi_i \neq \psi_j$.

It is easy to see that every weighted subset in such a PVC is an isolated separable element.

Proposition 3. Let K be a PVC which is inconsistent with lower quality. Then the degree of inconsistency of this PVC is as follows where $\max^{2\text{nd}}$ is a function that selects the 2nd largest value in a set of values $(\kappa_1, \dots, \kappa_p)$.

$$\text{Inc}(K) = \max^{2\text{nd}}\{\kappa_i | (\psi_i, \kappa_i)\}$$

However, these two definitions only describe the two extreme situations where in one case, all but one subset share some common elements, whilst in the other, all the subsets are separated from each other. In reality, many PVCs do not fall into these categories. We address this next.

3.2 Coherence Measures

Since an inconsistency degree alone is not sufficient to reflect the quality of an inconsistent PVC in terms of the coherence of its weighted subsets, we propose a method to further assess the quality of such a PVC.

In [DKP03], a coherence function which extends the coherence measure in [Hun02] was proposed to measure the quality of a possibilistic knowledge base when inconsistency exists. We adapt this function here in terms of weighted subsets and use our coherence measures for an inconsistent PVC.

Definition 7. Let K be a PVC.

$\text{OpinionBase}(K) = \{(\psi_i, \kappa_i) \mid \text{such that } (\psi_i, \kappa_i) \text{ is a weighted subset of } K\}$

$\text{ConflictBase}(K) = \{(\psi_i, \kappa_i) \in \text{OpinionBase}(K) \mid \exists (\psi_{i_t}, \kappa_{i_t}) \in \text{OpinionBase}(K), \text{ s.t. } \psi_i \cap \psi_{i_t} = \emptyset\}$

Then the **degree of coherence** of K is defined as follows where $A(S) = \sum_{(\psi_i, \kappa_i) \in S} \kappa_i$

$$\text{Coherence}(K) = 1 - \frac{A(\text{ConflictBase}(K))}{A(\text{OpinionBase}(K))}$$

Proposition 4. Let K be a PVC. If the possibility distribution associated with this PVC is normal, then $\text{Coherence}(K) = 1$.

When a PVC produces a normal possibility distribution, the weighted subsets in the PVC share at least one common element, therefore, the ConflictBase is empty which results in a degree of coherence of 1.

Proposition 5. Let K be a PVC. If K is inconsistent with low quality, then $\text{Coherence}(K) = 0$.

When a PVC is inconsistent with lower quality, every weighted subset in the PVC is selected in the ConflictBase, which is in turn equal to the OpinionBase, and therefore, the degree of coherence is 0.

Now, we use this new measure to examine the two PVCs in Example 2. Let K_1 and K_2 denote the two PVCs left and right respectively, the coherence measures of the two PVCs are

$$\text{Coherence}(K_1) = 1 - \frac{\sum_{i=1}^p \kappa_i}{\sum_{i=1}^p \kappa_i} = 0; \quad \text{Coherence}(K_2) = 1 - \frac{\sum_{i=1,2,4} \kappa_i}{\sum_{i=1}^4 \kappa_i} = 3/11$$

It is obvious that although the two PVCs have the same degree of inconsistency (e.g., 0.3), they have different degrees of coherence measure. The quality of K_2 is better than that of K_1 because the subsets that are assigned with degrees of belief (in terms of necessity measures) in K_2 largely overlap whilst the subsets with degrees of belief in K_1 are distinct which suggests that this knowledge is more contradicting internally.

The above defined coherence measure includes a weighted subset (e.g., (ψ_i, κ_i)) in the ConflictBase as long as there exists another weighted subset that the intersection of them is empty, although ψ_i may share some common elements with all other subsets.

Obviously, there can be many ways to define a conflict base, and the one defined in Definition 7 above is the largest in terms of cardinality.

On the other hand, the smallest conflict base possible is to include those weighted subsets which have no intersection with any other weighted subsets. This will surely result in a higher degree of coherence comparing to a larger conflictbase. Below, We give the definition of this conflict base and its corresponding coherence measure and call this measure the upper bound of the degree of coherence.

Definition 8. *Let K be a PVC.*

$$\text{OpinionBase}(K) = \{(\psi_i, \kappa_i) \mid \text{such that } (\psi_i, \kappa_i) \text{ is a weighted subset of } K\}$$

$$\begin{aligned} \text{UpperConflictBase}(K) = \{(\psi_i, \kappa_i) \in \text{OpinionBase}(K) \mid \\ \forall (\psi_{i_t}, \kappa_{i_t}) \in \text{OpinionBase}(K) \\ \text{if } \psi_i \neq \psi_{i_t} \text{ then } \psi_i \cap \psi_{i_t} = \emptyset\} \end{aligned}$$

Then the **upper bound of the degree of coherence** of K is defined as follows where $A(S) = \sum_{(\psi_i, \kappa_i) \in S} \kappa_i$.

$$\text{UpperCoherence}(K) = 1 - \frac{A(\text{UpperConflictBase}(K))}{A(\text{OpinionBase}(K))}$$

It is easy to verify that Propositions 5 and 6 are still valid with $\text{UpperCoherence}(K)$, since $\text{UpperCoherence}(K)$ is always greater than $\text{Coherence}(K)$. Interval

$$[\text{Coherence}(K), \text{UpperCoherence}(K)]$$

of a PVC defines the range of its coherence measure with following properties.

- when $[\text{Coherence}(K), \text{UpperCoherence}(K)] = [1, 1]$, the PVC is totally coherent. For example, when the associated possibility distribution of a PVC is normal, the corresponding coherence measure interval is $[1, 1]$. However, a $[1, 1]$ interval does not guarantee a PVC having a normal possibility distribution. For instance, a PVC with three weighted subsets $\{(\{1, 2, 4\}, 0.5), \{2, 3\}, 0.4), \{3, 4\}, 0.7)\}$ has interval $[1, 1]$, but its possibility distribution is not normal (where numerical numbers are the indexes for elements in the associated frame).
- when $[\text{Coherence}(K), \text{UpperCoherence}(K)] = [0, 0]$, the PVC is inconsistent with lower quality, see Proposition 6.
- when $[\text{Coherence}(K), \text{UpperCoherence}(K)] = [\alpha, 1]$ where $\alpha > 0$, the PVC has some weighted subsets that is not in conflict with any other subsets. An example is when a PVC is inconsistent with good quality and has no isolated separable elements. The right PVC in Example 5 specifies this case with the interval $[3/11, 1]$.
- when $[\text{Coherence}(K), \text{UpperCoherence}(K)] = [0, b < 1]$, the PVC has at least one isolated separable element.
- when $[\text{Coherence}(K), \text{UpperCoherence}(K)] = [0, 1]$. Any other situations not falling into the above categories.

For the last case where the pair gives $[0, 1]$ interval, there can be many situations to provoke this situation as illustrated by the next example.

<pre> <possibility> <ness value = "0.2"> <nessitem>ϕ_1</nessitem> <nessitem>ϕ_2</nessitem> </ness> <ness value = "0.3"> <nessitem>ϕ_2</nessitem> <nessitem>ϕ_3</nessitem> </ness> <ness value = "0.2"> <nessitem>ϕ_4</nessitem> <nessitem>ϕ_5</nessitem> </ness> <ness value = "0.3"> <nessitem>ϕ_5</nessitem> <nessitem>ϕ_6</nessitem> </ness> </possibility> </pre>	<pre> <possibility> <ness value = "0.2"> <nessitem>ϕ_1</nessitem> <nessitem>ϕ_2</nessitem> </ness> <ness value = "0.3"> <nessitem>ϕ_2</nessitem> <nessitem>ϕ_3</nessitem> </ness> <ness value = "0.3"> <nessitem>ϕ_3</nessitem> <nessitem>ϕ_4</nessitem> </ness> <ness value = "0.3"> <nessitem>ϕ_4</nessitem> <nessitem>ϕ_5</nessitem> </ness> </possibility> </pre>

Fig. 4. Two possibility-valid components (PVCs) ($\Omega = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6\}$)

Example 3. Consider Figure 4. Both of the PVCs have the same degree of inconsistency and the same interval of the degrees of coherence. The left PVC forms two separate clusters, whilst the right PVC forms a chain of subsets with each neighbouring pair sharing one comment element. At present, our methods for measuring coherence cannot distinguish the quality between these two situations.

Coherence measures are useful additions to the concept of degree of inconsistency, since they provide more information about the quality of an XML document when a degree of inconsistency is not sufficient. These measures can be used to rank information from multiple sources when no extra data is available about their reliability.

Definition 9. Let $\leq$ on the set $\{[1, 1], [\alpha, 1], [0, 1], [0, \beta], [0, 0]\}$ (where $1 > \alpha > 0$ and $1 > \beta > 0$) be a binary relation such that

$$\begin{aligned}
 &[0, 0] \leq [0, \beta]; [0, \beta] \leq [0, 1]; [0, 1] \leq [\alpha, 1]; [\alpha, 1] \leq [1, 1]; \\
 &[\alpha_1, 1] \leq [\alpha_2, 1] \text{ if } \alpha_1 \leq \alpha_2; \\
 &[0, \beta_1] \leq [0, \beta_2] \text{ if } \beta_1 \leq \beta_2.
 \end{aligned}$$

$\leq$ is a lex-ordering.

Proposition 6. Let K be a PVC with coherence interval $[\alpha, \beta]$. When $\alpha > 0$, $\beta = 1$ and when $\beta < 1$, $\alpha = 0$.

Proof: When $\text{Coherence}(K) = \alpha > 0$ is true, it implies that there exists at least one weighted subset, (ψ_i, κ_i) , such that for any other weighted subset (ψ_j, κ_j) , $\psi_i \cap \psi_j \neq \emptyset$, and ψ_i is not included in the $\text{ConflictBase}(K)$. It further implies that there is no isolated separable element in this component, otherwise, the intersection of ψ_i with this isolated

separable element would have been empty. Therefore, $\text{UpperConflictBase}(K) = \emptyset$, and $\text{UpperCoherence}(K) = \beta = 1$.

On the other hand, when $\beta < 1$ it implies that there is at least one isolated separable element, such that it has no common element with any other weighted subset. Therefore, every weighted subset is selected in $\text{ConflictBase}(K)$, and $\alpha = 0$. $\square$

With this proposition, together with the fact that the $\leq$ relation is a partial order relation, we see that Definition 9 is sufficient to cover all the possible intervals of coherence measures of PVCs.

Definition 10. *Let K_1 and K_2 be two PVCs with the same degree of inconsistency. Let I_{K_1} and I_{K_2} be two elements in the set in Definition 9 representing the intervals of coherence measures of K_1 and K_2 respectively. PVC K_2 is said to be more coherent than K_1 if $I_{K_1} \leq I_{K_2}$.*

Based on this partial order relation on X , it is possible to rank any number of information sources by ranking the quality of their PVCs.

4 Conclusion

In this paper, we have proposed some definitions and a coherence-based method to assess the quality of an inconsistent PVC when the degree of inconsistency alone is not adequate to serve the purpose. The coherence-based method can be used to rank information sources based on the quality of the information they provide. A potential application of the method is in information fusion where multiple PVCs need to be merged. When no preferences are given about information sources, information from highly ranked PVC could be merged before that of lower ranked ones if the sequence of merging is of an importance. Furthermore, the coherence measures can be used to select a more appropriate merging operator to merge a set of PVCs. For instance, given four PVCs which are pair-wise inconsistent, a disjunctive operator, e.g., **max**, is usually used to merge them which may result in an almost uniform possibility distribution. The merged result provides less information than the original sources. However, if the coherence measures of the conjunctively merged PVC suggest that the PVC is largely coherent, e.g., with a coherent interval $[\beta, 1]$, then applying the conjunctive operator may be of a better choice than the disjunctive one. The preliminary results of our investigation into this topic is summarized in [HL05c].

The measures of quality may also be used to assess whether a PVC should be rejected prior to merging. For example, suppose we have a set of news reports to merge, and suppose each news report is represented by a structured report, and further suppose each structured report contains a PVC with key information, then we may choose to ignore the structured reports with PVCs of low quality, or may send them back to their supplier with a request for clarification.

The two definitions on judging whether a PVC is of a *good* or *lower* quality, given that it is inconsistent, provides a way of assessing its quality without calculating its coherence intervals. A useful extension of the definition on *good* quality PVC is the new normalization rule that is best suited for this situation.

Our definitions of coherence measures can be seen as extensions of the coherence function in [DKP03] where this function is defined in a Quasi-possibilistic logic frame-

work. The definitions of the ConflictBase and the OpinionBase are based on the quasi-classical interpretations of the given knowledge base. We inherited the spirit of the function, but provided new definitions of ConflictBase and OpinionBase, as well as the UpperConflictbase in set-based situations.

Less closely related work is that on measuring the impression of a possibility distribution π ([DP87b], [HK83]), denoted as $\text{Imp}(\pi)$. This measure was defined only when the possibilistic knowledge base associated with π was consistent. For an inconsistent situation, $\text{Imp}(\pi)$ was recalculated as $\text{Imp}(\pi)/(1 - \text{Inc}(K))$.

References

- [BDP97] S Benferhat, D Dubois, and H Prade. From semantic to syntactic approach to information combination in possibilistic logic. In *Aggregation and Fusion of Imperfect Information*, pages 141-151. Physica Verlag, 1997.
- [BDKP00] S Benferhat, D Dubois, S Kaci, and H Prade. Encoding classical fusion in ordered knowledge bases framework. In *Linking Electronic Articles in Computer and Information Science*, Vol. 5, No. 027, 2000.
- [DKP03] D Dubois, S Konieczny and H Prade. Quasi-possibilistic logic and its measures of information and conflict. *Fundamenta Informaticae*, Vol. 57:101-125, 2003.
- [DP87a] D Dubois and H Prade. The principle of minimum specificity as a basis for evidential reasoning. *Uncertainty in Knowledge-Based Systems*, Bouchon and Yager (Eds.). Springer-Verlag, pages 75-84, 1987.
- [DP87b] D Dubois and H Prade. Properties of measures of information in evidence and possibility theories. *Fuzzy Sets and Systems*, Vol. 24:161-182, 1987.
- [DP88] D Dubois and H Prade. *Possibility theory: An approach to the computerized processing of uncertainty*. Plenum Press, 1988.
- [HK83] M Higashi and G Klir. Measures of uncertainty and information based on possibility distributions. *International Journal of General Systems*, Vol. 9: 43-58, 1983.
- [HL04] A Hunter and W Liu. Logical reasoning with multiple granularities of uncertainty in semi-structured information. *Proceedings of IPMU'04*, 1009-1016. 2004.
- [HL05a] A Hunter and W Liu. Fusion rules for merging uncertain information. *Information Fusion Journal*. (in press) 2005
- [HL05b] A Hunter and W Liu. Merging uncertain information with semantic heterogeneity in XML. *Knowledge and Information Systems*. (to appear) 2005.
- [HL05c] A Hunter and W Liu. Assessing the quality of merged information in possibilistic XML. Technical Report, Department of Computer Science, UCL. 2005.
- [HS04] A Hunter and R Summerton. Fusion rules for context-dependent aggregation of structured news reports. *Journal of Applied Non-classical Logic*, 14(3):329-366, 2004.
- [Hun02] A Hunter. Measuring inconsistency in knowledge via quasi-classical models. *Proceedings of AAI'2002*, 68-73, 2002.
- [KKA05] M van Keulen, A de Keijzer and W Alink A probabilistic XML approach to data integration. *Proceedings of ICDE'05*, 2005.
- [NJ02] A Nierman and H Jagadish. ProTDB: Probabilistic data in XML. In *Proceedings of VLDB'02, LNCS 2590*, pages 646-657. Springer, 2002.
- [SDK95] S Sandri, D Dubois, and H Kalfsbeek. Elicitation, assessment and polling of expert judgements using possibility theory. *IEEE Trans on Fuzzy Systems*, 3:313-335, 1995.

Remedying Inconsistent Sets of Premises

Philippe Besnard

IRIT, CNRS, Université Paul Sabatier,
118 route de Narbonne, 31062 Toulouse cedex, France
`besnard@irit.fr`

Abstract. The Lang-Marquis framework for reasoning in the presence of inconsistencies, which is based on the so-called forget operation, is generalized here. Despite extending the original proposal, a simpler structure is used. A notion of equivalence is introduced which is proven to provide extensionality for the framework. Some other formal properties are also given which illustrate the versatility of the definitions.

1 Introduction

In [Lang & Marquis 2002], a framework for reasoning from inconsistent belief bases is introduced. Despite its emphasis on a vector form of belief bases, the framework is shown to be general enough to capture various approaches such as reasoning from maximal consistent subsets, belief merging, belief revision and so on. It is based on variable forgetting as an operation for weakening beliefs in order to restore consistency, by means of specifying sets of variables whose forgetting enables the removal of inconsistency.

Here, the idea is generalized through a simplification of the original notion of a forgetting context while forgetting itself is replaced by a more general operation yielding formulas in a sublanguage of the belief base at hand. It is shown below that these new contexts allow more systematically for some properties mentioned in [Lang & Marquis 2002].

In a nutshell, a unified approach to vocabulary-based techniques for handling inconsistent belief bases is provided below.

2 Formal Preliminaries

2.1 Propositional Logic

$PROP_{PS}$ denotes the propositional language built from a finite set PS of propositional variables, the Boolean constants $\top$ (true) and $\perp$ (false), and the usual connectives. $Var(\varphi)$ denotes the set of propositional variables occurring in the formula φ .

Also, φ is consistent if and only if $\varphi \not\models \perp$ where $\varphi \models \psi$ denotes that φ entails ψ in propositional logic. That φ and ψ are equivalent in propositional logic (i.e., $\varphi \models \psi$ and $\psi \models \varphi$) is abbreviated by $\varphi \equiv \psi$.

2.2 Belief Bases and Remedies

Importantly, the belief bases we are interested in are not just collections of formulas: Here, a belief base *is also characterized* by the (finite) *number* of formulas that it contains and by the *order* in which they are presented.

Definition 1 (belief base). A belief base B of dimension n , where n is a positive integer, is a vector $\langle \varphi_1, \dots, \varphi_n \rangle$ of n formulas from $PROP_{PS}$.

Intuitively, each i ($1 \leq i \leq n$) identifies a source of information and φ_i denotes the piece of belief conveyed by source i . Note that a formula may occur more than once in B , which can be used to model the situation where two different sources (or more) give the same information.

A belief base $B = \langle \varphi_1, \dots, \varphi_n \rangle$ is conjunctively interpreted, so that it is said to be *consistent* if and only if $\varphi_1 \wedge \dots \wedge \varphi_n$ is consistent. Also, a belief base is *inconsistent* if and only if it fails to be consistent.

Definition 1 is the only item which is exactly as in [Lang & Marquis 2002]. Substituting the notion of a remedy for the forget operation, the next definition is still faithful to [Lang & Marquis 2002] but it definitely is a generalization.

Definition 2 (remedy). A remedy is a multi-mapping¹ over $PROP_{PS}$ such that for all $\varphi \in PROP_{PS}$ and for all $V \subseteq PS$

$$\varphi \mapsto \kappa V.\varphi \quad \text{where } \kappa V.\varphi \in PROP_{Var(\varphi) \setminus V}$$

A remedy is meant to amend inconsistent sets of premises by substituting formulas in a sublanguage for the original premises.

Remark 1. The notation $\kappa V.\varphi$ may look weird, it is adopted to follow the syntax of some well-known operations such as forgetting: $\exists V.\varphi$ [Lin & Reiter 1994]. In any case, κ does represent some transformation (for propositional formulas) parameterized by a set of variables V . That is, φ is transformed into a formula ψ denoted $\kappa V.\varphi$ (importantly, ψ depends on V even though no variable from V occurs in ψ).

A first illustration is with forgetting: $\exists \{v\}.\varphi[v]$ is defined as $\varphi[\top] \vee \varphi[\perp]$. I.e., $\varphi[v]$ is transformed into the greatest² formula in $PROP_{Var(\varphi) \setminus \{v\}}$ entailed by $\varphi[v]$. As just mentioned above, $\exists \{v\}.\varphi[v]$ depends on v despite the fact that v does not occur in $\exists \{v\}.\varphi[v]$. E.g., $\exists \{p\}.p \wedge q$ is q whereas $\exists \{q\}.p \wedge q$ is p .

Another illustration is with the dual of forgetting [Lin 2001] that transforms a formula $\varphi[v]$ into the least² formula in $PROP_{Var(\varphi) \setminus \{v\}}$ entailing $\varphi[v]$. In short, $\forall \{v\}.\varphi[v]$ is defined as $\varphi[\top] \wedge \varphi[\perp]$. For instance, $\forall \{p\}.p \vee q$ is q .

¹ It is the case that the pair (V, φ) yields a *unique* $\kappa V.\varphi$.

² θ is greater than σ and σ is less than θ if and only if $\sigma \models \omega$ implies $\theta \models \omega$ for all ω . Of course, the least and the greatest formulas are defined up to logical equivalence.

3 Cures and Recoveries

The main notions in our approach are *remedying contexts*, *cures* and *recoveries*. A remedying context consists of sets of variables to be ignored in each formula from a belief base B when B is being remedied to. These sets of variables need not be identical, but they should obey some constraints bearing on the remedying process:

Definition 3 (remedying context). A remedying context $\mathcal{C}$ of dimension n is a pair $\langle F, \mathfrak{R} \rangle$ where F is a subset of $PS \times \{1, \dots, n\}$ and $\mathfrak{R}$ is a binary relation over F .

Intuitively, $(v, i) \in F$ means that v is possibly to be ignored in φ_i as one may take v to be a variable which *must* be ignored in φ_i whereas $(v, i) \notin F$ means that v need not (but it may) be ignored in φ_i .

Intuitively, $(v, i)\mathfrak{R}(v', j)$ means that if v is ignored in φ_i then v' must be ignored in φ_j .

The definitions are such that a remedying context does not depend on a particular remedy and a remedying context of dimension n does not depend on a particular belief base, any belief base of dimension n can be considered.

Clearly, F and $\mathfrak{R}$ impose some constraints over the way the pieces of belief φ_i in B can be weakened. More on this below.

We first introduce the notation $\mathfrak{R}^*$ to stand for the reflexive-transitive closure of $\mathfrak{R}$. We also introduce the convenient abbreviation $F_i = \{v \in PS \mid (v, i) \in F\}$. That is, a variable v must be ignored in φ_i only if v is in F_i (i.e., a necessary condition but not a sufficient condition).

The terminology in the next definition slightly differs from the one presented in [Lang & Marquis 2002] but has been approved by the authors.

Definition 4 (cure/recovery). Consider a remedy, to be denoted by means of κ . Let $\mathcal{C} = \langle F, \mathfrak{R} \rangle$ be a remedying context of dimension n . Let $B = \langle \varphi_1, \dots, \varphi_n \rangle$ be a belief base. $\mathbf{V} = \langle V_1, \dots, V_n \rangle$ is a (κ) -solution, also said to be a (κ) -cure, for B w.r.t. $\mathcal{C}$ if and only if the following three conditions are satisfied:

- (i) for every $i \in \{1, \dots, n\}$, $V_i \subseteq F_i$
- (ii) for every i and j in $\{1, \dots, n\}$, every v and v' in PS , if $(v, i)\mathfrak{R}(v', j)$ then³ $v \in V_i$ implies $v' \in V_j$
- (iii) the recovery $B \mid \mathbf{V} \stackrel{\text{def}}{=} \langle \kappa(V_1 \cap \text{Var}(\varphi_1)).\varphi_1, \dots, \kappa(V_n \cap \text{Var}(\varphi_n)).\varphi_n \rangle$ is consistent⁴

$\mathcal{S}_{\mathcal{C}}^{\kappa}(B)$ denotes the set of all (κ) -solutions, or cures, for B wrt $\mathcal{C}$. B is said to be (κ) -curable wrt $\mathcal{C}$ if and only if $\mathcal{S}_{\mathcal{C}}^{\kappa}(B) \neq \emptyset$. Should there be no ambiguity, the abbreviated notation $\mathcal{S}_{\mathcal{C}}(B)$ can also be used.

³ Possibly, $v = v'$ or $i = j$.

⁴ Letting $B \mid \mathbf{V} \stackrel{\text{def}}{=} \langle \kappa V_1.\varphi_1, \dots, \kappa V_n.\varphi_n \rangle$ would call for $\kappa V.\varphi \equiv \kappa(V \cap \text{Var}(\varphi)).\varphi$ or any similar coherence constraint.

A noticeable consequence of Definition 3 is that it is always possible to find $\mathbf{V} = \langle V_1, \dots, V_n \rangle$ satisfying both (i) and (ii) in Definition 4.

Example 1. Consider the information about some tour-operator W that you can get from several rating agencies (F, M, S). These may deal with various matters about a company, that might well be represented by the variables below as follows:

- c “having enough cash flow”
- i “being under legal investigation”
- m “having a large part of the market”
- s “having reliable family-owned subcontractors”
- t “being a possible target for raiders”

Report by F may claim that W has enough cash flow, rehearse the well-known fact that W only has a niche as part of the market, and stress that W is under legal investigation. Report by M may claim that W has enough cash flow, has reliable family-owned subcontractors, but is under legal investigation. Report by S may claim that W has enough cash flow, is under legal investigation but is a possible target for raiders, and no longer has reliable family-owned subcontractors. Here is a belief base of dimension 3 recording the above information:

$$B = \langle \overbrace{c \wedge i \wedge \neg m}^F, \overbrace{c \wedge i \wedge s}^M, \overbrace{c \wedge i \wedge \neg s \wedge t}^S \rangle$$

Observe that B is inconsistent. How to remedy it? Presumably, the discrepancy about subcontractors could be explain by the fact that family-owned business is tough to accurately report upon. Moreover, it is not a major issue anyway. So, it might be a good idea to forget about the variable s . Formally, the remedy is chosen to be the above forget operation while the remedying context is as follows. $F = \{(s, 2), (s, 3)\}$ (it is also possible to require $(s, 1) \in F$ but it makes no difference in view of the above items). $\mathfrak{R}$ consists of two items: $(s, 2)\mathfrak{R}(s, 3)$ and $(s, 3)\mathfrak{R}(s, 2)$ (intuitively, information from M or S is regarded as equally sound when about family-owned subcontractors). Then, $\mathbf{V} = \langle V_1, V_2, V_3 \rangle$ is a cure for B where

$$\left. \begin{array}{l} V_1 = \emptyset \\ V_2 = \{s\} \\ V_3 = \{s\} \end{array} \right\} \Rightarrow B \mid \mathbf{V} = \langle c \wedge i \wedge \neg m, c \wedge i, c \wedge i \wedge r \rangle$$

In other words, the fact that the piece of information about having reliable family-owned subcontractors is ignored when coming from the M or S source is enough to get rid of the inconsistency. There are other ways: E.g., consider a remedying context with F as above and $\mathfrak{R}$ restricted to $(s, 2)\mathfrak{R}(s, 3)$ so that $(s, 3)\mathfrak{R}(s, 2)$ (intuitively, information from M about family-owned subcontractors is safer than information from S about family-owned subcontractors). An additional cure for B is now $\mathbf{V} = \langle V_1, V_2, V_3 \rangle$ where

$$\left. \begin{array}{l} V_1 = \emptyset \\ V_2 = \emptyset \\ V_3 = \{s\} \end{array} \right\} \Rightarrow B \mid \mathbf{V} = \langle c \wedge i \wedge \neg m, c \wedge i \wedge s, c \wedge i \wedge r \rangle$$

4 Equivalence and Extensionality

An important concept omitted from [Lang & Marquis 2002] is the notion of equivalence between remedying contexts. Not only does the next definition tackle the issue, it takes a simple form through a syntactical characterization.

Definition 5. *Two remedying contexts $\langle F, \mathfrak{R} \rangle$, $\langle F', \mathfrak{R}' \rangle$ are equivalent if and only if $F = F'$ and $\mathfrak{R}^* = \mathfrak{R}'^*$.*

Proposition 1. *Let $\mathcal{C}$ and $\mathcal{C}'$ be two remedying contexts of dimension n . If $\mathcal{C}$ and $\mathcal{C}'$ are equivalent, $\mathcal{S}_{\mathcal{C}}(B) = \mathcal{S}_{\mathcal{C}'}(B)$ for all belief bases B of dimension n .*

Proof. Let $\mathcal{C} = \langle F, \mathfrak{R} \rangle$ and $\mathcal{C}' = \langle F', \mathfrak{R}' \rangle$. Assume that $\mathcal{C}$ and $\mathcal{C}'$ are equivalent. Consider $\mathbf{V} = \langle V_1, \dots, V_n \rangle \in \mathcal{S}_{\mathcal{C}}(B)$. We now show that $\mathbf{V} \in \mathcal{S}_{\mathcal{C}'}(B)$, i.e. $\mathbf{V}$ satisfies all three conditions of Definition 4 wrt $\mathcal{C}'$.

First, $V_i \subseteq F_i$ for $i = 1..n$ due to $\mathbf{V} \in \mathcal{S}_{\mathcal{C}}(B)$. That is, $V_i \subseteq F'_i$ for $i = 1..n$ in view of $F = F'$ because $\mathcal{C}$ and $\mathcal{C}'$ are equivalent.

Second, let i and j be in $\{1, \dots, n\}$, let v and v' be in PS such that $(v, i)\mathfrak{R}'(v', j)$. Assume $v \in V_i$. We must show $v' \in V_j$, as is required by Condition (ii) in Definition 4. A trivial consequence of $(v, i)\mathfrak{R}'(v', j)$ is $(v, i)\mathfrak{R}^*(v', j)$. Since $\mathcal{C}$ and $\mathcal{C}'$ are equivalent, $\mathfrak{R}^* = \mathfrak{R}'^*$. Therefore, $(v, i)\mathfrak{R}^*(v', j)$. That is, there exists $k \geq 1$ such that

$$(v_0, i_0)\mathfrak{R}(v_1, i_1)\mathfrak{R} \dots \mathfrak{R}(v_{k-1}, i_{k-1})\mathfrak{R}(v_k, i_k)$$

for some $v_0, \dots, v_k$ in F and some $i_0, \dots, i_k$ in $\{1, \dots, n\}$ which satisfy $v_0 = v$, $v_k = v'$, $i_0 = i$, $i_k = j$.

By $\mathbf{V} \in \mathcal{S}_{\mathcal{C}}(B)$, it follows that $v_{h-1} \in V_{i_{h-1}}$ implies $v_h \in V_{i_h}$ for $h = 1..k$. Therefore, $v \in V_i$ implies $v' \in V_j$ and Condition (ii) is proven.

Condition (iii) is independent of the remedying context under consideration: It holds because $\mathbf{V}$ is a cure for B wrt $\mathcal{C}$.

Overall, we have proven $\mathcal{S}_{\mathcal{C}}(B) \subseteq \mathcal{S}_{\mathcal{C}'}(B)$. As $\mathcal{C}$ and $\mathcal{C}'$ play a symmetric rôle, $\mathcal{S}_{\mathcal{C}}(B) = \mathcal{S}_{\mathcal{C}'}(B)$ follows. ■

The converse requires a few constraints over the remedy, to be introduced in the next definition.

Definition 6 (conservative/normal/regular remedy). *Remedies of interest are as follows for all φ in $PROP_{PS}$, all V and V' subsets of PS :*

Conservative remedies: $\varphi \models \kappa V.\varphi$

Normal remedies: $\kappa V.\varphi \equiv \top$ for all $\varphi \not\models \perp$ s.t. $\text{Var}(\varphi) \subseteq V$

Regular remedies: $\kappa V.\varphi \models \kappa V'.\varphi$ whenever $V \subseteq V' \subseteq \text{Var}(\varphi)$

Lemma 1. *Consider a conservative remedy. Let $\mathcal{C} = \langle F, \mathfrak{R} \rangle$ be a remedying context of dimension n . For all belief bases B of dimension n , if $\mathcal{S}_{\mathcal{C}}(B) = \emptyset$ then B is inconsistent.*

Proof. Assume the contrary, i.e. there exists some consistent $B = \langle \varphi_1, \dots, \varphi_n \rangle$ such that $\mathcal{S}_C(B) = \emptyset$. Taking $V = \langle F_1, \dots, F_n \rangle$, Condition (i) in Definition 4 is satisfied. Condition (ii) in Definition 4 holds due to Definition 3. B being consistent means that $\varphi_1 \wedge \dots \wedge \varphi_n$ is consistent. Then, $\varphi \models \kappa V. \varphi$ for $\varphi \in PROP_{PS}$ and $V \subseteq PS$ yields that $\kappa(F_1 \cap Var(\varphi_1)).\varphi_1 \wedge \dots \wedge \kappa(F_n \cap Var(\varphi_n)).\varphi_n$ is consistent. Condition (iii) in Definition 4 is thus met, too. Overall, $F \in \mathcal{S}_C(B)$ which contradicts $\mathcal{S}_C(B) = \emptyset$. ■

A consequence of the previous lemma is that when dealing with a conservative remedy, there exists no remedying context $\mathcal{C}$ of dimension n such that $\mathcal{S}_C(B) = \emptyset$ for all belief bases B of dimension n .

Lemma 2. *Consider a regular remedy. Let $\mathcal{C} = \langle F, \mathfrak{R} \rangle$ be a remedying context of dimension n . For every belief base B of dimension n , if $\mathcal{S}_C(B) \neq \emptyset$ then $F \in \mathcal{S}_C(B)$.*

Proof. Due to $\mathcal{S}_C(B) \neq \emptyset$, there exists $V = \langle V_1, \dots, V_n \rangle$ obeying Conditions (i) to (iii) in Definition 4. For $F = \langle F_1, \dots, F_n \rangle$, Condition (i) trivially holds. As well, F meets Condition (ii) due to Definition 3. Turning to Condition (iii), it follows from $V \in \mathcal{S}_C(B)$ that $\kappa(V_1 \cap Var(\varphi_1)).\varphi_1 \wedge \dots \wedge \kappa(V_n \cap Var(\varphi_n)).\varphi_n$ is consistent. However, $\kappa V. \varphi \models \kappa V'. \varphi$ (for all $\varphi \in PROP_{PS}$, all V and V' s.t. $V \subseteq V' \subseteq Var(\varphi)$) implies that $\kappa(V_1 \cap Var(\varphi_1)).\varphi_1 \wedge \dots \wedge \kappa(V_n \cap Var(\varphi_n)).\varphi_n \models \kappa(F_1 \cap Var(\varphi_1)).\varphi_1 \wedge \dots \wedge \kappa(F_n \cap Var(\varphi_n)).\varphi_n$. So, $\kappa(F_1 \cap Var(\varphi_1)).\varphi_1 \wedge \dots \wedge \kappa(F_n \cap Var(\varphi_n)).\varphi_n$ is consistent. I.e., F also obeys Condition (iii). ■

Lemma 2 implies that in the case of regular remedies, a belief base B is curable wrt a remedying context $\mathcal{C} = \langle F, \mathfrak{R} \rangle$ if and only if F is a cure for B wrt $\mathcal{C}$. In symbols, $F \in \mathcal{S}_C(B)$ iff $\mathcal{S}_C(B) \neq \emptyset$.

Proposition 2. *Consider a conservative and regular remedy. Let $\mathcal{C} = \langle F, \mathfrak{R} \rangle$ and $\mathcal{C}' = \langle F', \mathfrak{R}' \rangle$ be two remedying contexts of dimension n . If $\mathcal{S}_C(B) = \mathcal{S}_{C'}(B)$ for all consistent B of dimension n then $F' = F$.*

Proof. We show the contrapositive: if $F \neq F'$ then $\mathcal{S}_C(B) \neq \mathcal{S}_{C'}(B)$ for some consistent B of dimension n . So, assume $F' \neq F$. As $\mathcal{C}$ and $\mathcal{C}'$ play a symmetrical rôle, it is enough to consider $F' \not\subseteq F$. By Lemma 1, $\mathcal{S}_C(B) \neq \emptyset$ for some consistent B . Condition (i) in Definition 4 yields $F' \notin \mathcal{S}_C(B)$. Due to Lemma 1, $\mathcal{S}_{C'}(B) \neq \emptyset$ hence $F' \in \mathcal{S}_{C'}(B)$ (by Lemma 2) which yields the expected conclusion $\mathcal{S}_C(B) \neq \mathcal{S}_{C'}(B)$. ■

Proposition 3. *Given a conservative remedy, let $\mathcal{C} = \langle F, \mathfrak{R} \rangle$ and $\mathcal{C}' = \langle F', \mathfrak{R}' \rangle$ be two remedying contexts of dimension n . If $\mathcal{S}_C(B) = \mathcal{S}_{C'}(B)$ for all consistent B of dimension n then $\mathfrak{R}'^* = \mathfrak{R}^*$.*

Proof. See the proof of Corollary 1 below. ■

Proposition 4. *Consider a conservative and regular remedy. Let $\mathcal{C}$ and $\mathcal{C}'$ be two remedying contexts of dimension n . $\mathcal{C}$ and $\mathcal{C}'$ are equivalent if and only if $\mathcal{S}_C(B) = \mathcal{S}_{C'}(B)$ for all consistent B of dimension n .*

Proof. Collating Proposition 1, Proposition 2, Proposition 3. ■

One way to put it is that the above notion of equivalence between remedying contexts is extensional:

Theorem 1. *Consider a conservative and regular remedy. Consider $\mathcal{C}$ and $\mathcal{C}'$, two remedying contexts of dimension n . $\mathcal{C}$ and $\mathcal{C}'$ are equivalent if and only if $\mathcal{S}_{\mathcal{C}}(B) = \mathcal{S}_{\mathcal{C}'}(B)$ for all belief bases B of dimension n .*

Theorem 1 trivially ensues from Proposition 4. It can more substantially be viewed as ensuing from Proposition 4 by the following property.

Proposition 5. *Given a conservative remedy, let $\mathcal{C} = \langle F, \mathfrak{R} \rangle$ and $\mathcal{C}' = \langle F', \mathfrak{R}' \rangle$ be two remedying contexts of dimension n . If $\mathcal{S}_{\mathcal{C}}(B) = \mathcal{S}_{\mathcal{C}'}(B)$ for every consistent B of dimension n then $\mathcal{S}_{\mathcal{C}}(B) = \mathcal{S}_{\mathcal{C}'}(B)$ for every inconsistent B of dimension n .*

Proof. Let $\mathbf{V} \in \mathcal{S}_{\mathcal{C}}(B)$ for any inconsistent B of dimension n . For every consistent B' of dimension n , $\kappa(V_1 \cap \text{Var}(\varphi'_1)) \cdot \varphi'_1 \wedge \dots \wedge \kappa(V_n \cap \text{Var}(\varphi'_n)) \cdot \varphi'_n$ is consistent because the remedy is conservative. Thus, $\mathbf{V}$ satisfies Condition (iii) (cf Definition 4) for being a cure for B' w.r.t. $\mathcal{C}$. Now, $\mathbf{V} \in \mathcal{S}_{\mathcal{C}}(B)$ implies that $\mathbf{V}$ satisfies Conditions (i) and (ii) for being a cure for B w.r.t. $\mathcal{C}$ and also for B' w.r.t. $\mathcal{C}$ because (i) and (ii) are independent of the belief base under consideration. So, $\mathbf{V} \in \mathcal{S}_{\mathcal{C}}(B')$. By the assumption, $\mathbf{V} \in \mathcal{S}_{\mathcal{C}'}(B')$. Therefore, $\mathbf{V}$ obeys Conditions (i) and (ii) for being a cure for B' w.r.t. $\mathcal{C}'$. By the same independence reason, $\mathbf{V}$ then obeys Conditions (i) and (ii) for being a cure for B w.r.t. $\mathcal{C}'$. Lastly, $\mathbf{V} \in \mathcal{S}_{\mathcal{C}}(B)$ means that $\mathbf{V}$ satisfies Condition (iii) for being a cure for B w.r.t. $\mathcal{C}$. By the fact that (iii) is independent from the remedying context under consideration, $\mathbf{V}$ satisfies Condition (iii) for being a cure for B w.r.t. $\mathcal{C}'$. All in all, $\mathbf{V} \in \mathcal{S}_{\mathcal{C}'}(B)$. Hence, $\mathcal{S}_{\mathcal{C}}(B) \subseteq \mathcal{S}_{\mathcal{C}'}(B)$. By symmetry, $\mathcal{S}_{\mathcal{C}}(B) = \mathcal{S}_{\mathcal{C}'}(B)$. ■

However, the class of inconsistent belief bases can still be relevant to the equivalence of remedying contexts and extensionality if the remedy is normal, in addition to being conservative:

Proposition 6. *Consider a remedy which is conservative and normal. Let $\mathcal{C} = \langle F, \mathfrak{R} \rangle$ and $\mathcal{C}' = \langle F', \mathfrak{R}' \rangle$ be two remedying contexts of dimension $n \geq 2$. If $\mathcal{S}_{\mathcal{C}}(B) = \mathcal{S}_{\mathcal{C}'}(B)$ for all inconsistent B of dimension n then $\mathfrak{R}^* = \mathfrak{R}'^*$.*

Proof. Assume that $\mathcal{S}_{\mathcal{C}}(B) = \mathcal{S}_{\mathcal{C}'}(B)$ for all inconsistent B of dimension n while there exist u and v in PS such that for some h and k in $\{1, \dots, n\}$, $(u, h)\mathfrak{R}^*(v, k)$ but $(u, h)\mathfrak{R}'^*(v, k)$. By Definition 3, $(u, h)\mathfrak{R}^*(v, k)$ requires $(u, h) \in F$ i.e. $u \in F_h$. Define $B = \langle \varphi_1, \dots, \varphi_n \rangle$ such that φ_i is $\top$ for $1 \leq i \leq n$ except φ_h being u and φ_l being $\neg u$ for some $l \neq h$. Define $\mathbf{V} = \langle V_1, \dots, V_n \rangle$ as follows: for all $w \in PS$, for $i = 1 \dots n$,

$$w \in V_i \text{ iff } (u, h)\mathfrak{R}'^*(w, i) \quad (\star\star)$$

By Definition 3, it follows that $V_i \subseteq F'_i$ for $i = 1 \dots n$. This takes care of Condition (i) in Definition 4. Also, for all w and w' in PS , for $i = 1 \dots n$, for $j = 1 \dots n$, it trivially follows from $(\star\star)$ that if $w \in V_i$ then $(u, h)\mathcal{R}'^*(w, i)$. If additionally $(w, i)\mathcal{R}'(w', j)$ then $(u, h)\mathcal{R}'^*(w', j)$. In view of $(\star\star)$, $w' \in V_j$ ensues. This takes care of Condition (ii) in Definition 4. For $i = 1 \dots n$ and $i \notin \{h, l\}$, φ_i is $\top$ hence the remedy being conservative implies that $\kappa(V_i \cap \text{Var}(\varphi_i)).\varphi_i$ is equivalent with $\top$. Letting $i = h$, reflexivity of $\mathcal{R}'^*$ together with $(\star\star)$ yield $u \in V_h$ so that $\kappa(V_h \cap \text{Var}(\varphi_h)).\varphi_h$ is $\kappa\{u\}.u$ which is equivalent with $\top$ as the remedy is normal. Letting $i = l$, it happens that $\kappa(V_l \cap \text{Var}(\varphi_l)).\varphi_l$ is $\kappa(V_l \cap \text{Var}(\varphi_l)).\neg u$ whose consistency trivially follows from the remedy being conservative. Therefore, $B \mid \mathbf{V}$ is consistent and Condition (iii) in Definition 4 is satisfied, too. As a result, $\mathbf{V} \in \mathcal{S}_{C'}(B)$.

By $\mathcal{S}_C(B) = \mathcal{S}_{C'}(B)$, it follows that $\mathbf{V} \in \mathcal{S}_C(B)$. By Condition (ii) in Definition 4, for all w and w' in PS , for $i = 1 \dots n$, for $j = 1 \dots n$, if $(w, i)\mathcal{R}(w', j)$ and $w \in V_i$ then $w' \in V_j$. Repeated application (according to $(u, h)\mathcal{R}^*(v, k)$) from $u \in V_h$ entails $v \in V_k$. Due to $(\star\star)$, $(u, h)\mathcal{R}'^*(v, k)$ which contradicts the assumption.

Therefore, we have proven that $\mathcal{R}^* \not\subseteq \mathcal{R}'^*$ entails a contradiction in the presence of $\mathcal{S}_C(B) = \mathcal{S}_{C'}(B)$. However, $\mathcal{R}$ and $\mathcal{R}'$ play a symmetrical rôle. It follows that the same conclusion holds when $\mathcal{R}'^* \not\subseteq \mathcal{R}^*$. Overall, it is also the case when $\mathcal{R}'^* \neq \mathcal{R}^*$ and the proof is over. ■

Corollary 1. *Consider a conservative remedy. Let $\mathcal{C} = \langle F, \mathcal{R} \rangle$ and $\mathcal{C}' = \langle F', \mathcal{R}' \rangle$ be two remedying contexts of dimension n . If $\mathcal{S}_C(B) = \mathcal{S}_{C'}(B)$ for all consistent B of dimension n then $\mathcal{R}^* = \mathcal{R}'^*$.*

Proof. Similar to the proof of Proposition 6, omitting the existence of l (so, $n \geq 1$ and no φ_i is $\neg u$). Then, the case $i = h$ is dealt with as follows: Due to φ_h being u , the fact that the remedy is conservative makes $\kappa(V_h \cap \text{Var}(\varphi_h)).\varphi_h$ to be consistent. ■

Anyway, the fundamental result above is that when considering a conservative and regular remedy, two remedying contexts are equivalent if and only if they have exactly the same solutions.

5 Homogeneous Contexts

In some situations, the variables to be ignored in each of the pieces of information must be identical so that all sources of information are considered on equal terms.

Definition 7 (explicitly homogeneous context). *A remedying context $\mathcal{C} = \langle F, \mathcal{R} \rangle$ of dimension n is said to be explicitly homogeneous if and only if $F = V \times \{1 \dots n\}$ for some $V \subseteq PS$ and $\mathcal{R}$ is a preorder (over F) such that:*

$$\text{for all } i \text{ and } j \quad \begin{cases} 1. \text{ for all } v, v', k, l & \text{if } (v, i)\mathcal{R}(v', j) \text{ then } (v, k)\mathcal{R}(v', l) \\ 2. \text{ for all } v \text{ in } F_i & (v, i)\mathcal{R}(v, j) \end{cases}$$

The above notion is not given in [Lang & Marquis 2002] where the authors directly define homogeneous forgetting contexts for which the next definition is the obvious counterpart.

Definition 8 (homogeneous context). *A remedying context is homogeneous if and only if it is equivalent with an explicitly homogeneous context.*

Trivially, any remedying context which is equivalent with an homogeneous context is homogeneous.

Definition 9 (preference). *A preference over a set of cures $\mathcal{S}_C^\kappa(B)$ is a pre-order $\sqsubseteq$ satisfying the inclusion property:*

$$\text{For all } \mathbf{V} \text{ and } \mathbf{V}' \text{ in } \mathcal{S}_C^\kappa(B), \text{ if } \mathbf{V} \subseteq \mathbf{V}' \text{ then } \mathbf{V} \sqsubseteq \mathbf{V}'$$

It is a gathering preference whenever it also satisfies the following property:

$$\langle V_1 \cup \dots \cup V_n, \dots, V_1 \cup \dots \cup V_n \rangle \sqsubseteq \langle V_1, \dots, V_n \rangle \text{ for all } \langle V_1, \dots, V_n \rangle \text{ in } \mathcal{S}_C^\kappa(B)$$

(The converse holds due to the inclusion property.)

The motivation for introducing (ordinary, gathering, ...) preferences in [Lang & Marquis 2002] is that some recoveries⁵ are more expected than others and should be distinguished.

Definition 10 (preferred cure). *Let $\sqsubseteq$ be a preference over some $\mathcal{S}_C^\kappa(B)$. A cure $\mathbf{V}$ is preferred if and only if $\mathbf{V}$ is minimal for $\sqsubseteq$ in $\mathcal{S}_C^\kappa(B)$.*

Regular remedies are well-behaved with respect to preferences, a first example being the next property about homogeneous contexts.

Proposition 7. *Consider a regular remedy. If $\mathcal{C}$ is an homogeneous context then for all belief base B , there always exists a gathering preference over $\mathcal{S}_C^\kappa(B)$.*

Proof. Clearly, it is always possible to define a gathering preference over any $\mathcal{S}_C^\kappa(B)$ which satisfy the following property: If $\langle V_1, \dots, V_n \rangle \in \mathcal{S}_C^\kappa(B)$ then $\langle \cup V_h, \dots, \cup V_h \rangle \in \mathcal{S}_C^\kappa(B)$ where $\cup V_h$ is an abbreviation for $V_1 \cup \dots \cup V_n$. In view of Proposition 1 and Definition 8, it is enough to prove the property for explicitly homogeneous contexts.

So, consider $\mathcal{C} = \langle F, \mathfrak{R} \rangle$ and $\mathbf{V} = \langle V_1, \dots, V_n \rangle \in \mathcal{S}_C^\kappa(B)$.

Since $\mathcal{C}$ is an explicitly homogeneous context, $F_h = F_k$ for $h = 1..n$ and $k = 1..n$. Also, $V_h \subseteq F_h$ for $h = 1..n$ because $\mathbf{V} \in \mathcal{S}_C^\kappa(B)$. Therefore, $V_h \subseteq F_k$ for all h and k in $\{1, \dots, n\}$. Thus, $\cup V_h \subseteq F_k$ for $k = 1..n$. I.e., Condition (i) in Definition 4 is satisfied.

Assume $v \in \cup V_h$ and $(v, i)\mathfrak{R}(v', j)$. Trivially, $v \in \cup V_h$ implies $v \in V_k$ for some k . By the definition of an explicitly homogeneous context (cf 1.), $(v, i)\mathfrak{R}(v', j)$

⁵ [Lang & Marquis 2002] mentions preferring minimal sets of omitted variables, ...

then yields $(v, k)\mathcal{R}(v', j)$. Since $v \in V_k$, the fact that $\mathbf{V} \in \mathcal{S}_{\mathcal{C}}^{\kappa}(B)$ implies $v' \in V_j$. So, $v' \in \cup V_h$. That is, Condition (ii) in Definition 4 is satisfied.

Let $k \in \{1, \dots, n\}$. Of course, $V_k \cap \text{Var}(\varphi_k) \subseteq (\cup V_h) \cap \text{Var}(\varphi_k) \subseteq \text{Var}(\varphi_k)$. Since the remedy is regular, $\kappa(V_k \cap \text{Var}(\varphi_k)).\varphi_k \models \kappa((\cup V_h) \cap \text{Var}(\varphi_k)).\varphi_k$. Then, $\langle \kappa((\cup V_h) \cap \text{Var}(\varphi_1)).\varphi_1, \dots, \kappa((\cup V_h) \cap \text{Var}(\varphi_n)).\varphi_n \rangle$ is consistent because $\mathbf{V} \in \mathcal{S}_{\mathcal{C}}^{\kappa}(B)$ is consistent. That is, Condition (iii) in Definition 4 is satisfied.

Overall, $\langle \cup V_h, \dots, \cup V_h \rangle \in \mathcal{S}_{\mathcal{C}}^{\kappa}(B)$. ■

One (of many) way to exploit preference by defining an inference relation is:

Definition 11 (preferential inference). $B \models_{\sqsubseteq}^{\mathcal{C}} \varphi$ if and only if every preferred $\mathbf{V}$ in $\mathcal{S}_{\mathcal{C}}^{\kappa}(B)$ satisfies $B \mid \mathbf{V} \models \varphi$.

In the case that $\sqsubseteq$ is exactly $\subseteq$, it happens that the corresponding inference amounts to preserving as much content as possible from the conclusions drawn from the original belief bases. If $\sqsubseteq$ differs from $\subseteq$, then some bias is introduced (e.g., favoring false for a variable v so that $\neg v$ is concluded).

Notation. Given $B = \langle \varphi_1, \dots, \varphi_n \rangle$ and $B' = \langle \varphi'_1, \dots, \varphi'_n \rangle$, the fact that $\varphi_i \models \varphi'_i$ for $i = 1..n$ is abbreviated $B \Vdash B'$.

Lemma 3. Consider a regular remedy. If $\mathbf{V} \subseteq \mathbf{V}'$ then $B \mid \mathbf{V} \Vdash B \mid \mathbf{V}'$.

Proof. If $\mathbf{V} \subseteq \mathbf{V}'$ then $V_i \subseteq V'_i$. Since the remedy is regular, it follows that $\kappa(V_i \cap \text{Var}(\varphi_i)).\varphi_i \models \kappa(V'_i \cap \text{Var}(\varphi'_i)).\varphi'_i$. ■

In the lemma, whether $B \mid \mathbf{V} \in \mathcal{S}_{\mathcal{C}}^{\kappa}(B)$ and whether $B \mid \mathbf{V}' \in \mathcal{S}_{\mathcal{C}}^{\kappa}(B)$ are independent in general.

Lemma 4. Consider a regular remedy. For every φ , $B \models_{\sqsubseteq}^{\mathcal{C}} \varphi$ if and only if $B \mid \mathbf{V} \models \varphi$ for all preferred $\mathbf{V}$ such that $\mathbf{V} \subset \mathbf{V}'$ holds for no preferred $\mathbf{V}'$.

Proof. Applying Lemma 3 for all preferred $\mathbf{V}'$ in $\mathcal{S}_{\mathcal{C}}^{\kappa}(B)$ such that $\mathbf{V} \subset \mathbf{V}'$ provides the if direction. The only if direction is trivial. ■

The next proposition shows that the proviso stated in [Lang & Marquis 2002] for gathering preferences to make sense is in fact automatically satisfied when the remedy is regular.

Proposition 8. Given a regular remedy, let $\mathcal{C} = \langle F, \mathcal{R} \rangle$ be a remedying context of dimension $n \geq 2$ that obeys the following property for all B of dimension n :

$$\text{for all } \langle V_1, \dots, V_n \rangle \in \mathcal{S}_{\mathcal{C}}^{\kappa}(B), \langle V_1 \cup \dots \cup V_n, \dots, V_1 \cup \dots \cup V_n \rangle \in \mathcal{S}_{\mathcal{C}}^{\kappa}(B)$$

Then, $F = V \times \{1, \dots, n\}$ for some $V \subseteq PS$.

Proof. Assume there exists x such that $x \in F_i \setminus F_j$ for some i and j in $\{1, \dots, n\}$. (I.e., $\mathcal{C}$ is not such that $F = V \times \{1, \dots, n\}$ for some $V \subseteq PS$.)

Of course, there exists some consistent $B = \langle \varphi_1, \dots, \varphi_n \rangle$ such that $x \notin \text{Var}(\varphi_i)$. Consequently, there exists some $V \in \mathcal{S}_{\mathcal{C}}^{\kappa}(B)$. Define V' to be just like V except that $V'_i = V_i \cup \{x\}$. Construct V'' from V' by enriching V' according to the requirements imposed by $\mathfrak{R}$ through Condition (ii) in Definition 4. Therefore, V'' satisfies Condition (i) and Condition (ii) from Definition 4. By Lemma 3, V'' also satisfies Condition (iii) in Definition 4 due to $V \subseteq V''$ and $V \in \mathcal{S}_{\mathcal{C}}^{\kappa}(B)$. Thus, $V'' \in \mathcal{S}_{\mathcal{C}}^{\kappa}(B)$.

While it additionally is the case that $x \in V''_i$, the property given in the statement of the proposition yields $\langle V''_1 \cup \dots \cup V''_n, \dots, V''_1 \cup \dots \cup V''_n \rangle \in \mathcal{S}_{\mathcal{C}}^{\kappa}(B)$ which by Condition (i) in Definition 4 requires its j th component (i.e., $V''_1 \cup \dots \cup V''_n$) to be a subset of F_j . However, $x \in V''_i$ trivially implies $x \in V''_1 \cup \dots \cup V''_n$ and a contradiction arises because $x \notin F_j$. ■

The next and final result shows that inference from a gathering preference can always be captured through homogeneous contexts.

Proposition 9. *Consider a regular remedy. If $\sqsubseteq$ is a gathering preference over $\mathcal{S}_{\mathcal{C}}^{\kappa}(B)$ then there always exists an homogeneous context $\mathcal{C}'$ such that $\models_{\sqsubseteq}^{\mathcal{C}}$ is $\models_{\sqsubseteq}^{\mathcal{C}'}$, for some preference $\sqsubseteq'$.*

Proof. Let $V = \langle V_1, \dots, V_n \rangle$ be preferred in $\mathcal{S}_{\mathcal{C}}^{\kappa}(B)$. As $\sqsubseteq$ is a gathering preference over $\mathcal{S}_{\mathcal{C}}^{\kappa}(B)$, it follows that $\langle \cup V_h, \dots, \cup V_h \rangle \sqsubseteq V$. By Lemma 4, $B \models_{\sqsubseteq}^{\mathcal{C}} \varphi$ if and only if $B \mid \langle \cup V_h, \dots, \cup V_h \rangle \models \varphi$ for all preferred $V = \langle V_1, \dots, V_n \rangle$ in $\mathcal{S}_{\mathcal{C}}^{\kappa}(B)$. Then, the consequences are the same when $\mathcal{C}$ is changed to $\mathcal{C}'$ where $\mathcal{S}_{\mathcal{C}'}^{\kappa}(B)$ is the set S consisting of the preferred V in $\mathcal{S}_{\mathcal{C}}^{\kappa}(B)$ such that $V = \langle \cup V_h, \dots, \cup V_h \rangle$. Next, define F' as follows: $F'_i = \bigcup_{V \in S} V_i$. Clearly, $F'_i = F_j$ for all i and j in $\{1, \dots, n\}$. Lastly, Definition 4 and Definition 8 show that $\mathfrak{R}'$ being the transitive closure of $\mathfrak{R}$ is as required for a homogeneous remedying context. ■

6 Concluding Remarks

The above framework captures various methods centered around inconsistency (be it consistency-based belief revision [Delgrande & Schaub 2000], belief merging [Konieczny & Pino-Perez 1998], ...) just by generalizing an approach introduced in [Lang & Marquis 2002]. The first dimension for generalization is what is called remedy here. In [Lang & Marquis 2002], there only is a fixed operation: Forgetting [Lin & Reiter 1994]. It is defined as follows: $\exists v_1, \dots, v_n. \varphi = \exists v_1. (\exists v_2, \dots, v_n. \varphi)$ for $n \geq 2$ while $\exists v. \varphi = \varphi_{v \leftarrow \perp} \vee \varphi_{v \leftarrow \top}$. More generally, $\exists V. \varphi = \exists v_1, \dots, v_n. \varphi$ when $V = \{v_1, \dots, v_n\}$ and $\exists \emptyset. \varphi = \varphi$. Clearly, forgetting is not only a remedy but it also is conservative, normal and regular.

Being conservative insures that overcoming inconsistency is not an excuse to introduce new information, obviously a wrong move. Being normal means that

the maximum amount of information is taken out when all the variables mentioned in the premises are to be ignored: In other words, only relevant items from the alphabet have an effect on losing information (if φ does not mention v , i.e., φ is not about v , then ignoring v in φ should not result in any loss of information). Regular remedies are reminiscent of the kind of comparison between conditions as discussed in [Lin 2001], they guarantee a logical behaviour when dealing with inconsistency (that is not to say that non-regular remedies are to be banned on the count of irrationality). Moreover, homogeneous contexts are now handled in a systematic way and various general results hold (one of them was loosely stated in [Lang & Marquis 2002]). Indeed, the notion of remedying contexts makes it possible to express more elegantly many properties and determine whether they hold. Lastly, there is room for further generalization as remedying contexts need not have a fixed relation $\mathcal{R}$, ... The framework given here does not only capture the methods mentioned in the previous paragraph, the generalization it brings out allows for discussing new work in the area such as [Su, Lv & Zhang 2004].

Acknowledgements

The author is grateful to Jérôme Lang and Pierre Marquis for many comments.

References

- [Coste-Marquis & al. 2004]
Sylvie Coste-Marquis, Jérôme Lang, Paolo Liberatore, Pierre Marquis. Expressive Power and Succinctness of Propositional Languages for Preference Representation. In *Proceedings of the 9th Conference on Knowledge Representation (KR-2004)*, pages 203-212. Morgan Kaufmann, 2004.
- [Delgrande & Schaub 2000]
James P. Delgrande, Torsten Schaub. A consistency-based approach for belief change. *Artificial Intelligence*, 151(1-2):1-41, 2003.
- [Konieczny & Pino Pérez 1998]
Sébastien Konieczny, Ramón Pino Pérez. On the Logic of Merging. In *Proceedings of the 6th Conference on Knowledge Representation (KR-1998)*, pages 488-498. Morgan Kaufmann, 1998.
- [Lang & Marquis 2002]
Jérôme Lang, Pierre Marquis. Resolving Inconsistencies by Variable Forgetting. In *Proceedings of the 8th Conference on Knowledge Representation (KR-2002)*, pages 239-250. Morgan Kaufmann, 2002.
- [Lang, Liberatore & Marquis 2003]
Jérôme Lang, Paolo Liberatore, Pierre Marquis. Propositional Independence: Formula-Variable Independence and Forgetting. *Journal of Artificial Intelligence Research* 18:391-443, 2003.
- [Lin 2001]
Fangzhen Lin. On strongest necessary and weakest sufficient conditions. *Artificial Intelligence*, 128(1-2):143-159, 2001.

[Lin & Reiter 1994]

Fangzhen Lin, Raymond Reiter. Forget it! In *Proceedings of the AAAI Fall Symposium on Relevance*, pages 154-159, 1994.

[Su, Lv & Zhang 2004]

Kaile Su, Guanfeng Lv, Yan Zhang. Reasoning about Knowledge by Variable Forgetting. In *Proceedings of the 9th Conference on Knowledge Representation (KR-2004)*, pages 576-586. Morgan Kaufmann, 2004.

Measuring Inconsistency in Requirements Specifications

Kedian Mu¹, Zhi Jin^{1,2}, Ruqian Lu^{1,2}, and Weiru Liu³

¹ Institute of Computing Technology,
Chinese Academy of Sciences, Beijing 100080, P.R.China

² MADIS, Academy of Mathematics and System Sciences,
Chinese Academy of Sciences, Beijing 100080, P.R.China

³ School of Computer Science,
Queen's University Belfast, BT7 1NN, Northern Ireland

Abstract. In the field of requirements engineering, measuring inconsistency is crucial to effective inconsistency management. A practical measure must consider both the degree and significance of inconsistency in specification. The main contribution of this paper is providing an approach for measuring inconsistent specification in terms of the priority-based scoring vector, which integrates the measure of the degree of inconsistency with the measure of the significance of inconsistency. In detail, for each specification Δ that consists of a set of requirements statements, if L is a m -level priority set, we define a m -dimensional priority-based significance vector $\vec{V}$ to measure the significance of the inconsistency in Δ . Furthermore, a priority-based scoring vector $\vec{S}_P: \mathcal{P}(\Delta) \rightarrow N^{m+1}$ has been defined to provide an ordering relation over specifications that describes which specification is “more essentially inconsistent than” others.

1 Introduction

It is widely recognized that inconsistency is unavoidable during the requirements stage, though most existing software development techniques or tools assume consistency [1–3]. A practical way of handling inconsistency is learning to live with inconsistency rather than parry it [3]. Furthermore, in many cases, it may be desirable to take the initiative in managing inconsistency to facilitate the requirements development and management [2]. Inconsistencies could be viewed as signals of problematical information about requirements.

Measuring inconsistency is crucial to effective inconsistency management [2, 1]. In general, customers and developers need to know the number and severity of inconsistencies in their requirements specifications. Often, developers need to use these measures to prioritize inconsistencies in order to identify inconsistencies that require urgent attentions, and to assess the progress after inconsistency-handling. In other words, the developers need to know if a set of requirements statements become more or less “consistent” after a particular inconsistency-handling action has been taken.

It is not surprising that techniques for measuring inconsistent specifications in classical logic are appealing [4]. In practical inconsistency-handling, customers and developers need to know both the significance and severity of inconsistency. The relative importance of a requirements statement always affects the evaluation of significance of an inconsistent specification. Therefore, central to measuring inconsistent specifications is the need to take the relative importance of requirements statements into account.

An approach to evaluating the significance of inconsistency in the framework of QC logic was proposed in [5] recently. It is based on specifying the relative significance of incoherent QC models using additional information, encoded as a mass assignment in Dempster-Shafer theory. This approach is not appropriate for measuring inconsistency in requirements specifications, though the QC logic is very appealing for representing inconsistent requirements specifications. Generally speaking, the relative importance of a requirements statement is implied by the relative priority of this statement in practical software development. But prioritization is just a strategy for differentiating requirements at a coarse granularity by relative importance and urgency. A common approach to prioritization is to group requirements statements into three priority categories, such as three-level scale of “*Essential*”, “*Conditional*”, and “*Optional*” [6, 7]. However, all such scales are subjective and imprecise, so it is difficult to specify the relative significance of inconsistency in the framework of Dempster-Shafer theory.

In this paper, we propose a new approach for measuring inconsistent specifications, which considers both the degree and significance of inconsistency based on the relative priorities of requirements statements. The rest of the paper is organized as follows. Section 2 introduces some preliminary notions. Section 3 presents the approach for measuring inconsistencies in requirements specifications. Finally, we conclude this paper in Section 4.

2 Preliminaries

As mentioned above, classical logic is appealing for representing the requirements specifications. We start this section with some notations of classical logic.

Let $\mathcal{L}_{\Phi_0}$ be the language composed from a set of classical atoms Φ_0 and logical connectives $\{\vee, \wedge, \neg, \rightarrow\}$ and let $\vdash$ be the classical consequence relation. Let $\alpha \in \mathcal{L}_{\Phi_0}$ be a classical formula and $\Delta \subseteq \mathcal{L}_{\Phi_0}$ a set of formulas in $\mathcal{L}_{\Phi_0}$. In this paper, we call Δ a requirements specification while each formula $\alpha \in \Delta$ represents a requirements statement.

Let $\text{Consequence}(\Delta) = \{\alpha \mid \Delta \vdash \alpha\}$. If $\exists \alpha$ such that $\Delta \vdash \alpha$ and $\Delta \vdash \neg \alpha$, then we call Δ is inconsistent and abbreviate $\alpha \wedge \neg \alpha$ by $\perp$.

Generally, both the “plausible” and “problematical” information in the inconsistent set of formulas is of interest. However, for any set of formulas, we may consider each of its maximal consistent subsets as the reflection of one of many plausible views of the requirements specification. Furthermore, we consider the common subset of all its maximal consistent subsets as the reflection of all the “uncontroversial” information in it. On the other hand, we consider the union of

all its minimal inconsistent subsets as the collection of all the “problematical” information [8].

Definition 1. Let Δ be a requirements specification. Then

$$\begin{aligned} \text{CON}(\Delta) &= \{\Gamma \subseteq \Delta \mid \Gamma \not\vdash \perp\}, \quad \text{INC}(\Delta) = \{\Gamma \subseteq \Delta \mid \Gamma \vdash \perp\} \\ \text{MC}(\Delta) &= \{\Phi \in \text{CON}(\Delta) \mid \forall \Psi \in \text{CON}(\Delta), \Phi \not\subset \Psi\} \\ \text{MI}(\Delta) &= \{\Phi \in \text{INC}(\Delta) \mid \forall \Psi \in \text{INC}(\Delta), \Psi \not\subset \Phi\} \\ \text{FREE}(\Delta) &= \bigcap_{\Phi \in \text{MC}(\Delta)} \Phi = \Delta - \bigcup_{\Psi \in \text{MI}(\Delta)} \Psi, \quad \text{CORE}(\Delta) = \Delta - \text{FREE}(\Delta) \end{aligned}$$

$\text{MC}(\Delta)$ is the set of maximal consistent subsets of Δ ; $\text{MI}(\Delta)$ is the set of minimal inconsistent subsets of Δ ; and $\text{FREE}(\Delta)$ is the set of formulas that appear in all the maximal consistent subsets of Δ .

Example 1. Let $\Delta = \{\alpha, \neg\gamma, \beta, \neg\beta \vee \gamma\}$, then $\text{MC}(\Delta) = \{\Phi_1, \Phi_2, \Phi_3\}$, where $\Phi_1 = \{\alpha, \neg\gamma, \neg\beta \vee \gamma\}$, $\Phi_2 = \{\alpha, \beta, \neg\beta \vee \gamma\}$, and $\Phi_3 = \{\alpha, \neg\gamma, \beta\}$, $\text{MI}(\Delta) = \{\{\neg\gamma, \beta, \neg\beta \vee \gamma\}\}$, and $\text{FREE}(\Delta) = \{\alpha\}$.

For a set of formulas Δ , a scoring function S is defined from $\mathcal{P}(\Delta)$ (the power set of Δ) into the natural numbers so that for any $\Gamma \in \mathcal{P}(\Delta)$, $S(\Gamma)$ gives the number of minimal inconsistent subsets of Δ that would be eliminated if the subset Γ was removed from Δ [8]. That is, for $\Gamma \subseteq \Delta$, $S(\Gamma) = |\text{MI}(\Delta)| - |\text{MI}(\Delta - \Gamma)|$. As such, sets of formulas could be compared using their scoring functions so that an ordering relation, which means “*more inconsistent than*”, over these sets can be defined.

Definition 2. (*score ordering* [8], $\leq$) Assume that Δ_i and Δ_j are of the same cardinality, S_i is the scoring function for Δ_i , and S_j the scoring function for Δ_j . $S_i \leq S_j$ holds iff there is a bijection $f : \mathcal{P}(\Delta_i) \mapsto \mathcal{P}(\Delta_j)$ such that the following condition can be satisfied:

$$\forall \Gamma \in \mathcal{P}(\Delta_i), S_i(\Gamma) \leq S_j(f(\Gamma))$$

Note that $S_i < S_j$ iff $S_i \leq S_j$ and $S_j \not\leq S_i$. Also, $S_i \simeq S_j$ iff $S_i \leq S_j$ and $S_j \leq S_i$. We say Δ_j is **more inconsistent than** Δ_i iff $S_i \leq S_j$.

3 Approach for Measuring Inconsistent Specification

Let m , a natural number, be the scale of the priority and L be $\{l_0^m, \dots, l_{m-1}^m\}$, a totally ordered finite set of m symbolic values of the priorities, i.e. $l_i^m < l_j^m$ iff $i < j$. Furthermore, each symbolic value in L could associate with a linguistic value. For example, for a three-level priority set, we have a totally ordered set L as $L = \{l_0^3, l_1^3, l_2^3\}$ where [6, 7]

$$l_0^3 : \text{Optional}, \quad l_1^3 : \text{Conditional}, \quad l_2^3 : \text{Essential}$$

In the rest of paper, we adopt this three-level priority set, though it is not obligatory. We may ignore the superscript m if no ambiguous arises. According to the convention in software engineering, the intuitive meaning of “*essential*” is that *the software product could not be acceptable unless all of the essential requirements are satisfied*; the meaning of “*conditional*” is that *these requirements would enhance the software product, but it is not unacceptable if absent*; the meaning of “*optional*” is that *these requirements may or may not be worthwhile*. In some sense, the priority could be seen as the abstraction of the requirements’ significance.

Prioritizing requirements statements in Δ is in essence to establish a prioritizing mapping $P : \Delta \mapsto L$ by balancing the business benefit that each requirements statement can provide against its cost and technique risk.

Definition 3. Let Δ be a requirements specification and L a m -level priority set. Let P be a prioritizing mapping $P : \Delta \mapsto L$. The priority-based partition of Δ under L can be defined as $\{\Delta^0, \dots, \Delta^{m-1}\}$, such that

$$\Delta^i = \{\alpha \in \Delta | P(\alpha) = l_i\}, \text{ for } i = 0, \dots, m-1.$$

Obviously, each component of the priority-based partition of Δ is a subset of Δ . We give an example to illustrate the priority-based partition.

Example 2. Let L be a three-level priority set, and $\Delta = \{\alpha, \neg\gamma, \beta, \neg\beta \vee \gamma\}$. P is the prioritizing mapping from Δ to L :

$$P(\alpha) = l_2, P(\neg\gamma) = l_2, P(\beta) = l_1, P(\neg\beta \vee \gamma) = l_0$$

Then, $\Delta^0 = \{\neg\beta \vee \gamma\}$, $\Delta^1 = \{\beta\}$, $\Delta^2 = \{\alpha, \neg\gamma\}$. Obviously $\Delta = \Delta^0 \cup \Delta^1 \cup \Delta^2$.

For the priority-based partition of Δ under L , $\{\Delta^0, \dots, \Delta^{m-1}\}$, Δ^i stands for the subset of Δ that is grouped to the category with priority level l_i . In other words, all of the requirements statements in Δ^i have the same level of relative importance and urgency. Note that, for Δ^l , the l -th component of its priority-based partition is itself, and others are $\emptyset$. For example, the priority-based partition of Δ^{m-1} is $\{\emptyset, \dots, \emptyset, \Delta^{m-1}\}$.

3.1 Priority-Based Score Ordering

Prioritizing requirements statements is in essence to differentiate the requirements statements by relative importance and urgency. In order to measure inconsistencies arising in requirements specifications, it is necessary to consider the relative priority of requirements statement in techniques. In fact, the approach based on scoring functions in [8] assumes that each formula has the same relative priority by default. In other words, it does not consider the significance of inconsistency. For the specifications consisting of requirements statements with different priorities as we have defined above, to consider their significance, we need to define a **priority-based score ordering** as follows to compare the inconsistent specifications.

Definition 4. (priority-based score ordering, $\leq_P$) Let L be a m -level priority set. Let Δ_i and Δ_j be two specifications with the same cardinality. Let $\{\Delta_i^0, \dots, \Delta_i^{m-1}\}$ and $\{\Delta_j^0, \dots, \Delta_j^{m-1}\}$ be the priority-based partitions under L of Δ_i and Δ_j , respectively. Let S_i be the scoring function for Δ_i and S_j be the scoring function for Δ_j . $S_i \leq_P S_j$ holds iff there is a bijection $f : \mathcal{P}(\Delta_i) \mapsto \mathcal{P}(\Delta_j)$ such that the following conditions can be satisfied:

- f is a bijection from $\mathcal{P}(\Delta_i^l)$ to $\mathcal{P}(\Delta_j^l)$ ($l \in \{0, \dots, m-1\}$);
- $\forall \Gamma \in \mathcal{P}(\Delta_i)$, $S_i(\Gamma) \leq S_j(f(\Gamma))$

We call $\leq_P$ the priority-based score ordering. Note that $S_i <_P S_j$ iff $S_i \leq_P S_j$ and $S_j \not\leq_P S_i$, $S_i \simeq_P S_j$ iff $S_i \leq_P S_j$ and $S_j \leq_P S_i$. We say Δ_j is **more truly inconsistent** than Δ_i iff $S_i \leq_P S_j$.

The priority-based score ordering emphasizes the bijection from $\mathcal{P}(\Delta_i^l)$ to $\mathcal{P}(\Delta_j^l)$, which provides a basis for comparing the scoring functions under the same level of priority. In other words, the significance of inconsistency is considered in the priority-based score ordering in an indirect way. Let us look at the following example to see how to compare inconsistent specifications via the priority-based score ordering.

Example 3. Let L be a three-level priority set. Consider $\Delta_1 = \{\neg\alpha, \alpha, \beta\}$ and $\Delta_2 = \{\alpha \wedge \neg\alpha, \beta, \gamma\}$. Assume that $\Delta_1^0 = \{\neg\alpha\}$, $\Delta_1^1 = \{\alpha\}$, $\Delta_1^2 = \{\beta\}$, $\Delta_2^0 = \{\alpha \wedge \neg\alpha\}$, $\Delta_2^1 = \{\beta\}$, and $\Delta_2^2 = \{\gamma\}$. Let S_1 and S_2 be the scoring functions for Δ_1 and Δ_2 respectively, as detailed below,

$$\begin{aligned} S_1(\{\neg\alpha\}) &= 1, \quad S_1(\{\alpha\}) = 1, \quad S_1(\{\beta\}) = 0, \quad S_1(\{\neg\alpha, \alpha\}) = 1 \\ S_1(\{\neg\alpha, \beta\}) &= 1, \quad S_1(\{\alpha, \beta\}) = 1, \quad S_1(\{\neg\alpha, \alpha, \beta\}) = 1 \\ S_2(\{\alpha \wedge \neg\alpha\}) &= 1, \quad S_2(\{\beta\}) = 0, \quad S_2(\{\gamma\}) = 0, \quad S_2(\{\alpha \wedge \neg\alpha, \beta\}) = 1 \\ S_2(\{\alpha \wedge \neg\alpha, \gamma\}) &= 1, \quad S_2(\{\beta, \gamma\}) = 0, \quad S_2(\{\alpha \wedge \neg\alpha, \beta, \gamma\}) = 1 \end{aligned}$$

Then we have $S_2 < S_1$. Therefore, if we ignore the relative significance of each formula in $\Delta_1 \cup \Delta_2$, we conclude that Δ_1 is *more inconsistent than* Δ_2 . But if we consider the relative significance of each formula, then $S_2 <_P S_1$. We may say that Δ_1 is *more truly inconsistent than* Δ_2 .

The priority-based score ordering considers both the degree and significance of inconsistency in some sense. It is more strict than the score ordering. That could be shown by the following proposition.

Proposition 1. Let Δ_i and Δ_j be of the same cardinality. If S_i and S_j are the scoring functions for Δ_i and Δ_j respectively, then $S_i \leq_P S_j$ implies $S_i \leq S_j$. But the converse does not hold.

Example 4. (a counterexample for the converse) Let L be a three-level priority set. Consider $\Delta_1 = \{\neg\alpha, \alpha, \beta\}$ and $\Delta_2 = \{\alpha \wedge \neg\alpha, \beta, \gamma\}$. Assume that $\Delta_1^0 = \{\neg\alpha, \alpha\}$, $\Delta_1^1 = \{\beta\}$, $\Delta_2^0 = \{\alpha \wedge \neg\alpha\}$, $\Delta_2^1 = \{\beta\}$, and $\Delta_2^2 = \{\gamma\}$. There is no bijection from $\mathcal{P}(\Delta_1^0)$ to $\mathcal{P}(\Delta_2^0)$, thus $S_2 < S_1$ but $S_2 \not\leq_P S_1$.

3.2 Measuring Significance of Inconsistent Specification

The priority-based score ordering does not provide a direct approach for measuring the significance of inconsistency based on the priority. It just provides a basis for comparing the scoring functions under the same level of priority. As mentioned above, the priority associated with each requirements statement is some kind of abstraction of this statement's significance. We may easily think up the following intuitive assumptions: (1) the requirements statements with the same priority have the same significance; (2) any requirements statement with higher priority is more significant than all of those with lower priorities; (3) those requirements statements with higher priorities play dominant roles in measuring the significance of the inconsistencies in requirements specifications. This is the reason why we have to take the priority into account. To achieve this objective, we first introduce a priority-based cardinality vector for Δ .

Definition 5. Let L be a m -level priority set. $\forall \Delta \subseteq \mathcal{L}_{\Phi_0}$, the priority-based cardinality vector of Δ , denoted $\vec{C}(\Delta)$, is defined as $\vec{C}(\Delta) = (|\Delta^0|, \dots, |\Delta^{m-1}|)$, where $\{\Delta^0, \dots, \Delta^{m-1}\}$ is the priority-based partition of Δ under L and $|\Delta^l|$ is cardinality of Δ^l for each $l \in \{0, \dots, m-1\}$.

Example 5. Consider $\Delta = \{\alpha, \beta, \neg\beta \vee \neg\alpha, \gamma\}$. Let L be a three-level priority set. Let $\{\Delta^0, \Delta^1, \Delta^2\}$ be the priority-based partition of Δ under L , where $\Delta^0 = \{\beta, \neg\beta \vee \neg\alpha\}$, $\Delta^1 = \{\alpha\}$, and $\Delta^2 = \{\gamma\}$, then $\vec{C}(\Delta) = (2, 1, 1)$.

Definition 6. (cardinality vector ordering, $\preceq_P$) Let $\Delta \subseteq \mathcal{L}_{\Phi_0}$, $\Gamma_i, \Gamma_j \subseteq \Delta$, and L a m -level priority set. Let $\vec{C}(\Gamma_i)$ and $\vec{C}(\Gamma_j)$ be the priority-based cardinality vectors under L of Γ_i and Γ_j respectively. The cardinality vector ordering, denoted $\preceq_P$, is defined as: $\vec{C}(\Gamma_i) \preceq_P \vec{C}(\Gamma_j)$ iff $\exists 0 \leq l \leq m-1$ such that $|\Gamma_i^l| \leq |\Gamma_j^l|$ and $\forall l < k \leq m-1, |\Gamma_i^k| = |\Gamma_j^k|$. Furthermore, $\vec{C}(\Gamma_i) \prec_P \vec{C}(\Gamma_j)$ iff $\vec{C}(\Gamma_i) \preceq_P \vec{C}(\Gamma_j)$ and $\vec{C}(\Gamma_j) \not\preceq_P \vec{C}(\Gamma_i)$; $\vec{C}(\Gamma_i) = \vec{C}(\Gamma_j)$ iff $\vec{C}(\Gamma_i) \preceq_P \vec{C}(\Gamma_j)$ and $\vec{C}(\Gamma_j) \preceq_P \vec{C}(\Gamma_i)$.

In this sense, the priority-based cardinality vector $\vec{C}(\Delta)$ gives a measure of priority-based significance of Δ . The l -th component of $\vec{C}(\Delta)$ denote the number of the requirements with the l -th level of priority.

Proposition 2. Let L be a m -level priority set. Let $\Delta \subseteq \mathcal{L}_{\Phi_0}$. If $\vec{C}$ denotes the priority-based cardinality vector under L , then for $\Gamma_i, \Gamma_j \subseteq \Delta$,

$$\begin{aligned} \vec{C}(\Gamma_i \cap \Gamma_j) &\preceq_P \min_{\preceq_P}(\vec{C}(\Gamma_i), \vec{C}(\Gamma_j)) \\ \max_{\preceq_P}(\vec{C}(\Gamma_i), \vec{C}(\Gamma_j)) &\preceq_P \vec{C}(\Gamma_i \cup \Gamma_j) \end{aligned}$$

where $\min_{\preceq_P}(\vec{C}(\Gamma_i), \vec{C}(\Gamma_j)) = \vec{C}(\Gamma_i)$ if $\vec{C}(\Gamma_i) \preceq_P \vec{C}(\Gamma_j)$, or $\vec{C}(\Gamma_j)$ otherwise; $\max_{\preceq_P}(\vec{C}(\Gamma_i), \vec{C}(\Gamma_j)) = \vec{C}(\Gamma_j)$ if $\vec{C}(\Gamma_i) \preceq_P \vec{C}(\Gamma_j)$, or $\vec{C}(\Gamma_i)$ otherwise.

Now we can use the priority-based cardinality vector to describe the significance of inconsistency. Let N be a set of natural numbers, then N^m is a m -dimensional space.

Definition 7. Let L be a m -level priority set and $\Delta \subseteq \mathcal{L}_{\Phi_0}$. The priority-based significance vector for Δ under L , $\vec{V} : \mathcal{P}(\Delta) \mapsto \mathbb{N}^m$, can be defined as that for $\Gamma \in \mathcal{P}(\Delta)$,

$$\vec{V}(\Gamma) = \vec{C}(\text{CORE}(\Delta)) - \vec{C}(\text{CORE}(\Delta - \Gamma))$$

If we use $V^l(\Gamma)$ to denote $|\text{CORE}(\Delta)|^l - |\text{CORE}(\Delta - \Gamma)|^l$ for each $l \in \{0, \dots, m-1\}$, then $\vec{V}(\Gamma) = (V^0(\Gamma), \dots, V^{m-1}(\Gamma))$.

Intuitively, for $\Gamma \in \mathcal{P}(\Delta)$, $\vec{V}(\Gamma)$ captures the reduction of the significance of those “problematical” statements in Δ after Γ were removed from Δ . Based on $\vec{V}$, we may introduce another ordering relation, the *priority-based significance ordering*, for comparing the significance of inconsistencies in specifications.

Definition 8. (*priority-based significance ordering*, $\preceq_P^S$) Let L be a m -level priority set. Assume that Δ_i and Δ_j are of the same cardinality. Let $\vec{V}_i$ and $\vec{V}_j$ be the priority-based significance vectors under L for Δ_i and Δ_j respectively. Then $\vec{V}_i \preceq_P^S \vec{V}_j$ holds iff there is a bijection $f : \mathcal{P}(\Delta_i) \mapsto \mathcal{P}(\Delta_j)$ such that the following condition can be satisfied:

$$\forall \Gamma \in \mathcal{P}(\Delta_i), \vec{V}_i(\Gamma) \preceq_P \vec{V}_j(f(\Gamma))$$

We call $\preceq_P^S$ the priority-based significance ordering. Furthermore, $\vec{V}_i \prec_P^S \vec{V}_j$ iff $\vec{V}_i \preceq_P^S \vec{V}_j$ and $\vec{V}_j \not\preceq_P^S \vec{V}_i$; $\vec{V}_i \simeq_P^S \vec{V}_j$ iff $\vec{V}_i \preceq_P^S \vec{V}_j$ and $\vec{V}_j \preceq_P^S \vec{V}_i$. We say the inconsistency in Δ_j is **more significant than** that in Δ_i iff $\vec{V}_i \preceq_P^S \vec{V}_j$.

Let us give an example to illustrate how to compare two inconsistent specifications from the significance of inconsistency via the priority-based significance ordering.

Example 6. Consider $\Delta_1 = \{\alpha, \neg\alpha\}$ and $\Delta_2 = \{\beta, \neg\beta\}$. Let L be a three-level priority set. Assume that $\Delta_1^0 = \{\alpha\}$, $\Delta_1^1 = \{\neg\alpha\}$, $\Delta_2^1 = \{\beta\}$, and $\Delta_2^2 = \{\neg\beta\}$. If $\vec{V}_1$ and $\vec{V}_2$ are priority-based significance vectors under L for Δ_1 and Δ_2 respectively, then

$$\begin{aligned} \vec{V}_1(\Delta_1) &= (1, 1, 0), \vec{V}_1(\{\alpha\}) = (1, 1, 0), \vec{V}_1(\{\neg\alpha\}) = (1, 1, 0) \\ \vec{V}_2(\Delta_2) &= (0, 1, 1), \vec{V}_2(\{\beta\}) = (0, 1, 1), \vec{V}_2(\{\neg\beta\}) = (0, 1, 1) \end{aligned}$$

Therefore, $\vec{V}_1 \prec_P^S \vec{V}_2$, and we conclude that the inconsistency in Δ_2 is **more significant than** that in Δ_1 . However, if we apply the scoring function, S , to Δ_1 and Δ_2 , we can not tell the difference of their inconsistencies.

Proposition 3. Let L be a m -level priority set. Let $\Delta_i, \Delta_j \subseteq \mathcal{L}_{\Phi_0}$. If $\vec{V}_i$ and $\vec{V}_j$ are the priority-based significance vectors under L for Δ_i and Δ_j respectively, then $\vec{V}_i \preceq_P^S \vec{V}_j$ implies $\vec{C}(\text{CORE}(\Delta_i)) \preceq_P \vec{C}(\text{CORE}(\Delta_j))$.

The priority-based significance vector provides a concise means for articulating the significance of inconsistency in specifications. For inconsistent specifications, it is easy to get the following relation between the degree and significance of inconsistency.

Proposition 4. Let L be a m -level priority set and $\Delta \subseteq \mathcal{L}_{\Phi_0}$. Let $\vec{0}$ be a zero vector. If S is the scoring function for Δ and $\vec{V}$ the priority-based significance vector for Δ under L , then for $\Gamma \subseteq \Delta$, $S(\Gamma) = 0$ iff $\vec{V}(\Gamma) = \vec{0}$.

3.3 Priority-Based Scoring Vector

As mentioned earlier, the scoring function S for Δ reveals the degree of inconsistency arising in Δ , while the priority-based significance vector $\vec{V}$ for Δ measures the significance of inconsistency. We also give two ordering relations for comparing inconsistent specifications from the perspectives of the degree and the significance of inconsistency, respectively. Actually, in many cases, we need to consider both of them. In software engineering, we might define this integrated measure by combining the scoring function with the priority-based significance vector.

Definition 9. Let L be a m -level priority set and $\Delta \subseteq \mathcal{L}_{\Phi_0}$. Let $\vec{V}$ be the priority-based significance vectors under L for Δ . The priority-based scoring vector for Δ under L , $\vec{S}_P: \mathcal{P}(\Delta) \mapsto \mathbb{N}^{m+1}$, can be defined as that for $\Gamma \in \mathcal{P}(\Delta)$,

$$\vec{S}_P(\Gamma) = (V^0(\Gamma), \dots, V^{m-1}(\Gamma), S(\Gamma))$$

Actually, for $\Gamma \in \mathcal{P}(\Delta)$, the priority-based scoring vector for Δ consists of $\vec{V}(\Gamma)$ concatenated with value $S(\Gamma)$. It could be viewed as the integrated measure of inconsistent information of Δ that would be reduced if Γ were removed from Δ . Furthermore, we can compare these inconsistent specifications using the priority-based scoring vector for each specification from an integrated view.

Definition 10. (scoring vector ordering, $\preceq_P^*$) Let $\Delta \subseteq \mathcal{L}_{\Phi_0}$, $\Gamma_i, \Gamma_j \subseteq \Delta$, and L a m -level priority set. Let $\vec{S}_P(\Gamma_i)$ and $\vec{S}_P(\Gamma_j)$ be the priority-based scoring vectors under L of Γ_i and Γ_j respectively. The scoring vector ordering, denoted $\preceq_P^*$, is defined as: $\vec{S}_P(\Gamma_i) \preceq_P^* \vec{S}_P(\Gamma_j)$ iff $\vec{V}(\Gamma_i) \preceq_P \vec{V}(\Gamma_j)$ and $S(\Gamma_i) \leq S(\Gamma_j)$. Furthermore, $\vec{S}_P(\Gamma_i) \prec_P^* \vec{S}_P(\Gamma_j)$ iff $\vec{S}_P(\Gamma_i) \preceq_P^* \vec{S}_P(\Gamma_j)$ and $\vec{S}_P(\Gamma_j) \not\preceq_P^* \vec{S}_P(\Gamma_i)$; $\vec{S}_P(\Gamma_i) = \vec{S}_P(\Gamma_j)$ iff $\vec{S}_P(\Gamma_i) \preceq_P^* \vec{S}_P(\Gamma_j)$ and $\vec{S}_P(\Gamma_j) \preceq_P^* \vec{S}_P(\Gamma_i)$;

Definition 11. (priority-based score vector ordering, $\preceq_P^E$) Let L be a m -level priority set. Assume that Δ_i and Δ_j are of the same cardinality. Let $\vec{S}_{P_i}$ and $\vec{S}_{P_j}$ be the priority-based scoring vectors under L for Δ_i and Δ_j , respectively. $\vec{S}_{P_i} \preceq_P^E \vec{S}_{P_j}$ holds iff there is a bijection $f: \mathcal{P}(\Delta_i) \mapsto \mathcal{P}(\Delta_j)$ such that the following condition can be satisfied: $\forall \Gamma \in \mathcal{P}(\Delta_i)$, $\vec{S}_{P_i}(\Gamma) \preceq_P^* \vec{S}_{P_j}(f(\Gamma))$. We call $\preceq_P^E$ the priority-based score vector ordering. Furthermore, $\vec{S}_{P_i} \prec_P^E \vec{S}_{P_j}$ iff $\vec{S}_{P_i} \preceq_P^E \vec{S}_{P_j}$ and $\vec{S}_{P_j} \not\preceq_P^E \vec{S}_{P_i}$; $\vec{S}_{P_i} \simeq_P^E \vec{S}_{P_j}$ iff $\vec{S}_{P_i} \preceq_P^E \vec{S}_{P_j}$ and $\vec{S}_{P_j} \preceq_P^E \vec{S}_{P_i}$. We say Δ_j is **more essentially inconsistent** than Δ_i iff $\vec{S}_{P_i} \preceq_P^E \vec{S}_{P_j}$.

Proposition 5. Let L be a m -level priority set, and $\Delta_i, \Delta_j \subseteq \mathcal{L}_{\Phi_0}$. Let S_i and S_j be the scoring functions for Δ_i and Δ_j respectively. If $\vec{S}_{P_i}$ and $\vec{S}_{P_j}$ are the priority-based scoring vectors under L for Δ_i and Δ_j respectively, then $\vec{S}_{P_i} \preceq_P^E \vec{S}_{P_j}$ implies $\vec{V}_i \preceq_P^S \vec{V}_j$ and $S_i \leq S_j$.

Let us look at the following example to see how to compare two inconsistent specifications from two different perspectives, i.e. the degree and the significance of inconsistency.

Example 7. Consider $\Delta_1 = \{\alpha, \beta, \neg\alpha, \neg\beta\}$ and $\Delta_2 = \{\alpha, \gamma, \neg\alpha, \neg\gamma\}$. Let L be a three-level priority set. And let $\{\Delta_1^0, \Delta_1^1, \Delta_1^2\}$ and $\{\Delta_2^0, \Delta_2^1, \Delta_2^2\}$ be the priority-based partitions under L of Δ_1 and Δ_2 , respectively, where $\Delta_1^0 = \{\alpha\}$, $\Delta_1^1 = \{\neg\beta\}$, $\Delta_1^2 = \{\neg\alpha, \beta\}$, $\Delta_2^0 = \{\alpha, \neg\alpha\}$, $\Delta_2^1 = \{\neg\gamma\}$, and $\Delta_2^2 = \{\gamma\}$.

If S_1 and S_2 are the scoring functions for Δ_1 and Δ_2 respectively, then $S_1 \simeq S_2$. Therefore, we may say Δ_1 is **as inconsistent as** Δ_2 from the perspective of the **degree of inconsistency**. On the other hand, from the perspective of the **significance of inconsistency**, we may say the inconsistency in Δ_1 is **more significant than** that in Δ_2 since $\vec{V}_2 \prec_P^S \vec{V}_1$, where $\vec{V}_1$ and $\vec{V}_2$ are the priority-based significance vectors under L for Δ_1 and Δ_2 respectively. Furthermore, if $\vec{S}_{P1}$ and $\vec{S}_{P2}$ are the priority-based scoring vectors under L for Δ_1 and Δ_2 respectively, we have $\vec{S}_{P2} \prec_P^E \vec{S}_{P1}$. That is, from the integrative perspective, Δ_1 is **more essentially inconsistent than** Δ_2 .

However, as illustrated by the following propositions, the priority-based scoring vector is also a concise and yet expressive articulation of the inconsistencies that arise in requirements specifications from both the severity and significance.

Proposition 6. Let $\Delta \subseteq \mathcal{L}_{\Phi_0}$ and L a m -level priority set. If $\vec{S}_P$ is the priority-based scoring vector under L for Δ , then

$$\vec{S}_P(\text{FREE}(\Delta)) = \vec{0}, \vec{S}_P(\text{CORE}(\Delta)) = (|\text{CORE}(\Delta)^0|, \dots, |\text{CORE}(\Delta)^{m-1}|, |\text{MI}(\Delta)|);$$

Proposition 7. Let $\Delta \subseteq \mathcal{L}_{\Phi_0}$ and L a m -level priority set. If $\vec{S}_P$ is the priority-based scoring vector under L for Δ , then for $\alpha \in \Delta$,

$$\alpha \in \text{FREE}(\Delta) \text{ iff } \vec{S}_P(\{\alpha\}) = \vec{0}; \text{ and } \alpha \in \text{CORE}(\Delta) \text{ iff } \vec{0} \prec_P^* \vec{S}_P(\{\alpha\})$$

Proposition 8. Let $\Delta \subseteq \mathcal{L}_{\Phi_0}$ and L a m -level priority set. If $\vec{S}_P$ is the priority-based scoring vector under L for Δ , then for $\Gamma_i, \Gamma_j \subseteq \Delta$,

$$\begin{aligned} \vec{S}_P(\Gamma_i \cap \Gamma_j) &\preceq_P^* \min_{\preceq_P^*}(\vec{S}_P(\Gamma_i), \vec{S}_P(\Gamma_j)) \\ \max_{\preceq_P^*}(\vec{S}_P(\Gamma_i), \vec{S}_P(\Gamma_j)) &\preceq_P^* \vec{S}_P(\Gamma_i \cup \Gamma_j) \end{aligned}$$

where $\min_{\preceq_P^*}(\vec{S}_P(\Gamma_i), \vec{S}_P(\Gamma_j)) = \vec{S}_P(\Gamma_i)$ if $\vec{S}_P(\Gamma_i) \preceq_P^* \vec{S}_P(\Gamma_j)$, or $\vec{S}_P(\Gamma_j)$ otherwise; $\max_{\preceq_P^*}(\vec{S}_P(\Gamma_i), \vec{S}_P(\Gamma_j)) = \vec{S}_P(\Gamma_j)$ if $\vec{S}_P(\Gamma_i) \preceq_P^* \vec{S}_P(\Gamma_j)$, or $\vec{S}_P(\Gamma_i)$ otherwise.

Proposition 9. Let L be a m -level priority set and $\Delta_i, \Delta_j \subseteq \mathcal{L}_{\Phi_0}$. If $\vec{S}_{P_i}$ and $\vec{S}_{P_j}$ are the priority-based scoring vectors under L for Δ_i and Δ_j respectively, then $\vec{S}_{P_i} \preceq_P^E \vec{S}_{P_j}$ implies $|\text{FREE}(\Delta_i)| \geq |\text{FREE}(\Delta_j)|$. But the converse does not hold.

Example 8. (a counterexample for the converse). Consider $\Delta_1 = \{\alpha, \neg\alpha, \beta\}$ and $\Delta_2 = \{\alpha \wedge \neg\alpha, \beta, \gamma\}$. Let L be a three-level priority set. Let $\Delta_1^0 = \{\alpha, \neg\alpha\}$, $\Delta_1^1 = \{\beta\}$, $\Delta_1^2 = \{\beta, \gamma\}$, and $\Delta_2^2 = \{\alpha \wedge \neg\alpha\}$. So, $|\text{FREE}(\Delta_2)| > |\text{FREE}(\Delta_1)|$, but $\vec{S}_{P2} \not\preceq_P^E \vec{S}_{P1}$.

3.4 Case Study

Example 9. Let L be a three-level priority set. Consider a scenario in a close residential area management system. Developer A , who is in charge of gathering information about vehicle's application for entrance, supplies the "essential" requirements as follows: *The vehicles with authorization (Auth) of residential area can enter (Enter) the area; The vehicles without authorization can not enter.* He also gathers a legal rule about *fire engine* as follows: *the fire engine (Fire) can enter the area without authorization.* If we use Δ_A to represent the specification from A , then Δ_A contains:

$$Fire(v) \rightarrow Enter(v), Fire(v) \rightarrow \neg Auth(v), \neg Auth(v) \rightarrow \neg Enter(v), Fire(v)$$

The priority-based partition of Δ_A is: $\Delta_A^2 = \Delta_A$.

Developer B , who is in charge of managing renting garages, supplies the "essential" requirements as follows: *A garage is available (Available) if it is unoccupied (Unoccupied).* A further "conditional" requirements is: *If a garage should be repaired (Repaired), then it is not available; If a garage can be repaired, then it is unoccupied.* Then specification Δ_B contains the following statements:

$$Unoccupied(a) \rightarrow Available(a), Repaired(a) \rightarrow \neg Available(a), \\ Repaired(a) \rightarrow Unoccupied(a), Repaired(a)$$

The priority-based partition of Δ_B is: $\Delta_B^1 = \{Repaired(a) \rightarrow \neg Available(a), Repaired(a) \rightarrow Unoccupied(a)\}$, $\Delta_B^2 = \Delta_B - \Delta_B^1$. Obviously, both Δ_A and Δ_B are inconsistent. If $\vec{S}_{PA}$ and $\vec{S}_{PB}$ are the priority-based scoring vectors under L of Δ_A and Δ_B , respectively, then $\vec{S}_{PB} \prec_P^E \vec{S}_{PA}$. It signifies that the developers should give Δ_A priority based on integrated measure of inconsistency. However, if we use the scoring functions in [8], we can't distinguish the inconsistencies of the two specifications.

The approach could also be applied to other scenarios such as negotiation between agents and the comparison of heterogeneous sources of information, since the relative importance of knowledge in certain scenario may affect the measure of inconsistency, especially in competitive negotiation.

Example 10. Consider the competition of Japan and China for Russia's oil and gas pipeline routes. Generally, large amount of the export of oil, dominant role in export, the length and cost of routes are viewed as factors that may contribute to Russia's choice of routes.

Let Δ_R be Russia's perspective about routes. $\Delta_R = \{short, cheap, large, dominant\}$.

Let the descriptions of routes proposed by China and Japan be represented by Δ_C and Δ_J respectively. $\Delta_C = \{short, cheap, \neg large, \neg dominant\}$, $\Delta_J = \{\neg short, \neg cheap, large, dominant\}$.

Hence, the negotiation between Russia and China is captured by Δ_{RC} .

$$\Delta_{RC} = \{short, cheap, \neg large, \neg dominant, large, dominant\}$$

The negotiation between Russia and Japan is captured by Δ_{RJ} .

$$\Delta_{RJ} = \{short, cheap, large, dominant, \neg short, \neg cheap\}$$

Let L be a three-level priority set. As for the items that contribute to Russia's choice of routes, large amount of the export of oil and dominant role in export are essential factors, while the length and cost of route are significant but less essential factors. Therefore, the priority-based partition of Δ_{RC} is captured as follows:

$$\Delta_{RC}^1 = \{short, cheap\}, \Delta_{RC}^2 = \{\neg large, \neg dominant, large, dominant\}$$

The priority-based partition of Δ_{RJ} is captured as follows:

$$\Delta_{RJ}^1 = \{short, cheap, \neg short, \neg cheap\}, \Delta_{RJ}^2 = \{large, dominant\}$$

If $\vec{S}_{PRC}$ is the priority-based scoring vector for Δ_{RC} and $\vec{S}_{PRJ}$ is the priority-based scoring vector for Δ_{RJ} , then $\vec{S}_{PRJ} \prec_P^E \vec{S}_{PRC}$. It implies that Japanese proposal of pipeline route is more attractive to Russia than that of China.

4 Conclusions

In terms of the relative priorities of requirements statements, this paper presents a set of priority-based strategies to measure the inconsistencies arising in requirements specifications. First, the priority-based score ordering is proposed to compare the degree of inconsistencies under the same level of priority. And then the priority-based significance vector is given to assess the significance of inconsistency. And finally, the priority-based score vector ordering, which is based on the priority-based scoring vector, is defined to compare the inconsistent specifications from an integrated view, i.e. according to both the degree and the significance of inconsistency.

Measuring inconsistency is still an important issue in developing requirements specifications as well as intelligent systems. Some recent techniques for measuring inconsistent information have been reviewed in [9]. The overwhelming majority of these techniques focus on different measures of the degree of inconsistency [10–13]. At present, the scoring function [8] is one of the most appropriate tools for summarizing the degree of inconsistency. However, researchers have begun to study the significance of inconsistency. For example, Hunter provided a approach for measuring the significance of inconsistency arising in QC models [5]. This approach is based on specifying the relative significance of incoherent models using additional information, encoded as a mass assignment. But, the priority of a requirements statement is just an imprecise measure of relative importance. It is difficult to determine the precise measure of relative significance for each statement during the requirements stage in many cases. That might be the main obstacles in putting this approach into practical applications.

In contrast, the approach described in this paper uses the priority-based significance vector to measure the significance of inconsistency. The priority-

based partition of specification is available during the requirements stage [14]. It could be viewed as a partition of requirements by relative importance and urgency. Moreover, in general cases, the priority-based partition of specification is accepted by all stakeholders. That is, each stakeholder gives the same meaning of the same level of significance. It shows that this approach may be more feasible to requirements engineering practices.

Acknowledgements

This work was partly supported by the National Natural Science Foundation of China (No.60233010 and No.60496324), the National Key Research and Development Program (Grant No. 2002CB312004) of China, the Knowledge Innovation Program of the Chinese Academy of Sciences and the British Royal Society China-UK Joint Project. We are grateful to the reviewers for their constructive comments, which helped to improve our work.

References

1. Nuseibeh, B., Easterbrook, S., Russo, A.: Leveraging inconsistency in software development. *IEEE Computer* **33** (2000) 24–29
2. Nuseibeh, B., S.Easterbrook, A.Russo: Making inconsistency respectable in software development. *Journal of Systems and Software* **58** (2001) 171–180
3. Easterbrook, S., M.Chechik: 2nd international workshop on living with inconsistency. *Software Engineering Notes* **26** (2001) 76–78
4. Hunter, A., B.Nuseibeh: Managing inconsistent specification. *ACM Transactions on Software Engineering and Methodology* **7** (1998) 335–367
5. A.Hunter: Evaluating the significance of inconsistency. In: *Proceedings of the International Joint Conference on AI (IJCAI'03)*. (2003) 468–473
6. Wiegers, K.E.: *Software Requirements*, 2nd ed. Microsoft Press (2003)
7. 830–1998, I.S.: *IEEE Recommended Practice for Software Requirements Specifications*. Los Alamitos, CA:IEEE Computer Society Press (1998)
8. A.Hunter: Logical comparison of inconsistent perspectives using scoring functions. *Knowledge and Information Systems Journal* **6** (2004) 528–543
9. Hunter, A., Konieczny, S.: Approaches to measuring inconsistent information. In: *Inconsistency Tolerance, LNCS*. Volume 3300. Springer-Verlag (2004) 189–234
10. Hunter, A.: Measuring inconsistency in knowledge via quasi-classical models. In: *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI'2002)*, MIT Press (2002) 68–73
11. D.Dubois, Lang, J., Prade, H.: Possibilistic logic. In: *Handbook of logic in artificial intelligence and logic programming*. Oxford University Press (1994) 439–531
12. S.Benferhat, D.Dubois, S., H.Prade: Encoding information fusion in possibilistic logic:a general framework for rational syntactic merging. In: *Proceedings of ECAI'2000*, IOS Press (2000) 3–7
13. S. Konieczny, Lang, J., P.Marquis: Quantifying information and contradiction in propositional logic through test actions. In: *Proceedings of IJCAI2003*, Morgan Kaufmann (2003) 106–111
14. K.Wiegers: First things first:prioritizing requirements. *Software Development* **7** (1999) 48–53

Belief Revision of GIS Systems: The Results of REV!GIS

Salem Benferhat³, Jonathan Bennaïm⁴, Robert Jeansoulin²,
Mahat Khelfallah², Sylvain Lagrue³, Odile Papini¹,
Nic Wilson⁵, and Eric Würbel¹

¹ LSIS-CNRS, université du Sud Toulon -Var. BP 132. 83957 La Garde Cedex France
papini, wurbel@univ-tln.fr

² LSIS-CNRS, CMI technopôle de Château Gombert. 13353.,
Marseille cedex 13. France
jeansoulin, mahat@cmi.univ-mrs.fr

³ CRIL-CNRS, université d'Artois. Rue Jean Souvraz. 62307 Lens Cedex. France
benferhat, lagrue@cril.univ-artois.fr

⁴ LIF-CNRS, CMI technopôle de Château Gombert. 13353.,
Marseille cedex 13. France
jbennaïm@lif.univ-mrs.fr

⁵ University College Cork. Cork. Ireland
n.wilson@4C.ucc.ie

Abstract. This paper presents a synthesis of works performed on the practical tractability of revision on geographic information within the european REV!GIS project¹. It surveys different representations of the revision problem as well as different implementations of the adopted strategy: Removed Set Revision (RSR). A comparison of the representation formalisms is provided, a formal and an experimental comparison is conducted on the various implementations on real scale applications in the context of GIS.

1 Introduction

One of the aim of the REV!GIS project was to investigate how artificial intelligence tools could be used to perform revision in the case of spatially referenced information. Within this project different formalisms have been proposed for representing geographic information with a special focus on practical tractability for symbolic change operations. The present paper provides a synthesis of works done during the project. It presents a comparison and a discussion on the different symbolic formalisms to represent geographic information as well as on the various implementations of the Removed Set Revision (RSR) experimented on real scale applications.

¹ This work was supported by European Community project IST-1999-14189 REVIGIS.

The paper is organized as follows. After a short reminder on the specificity of geographic information in Section 2 and on RSR in Section 3, we survey and discuss in Section 4 different symbolic representations of the revision problem on geographic information. For each representation, we briefly recall in Section 5 adjustments of existing strategies, taking advantage of the nature of geographic information, to perform revision. The results of the implementations of the different revision approaches are discussed in Section 6 before concluding in Section 7.

2 Geographic Information

Geographic information is made of numerous items gathered from different sources (instruments, surveys, images), and recorded as values on some specified domain, after a theory-based interpretation. Therefore, all these items can be qualified by some confidence level, depending on the supposed fitness of the interpretative process, for the actual situation, or some preference which expresses the subjective vision of what the world should be. These items of information are uncertain, incomplete or inaccurate, and they can conflict with each other. Hence they may require corrective operations: revision, update or fusion according to the context.

In terms of formal representation, the huge amount of data raises tractability problems. For instance, a small problem involving a hundred spatial regions, with ten attributes defined on finite domains of low cardinality, is represented by about one hundred thousand propositional clauses. Another problem is that what is observed differs from the variables expected in the model built by the user: we need to apply (uncertain) inference rules, for deriving such variables from several observed variables, and inconsistency can result. The size and the variety of the data seem to prevent any reasonable implementation of belief change operations when reasoning with geographic information.

Three main considerations can help us. First, the information relies on space where everything get situated, overlaps or coexists according to definite relations, topologic, metric or temporal: these constraints can reduce the size of the problem. Second, in case of inconsistency, the conflicts are local, and their detection and resolution can be carried out over restricted parts of the data set. Finally, the spatial relations translate into a particular syntax, which allows us to adapt existing algorithms into faster versions. In this work we consider the case, particular but very frequent, where the information is linked to non ubiquitous spatial locations, (1) either distinct, as a set of non overlapping spots, (2) or elements of a space partition (full coverage, no intersection). This limits the topology to only three relations: same, different, adjacent. The case of partially overlapping zones, or with undetermined limits, should be treated in a separate work, for further integration. We use the general term of parcel for referring to such locations, and we use the capital letters A, B, C etc. to denote them. Throughout the paper, the following two examples, extracted from real scale applications, will be used for illustration and comparison.

E 1. *F* . The aim is to assess the water height in a flooded valley, which is segmented into parcels. We assess a minimum/maximum interval of water height for each parcel, where this height can be considered as constant. We have two sources of information (aside from the geographical layout): (1) a set of hydraulic relations between neighbouring parcels; (2) a set of initial assessments of minimal and/or maximal submersion heights for some parcels, for more details see [13] and [18]. For the illustration, we consider only 3 parcels, and 2 flow relations: from *A* to *B* and from *A* to *C*. The sampled observation domain is $D = \{1, 2, 3, 4\}$, and the assessments are $\{A : 2, B : 3, C : 4\}$, for the maximum submersion height and $\{A : 1, B : 1, C : 3\}$, for the minimum submersion height.

E 2. *B* . The aim is to find the best location(s) for building a construction according to some constraints, which can be partially or totally ordered [14]. We consider 3 parcels, and 3 constraints: (C_1) to be near a fire hydrant, the domain for the distance being $D_d = \{very\ close, close, far, very\ far\}$, (C_2) to be far from a street intersection, same domain D_d , (C_3) to be built-free, in the domain $D_b = \{yes, no, may\ be\}$.

3 Removed Set Revision

We briefly recall RSR. We first transform the initial set of formulas in CNF for dealing with clauses. Let K and A be finite sets of clauses, the method focuses on the minimal subsets of clauses to remove from K , called [12], in order to restore consistency of $K \cup A$. More formally:

Definition 1. Let K and A be two sets of clauses such that $K \cup A$ is inconsistent. R a subset of clauses of $K \cup A$, is a removed set iff (i) $R \subseteq K$; (ii) $(K \cup A) \setminus R$ is consistent; iii) $\forall R' \subseteq K$, if $(K \cup A) \setminus R'$ is consistent then $|R| < |R'|$ ².

Definition 2. Let K , K' and A be finite sets of clauses. $K \circ_R A = \{K' \cup A, \text{ such that } K' = K \setminus R, \text{ where } R \text{ is a removed set}\}$.

The removed sets define a family of revision operations $\circ_R$ which satisfy the AGM postulates and it can be checked that if R is a removed set then $(K \cup A) \setminus R$ is a so-called cardinality-based maximal consistent subbase of $(K \cup A)$ [2], [6] [10].

4 Representing Revision on Geographic Information

4.1 Propositional Clausal Form Representation

The most basic representation is the propositional clausal form. Representing geographic information with propositional calculus takes advantage of the simplicity of expression of this language and, from a computational point of view,

² $|R|$ denotes the number of clauses of R .

takes advantage of the decidability of this logic. Moreover most of the change operations defined in the area of knowledge representation are defined in propositional calculus.

The nature of geographic information knowledge leads to a special propositional clausal form representation. Any proposition refers to some phenomenon linked to one parcel, and we use the propositional variables a^k to denote the propositions concerning a phenomenon k which are linked to the parcel A . The arbitrary rank k varies within a finite set.

For measurable observations, the propositions represent the numerical response of some phenomenon against a finite, sampled, domain of values $D = \{v_1, \dots, v_n\}$, the notation becomes a_i^k , $1 \leq i \leq n$, to remind us about the phenomenon (ranked by k) and the questioned value (phenomenon $k = v_i$). Therefore, the representation of a particular set of observations is encoded by the n -ary clause $a_1^k \vee a_2^k \vee \dots \vee a_n^k$ and $n(n-1)/2$ binary mutual exclusive clauses: $\neg a_i^k \vee \neg a_j^k$, $1 \leq i \leq n$ and $j > i$. The binary relations between measures, for instance a simple inequality, a linear equation, or a more complex mathematical formula, can be represented by couples of forbidden values which are encoded in binary negative clauses. From now on, S^O , S^D and S^C denote the set of clauses representing the observations, the domain and the relations respectively.

Revision of a Set of Clauses by a Set of Clauses. The revision problem amounts to revising the set of clauses S^O by the set of clauses $S^D \cup S^C$.

Example 3. In the flooding application, for each parcel A , B and C , we define the propositional variables a^+ and a^- , b^+ and b^- , c^+ and c^- for maximal and minimal submersion height respectively³. These variables are defined on a domain $D = \{1, 2, 3, 4\}$. The set of clauses S^D representing the finite domain consists, for each variable, in one 4-ary clause and 6 binary negatives clauses. For instance, the clauses corresponding to the variable a^+ are the 4-ary clause $a_1^+ \vee a_2^+ \vee a_3^+ \vee a_4^+$ and the 6 binary negatives clauses $\neg a_i^+ \vee \neg a_j^+$, $i \in D$, $i \neq j$. The set of clauses representing the observations is $S^O = \{a_1^-, a_2^-, b_1^+, b_3^+, c_3^-, c_4^+\}$. The set of clauses S^C representing the flow relations between parcels is the set of forbidden couples for the inequalities representing the flow relations. For example the relation $a^+ \geq b^+$ is represented by the clauses $\neg a_1^+ \vee \neg b_2^+$, $\neg a_1^+ \vee \neg b_3^+$, $\neg a_1^+ \vee \neg b_4^+$, $\neg a_2^+ \vee \neg b_3^+$, $\neg a_2^+ \vee \neg b_4^+$, $\neg a_3^+ \vee \neg b_4^+$. The revision problem amounts to revising the set of clauses S^O by the set of clauses $S^D \cup S^C$.

Translation Into a Satisfiability Problem. We use the transformation proposed by De Kleer for ATMS [9]. Each clause c of S^O is replaced by the formula $\phi_c \rightarrow c$, where ϕ_c is a new variable, called hypothesis variable. If ϕ_c is assigned true then $\phi_c \rightarrow c$ is true iff c is true, this enforces c . On contrast if ϕ_c is assigned false then $\phi_c \rightarrow c$ is true whatever the truth value of c , the clause c is ignored. Let us denote $\mathcal{H}(S^O)$ the transformed set. The revision problem corresponds to the satisfiability of the set of clauses $\mathcal{H}(S^O) \cup (S^D \cup S^C)$ with some conditions on hypothesis variables ϕ_c according to the revision method, for instance minimizing the number of falsified hypothesis variables ϕ_c .

³ For a better understanding we denote a^+ instead of a^1 and a^- instead of a^2 .

Translation Into a ROBDD. A set of clauses can be compactly encoded in a Reduced Ordered Binary Decision Diagram (ROBDD), which is a labeled acyclic directed graph [4]. Using the transformation $\mathcal{H}$ defined above, the revision problem amounts to find the shortest path in the ROBDD corresponding to the set of clauses $\mathcal{H}(S^O) \cup (S^D \cup S^C)$ as described in [17].

4.2 Logic Programming Representation

Standard Logic Programming. In a standard logic programming approach (PROLOG or DATALOG), the observations are represented by facts, and relations between observations are represented by facts and rules. Inconstency rules have to be explicitly provided. The revision problem amounts to defining rules involving the facts representing the observations to solve the inconsistencies.

Logic Programming with Answer Set Semantics. In this approach we directly translate the revision problem into a logic program with answer set semantics (ASP) [3]. This translation is suitable for Removed Set Revision. Firstly, for each clause c of S^O , we introduce a new atom in V the set of atoms occurring in $S^O \cup (S^D \cup S^C)$. We then construct a logic program $P_{S^O \cup (S^D \cup S^C)}$ whose answer sets correspond to subsets R of S^O such that $(S^O \cup (S^D \cup S^C)) \setminus R$ is consistent. This construction stems from the enumeration of interpretations of V and a progressive elimination of interpretations. For more details see [3].

4.3 Constraint Satisfaction Problem Representation

Let $\mathcal{X}$ be a set of variables and $\mathcal{D}$ a set of sampled, hence discrete domains. The observations and the relations are encoded by the following CSPs denoted by $\mathcal{P}_O = \{\mathcal{X}, \mathcal{D}, \mathcal{C}_O, \mathcal{R}_O\}$ and $\mathcal{P}_C = \{\mathcal{X}, \mathcal{D}, \mathcal{C}_C, \mathcal{R}_C\}$ respectively where $\mathcal{C}_O$ and $\mathcal{C}_C$ are the constraints on the variables and $\mathcal{R}_O$ and $\mathcal{R}_C$ the relations [17].

4.4 Linear Constraint Representation

When the variable domain is continuous and the relations between the variables can be represented by linear constraints, another representation stems from the Logic of Linear Constraints (LLC) [15]. Within this framework, a variable X_i is associated with each parcel i . The measures and observations are given as intervals $[l_i, u_i]$ of possible values for variables X_i , where l_i and u_i are real scalars. The set of measures and observations is represented by a set $\mathcal{L}^O$ of linear constraints of the form $X_i \geq l_i$, called the $\mathcal{L}^O$ constraints or constraints of the form $X_i \leq u_i$, called the $\mathcal{L}^O$ constraints. The set of relations between variables is represented by a set $\mathcal{L}^C$ of linear constraints of the form $X_i - X_j \geq a_{ij}$ called $\mathcal{L}^C$ constraints, where a_{ij} is a real scalar. If $D = [L, U]$ is the domain of variables, then the variable domain represented by the set of linear constraints $\mathcal{L}^D$ consists, for each parcel i , in the constraint $L \leq X_i \leq U$. If the set of constraints $\mathcal{L}^O \cup \mathcal{L}^C \cup \mathcal{L}^D$ is inconsistent, the revision amounts to identifying constraints of $\mathcal{L}^O$ whose removal is sufficient to restore the consistency.

In the following section we present how revision is performed according to the different representations.

5 Performing Revision on Geographic Information

Revision in the framework of geographic information has been performed according to RSR strategy with suitable adjustments in order to take advantage of the spatial knowledge representation [7].

5.1 Removed Set Revision Using Hitting Sets (REM Algorithm)

The direct computation of removed sets consists in removing a clause from each element of the collection of minimal inconsistent subsets of $S^O \cup S^D \cup S^C$ without listing all elements of this collection. This strategy stems from the notion of minimal hitting set which is a minimal set of clauses that intersects with each minimal inconsistent subset. R is a removed set iff it is a minimal hitting set of the collection $\mathcal{I}(S^O \cup S^D \cup S^C)$ of the inconsistent subsets of $S^O \cup S^D \cup S^C$. This is described in [18] and [19].

5.2 Removed Set Revision as a SAT Problem

Using the representation proposed in 4.1 the Removed Set Revision of S^O by $S^D \cup S^C$ corresponds to the problem of looking for a model of the set of clauses $\mathcal{H}(S^O) \cup (S^D \cup S^C)$ which minimizes the number of falsified hypothesis variables ϕ_c . This leads to the definition of a preference relation between interpretations stemming from the number of hypothesis variables they falsify, denoted by H_{So} -preference. Let M be a model of $\mathcal{H}(S^O) \cup (S^D \cup S^C)$ generated by a removed set R , then R is a removed set iff M is a H_{So} -preferred model of $\mathcal{H}(S^O) \cup (S^D \cup S^C)$ [3]. Performing Removed Set Revision amounts to looking for the H_{So} -preferred model of $\mathcal{H}(S^O) \cup (S^D \cup S^C)$. This can be achieved using a SAT-solver. In order to compare different implementations of Removed Set Revision we used the SAT-solver MiniSat[5].

5.3 Removed Set Revision with ROBDD

As shown in section 4, we can build a ROBDD representing $\mathcal{H}(S^O) \cup (S^D \cup S^C)$. In this context, minimizing the number of clauses to remove from S^O amounts to minimizing the number of hypothesis variables ϕ_c assigned false, see [19].

5.4 Revision in the Framework of Constraint Satisfaction Problems

In section 4, we described how to represent geographic information using the CSP framework. In this context, a revision situation arises when the problem $\mathcal{P}_{OUC} = \{\mathcal{X}, \mathcal{D}, \mathcal{C}_{OUC}, \mathcal{R}_{OUC}\}$ has no solution (we say that $\mathcal{P}_{OUC}$ is *unsatisfiable*), that is, there is no affectation of the variables which simultaneously satisfies all the constraints. This is a static aspect of CSP, which is a limitation of the use of CSP in real situations [1]. This situation can be mainly addressed by two kind of approaches, Partial CSP (PCSP) and Flexible CSP (FCSP).

5.5 Prioritized Removed Set Revision with ASP

We now present the Prioritized Removed Set Revision (PRSR) which generalizes the Removed Set Revision to the case of prioritized belief bases. Let S^O be a prioritized finite set of clauses, where S^O is partitioned into n strata, i. e. $S^O = S_1^O \cup \dots \cup S_n^O$, such that clauses in S_i^O have the same level of priority and are more prioritary than the ones in S_j^O where $j > i$. S_1^O contains the clauses which are the most prioritary beliefs in S^O , and S_n^O contains the ones which are the least prioritary in S^O [2].

When S^O is prioritized in order to restore consistency the principle of minimal change stems from removing the minimum number of clauses from S_1^O , then the minimum number of clauses in S_2^O , and so on. We generalize the notion of removed set in order to perform Removed Sets Revision with prioritized sets of clauses⁴. This generalization first requires the introduction of a preference relation between subsets of S^O and leads the definition of prioritized removed sets detailed in [2]. This definition of removed sets generalizes the definition 1. We directly translate the revision problem into an a logic program with answer set semantics. We build a logic program denoted by $P_{S^O \cup (S^D \cup S^C)}$ such that the answer sets of $P_{S^O \cup (S^D \cup S^C)}$ correspond to removed sets of $S^O \cup ((S^D \cup S^C))$. We then define a preference relation between answer sets stemming from the preference relation between subsets of S^O and we establish the correspondence between prioritized removed sets and preferred answer sets. The computation of Prioritized Removed Sets Revision is based on the adaptation of the smodels system. This is achieved in two steps. The first step, Prio, is an adaptation of smodels [11] system which computes the set of subsets of literals of R_{S^O} which lead to preferred answer sets and which minimize the number of clauses to remove from each stratum. The second step, Rens, computes the prioritized removed sets of $S^O \cup (S^D \cup S^C)$ stratum by stratum [3].

5.6 Revision in the Framework of Logic of Linear Constraints

In this approach we revise the set of bound constraints $\mathcal{L}^O$ by the set of rules $\mathcal{L}^C$. The revision method consists in first checking the consistency of the set of constraints $\mathcal{L}^O \cup \mathcal{L}^C$, this is performed by propagation of upper and lower bound constraints. In case of inconsistency, we have to identify the best subset(s), in terms of cardinality, of bound constraints $\mathcal{L}^O$ whose removal is sufficient to restore consistency. This is achieved by assigning each bound constraints in conflict $X_i \leq u_i$ (resp. $X_j \geq l_j$) a propositional variable U_i (resp. L_j) and to look for the models of $\neg(\bigvee_{i,j}(U_i \wedge L_j))$. For more details see [16].

6 Comparison

6.1 Comparison Between the Different Representations

We need to design a comparison framework suitable for geographic information. As specified in table 1, a first classification stems from the different levels of

⁴ When there is no stratification PRSR amounts to RSR.

Table 1. Comparison between representation formalisms

available information	epistemic state	representation formalism	logic
unordered information	belief set	propositional representation, ROBDD, SAT, ASP,	PL
		PROLOG	FL
partially ordered information	partial pre-order	propositional representation + partially ordered information	PL
totally ordered information	total pre-order	propositional representation + totally ordered information	PL
		propositional representation + quality	PL
	total order	Flexible CSP	HL
	dense total order	LLC	HL

representation of the epistemic states, depending on the available information on the relations between observations. Another classification can be made according to the different levels of the underlying logical formalisms, propositional logic (PL), first order logic (FL) or high order logics (HL).

The propositional logic involves a huge amount of propositional variables and clauses, though it takes advantage of the existing algorithms for revision in the propositional case, of possible translation into SAT problem and of compact representation with ROBDD approaches. The inconsistency is not explicit but comes out from the resolution of the satisfaction. The main drawback is the loss of the structure of the initial problem. However representing the quality of data, with, for example, a total pre-order on propositional variables allows us to give again a certain structure to the representation. Consequently, this reduces the search space.

The standard logic programming approach is very close to natural languages and directly representable in relational database. However the difficulties are twofold. Inconsistency rules have to be explicated, but these rules depend on the problem and there is no general formulation. Besides, revision rules have to be defined, these rules also depend on the problem and on the strategy used to solve the revision problem, the formulation of such rules is, in general, very difficult.

On contrast, the propositional clausal representation of the problem can be translated into a normal logic program with answer set semantics (ASP) stemming from the used revision strategy, like the proposed translation for Removed Set revision [3]. This is not suprising, since there is an equivalence between revision and non-monotonic inference. The inference relation used in standard logic programming is a monotonic inference relation whereas normal logic programming with answer set semantics uses non-monotonic inference.

The CSP representation provides a compact representation since it involves a smaller number of variables. Moreover, this representation is more expressive, because the relations capture part of the stucture of the problem. In the example from the flooding application, when dealing with 3 parcels there are 6 variables while there are 24 variables for the clausal representation and the set of rela-

tions given in intension, expresses the flow. Since standard CSP uses monotonic inference, Flexible or Fuzzy CSP is suitable for representing revision, however the relaxation of constraints may modify a lot of conflicts. The minimality of change takes the form of minimality in terms of optimization and compromises the principle of minimal change in terms of minimal change of explicit beliefs.

The LLC formalism also provides a compact representation, since it uses real valued variables. The domain consisting in the real numbers is continuous and dense, and given in intension as well as the relations that express in a very natural way the structure of the problem. In the flooding application, when dealing with 3 parcels, there are 3 real variables and the relations are $A \geq B$ and $A \geq C$, which is a very natural and simple way for expressing the flow relations. However the LLC representation is not general, it is suitable for linear problems, but not for non-linear problems. This is not always the case when dealing with geographical information because we also have to deal with qualitative data defined on discrete domains, like shapes or colors, for example, or boolean data, and not every problem can be represented in terms of linear constraints as illustrated by example 2.

The expression of the revision problem is different in the different representations, however the revision problem is the same. The revision problem consists in identifying the conflicting observations to modify in order to restore consistency.

6.2 Comparison Between the Different Revision Approaches

The different approaches of revision presented in Section 5 can be classified according to the different levels of the underlying logical formalisms. In propositional approaches and first order logic representations the loss of structure put all conflicts at the same level whereas in higher order logic representations some conflicts can be solved by constraint propagation. This leads to a classification of the approaches into two categories. The first category encompasses all revision operations which concentrate on the detection of the conflicts between different sources of information. The second category contains all approaches which concentrate on the direct resolution of the conflicts by means of propagation mechanisms.

Comparison Between the Approaches Stemming from Conflict Detection. The first category (i.e. conflict identification) contains all approaches based on propositional logic. They perform RSR using the previously described representations. The main part of the work on these approaches had been to provide an adequate revision machinery in order to break down the complexity inherent to logical based reasoning. More precisely, the “complexity break down” work has been tackled using two different points of view. On one hand encoding the knowledge by means of propositional clauses and finding heuristics lowering the complexity of the satisfiability tests needed during the revision process. On the other hand using knowledge compilation techniques to perform all computationally heavy tasks during a compilation phase, yet allowing us to work further on lighter representations of our knowledge.

Typically, compiled forms of the knowledge allows satisfiability test to be done with a worst case time complexity linear in the number variables or even constant. These approaches are summarized in the following table:

Approach	Type	Comments
RSR with REM	Clauses	hitting sets
RSR with SAT	Clauses	preferred models
RSR with ASP	Clauses	preferred models
PRSR with ASP	Clauses	strat., preferred models, the most efficient
RSR with ROBDD	Knowledge compilation	Compilation stage size problems

These approaches have been shown to be equivalent [19, 3], they provide the same removed sets, except, of course, in the PRSR case.

Experimental Comparison. All experimental comparison and measures have been presented in previously published work [19, 3]. We just recall here the main results. In [19] it has been shown that the REM algorithm described in subsection 5.1 which computes the removed sets by using a modification of Reiter’s algorithm for the computation of minimal hitting set gives better results than the ROBDD approach. A comparison between the REM algorithm and the Rens algorithm which is an adaptation of the smodels system for RSR with ASP in [3] showed that the adaptation of the smodels system for RSR with ASP gave the best results. In [3], we compared the SAT approach which uses the efficient SAT-solver MiniSat and to the Rens algorithm which is an adaptation of the smodels system for RSR with ASP. This test showed that Rens gave the best results.

However, RSR with ASP can deal with 60 parcels with a reasonable running time (few minutes) but reaches a CPU time limit (10 hours) around 64 parcels. In the flooding application we have to deal with a block consisting of 120 parcels and the stratification is useful to deal with the whole area. A stratification of S_1 is induced from the geographic position of parcels. Parcels located in the upstream part of the valley are preferred to the parcels located in the downstream part of the valley. Using a stratification of S_1 , we observed that Rens algorithm for PRSR with ASP can deal with the whole area with a reasonable running time [3].

Comparison Between the Approaches Stemming from Propagation. The second category of approaches is the “propagation” set of approaches. This category contains the original method used by CEMAGREF to solve the problem before we start our common work on this project. It is a purely numerical method, which tries to correct conflicting information as soon as it is discovered. The search space of the conflicts is reduced by using the upstream/downstream orientation of the flooded valley. The complexity of this method is very low (almost linear in the number of parcels).

The second method contained in this “conflict correction” category is based on the Logic of Linear Constraints (LLC) and a directed propagation algorithm, proposed in [8]. This approach is a logical framework for the original approach developped by the CEMAGREF and follows 2 steps. The first step, the conflict detection, stems from the propagation of the upper bounds or lower bounds constraints (worst case time complexity : $O(n^2)$, n being the # of parcels). In a second step, a logical formula is then constructed from the list of detected conflicts according to the process described in Section 5.6. Therefore the determination of the models of this formula which falsify the least number of literals

of this formula corresponds to the determination of the subsets of constraints to revise. Since a Davis and Putnam procedure was used to compute the models, the complexity in the worst case is exponential in the number of detected conflicts. However, an experimental study has shown that revision using LLC is efficient because the number of detected conflicts is generally low. In the flooding application, for the whole area consisting of the 3 blocks, that is 200 parcels only 15 conflicts are detected and the algorithm provides 128 subsets of constraints to revise, each subset consisting in 10 constraints. The global running time for revision for LLC is around 2 seconds.

The main differences between LLC and CSP approaches are the following. FCSP approach deals with finite discrete domains which is not the case of LLC which deals with variables defined on continuous domains. The FCSP approach uses constraints defined with a degree of uncertainty (or degree of satisfaction) that allows us to represent uncertain data and preferences. On contrast, in LLC the constraints are satisfied or not. LLC follows the principle of minimal change in minimizing the number of constraints to revise, in a similar way as RSR while the minimality of change in the FCSP approach amounts to the min-max optimization (maximizing the degree of satisfaction of the less satisfied constraint).

Comparison Between Conflict Detection and Propagation. Directly comparing the two preceeding classes of approaches is rather difficult since they tackle the problem from different points of view. On one hand purely logical approaches concentrate on the detection of minimal sets of conflicts. On the other hand propagation approaches try to detect conflicts while solving the problem at the same time by the mean of constraint propagation.

The minimal change principle is not the same in the two classes of approaches as stated above.

The constraints propagation approaches provide best running times since they take into account the structure of the problem, while in the detection approaches the loss of structure of the in initial problem put all conflicts at the same level. However, the propagation approaches are not general, they are suitable only for linear problems, while non-linear problems can be dealt with detection approaches. By the way, we can list the pros and the cons of the two families of approaches:

	Conflict detection	Propagation
Pros	Focuses on the explanation of the conflicts suitable for non linear problems	Directly delivers a solution Low worst case time complexity (quadratic)
Cons	No numerical results (no refinement of initial assessments) High worst case complexity in the general case	Less general (bound to linear problems) Computation of minimal sets of conflicts is "ad-hoc" if we do not use a logical revision framework

7 Conclusion

We studied different representations of the revision problem on geographic information. We then discussed the advantages and the drawbacks of the different representations and we illustrated the revision problem by examples extracted from real scale applications. According to each representation, we then proposed adjustments of existing strategies, taking advantage of the nature of geographic information, to perform revision. We implemented the different revision approaches and we conducted an experimental study on the flooding application. The comparison between the different approaches leads to a classification into two classes of approaches, the propagation approaches which are not general but suitable and very efficient for linear revision problems and the logical approaches which are less efficient for linear revision problems, but more general and suitable for non-linear problems.

The problem of merging multiple sources of information is central in GIS. Since revision is a special case of fusion with two sources where one source is preferred to the other, it could be interesting to investigate how we could generalize the adjustments proposed for revision to fusion.

References

1. A. Bellicha and al. Autour du problème de satisfaction de contraintes. In *Actes des 5èmes journées nationales du PRC GDR Intelligence Artificielle*, pages 159–178, 1995.
2. S. Benferhat, Cayrol C, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *Proc. of IJCAI'93*, pages 640–645, 1993.
3. J. Bennaïm, S. Benferhat, O. Papini, and E. Würbel. An answer set programming approach of prioritized removed sets revision : Application to gis. In *Proc. of JELIA'04*, pages 604–616. LNAI, 2004.
4. R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on computers*, C-35(8):677–691, Aout 1986.
5. N. Eén and N. Sörensson. An Extensible SAT-solver. In *Proc. of ICTAS'03*, 2003.
6. De Kleer J. Using crude probability estimates to guide diagnosis. *Artificial Intelligence*, 45:381–392, 1990.
7. R. Jeansoulin and O. Papini. Révision et information spatiale. In *Le temps, l'espace et l'évolutif en sciences du traitement de l'information*, pages 293–304. Cépadués-éditions, 2000.
8. M. Khelfallah, N. Wilson, and R. Jeansoulin. Geographic Information Revision based on linear constraints. In *Tech. Rep., Annex 313.4 of report R313, REVIGIS project*, 2003.
9. J. De Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.
10. Bernhard Nebel. Syntax based approaches to belief revision. In Peter Gärdenfors, editor, *Belief Revision*, pages 52–88. Cambridge University Press, UK, 1992.
11. I. Niemela and P. Simons. An implementation of stable model and well-founded semantics for normal logic programs. In *Proc. of LPNMR'97*, pages 420–429, 1997.
12. O. Papini. A complete revision function in propositional calculus. In B. Neumann, editor, *Proc. of ECAI'92*, pages 339–343. John Wiley and Sons. Ltd, 1992.

13. D. Raclot and C. Puech. Photographies aériennes et inondation : globalisation d'informations floues par un système de contraintes pour définir les niveaux d'eau en zone inondée. *Revue internationale de géomatique*, 8(1):191–206, 1998.
14. S.Lagrué, R. Devillers, and J-Y. Besqueut. Partially ordered preferences applied to the site location problem in urban planning. In *Proc. of DEXA'04*, 2004.
15. N. Wilson. The logic of linear constraints and its application to the flooding problem. Technical report, REVIGIS project report, 2002.
16. N. Wilson, M.Khelfallah, and R. Jeansoulin. Geographic information revision based on linear constraints. Technical report, REVIGIS project- Annex 313.4, 2003.
17. E. Würbel, R. Jeansoulin, and O.Papini. Spatial information revision : A comparison between 3 approaches. In *Proc. of ECSQARU'2001*, pages 454–465. L NA I, 2143, Springer, 2001.
18. E. Würbel, R. Jeansoulin, and O. Papini. Revision : An application in the framework of gis. In *Proc. of KR'2000*, pages 505–516. Morgan Kaufmann, 2000.
19. E. Würbel, R. Jeansoulin, and O. Papini. Spatial information revision : A comparison between 3 approaches. In *Proc. of ECSQARU'2001*, number 2143 in LNAI, pages 454–465. Springer Verlag, 2001.

Multiple Semi-revision in Possibilistic Logic

Guilin Qi, Weiru Liu, and David A. Bell

School of Computer Science, Queen's University Belfast,
Belfast, BT7 1NN, UK
{G.Qi, W.Liu, DA.Bell}@qub.ac.uk

Abstract. Semi-revision is a model of belief change that differs from revision in that a new formula is not always accepted. Later, Fuhrmann defined multiple semi-revision by replacing a new formula with a set of formulae as the new information, which results in a merging operator called a partial meet merging operator. The problem for the partial meet merging operator is that it needs additional information to define a *selection function* which selects a subset from a set of maximal consistent subbases of an inconsistent knowledge base. In this paper, we will extend multiple semi-revision in the framework of possibilistic logic. The advantage of possibilistic logic is that it provides an ordering relation on formulae in knowledge bases, which makes it easy to define a *selection function* practically.

1 Introduction

The problem of belief revision has been widely discussed in the past twenty years [1, 8, 11, 12, 19, 24]. In belief revision theory, new information (a new formula) must be adopted and some existing information will be dropped to accommodate it. However, many researchers argued that new information should not always have the priority over the existing information and some non-prioritized belief revision methods have been proposed in which new information is not necessarily accepted [11, 17, 18]. For example, the semi-revision introduced by Hansson [17] differs from belief revision in two aspects: first, original information is represented as a belief base rather than a belief set, and second, new information is not always accepted. The semi-revision can be related to belief merging which deals with the problem of deriving a coherent belief base from a set of inconsistent belief bases [2, 3, 4, 5, 11, 13, 14, 15, 19, 23]. Fuhrmann in [11] considered a multiple semi-revision by replacing the new formula with a set of formulae as new information, which results in a merging operator which he called a *partial meet merging operator*. Both the semi-revision and the partial meet merge methods consist of two steps. The first step is to conjoin original information and new information and the second step is to restore consistency using a contraction function defined in [1, 16].

Two problems exist in semi-revision and partial meet merge. First, it is not advisable to conjoin an original knowledge base with a new formula (or a set of formulae) because some information may be lost. Let us look at an example.

Let $K_1 = \{\phi, \psi\}$ be the original knowledge base and $K_2 = \{\phi, \psi\}$ be the new information. Conjoining K_1 and K_2 results in $K_3 = \{\phi, \psi\}$, which is consistent. It is the result of partial meet merge of K_1 and K_2 . If new information $K_4 = \{\neg\phi\}$ is obtained, conjoining K_3 and K_4 results in a knowledge base $K = \{\phi, \neg\phi, \psi\}$. Since ϕ and $\neg\phi$ are equally reliable, it is not possible to decide which formula should be dropped, both ϕ and $\neg\phi$ have to be deleted. However, since both K_1 and K_2 support ϕ and only K_4 supports $\neg\phi$, by majority principle, ϕ should be kept and $\neg\phi$ should be deleted.

The second problem is that we need a method to define a contraction function in a practical way. In belief revision [1, 12], this problem is solved by considering a notion of epistemic entrenchment. An epistemic entrenchment is an ordering that envisages the logical dependencies of the formulae in the belief set. It is the epistemic entrenchment of a formula in a belief set that determines the formula's fate when the belief set is contracted.

In this paper, we will resolve above problems by considering the multiple semi-revision in possibilistic logic. In [9], a corresponding relationship between epistemic entrenchment and possibilistic logic has been established. It has been shown that the only numerical counterparts of epistemic entrenchment relations are necessity measures. Possibilistic logic is an extension of classical logic. Each formula in possibilistic logic is attached with a weight denoting its necessity degree. Possibilistic logic has been shown to be a good framework for belief revision and belief merging [3, 4, 5].

Multiple semi-revision in possibilistic logic is carried out in two steps: a combination step and an inconsistency handling step. In the combination step, each belief base is split into two subbases: one consists of *conflict* formulae and the other consists of *free* formulae in the union of all the belief bases. The weights of formulae in the subbases with *free* formulae are either increased or unchanged and the weights of formulae in the subbases with *conflict* formulae are decreased. That is, we have a *reinforcement* effect on the *free* formulae and a *counteract* effect on the *conflict* formulae. This method is more reasonable than the conjoining method because it does not ignore any information in both sources. Then in the inconsistency handling step, we will restore consistency of the resulting belief base if it is inconsistent by dropping some conflict formulae according to their priorities.

This paper is organized as follows. Section 2 gives a brief review of possibilistic logic. We then introduce Hansson's semi-revision and Fuhrmann's partial meet merging in Section 3. In Section 4 we will define the stratified semi-revision. We compare the stratified semi-revision and some other merging methods in possibilistic logic in Section 5. Finally, we conclude this paper in Section 6.

2 Possibilistic Logic

In this paper, we only consider a finite propositional language denoted by $\mathcal{L}$. The classical consequence relation is denoted as $\vdash$. $\phi, \psi, \gamma, \dots$ represent classical formulas. $\top$ and $\perp$ represent constant truth and constant false respectively. A

(classical) *knowledge base* K is a finite set of propositional formulas. Knowledge bases are denoted by capital letters $A, B, C, K \dots$

Possibilistic logic [10] is an extension of classical logic. It is a weighted logic where each classical formula is associated with a level of priority. A possibilistic knowledge base is the set of possibilistic formulae of the form $\mathcal{B} = \{(\phi_i, a_i) : i = 1, \dots, n\}$. The *possibilistic formula* (ϕ_i, a_i) means that the necessity degree of ϕ_i is at least equal to a_i . Let $\mathcal{KB}$ denote the set of all the possibilistic knowledge bases. In this paper, we only consider possibilistic knowledge bases where every formula ϕ is a classical propositional formula. The classical base associated with $\mathcal{B}$ is denoted as $\mathcal{B}^*$, namely $\mathcal{B}^* = \{\phi_i | (\phi_i, a_i) \in \mathcal{B}\}$. A *possibilistic base* $\mathcal{B}$ is consistent if and only if its classical base $\mathcal{B}^*$ is consistent. The formulas in $\mathcal{B}$ can be rearranged according to their weights in the descending order, such that $a_0 = 1 \geq a_1 \geq \dots \geq a_n > 0$. Suppose there are m distinct a_i values (weights) $a_{i_1}, \dots, a_{i_m}$, where $a_{i_j} > a_{i_{j+1}}$. Then $\mathcal{B}$ can be equivalently expressed as a layered belief base $\Sigma_{\mathcal{B}} = S_1 \cup \dots \cup S_m$, where $S_k = \{\phi : (\phi, a_{i_k}) \in \mathcal{B}\}$. $\Sigma_{\mathcal{B}}$ is called the stratification of $\mathcal{B}$.

In possibilistic logic, a *possibility distribution*, denoted by π , is a mapping from a set of possible worlds $\mathcal{W}$ to the interval $[0, 1]$. $\pi(\omega)$ represents the possibility degree of the interpretation ω with the available beliefs. From a *possibility distribution* π , two measures defined on a set of propositional or first order formulas can be determined. One is the possibility degree of formula ϕ , denoted as $\Pi(\phi) = \max\{\pi(\omega) : \omega \models \phi\}$. The other is the necessity degree of formula ϕ , and is defined as $N(\phi) = 1 - \Pi(\neg\phi)$.

Definition 1. [10] Let $\mathcal{B}$ be a PKB, and $\alpha \in [0, 1]$. The α -cut of $\mathcal{B}$ is $\mathcal{B}_{\geq \alpha} = \{\phi \in \mathcal{B}^* | (\phi, a) \in \mathcal{B} \text{ and } a \geq \alpha\}$.

Definition 2. [4] A subbase $\mathcal{A}$ of $\mathcal{B}$ is said to be minimally inconsistent if and only if it satisfies the following two requirements:

- $(\mathcal{A})^* \models \perp$, where $(\mathcal{A})^*$ is the classical base of $\mathcal{A}$, and
- $\forall \phi \in (\mathcal{A})^*, (\mathcal{A})^* - \{\phi\} \not\models \perp$.

Definition 3. [4] A *possibilistic formula* (ϕ, a) is said to be free in $\mathcal{B}$ if it does not belong to any minimally inconsistent subbase of $\mathcal{B}$ and (ϕ, a) is said to be conflict in $\mathcal{B}$ otherwise. Conflict($\mathcal{B}$) to denote the set of formulae in $\mathcal{B}$ which are in conflict.

The *inconsistency degree* of $\mathcal{B}$, which defines the level of inconsistency of $\mathcal{B}$, is defined as [10]:

$$Inc(\mathcal{B}) = \max\{\alpha_i | \mathcal{B}_{\geq \alpha_i} \text{ is inconsistent}\}.$$

Suppose $\Sigma_{\mathcal{B}}$ is the stratification of $\mathcal{B}$, then the degree of inconsistency of $\Sigma_{\mathcal{B}}$ is defined as the degree of inconsistency of $\mathcal{B}$.

Definition 4. [10] Let $\mathcal{B}$ be a possibilistic base. Let (ϕ, α) be a piece of information with $\alpha > Inc(\mathcal{B})$. (ϕ, α) is said to be a consequence of $\mathcal{B}$, denoted by $\mathcal{B} \vdash_{\pi} (\phi, \alpha)$, iff $\mathcal{B}_{\geq \alpha} \vdash \phi$.

3 Semi-revision

The main difference between semi-revision [17] and traditional belief revision [1, 12] is that a new formula is not necessarily accepted. The basic idea of semi-revision is to conjoin the original belief base and the new formula and then drop some formulae in the resulting base to restore consistency.

Definition 5. [1] Let A be a set of formulae and ϕ a formula. The set $A \perp \phi$ ¹ (“ A less ϕ ”) is the set of sets such that $B \in A \perp \phi$ if and only if:

- (1) $B \subseteq A$
- (2) $B \not\models \phi$
- (3) $\forall B' \subset A$, if $B \subset B'$, then $B' \vdash \phi$

Definition 6. [1] A selection function for a set A of formulae is a function γ such that for every formula ϕ :

- (1) If $A \perp \phi$ is non-empty, then $\gamma(A \perp \phi)$ is a non-empty subset of $A \perp \phi$, and
- (2) If $A \perp \phi$ is empty, then $\gamma(A \perp \phi) = \{A\}$.

Definition 7. [1] Let A be a set of formulae and γ a selection function for A . The partial meet contraction on A that is generated by γ is the operation $\sim \gamma$ such that for every formula ϕ :

$$A \sim \gamma \phi = \cap \gamma(A \perp \phi)$$

Partial meet semi-revision [17] is based on the partial meet contraction. It first adds the belief ϕ to the base, and then the resulting base is contracted by the constant false $\perp$.

Definition 8. The partial meet semi-revision of B based on a selection function γ is the operator $?_\gamma$ such that for every formula ϕ :

$$B ?_\gamma \phi = (B \cup \{\phi\}) \sim \gamma \perp = \cap \gamma((B \cup \{\phi\}) \perp \{\perp\})$$

In [11], Fuhrmann generalized the semi-revision by replacing the input as a set of formulae, which results in a merging operator.

Definition 9. Let A and B be two belief bases. The partial meet merge of A and B is defined as:

$$A \circ B = (A \cup B) \sim \gamma \perp$$

Fuhrmann also gave the axiomatic characterization of the partial meet merge [11].

¹ We use $\perp$ to denote both the constant false and the operation to obtain the set of maximal subbases of A which do not imply ϕ as in belief revision literature. Hopefully it will not make confusion.

Theorem 1. $\circ$ is an operator of partial meet merge if and only if it satisfies:

- (M1) $A \circ B$ is consistent (strong consistency)
- (M2) $A \circ B \subseteq A \cup B$ (inclusion)
- (M3) If $\phi \in (A \cup B) \setminus (A \circ B)$, Then $\exists D : A \circ B \subseteq D \subseteq A \cup B$ and $D \not\vdash \perp$ but $D \cup \{\phi\} \vdash \perp$ (relevance)
- (M4) If $A \cup B = A' \cup B'$, then $A \circ B = A' \circ B'$ (congruence)

4 Multiple Semi-revision: A Possibilistic Approach

Priority plays a very important role in belief revision [12, 19]. Possibilistic logic is a good framework to represent priority [4]. In this section, we extend multiple semi-revision in the framework of possibilistic logic. Multiple semi-revision consists of two steps: one is the combination step and the other is the inconsistency handling step. In the combination step, the original knowledge bases are combined that may produce a possibly inconsistent knowledge base. In the inconsistency handling step, some conflicting formulae are dropped to restore consistency.

4.1 Combination Step

In semi-revision and partial meet merge, the intermediate combination step is to conjoin original information and new information. Recall the example given in the Section 1, a disadvantage of conjoining the knowledge bases is that some important information may be lost.

It is also not always advisable to conjoin two possibilistic knowledge bases in the intermediate combination step in multiple semi-revision in possibilistic logic. Let us look at an example to illustrate the reason for it.

Example 1. Let $\mathcal{B}_1 = \{(\neg\phi, 0.7), (\gamma, 0.8)\}$ and $\mathcal{B}_2 = \{(\phi, 0.6), (\gamma, 0.8)\}$ be two possibilistic belief bases. By conjoining them we obtain a knowledge base $\mathcal{B} = \{(\neg\phi, 0.7), (\phi, 0.6), (\gamma, 0.8)\}$. Since the weight of $\neg\phi$ is greater than that of ϕ , it is reasonable to delete ϕ , so the result of merging is $\mathcal{B}_3 = \{(\neg\phi, 0.7), (\gamma, 0.8)\}$. Suppose later we receive another source represented as $\mathcal{B}_4 = \{(\phi, 0.7), (\neg\gamma, 0.8)\}$. By conjoining $\mathcal{B}_3$ and $\mathcal{B}_4$ we get $\mathcal{B}' = \{(\neg\phi, 0.7), (\phi, 0.7), (\gamma, 0.8), (\neg\gamma, 0.8)\}$. Since $\phi, \neg\phi$ have the same weights and $\gamma, \neg\gamma$ have the same weights, we have to drop all the formulae in $\mathcal{B}'$. So the final result is a knowledge base with no information. This is not reasonable! For ϕ , there are two sources $\mathcal{B}_1$ and $\mathcal{B}_2$ supporting it with weights 0.7 and 0.6 respectively. Whilst there is only one source supporting $\neg\phi$ with weight 0.7. So we may prefer to retain ϕ and drop $\neg\phi$. For the same reason, it is more reasonable to retain γ and drop $\neg\gamma$. The problem for the example above is that when we combine $\mathcal{B}_1$ and $\mathcal{B}_2$ by conjoining them, after restoring consistency, information provided by $\mathcal{B}_2$ is ignored.

Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two possibilistic knowledge bases from two different sources. For those formulas that are involved in the conflict in $\mathcal{B}_1 \cup \mathcal{B}_2$, their necessity

degrees should decrease after combination because they will counteract with each other. In contrast, the necessity degree should increase for those formulas that are supported by both sources.

Definition 10. [6] An operator $\oplus_{SC}$ is said to be strongly conjunctive on $[0,1]$ if for all $(a_1, \dots, a_n)$

$$\oplus_{SC}(a_1, \dots, a_n) \geq \max(a_1, \dots, a_n).$$

A strongly conjunctive operator is used to increase the weight of a formula after combination. Many operators belong to this class, such as the probabilistic sum $\oplus(a, b) = \min(a + b - ab, 1)$ and bounded sum $\oplus(a, b) = \min(a + b, 1)$.

Definition 11. [21] An operator $\oplus_{UA}$ is said to be an up-averaging operator if for all $(a_1, \dots, a_n)$

$$\oplus_{UA}(a_1, \dots, a_n) \leq \max(a_1, \dots, a_n).$$

This operator reflects that a combination result cannot be greater than the greatest of all. An example of up-average operator is the standard average operator $\oplus(a, b) = (a + b)/2$. Another up-average operator, called *max-product* operator, is defined as follows:

$$\oplus_{max,pro}(a, b) = \begin{cases} \max(a, b) & \text{if } a, b \neq 0, \\ \max(a^2, b^2) & \text{otherwise.} \end{cases}$$

This operator reflects that if a formula is supported by two sources with weights greater than 0, then we keep the maximum weight as the result of combination of two weights a and b , otherwise the weight of the formula will be decreased after combination.

Now we give a combination method based on the operators defined above.

Given two knowledge bases $\mathcal{B}_1$ and $\mathcal{B}_2$, we use two operators, one is a strongly conjunctive operator and the other is an up-averaging operator. For those formulas that are not in conflict in $\mathcal{B}_1 \cup \mathcal{B}_2$, we choose the strongly conjunctive operator to combine them. But for those formulas that are in conflict, we use the up-averaging operator to combine them. We always assume that if a formula ϕ does not appear in a possibilistic knowledge base $\mathcal{B}$, then $(\phi, 0)$ has been added to $\mathcal{B}$ implicitly if necessary. Moreover, we assume that each formula in a possibilistic knowledge base appears only once with a unique weight.

Definition 12. Let $\mathcal{B}_1 = \{(\phi_i, a_i) : i = 1, \dots, n\}$ and $\mathcal{B}_2 = \{(\psi_j, b_j) : j = 1, \dots, m\}$ be two self-consistent possibilistic knowledge bases. Let $\oplus_{SC}$ and $\oplus_{UA}$ be a strong conjunctive operator and an up-averaging operator respectively. The combination of $\mathcal{B}_1$ and $\mathcal{B}_2$ is defined as $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2) = \mathcal{C} \cup \mathcal{D}$, where

$$\mathcal{C} = \{(\phi, \oplus_{UA}(a, b)) \mid \phi \in (\text{Conflict}(\mathcal{B}_1 \cup \mathcal{B}_2))^*, (\phi, a) \in \mathcal{B}_1, (\phi, b) \in \mathcal{B}_2\},$$

$$\mathcal{D} = \{(\phi, \oplus_{SC}(a, b)) \mid \phi \notin (\text{Conflict}(\mathcal{B}_1 \cup \mathcal{B}_2))^*, (\phi, a) \in \mathcal{B}_1 \text{ and } (\phi, b) \in \mathcal{B}_2\}$$

Example 2. (Continue Example 1) Since γ is supported by both sources, its certainty degree should increase, i.e., there is a reinforcement between $\mathcal{B}_1$ and $\mathcal{B}_2$ for γ . For formulas ϕ and $\neg\phi$, they are involved in the inconsistency of $\mathcal{B}_1 \cup \mathcal{B}_2$, so their necessity degrees should decrease. Let $\oplus_{SC}$ be the probabilistic sum and $\oplus_{UA}$ be the *max-product* operator. By Definition 12, the combination of $\mathcal{B}_1$ and $\mathcal{B}_2$ is $\mathcal{B} = \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2) = \{(\neg\phi, 0.49), (\phi, 0.36), (\gamma, 0.96)\}$.

4.2 Inconsistency Handling Step

The knowledge base obtained by the combination step is inconsistent if the original knowledge bases are in conflict. As in semi-revision and partial meet merge, we will drop some formulae in the knowledge base to restore consistency. Since a possibilistic knowledge base provides explicit priorities between formulae, we can drop those formulae in conflict according to their weights or priorities.

As most inconsistency handling methods in possibilistic logic [4], first we need to stratify the possibilistic knowledge bases. A very common approach to handling inconsistency in a stratified knowledge base is to keep as much information in a higher layer as possible.

Definition 13. [4] Let $\Sigma = S_1 \cup \dots \cup S_n$ be a layered belief base. A subbase $\Sigma' = A_1 \cup \dots \cup A_n$ of Σ is a strongly maximal consistent subbase (*SMC-subbase* for short) iff for all k ($1 \leq k \leq n$) $A_1 \cup \dots \cup A_k$ is a maximal consistent subset of $S_1 \cup \dots \cup S_k$. The set of all SMC-subbases of Σ is denoted by $SMC(\Sigma)$.

The SMC-subbase was also defined in [7], with the name “preferred subbases”. It can be constructed by starting with a maximal consistent subset of S_1 , then adding to the maximal consistent subset as many formulas of S_2 as possible (while preserving consistency), and so on. So a SMC-subbase Σ' of a stratified belief base Σ must be a maximal subbase of it, i.e., $\Sigma' \in \Sigma \perp \{\perp\}$. The following proposition suggests that SMC-subbases are acceptable in the sense of the best out selection.

Proposition 1. [4] Let $\Sigma_{\mathcal{B}}$ be the stratification of a possibilistic knowledge base $\mathcal{B}$. A SMC-subbase of $\Sigma_{\mathcal{B}} = S_1 \cup \dots \cup S_n$ is $\Sigma' = A_1 \cup \dots \cup A_n$ such that the degree of inconsistency of $\Sigma' \cup \{\phi\}_i$ is a_i , $\forall \phi \in S_i - A_i$, where $\Sigma' \cup \{\phi\}_i$ is the new stratified knowledge base obtained by adding ϕ to the layer S_i in Σ' .

Now suppose we have two possibilistic knowledge bases $\mathcal{B}_1$ and $\mathcal{B}_2$, where $\mathcal{B}_1$ is the original knowledge base the $\mathcal{B}_2$ is a new knowledge base. Then the multiple semi-revision is processed as follows. First we combine $\mathcal{B}_1$ and $\mathcal{B}_2$ as $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$. Let Σ be the stratification of $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$, then in the second step, we delete those elements of $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$ that do not belong to any of the elements of $SMC(\Sigma)$.

Definition 14. Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two possibilistic knowledge bases. $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$ is the possibilistic knowledge base obtained by the combination step. Suppose Σ is the stratification of $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$. Let $\Sigma' = A_1 \cup \dots \cup A_n = \cap \{\Sigma_i \subseteq \Sigma : \Sigma_i \in SMC(\Sigma)\}$. Then the SMC-subbases based merging is defined as $\mathcal{B}_1 \circ_{\oplus_{SC}, \oplus_{UA}}^{SMC} \mathcal{B}_2 = \{(\phi, a_i) \in \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2) : \phi \in A_i\}$.

Example 3. (Continue Example 2) In Example 2, the combination of $\mathcal{B}_1$ and $\mathcal{B}_2$ is $\mathcal{B} = \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2) = \{(\neg\phi, 0.49), (\phi, 0.36), (\gamma, 0.96)\}$. The stratification of $\mathcal{B}$ is $\Sigma_{\mathcal{B}} = \{\{\gamma\}, \{\neg\phi\}, \{\phi\}\}$, and the only *SMC*-subbase of $\Sigma_{\mathcal{B}}$ is $\{\{\gamma\}, \{\neg\phi\}\}$. So the result of merging of $\mathcal{B}_1$ and $\mathcal{B}_2$ is $\mathcal{B}_3 = \{(\neg\phi, 0.49), (\gamma, 0.96)\}$. Now suppose another source $\mathcal{B}_4 = \{(\phi, 0.7), (\neg\gamma, 0.8)\}$ is received. By combining $\mathcal{B}_3$ and $\mathcal{B}_4$ we get $\mathcal{B}' = \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_3, \mathcal{B}_4) = \{(\neg\phi, 0.24), (\phi, 0.49), (\gamma, 0.92), (\neg\gamma, 0.64)\}$. The stratification of $\mathcal{B}'$ is $\Sigma_{\mathcal{B}'} = \{\{\gamma\}, \{\neg\gamma\}, \{\phi\}, \{\neg\phi\}\}$. The only *SMC*-subbase of $\Sigma_{\mathcal{B}'}$ is $\{\{\gamma\}, \{\phi\}\}$. So the final result of merging is $\mathcal{B}_5 = \{(\phi, 0.49), (\gamma, 0.92)\}$. So both ϕ and γ can be inferred from $\mathcal{B}_5$, which is consistent with our analysis in Example 1.

Example 4. Let $\mathcal{B}_1 = \{(\phi, 0.8), (\neg\phi \vee \psi, 0.7), (\gamma, 0.6), (\psi \vee \varphi, 0.5)\}$ and $\mathcal{B}_2 = \{(\neg\phi, 0.8), (\neg\psi, 0.7), (\gamma, 0.7)\}$. Let $\oplus_{SC}$ be the probabilistic sum and $\oplus_{UA}$ be the *max-product* operator. The knowledge base obtained by the combination step is $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2) = \{(\gamma, 0.88), (\phi, 0.64), (\neg\phi, 0.64), (\neg\phi \vee \psi, 0.49), (\neg\psi, 0.49), (\varphi \vee \psi, 0.5)\}$. The stratification of $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$ is $\Sigma = \{\{\gamma\}, \{\phi, \neg\phi\}, \{\varphi \vee \psi\}, \{\neg\phi \vee \psi, \neg\psi\}\}$. There are three *SMC*-subbases in Σ : $\{\{\gamma\}, \{\phi\}, \{\varphi \vee \psi\}, \{\neg\phi \vee \psi\}\}$, $\{\{\gamma\}, \{\phi\}, \{\varphi \vee \psi\}, \{\neg\psi\}\}$, $\{\{\gamma\}, \{\neg\phi\}, \{\varphi \vee \psi\}, \{\neg\phi \vee \psi, \neg\psi\}\}$. The intersection of the *SMC*-subbases is $\{\{\gamma\}, \{\varphi \vee \psi\}\}$. So the result of *SMC*-subbases based merge of $\mathcal{B}_1$ and $\mathcal{B}_2$ is $\mathcal{B} = \{(\gamma, 0.88), (\varphi \vee \psi, 0.5)\}$.

The *SMC*-subbases based merge discards too much information. In Example 4, all the formulae involved in conflict are dropped after merging. As in the semi-revision and partial meet merge, we can select a subset of *SMC*-subbases. This can be done by defining a selection function as follows.

Definition 15. A selection function for a layered belief base Σ is a function γ such that:

- (1) If $SMC(\Sigma)$ is non-empty, then $\emptyset \subset \gamma(SMC(\Sigma)) \subseteq SMC(\Sigma)$, and
- (2) If $SMC(\Sigma)$ is empty, then $\gamma(SMC(\Sigma)) = \{\Sigma\}$.

The merging operator based on a selection function is defined as follows.

Definition 16. Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two possibilistic knowledge bases. $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$ is the possibilistic knowledge base obtained by the combination step. Suppose Σ is the stratification of $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$. Let γ be a selection function for Σ . Let $\Sigma' = A_1 \cup \dots \cup A_n = \cap \{\Sigma_i \subseteq \Sigma : \Sigma_i \in \gamma(SMC(\Sigma))\}$. The partial *SMC*-subbases based merging is defined as $\mathcal{B}_1 \circ_{\oplus_{SA}, \oplus_{UA}}^{PSMC} \mathcal{B}_2 = \{(\phi, a_i) \in \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2) : \phi \in A_i\}$.

A particular selection function can be defined by selecting the lexicographically maximal consistent subbases [4].

Definition 17. [4] Let Σ be a stratified knowledge base. Suppose $SMC(\Sigma)$ is the set of *SMC*-subbases of Σ , then any $\Sigma' = A_1 \cup \dots \cup A_n \in SMC(\Sigma)$ is said to be a lexicographically maximal consistent (LMC) subset of Σ if and only if $\forall \Sigma'' = B_1 \cup \dots \cup B_n \in SMC(\Sigma)$,

$$\nexists i, \text{ such that } |B_i| > |A_i| \text{ and } \forall j < i, |B_j| = |A_j|$$

The set of all lexicographically maximal consistent subsets of Σ is denoted as $Lex(\Sigma)$.

Definition 18. Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two possibilistic knowledge bases. $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$ is the possibilistic knowledge base obtained by the combination step. Suppose Σ is the stratification of $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$. Let $\Sigma' = A_1 \cup \dots \cup A_n = \cap \{ \Sigma_i \subseteq \Sigma : \Sigma_i \in Lex(\Sigma) \}$. Then the *Lex*-subbases based merging is defined as $\mathcal{B}_1 \circ_{\oplus_{SA}, \oplus_{UA}}^{Lex} \mathcal{B}_2 = \{(\phi, a_i) \in \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2) : \phi \in A_i\}$.

Example 5. (Continue Example 4) The lexicographically maximal consistent subbase of Σ is $\{\{\gamma\}, \{\neg\phi\}, \{\varphi \vee \psi\}, \{\neg\phi \vee \psi, \neg\psi\}\}$. So the result of *Lex*-subbases based merge is $\mathcal{B} = \{(\gamma, 0.88), (\neg\phi, 0.64), (\neg\phi \vee \psi, 0.49), (\neg\psi, 0.49), (\varphi \vee \psi, 0.5)\}$, which is equivalent to $\mathcal{B}' = \{(\gamma, 0.88), (\neg\phi, 0.64), (\neg\psi, 0.49), (\varphi \vee \psi, 0.5)\}$.

In Example 5, $\mathcal{B}'$ contains two more formulae $(\neg\phi, 0.64)$ and $(\neg\psi, 0.49)$ than $\mathcal{B}$ in Example 4. Although ϕ and $\neg\phi$ have the same priority, both formulae $\neg\phi \vee \psi$ and $\neg\psi$ from the lower level give support to $\neg\phi$. So we still accept $\neg\phi$ and drop ϕ .

5 Postulates for Partial *SMC*-Subbases Based Merge

In this section, we will propose the postulates to characterize the partial *SMC*-subbases based merge by adapting the postulates for partial meet merge in Theorem 1.

First, by Definition 16, the condition *strong consistency* still holds for the partial *SMC*-subbases based merging operator. However, other postulates should be changed because we do not conjoin the knowledge bases in the combination step. There are two main differences between the partial *SMC*-subbases based merging operator and the partial meet merging operator. First, given two possibilistic knowledge bases $\mathcal{B}_1$ and $\mathcal{B}_2$, instead of conjoining them, we take $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$ as the result of combination step. Second, the partial *SMC*-subbases based merging operator is based on a selection function which selects a subset of the set of *SMC*-subbase of Σ , the stratification of $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$. So we have the following postulates for the partial *SMC*-subbases based merging operator.

Theorem 2. Let $\oplus_{SA}$ and $\oplus_{UA}$ be a strongly conjunctive operator and an upper-averaging operator respectively. An operator $\circ: \mathcal{KB} \times \mathcal{KB} \rightarrow \mathcal{KB}$ is a partial *SMC*-subbases based merging operator with regard to $\oplus_{SA}$ and $\oplus_{UA}$ iff for every two possibilistic knowledge bases $\mathcal{B}_1$ and $\mathcal{B}_2$, it satisfies the following conditions:

1. $(\mathcal{B}_1 \circ \mathcal{B}_2)^* \not\models \perp$ (consistency)
2. $\mathcal{B}_1 \circ \mathcal{B}_2 \subseteq \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$ (inclusion)
3. If $(\phi, a) \in \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$ and $(\phi, a) \notin \mathcal{B}_1 \circ \mathcal{B}_2$, then $\exists \mathcal{E}$ such that $\mathcal{B}_1 \circ \mathcal{B}_2 \subseteq \mathcal{E} \subseteq \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$, and $\mathcal{E}^* \not\models \perp$ and $Inc(\mathcal{E} \cup \{(\phi, a)\}) = a$.
4. If $\Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2) = \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}'_1, \mathcal{B}'_2)$, then $\mathcal{B}_1 \circ \mathcal{B}_2 = \mathcal{B}'_1 \circ \mathcal{B}'_2$.

Proof. We only prove the “only if” part, the proof of “if” part is similar to that of Theorem 1 [11].

($\implies$) Conditions 1, 2 and 4 clearly hold. To prove Condition 3, let us assume $(\phi, a) \in \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$ and $(\phi, a) \notin \mathcal{B}_1 \circ \mathcal{B}_2$. Let $\Sigma = S_1 \cup \dots \cup S_n = \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$ and $\phi \in S_k$. By Definition 16, there is some $\Sigma' \in \gamma(\text{SMC}(\Sigma))$ such that $\Sigma' = A_1 \cup \dots \cup A_n$ and $\phi \notin A_k$. Let $\mathcal{E} = \{(\phi, a_i) \in \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2) : \phi \in A_i, 1 \leq i \leq n\}$. It is clear that $\mathcal{B}_1 \circ \mathcal{B}_2 \subseteq \mathcal{E} \subseteq \Delta_{\oplus_{SC}, \oplus_{UA}}(\mathcal{B}_1, \mathcal{B}_2)$, and $\mathcal{E}^* \not\models \perp$. Since $\phi \in S_k - A_k$, by Proposition 1, $\text{Inc}(\mathcal{E} \cup \{(\phi, a)\}) = a$.

By Condition 3 above, a formula which is deleted after merging must be in conflict in $\mathcal{B}_1 \cup \mathcal{B}_2$. Since in the combination step, the weights of free formulae will increase or keep intact, we have the following corollary.

Corollary 1. Let $\oplus_{SA}$ and $\oplus_{UA}$ be a strongly conjunctive operator and an upper-averaging operator respectively. Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two possibilistic knowledge bases, with $\mathcal{B}_1 \circ \mathcal{B}_2$ as the result of merging by a partial SMC-subbase based merging operator with regard to $\oplus_{SA}$ and $\oplus_{UA}$. If (ϕ, a) is a free formula in $\mathcal{B}_1 \cup \mathcal{B}_2$, then $(\phi, b) \in \mathcal{B}_1 \circ \mathcal{B}_2$ and $b \geq a$.

The following Corollary tells us that our partial SMC-subbases based merging operator is a generalization of Fuhrmann’s partial meet merging operator.

Corollary 2. Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two classical knowledge bases. Let $\oplus_{SC}(a, b) = \oplus_{UA}(a, b) = \max(a, b)$. Then the partial SMC-subbases based merging operator $\circ_{\oplus_{SA}, \oplus_{UA}}$ and partial meet merging operator $\circ$ have the same result, i.e. $\mathcal{B}_1 \circ_{\oplus_{SA}, \oplus_{UA}} \mathcal{B}_2 = \mathcal{B}_1 \circ \mathcal{B}_2$.

6 Related Work

Many merging operators have been proposed in the framework of possibilistic logic [3, 4, 5, 6, 20, 21].

The merging operators in [3, 5] are defined semantically and syntactically, i.e. the fusion of two possibilistic knowledge bases are defined semantically by combining their possibility distributions using an operator which is weakly constrained (the result is a new possibility distribution) and then a possibilistic knowledge base is recovered from the new possibility distribution. A problem is, if the result of merging is required to be consistent, *disjunctive* operators are usually chosen, which was criticized to be too cautious.

In [20], we proposed a split-combination method for merging possibilistic knowledge bases which combines formulae in conflict using a *disconjunctive* operator and formulae that are free using a *conjunctive* operator. We showed that this method improves the *disjunctive*-operator based methods because more information was kept after merging. A common point between the partial SMC-subbases based method and the split-combination method is that they both differentiate conflict formulae from free formulae and combines them using different operators. The difference among them is that the partial SMC-subbases based method

resolves inconsistency by deleting some formulae that are in conflict whilst the split-combination method does this by weakening conflict information instead of deleting some of them.

Some inconsistency-tolerant consequence relations were proposed to deal with inconsistency in [4], merging uncertain sources of information is done in two steps: the first step is simply to conjoin the original knowledge bases, and then in the second step, an inconsistency-tolerant consequence will be applied to handle inconsistency. This method does not require to restore consistency after combination. Moreover, it conjoins the original knowledge bases, which is different from our first step of merging.

7 Conclusion

In this paper, we extend Fuhrmann's partial meet merge in possibilistic logic. The merge is processed in two steps: a combination step and an inconsistency handling step. In the combination step, we combine *free* formulae and *conflict* formulae using different operators. The result of combination in the first step may be an inconsistent knowledge base. Then in the inconsistency handling step, we delete those formulae that are in conflict and do not belong to some *strongly maximal consistent* subbase.

We only defined the merging operator for two knowledge bases. A future work is to extend it to merge more than two knowledge bases. A problem with it is that the order of merging will influence the final result. This problem exists in most merging methods. We will deal with this problem by introducing some criterion to decide which two knowledge bases should be merged first. For example, we can choose two knowledge bases which are "closest" to each other to merge each time.

Another important issue is how to choose the appropriate operators in the combination step. We have discussed some criteria to choose operators in [22]. More work will be done on this problem in the future.

References

1. Alchourrón, C.E., Gärdenfors, P., Markinson, D.: On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, vol. 50, pp. 510-530, 1985.
2. Baral, C., Kraus, S., Minker J., and Subrahmanian, V.S.: Combining knowledge bases consisting in first order theories. *Computational Intelligence*, vol. 8, pp. 45-71, 1992.
3. Benferhat, S., Dubois, D., Prade, H., and Williams, M.A.: A practical approach to fusing prioritized knowledge bases. In *Proc. of EPIA'99*, pp. 223-236, 1999.
4. Benferhat, S., Dubois, D., Prade, H.: Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study. Part 2: The prioritized case. In *Logic at work : essays dedicated to the memory of Helena Rasiowa / Ewa Orow.* - New York : Physica-Verlag, pp. 473-511, 1998.

5. Benferhat, S., Dubois, D., Kaci, S., and Prade, H.: Possibilistic merging and distance-based fusion of propositional information. *Annals of Mathematics and Artificial Intelligence*, vol. 34, pp. 217-252, 2002.
6. Benferhat S.; Kaci S.: Fusion of possibilistic knowledge bases from a postulate point of view. *International Journal of Approximate Reasoning*, vol. 33(3), pp. 255-285, 2003.
7. Brewka G.: *Preferred subtheories: an extended logical framework for default reasoning*. In *Proc. of IJCAI'89*, pp. 1043-1048, 2003.
8. Chopra, S., Ghose A., Meyer T.: Non-prioritized ranked belief change. *Journal of Philosophical Logic*, vol. 32(4), pp. 417-443, 2003.
9. Dubois, D., Prade, H.: Epistemic entrenchment and possibilistic logic. *Artificial Intelligence*, vol. 50, pp. 223-239, 1991.
10. Dubois, D., Lang, J., and Prade, H.: Possibilistic logic. In *Handbook of logic in Artificial Intelligence and Logic Programming*, Volume 3. Oxford University Press, pp. 439-513, 1994.
11. Fuhrmann, A.: *An Essay on Contraction*. Stanford University: CSLI Publications & FoLLI. 1996.
12. Gärdenfors P.: *Knowledge in Flux-Modeling the Dynamic of Epistemic States*. Mass.: MIT Press. 1988.
13. Konieczny, S., Pino Pérez, R.: On the logic of merging. In *Proc. of KR'98*, pp. 488-498, 1998.
14. Liberatore, P., and Schaerf, M.: Arbitration (or How to Merge Knowledge Bases). *IEEE Transaction on Knowledge and Data Engineering*, vol. 10(1), pp. 76-90, 1998.
15. Lin, J., Mendelzon, A.: Merging databases under constraints. *International Journal of Cooperative information Systems* vol. 7(1), pp. 55-76, 1998.
16. Hansson S.: Kernel contraction. *Journal of Symbolic Logic*, vol. 59(3), 1994.
17. Hansson S.: Semi-revision. *Journal of Applied Non-Classical Logic*, pp. 151-175, 1997.
18. Hansson S.: A survey of non-prioritized belief revision. *Erkenntnis*, vol. 50, pp. 413-427, 1999.
19. Nelbel, B.: Syntax-Based Approaches to Belief Revision. In *Belief Revision*, P. Gärdenfors (eds.), Cambridge Tracts in Theoretical Computer Science 29, Cambridge University Press, Cambridge, UK, pp. 52-88, 1992.
20. Qi, G., Liu, W., Glass, D. H.: A Split-Combination Method for Merging Inconsistent Possibilistic Knowledge Bases. In *Proc. KR'04*, pp. 348-356, 2004.
21. Qi, G., Liu, W., Glass, D.: Combining Individually Inconsistent Prioritized Knowledge Bases. In *Proc. of NMR'04*, pp. 342-349, 2004.
22. Qi, G., Liu, W., Bell, David.A.: Measureing conflict and agreement in a prioritized knowledge base. In *Proc. of IJCAI'05*, to appear, 2005.
23. Revesz, P. Z.: On the semantics of arbitration. *International Journal of Algebra and Computation*, vol. 7(2), pp. 133-160, 1997.
24. Williams, M. A.: A practical approach to belief revision: reason-based change. In *Proc. of KR'96*, pp. 412-421, 1996.

A Local Fusion Method of Temporal Information

Mahat Khelfallah and Belaïd Benhamou

LSIS - UMR CNRS 6168,
CMI Technopôle de Château Gombert.,
13453 Marseille Cedex 13. France
{mahat, Belaid.Benhamou}@cmi.univ-mrs.fr

Abstract. Information often comes from different sources and merging these sources usually leads to apparition of inconsistencies. Fusion is the operation which consists in restoring the consistency of the merged information by changing a minimum of the initial information. There are many fields or applications where the information can be represented by simple linear constraints. For instance in scheduling problems, some geographic information can be also expressed by linear constraints. In this paper, we are interested in linear constraints fusion in the framework of simple temporal problems (STPs). We propose a fusion method and we experiment with it on random temporal problem instances.

1 Introduction

Information often comes from different sources and merging these sources usually leads to apparition of inconsistencies. Fusion is the operation which consists in restoring the consistency of the merged information by keeping a maximum of the initial information unchanged.

Information fusion is an important area in artificial intelligence. Several fusion methods have been proposed in the literature [12, 1, 8]. Most of them was done in the framework of propositional logic or other logic-based formalisms.

There are many fields or applications where the information can be represented by simple linear constraints. For instance in scheduling problems [7], some geographic information can be also expressed by spatial and/or temporal constraints which are sometimes linear constraints [9, 13, 5, 6].

In this paper, we are interested in linear constraints fusion in the framework of simple temporal problems (STPs). We consider p STPs coming from different agents or sources, and which we want to merge. We consider their union, i.e., the STP S whose set of variables is the union of the sets of variables of the p considered STPs and its set of constraints is obtained from the union of the sets of constraints of the p considered STPs. If the STP S is consistent, then the fusion is done. Otherwise, conflicts appear in the STP S and some constraints of the p STPs have to be corrected. This amounts to restoring the consistency of the STP S by correcting some of its constraints. The restoration of consistency has two main steps: the detection of conflicts, and the elimination of these conflicts. First, we present a general principle of fusion, and then we propose a fusion

method called the *Good Local Fusion* method which starts by detecting a bundle of conflicts of the STP S , and then eliminates these conflicts by correcting some of the constraints involving in them. This process is repeated until the restoration of the consistency of the STP S . We implement and experiment the proposed fusion method with random simple temporal problem instances.

In [5, 6] we have developed revision methods for linear constraints in the framework of a real-world geographic application which is a flooding problem. We extend in this paper the work presented in [5, 6] by first considering more general linear constraints since the flooding problem was represented by a particular STP. Secondly, all the STP constraints are subject to correction in this work whereas only constraints involving the origin variable were corrected when revising the flooding problem. The rest of this paper is organized as follows. In section 2, we recall some background on simple temporal problems STPs. We present, in section 3, the fusion principle and we propose a fusion method in section 4. This method is experimented with random instances of STP, and the obtained results are given in section 5, before concluding in section 6.

2 Background

A *Simple Temporal Problem* (STP) S is defined by $S=(\mathcal{X}, C)$ where $\mathcal{X}$ is a finite set of variables $X_0, \dots, X_n$, having continuous domains. These variables represent temporal events (time points) and X_0 usually represents the origin of time. C is the set of constraints of the form $X_j - X_i \leq a_{ij}$ defined on these variables, where a_{ij} is a scalar. Each constraint expresses a distance between two temporal events. Constraints of the form $X_j - X_i \geq a_{ij}$ can be also represented since $X_j - X_i \geq a_{ij}$ is equivalent to the constraint $X_i - X_j \leq -a_{ij}$. A tuple $x = (x_1, \dots, x_n)$ of real values is a *solution* of the STP S if the instantiation $\{X_1=x_1, \dots, X_n=x_n\}$ satisfies all its constraints. The STP S is *consistent* if and only if it has a solution.

The STP $S=(\mathcal{X}, C)$ is associated with a directed edge-weighted graph, $G_d = (\mathcal{X}, E_d)$, called its *distance graph* where $\mathcal{X}$ the set of vertices is the set of variables of the STP S , and E_d is the set of weighted arcs representing the set of constraints C . Each constraint $X_j - X_i \leq a_{ij}$ of C is represented by the arc $i \rightarrow j^1$, which is weighted by a_{ij} . For more details see [3].

3 Fusion of Constraints

When different STPs are merged some conflicts could appear even if each of the considered STPs is consistent separately. Let $S_1 = (\mathcal{X}_1, C_1)$, $\dots$, $S_p = (\mathcal{X}_p, C_p)$ be p STPs obtained from different sources and defined respectively on the set of variables $\mathcal{X}_1, \dots, \mathcal{X}_p$. We want to fuse these STPs but there are some conflicts between them, i.e., their union is inconsistent. To eliminate these conflicts, some

¹ For simplicity a vertex X_i of the graph G_d is denoted by its index i .

constraints of the STPs S_i ($1 \leq i \leq p$) have to be corrected. This amounts to restore the consistency of the STP obtained from the union of the STPs S_i ($1 \leq i \leq p$). Let $S = (\mathcal{X}, C)$ where $\mathcal{X} = \bigcup_{1 \leq i \leq p} \mathcal{X}_i$ and $C = \bigcup_{1 \leq i \leq p} C_i$ be the STP resulting from the union of the STPs $S_1, \dots, S_p$. Without loss of generality, we suppose in the sequel that the STP S contains at most one constraint between each ordered pair of variables².

The aim of the STP fusion operation is to restore the consistency of an STP by keeping a maximum of its initial constraints unchanged. This means that fusion has to correct as few constraints as possible when restoring the consistency.

Example 1. (Inspired from [3]'s example) We have two persons Nana and Sissi which go to work every morning, and we have two scenarios. The first one informs us that: Nana leaves home before 7:05, and arrives at work between 7:20 and 7:30. Sissi leaves home at most 5 mn after Nana does so and arrives at work at least 10 mn before Nana. The second scenario states that: it takes to Nana at most 10 mn to get at work, whereas it takes to Sissi between 10 and 20 mn to get at work, and Nana arrives at work at most 5 mn after Sissi leaves home. Information of each scenario can be represented by an STP. Let S_1 be the STP representing the first scenario, and S_2 be the STP representing the second one. Let X_1, X_2, X_3, X_4 be the variables representing the temporal events "Nana leaves home", "Nana arrives at work", "Sissi leaves home" and "Sissi arrives at work" respectively. Let X_0 be the variable representing the time 7:00 a.m. The information "Nana leaves home before 7:05" can be represented by the constraint $X_1 - X_0 \leq 5$. "Nana arrives at work between 7:20 and 7:30" is represented by $20 \leq X_2 - X_0 \leq 30$, and so on. We obtain the STPs S_1 and S_2 such that: $S_1 = (\mathcal{X}_1, C_1)$ where $\mathcal{X}_1 = \{X_0, X_1, X_2, X_3, X_4\}$ and $C_1 = \{X_1 - X_0 \leq 5, X_2 - X_0 \leq 30, X_0 - X_2 \leq -20, X_3 - X_1 \leq 5, X_4 - X_2 \leq -10\}$, and $S_2 = (\mathcal{X}_2, C_2)$ where $\mathcal{X}_2 = \{X_1, X_2, X_3, X_4\}$ and $C_2 = \{X_2 - X_1 \leq 10, X_4 - X_3 \leq 20, X_3 - X_4 \leq -10, X_2 - X_3 \leq 5\}$. Each of them is consistent separately. However, their union $S = (\mathcal{X}, C)$ where $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 = \{X_0, X_1, X_2, X_3, X_4\}$ and $C = C_1 \cup C_2 = \{X_1 - X_0 \leq 5, X_2 - X_0 \leq 30, X_0 - X_2 \leq -20, X_3 - X_1 \leq 5, X_4 - X_2 \leq -10, X_2 - X_1 \leq 10, X_4 - X_3 \leq 20, X_3 - X_4 \leq -10, X_2 - X_3 \leq 5\}$ is inconsistent.

Throughout this paper, we will manipulate the STP S instead of the STPs $S_1, \dots, S_p$. Let n and m be respectively the number of variables and the number of constraints of the STP S . Let G_d be the distance graph associated with the STP S . Thus, n and m are also respectively the number of vertices and the number of arcs of the distance graph G_d . Restoring the consistency of the STP S needs performing the following steps: (1) Detection of conflicts of the STP S ; (2) Representation of the detected conflicts; (3) Identification of a subset of constraints whose correction is sufficient to eliminate the detected conflicts of S , and (4) Correction of these constraints. For space lack, the proofs of theorems and propositions of this paper are omitted.

² If there are two constraints $X_i - X_j \leq a_{ij}$ and $X_i - X_j \leq b_{ij}$ in the STP S such that $a_{ij} < b_{ij}$, then only the constraint $X_i - X_j \leq a_{ij}$ is considered in S .

3.1 Detection of Conflicts

The first step of the fusion operation is the detection of conflicts of the STP S . The method which we propose is based on a variant of the well known following result.

Theorem 1. (*[14, 11, 10]*) *An STP is consistent if and only if its corresponding distance graph does not contain negative circuits³.*

We can deduce from Theorem 1 that for restoring the consistency of an STP, we need to remove all the negative circuits of its distance graph. Actually, it is sufficient to remove all the elementary⁴ negative circuits in the distance graph to restore the consistency of an STP. This weakens the conditions of Theorem 1 and results in the following variant on which is based our fusion method.

Theorem 2. *An STP is consistent if and only if its corresponding distance graph does not contain elementary negative circuits.*

The presence of elementary negative circuits in G_d means that the STP S contains conflicts. We associate a conflict with each elementary negative circuit of G_d , such a conflict is defined as follows.

Definition 1. *Let S be an STP and G_d be its distance graph. A conflict of S is a pair (σ, d) where σ is an elementary negative circuit of the distance graph G_d and d is the distance⁵ of the circuit σ .*

Now, we define the *Conflict-Detection* procedure which detects a subset of conflicts of the distance graph. The *Conflict-Detection* procedure is an extension of the *Bellman-Ford* algorithm which computes the shortest paths of a graph [2]. The main idea is to compute for each pair (i, j) of vertices the shortest path from i to j in the distance graph G_d . In particular, if $i = j$ then the procedure will compute the shortest circuit visiting the vertex i .

The *Conflict-Detection* procedure is given in Algorithm 1. It consists in two steps. First, it constructs a matrix $mat^{(0)}$ whose elements are pairs defined by: $mat_{ij}^{(0)} = (p_{ij}, d_{ij})$ where p_{ij} represents a path of length 1 from i to j in G_d , and d_{ij} is the distance of the path p_{ij} . The matrix $mat^{(0)}$ is copied in the matrix mat . This terminates the initialization step. The second step is a loop which computes the shortest paths between each pair (i, j) of vertices. At each iteration l of the loop, a call to the *Shortest-Path-Extension* function, given in Algorithm 2, is made to compute the shortest paths of length l . The loop stops either when a conflict is detected or when the length of the computed paths reaches n .

The *Shortest-Path-Extension* function is based on the following: a shortest path p_{ij} of length l from i to j is composed from a shortest path p_{ik} of length $l-1$

³ A negative circuit is a circuit whose the sum of its arc weights is negative.

⁴ An elementary circuit is a circuit which does not contain any smaller circuit with respect to the number of vertices.

⁵ The distance of a path is the sum of its arc weights.

Algorithm 1. Conflict Detection

Procedure Conflict-Detection(G_d : the distance graph, **Var** $Conf$: a set of conflicts)
Var $mat^{(0)}$, mat : matrices of pairs (path,distance)
Begin
{ *Initialization* }
 $Conf := \emptyset$
 $l := 2$
for $i, j := 1$ to n **do**
 if there is an arc $i \rightarrow j$ in G_d weighted by a_{ij} **then** $mat_{ij}^{(0)} := ((i, j), a_{ij})$
 else if $i \neq j$ **then** $mat_{ij}^{(0)} := (\emptyset, \infty)$
 else $mat_{ij}^{(0)} := (\emptyset, 0)$
 $mat := mat^{(0)}$
{ *Path extension and conflict detection* }
while $l \leq n$ **and** $Conf = \emptyset$ **do**
 begin
 $mat := \text{Shortest-Path-Extension}(mat^{(0)}, mat, Conf)$
 $l := l + 1$
 end
End

from i to k and an arc from k to j . When the *Shortest-Path-Extension* function is called at the iteration l in the loop of the *Conflict-Detection* procedure, it takes as arguments: $mat^{(0)}$ the initial matrix of pairs (path,distance) corresponding to the distance graph G_d , mat the matrix of pairs (path,distance) corresponding to the shortest paths of length $l-1$ in G_d . It returns the matrix mat' corresponding to the shortest paths of G_d of length l . In particular, mat'_{ii} will contain a negative circuit of length l including the vertex i , if it exists. Furthermore, the detected negative circuits are elementary and are added to $Conf$. The *Shortest-Path-Extension* function returns in $Conf$ the set of conflicts whose negative circuits are of length l . If $Conf = \emptyset$, then there is no negative circuit of length l in G_d .

Now, We evaluate the complexity of the *Shortest-Path-Extension* procedure. The initialization phase is performed in $O(n^2)$. The second phase is composed from three loops. Each iteration of the internal loop can be performed at most in $O(n)$ since the path and distance tests are done in a constant time and path concatenation is done at most in $O(n)$. Thus, the second phase can be performed in $O(n^4)$ in the worst case. Therefore, the time complexity of the *Shortest-Path-Extension* procedure is $O(n^4)$ in the worst case.

We evaluate the complexity of the *Conflict-Detection* procedure. The initialization phase can be performed in $O(n^2)$. The procedure performs at most $n-1$ iterations, and the complexity of each iteration is identical to the complexity of the *Shortest-Path-Extension* procedure which is equal to $O(n^4)$. Therefore, the complexity of the *Conflict-Detection* procedure is $O(n^5)$ in the worst case.

Algorithm 2. Shortest Path Extension

Function *Shortest-Path-Extension*($mat^{(0)}$: the initial matrix of (path,distance),
 mat : the matrix of (path,distance) to extend, **Var** $Conf$: the set of detected conflicts):
the extended matrix of (path,distance)
Var mat' the extended matrix of (path,distance)
Begin
{ *Initialization* }
 for $i, j := 1$ to n , $i \neq j$ **do** $mat'_{ij} := (\emptyset, \infty)$
 for $i := 1$ to n **do** $mat'_{ii} := (\emptyset, 0)$
{ *Extension of the paths of mat* }
 for $i, j := 1$ to n **do**
 for $k := 1$ to n **do**
 if ($mat_{ik}.path \neq \emptyset$ **and** $mat_{kj}^{(0)}.path \neq \emptyset$ **and**
 $mat_{ik}.distance + mat_{kj}^{(0)}.distance < mat'_{ij}.distance$) **then**
 begin
 $mat'_{ij} := ((mat_{ik}.path \bullet mat_{kj}^{(0)}.path), (mat_{ik}.distance + mat_{kj}^{(0)}.distance))$
 if ($i = j$) **then** $Conf := Conf \cup mat'_{ii}$
 end
 $Shortest-Path-Extension := mat'$
 End

3.2 Representation of Conflicts

Each conflict of the STP S is identified by a pair (σ, d) where σ is an elementary negative circuit of the distance graph G_d and d is the distance of σ . We recall that each arc $i \rightarrow j$ in G_d , weighted by a_{ij} , represents the constraint $c_{ij} : X_j - X_i \leq a_{ij}$ of the STP S . We define now the notion of conflicting constraint.

Definition 2. Let $S = (\mathcal{X}, C)$ be an STP, G_d its distance graph, and $Conf$ the set of detected conflicts of S . A constraint $c_{ij} \in C$ is a conflicting constraint if and only if there is a conflict $c = (\sigma, d)$ in $Conf$ such that the arc $i \rightarrow j$ belongs to the elementary negative circuit σ of G_d .

Let $ConfConst$ be the function which associates to each conflict $c = (\sigma, d)$ the set of conflicting constraints involved in it. That is, $ConfConst(c) = \{c_{ij} \in C : i \rightarrow j \text{ is an arc of } \sigma\}$.

The set of detected conflicts $Conf$ is represented by a hypergraph which is defined as follows: $H_c = (V, E_c)$ where V is the set of vertices corresponding to the set of all conflicting constraints defined by: $V = \bigcup_{c \in Conf} ConfConst(c)$, and E_c is the set of hyperedges defined as follows: each hyperedge e represents a conflict c of $Conf$ which itself is represented by its conflicting constraints, i.e., $e = ConfConst(c)$. Thus $E_c = \{ConfConst(c) : c \in Conf\}$. H_c is called the *hypergraph of conflicts* of the STP S .

Example 2. The distance graph of the STP S defined in Example 1 is represented in Figure 1.a. The elementary negative circuit $\{(0,1),(1,2),(2,0)\}$ shows a conflict

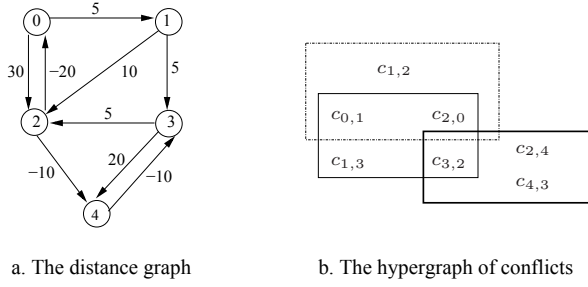


Fig. 1. The graph of distances and the hypergraph of conflicts of the STP S defined in Example 1

between the constraints $c_{0,1}$, $c_{1,2}$ and $c_{2,0}$ (That is, there is a conflict between the statements : "Nana leaves home after 7:05", "It takes to Nana less than 10 mn to get at work" and "Nana arrives at work after 7:20"). This adds the hyperedge $\{c_{0,1}, c_{1,2}, c_{2,0}\}$ in the hypergraph of conflicts. By considering all the elementary negative circuits of the distance graph of Figure 1.a, we obtain the hypergraph of conflicts of Figure 1.b.

3.3 Identification of a Subset of Constraints to Correct

To remove all the detected conflicts, some constraints involved in them have to be corrected. More precisely, we have to identify a subset of constraints whose correction is sufficient to remove all the detected conflicts. In order to guarantee the elimination of all the detected conflicts, at least one conflicting constraint of each conflict has to be corrected. In other words, the intersection of the set of the corrected constraints and the set of conflicting constraints of each conflict has to be not empty. Therefore, the subset of corrected constraints is a *transversal* of the hypergraph of conflicts H_c representing the conflicts of the STP S . The minimization of the corrected constraint number needs to find a minimal transversal of the hypergraph of conflicts H_c . We recall the definitions of a transversal and a minimal transversal.

Definition 3. Let H be a hypergraph defined by $H = (V, E)$. T is a transversal of the hypergraph H if and only if $T \subseteq V$ and for each hyperedge e of E , $T \cap e \neq \emptyset$. A transversal T_m of a hypergraph H is minimal (according to cardinality) if and only if for each transversal T of H , if $|T| \leq |T_m|$ then $|T| = |T_m|$.

Example 3. The hypergraph of conflicts shown in Figure 1.b has many transversals. For instance $T = \{c_{1,2}, c_{1,3}, c_{4,3}\}$. It has seven minimal transversals $T_{m_1} = \{c_{0,1}, c_{3,2}\}$, $T_{m_2} = \{c_{0,1}, c_{4,3}\}$, $T_{m_3} = \{c_{0,1}, c_{2,4}\}$, $T_{m_4} = \{c_{2,0}, c_{3,2}\}$, $T_{m_5} = \{c_{2,0}, c_{4,3}\}$, $T_{m_6} = \{c_{2,0}, c_{2,4}\}$, $T_{m_7} = \{c_{1,2}, c_{3,2}\}$.

Looking for a transversal of a fixed size is an NP-Complete problem [4], and looking for minimal transversal is NP-Hard. We can reduce substantially

Algorithm 3. The Good Transversal**Procedure** Good-Transversal($H_c = (V, E_c)$): the hypergraph of conflicts,**Var** T : a transversal of H_c)**Begin** **for each** vertex v of V **do** Compute the degree $\deg(v)$ **repeat** Select v the vertex having the highest degree in H_c $T := T \cup \{v\}$ **for each** hyperedge e of E_c such that $v \in e$ **do** **begin** remove e from E_c **for each** vertex $w \in e$ **do** $\deg(w) := \deg(w) - 1$ **end** **until** there is no hyperedge in H_c (i.e., $E_c = \emptyset$)**End**

this complexity just by considering a "good" transversal of the hypergraph of conflicts instead of a minimal one. For doing that, we define the *Good-Transversal* procedure (Algorithm 3) based on a heuristic which considers first the vertices having the highest degrees in the hypergraph of conflicts.

Let n_c and m_c be respectively the number of vertices and the number of hyperedges of the hypergraph of conflicts H_c . The *Good-Transversal* procedure starts by computing the degree of each vertex of the hypergraph of conflicts H_c . The complexity of this operation is in $O(m_c n_c)$. The *Good-Transversal* algorithm performs at most n_c iterations since we can at most consider all the vertices of the hypergraph H_c . In each iteration, the vertex having the highest degree in H_c is selected. This operation is performed in $O(n_c)$. The removal of all the hyperedges incident to the vertex v is performed in $O(n_c)$ and the update of the involved vertex degrees can be done in $O(n_c m_c)$. Thus, the complexity of an iteration is in $O(n_c m_c)$. Therefore, the complexity of the *Good-Transversal* algorithm is $O(m_c n_c^2)$ in the worst case.

3.4 Correction of the Conflicting Constraints

Now, we shall see how to perform the corrections. Let $c = (\sigma, d)$ be a conflict of the STP S . The elimination of the conflict c needs the elimination of its associated elementary negative circuit σ . This implies the correction of at least one of the constraints involved in σ , i.e., at least one of the constraints of $\text{ConfConst}(c)$. The following proposition shows how this correction is made.

Proposition 1. *Let S be an STP and $c = (\sigma, d)$ be a conflict of S .*

Let $c_{ij} : X_j - X_i \leq a_{ij}$ be a conflicting constraint of c ($c_{ij} \in \text{ConfConst}(c)$). Replacing the constraint $c_{ij} : X_j - X_i \leq a_{ij}$ by the constraint $X_j - X_i \leq a_{ij} - d$ eliminates the conflict c .

Example 4. In Figure 1.a, the elementary negative circuit $\sigma = \{(0, 1), (1, 2), (2, 0)\}$ whose distance is -5 identifies the conflict $(\sigma, -5)$ between the constraints $X_1 - X_0 \leq 5$, $X_2 - X_1 \leq 10$ and $X_0 - X_2 \leq -20$. This conflict can be removed by either replacing the constraint $X_1 - X_0 \leq 5$ ("Nana leaves home before 7:05") by the constraint $X_1 - X_0 \leq 10$ ("Nana leaves home before 7:10"), or replacing the constraint $X_2 - X_1 \leq 10$ ("it takes to Nana at most 10 mn to get at work") by the constraint $X_2 - X_1 \leq 15$ ("it takes to Nana at most 15 mn to get at work") or replacing the constraint $X_0 - X_2 \leq -20$ ("Nana arrives at work after 7:15") by the constraint $X_0 - X_2 \leq -15$ ("Nana arrives at work after 7:20").

When correcting a constraint no new conflicts are generated and the following theorem states that the correction of the constraints corresponding to a transversal of the hypergraph of conflicts representing the detected conflicts, eliminates these conflicts.

Theorem 3. *Let S be an STP and let $Conf$ be a set of detected conflicts of S . Let H_c be the hypergraph of conflicts representing the set $Conf$. The conflicts of $Conf$ are removed from the STP S if and only if the constraints corresponding to a transversal of the hypergraph of conflicts H_c are corrected.*

4 Good Local Fusion Algorithm

Since the number of elementary negative circuits of the distance graph of an STP is potentially high, the exhaustive detection of conflicts can be impossible. A local handling of the problem seems to be a good alternative. That is, if the conflicts are detected and corrected bundle by bundle, the complexity of the fusion operation decreases. On other hand, if a detected conflict c , of the STP S , involves for instance a constraint c_{ij} , and if this constraint participates in another not yet detected conflict c' , then the correction of the constraint c_{ij} could eliminate the conflict c' . If a bundle of conflicts is detected and corrected, this could eliminate not yet detected conflicts.

The *Good Local Fusion* algorithm consists in detecting a bundle of conflicts, then in eliminating them by correcting the conflicting constraints corresponding to a "good" transversal of the hypergraph of conflicts. It repeats these operations until the restoration of the consistency, in other words, until the removal of all conflicts of S . The *Good-Local-Fusion* procedure is sketched in Algorithm 4.

Theorem 4. *The Good Local Fusion algorithm, applied to G_d , terminates and restores the consistency of the STP S .*

To evaluate the complexity of the *Good-Local-Fusion* algorithm, we proceed step by step. Let m_c be the number of conflicts of the STP S . This number is bounded by the number of possible elementary circuits of the distance graph G_d which is itself bounded by $\sum_{k=2}^n A_n^k$ where $A_n^k = \frac{n!}{(n-k)!}$. At each iteration, n conflicts can be detected (one for each vertex of the distance graph G_d) and

Algorithm 4. The Good Local Fusion**Procedure** *Good-Local-Fusion*(**Var** G_d : the distance graph)**Begin** **repeat** *Detection-Conflict*($G_d, Conf$) Construct H_c the hypergraph of conflicts corresponding to $Conf$ $T := \emptyset$ *Good-Transversal*(H_c, T) Correct the constraints corresponding to the transversal T **until** $Conf = \emptyset$ **End**

then corrected. Thus the number of iterations that the *Local-Fusion* algorithm performs is bounded by $\frac{m_c}{n}$. In practice, the number of iterations does never reach the worst case, since the correction of a conflict could eliminate other not yet detected conflicts.

Now, we evaluate the complexity of each iteration. The complexity of the *Conflict-Detection* procedure is $O(n^5)$ in the worst case. Since the number of detected conflicts at each iteration is at most n and each conflict can involve at most n conflicting constraints, then the construction of the hypergraph of conflicts H_c corresponding to $Conf$ is in $O(n^2)$. The transversal T of the hypergraph of conflicts H_c is computed in $O(m'_c n'^2_c)$ where m'_c is the number of handled conflicts and n'_c is the number of conflicting constraints. The number m'_c is bounded by n because at most n conflicts are handled in each iteration and n'_c is bounded by n^2 . Thus, the good transversal search is performed in $O(n^5)$ in the worst case. The correction of the constraints of T is performed in $O(n)$ since at most n constraints can be corrected. Therefore, each iteration is performed in $O(n^5 + n^2 + n^5 + n)$, i.e., $O(n^5)$ in the worst case. The *Good Local Fusion* algorithm performs at most $\frac{m_c}{n}$ iterations. Therefore, its complexity is $O(m_c n^4)$ in the worst case.

5 Experimental Results

The fusion algorithm presented in this paper is implemented in *C* and tested on randomly generated problems. The program is run on a P4 with 2.2 MHz and 512 Mb of RAM.

For the generation of the p STPs S_i , $1 \leq i \leq p$, to fuse, it is sufficient to generate only their union. Generation of random STPs is based on two parameters: the number of variables n , and the constraint density d which is a ratio of the number of constraints to the number of possible constraints, $d = \frac{\text{number of constraints}}{n(n-1)}$. The tightness t of the constraints is represented by the interval $[a, b]$ where the constraint weights are generated. A sample of 50 problems is generated for each pair (n, d) and the measures are taken in average. The experimental results ob-

Table 1. Experimental results obtained by the application of the Good-Local-Fusion algorithm on random STP instances having $n = 20, 50, 100, 200$ variables

n	# conflicts			# corrected const.			# iterations			Run. time (s)		
	Density			Density			Density			Density		
	0.2	0.5	0.8	0.2	0.5	0.8	0.2	0.5	0.8	0.2	0.5	0.8
20	46	168	235	22	91	141	9	20	25	0	0	0
50	486	1066	1454	275	712	1026	31	48	60	5	5	6
100	2000	4044	5708	1283	2961	4357	54	84	108	99	106	113
200	7638	15885	23973	5458	12471	18876	91	152	215	461	537	691

tained by the application of the *Good Local Fusion* algorithm on random STP instances are shown in Table 1.

We can see in Table 1 that when the density grows, the number of conflicts grows. This increases the number of corrected constraints, the number of iterations and the running time. The number of corrected constraints in all cases is smaller than the number of detected conflicts. This is due to the minimization policy applied in the good local fusion method when computing a "good" transversal of the hypergraph of conflicts. We can see that the *Good Local Fusion* algorithm succeeds in fusing large scale STPs (200 variables) in reasonable time.

6 Conclusion

In this paper, we investigated fusion of Simple Temporal Problems STPs. That is, giving a set of STPs to merge, we considered the STP S resulting from the union of these STPs. If the STP S is consistent, then the fusion is done. Otherwise, the consistency of S had to be restored, and this paper was focusing on this case. First, we presented the general principle of a fusion method which consists in detecting conflicts of the STP S . This operation is based on the detection of elementary negative circuits of the distance graph associated with the STP S . The second step is the representation of the detected conflicts by an hypergraph of conflicts whose vertices represent the conflicting constraints and each hyperedge represents the set of conflicting constraints of a conflict. After that, the identification of a subset of constraints whose correction eliminates the detected conflicts amounts to searching a transversal of the hypergraph of conflicts. The final step is the correction of the constraints corresponding to the computed transversal. A fusion method which is called the *Good Local Fusion* method is proposed in this paper.

Two justifications motivate the local fusion strategy. The first one is the high complexity of an exhaustive detection of the conflicts. The second justification is related to the conflict nature. If a bundle of conflicts is detected and corrected, this could eliminate not yet detected conflicts which speeds up the fusion operation. Experiments have shown that the proposed fusion method succeeds in

handling STPs having more than 200 variables with high constraint densities in reasonable time.

In the future, we hope to extend this work to handle prioritized fusion of temporal constraint problems. The priority can represent either preferences on STPs or preferences on the constraints of the same STP. We hope also to handle the fusion of disjunctive temporal problems.

References

1. S. Benferhat, D. Dubois, and H. Prade. Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study part 1: The flat case. *Studia Logica*, 58:17–45, 1997.
2. T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, 1990.
3. R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.
4. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and co., 1979.
5. M. Khelfallah and B. Benhamou. Geographic information revision based on constraints. In *Proc. of the 14th Eur. Conf. on Artificial Intelligence, ECAI'04*, pages 828–832, 2004.
6. M. Khelfallah and B. Benhamou. Two revision methods based on constraints: Application to a flooding problem. In *Proc. of the 7th Int. Conf. of Artificial Intelligence and Symbolic Computation AISC'04*, volume 3249 of *LNAI*, pages 265–270, 2004.
7. R. Kolisch and R. Padman. An integrated survey of deterministic project scheduling. *Omega*, 29:249–272, 2001.
8. S. Konieczny, J. Lang, and P. Marquis. Distance based merging: A general framework and some complexity results. In *Proc. of the 8th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'02)*, pages 97–108, 2002.
9. G. Kuper, G. L. Libkin, and J. Paradaens, editors. *Constraint Databases*. Springer-Verlag, 2000.
10. C.E. Leiserson and J.B. Saxe. A mixed-integer linear programming problem which is efficiently solvable. In *Proc. of the 21st annual Allerton conference on Communications, Control, and Computing*, pages 204–213, 1983.
11. Y.Z. Lia and C.K. Wong. An algorithm to compact a vlsi symbolic layout with mixed constraints. In *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, volume 2, pages 62–69, 1983.
12. J. Lin and A. Mendelzon. *Dynamic Worlds: From the Frame Problem to Knowledge Management*, volume 12 of *Applied Logic Series*, chapter Knowledge Base Merging by Majority. Kluwer, 1999.
13. P. Rigaux, M. Scholl, and A. Voisard. *Spatial Databases with Application to GIS*. Morgan Kaufmann, 2002.
14. R. Shostak. Deciding linear inequalities by computing loop residues,. *Journal of ACM*, 28(4):769–779, 1981.

Mediation Using m -States

Thomas Meyer¹, Pilar Pozos Parra², and Laurent Perrussel³

¹ National ICT Australia and CSE, UNSW, Sydney, Australia

`thomas.meyer@nicta.com.au`, `tmeyer@cse.unsw.edu.au`

² Department of Computing, Macquarie University, Sydney, Australia

`pilar@ics.mq.edu.au`

³ IRIT-Université Toulouse 1, Manufacture des Tabacs 21, allée de Brienne,

F-31042 Toulouse Cedex - France

`laurent.perrussel@univ-tlse1.fr`

Abstract. *Model-based* propositional belief merging operators are constructed from distances between the interpretations, or states, of the logic under consideration. In this paper we extend the notion of a distance between interpretations to generalised versions of propositional interpretations referred to as m -states. m -states allow for the definition of m -merging operators, which are generalisations of classical model-based merging operators. We show how m -merging, combined with appropriate measures of satisfaction, can be used to construct a logical framework for *agent mediation*: a process of intervening between parties with conflicting demands to facilitate a compromise.

1 Introduction

Belief merging is concerned with the process of combining the information contained in a set of (possibly inconsistent) belief bases obtained from different sources to produce a single consistent belief base [1, 2, 3]. Techniques for solving this problem vary considerably. However, most can be placed in one of two main families of merging operators [2, 4]: (1) *model-based* operators which obtain a belief base from a set of interpretations selected with the help of a distance measure on interpretations and an aggregation function, and (2) *syntax-based* operators which select some consistent set of formulas in the union of the bases.

In this paper a new class of model-based merging operators is presented. We introduce the notion of an m -state, and use it to generalise classical model-based merging to obtain the class of m -merging operators. m -states are generalisations of classical interpretations, and it is possible to define generalised distances between m -states and interpretations, belief bases and belief sets, given the classical definition of a distance. Whenever m -states correspond to propositional interpretations, the propositional merging operators found in the literature are recovered. One of the main advantages of m -merging is that it provides a very natural way to define a framework for logic-based agent *mediation*, a topic which

has, thus far, received little attention. Mediation is the process of actively intervening between parties with conflicting demands to ensure that they are able to reach a compromise acceptable to all. We give a formal definition of the mediation framework and discuss some of its properties.

The rest of the paper is organised as follows. After providing some technical preliminaries, Section 2 reviews model-based merging. Then Section 3 introduces the notion of an m -state and the associated m -merging operators, while Section 4 proposes a mediation framework based on m -merging operators. Section 5 discusses related work, and Section 6 concludes with a discussion of future work.

2 Model-Based Merging Operators

Let $\mathcal{L}$ be a propositional language generated by n propositional atoms $\{p_1, p_2, \dots, p_n\}$ and with the usual propositional connectives. A *state* or *interpretation* is a function from $\{p_1, p_2, \dots, p_n\}$ to $\{0, 1\}$, with 0 denoting falsity and 1 denoting truth. Satisfaction of a sentence φ in a state w is determined in the usual truth-functional way. We frequently denote a state w by a vector of the form $l_1 l_2 \dots l_n$ where $l_i = p_i$ if $w(p_i) = 1$ and $l_i = \bar{p}_i$ if $w(p_i) = 0$. A *model* of a formula φ is a state w such that φ is satisfied by w , and $\text{mod}(\varphi)$ is the set of all the models of φ . The set of all states is denoted by $\mathcal{W}$ and classical entailment by $\models$. A *belief set* $E = \{K_1, \dots, K_N\}$ is a multi-set of *consistent* belief bases where each member K_i is a propositional formula representing the beliefs of agent i .

In the model-based approach to merging the models of the belief base obtained when merging the bases in a belief set E , given some integrity constraints μ , are the models of μ that are as close as possible to E . Closeness is defined in terms of distances between states. The integrity constraints are formulas which must be entailed by the merged base [5, 6, 7, 1, 8]. For the purposes of this paper we ignore the introduction of integrity constraints, as was initially done in [1]. Clearly this corresponds to the special case where $\mu \equiv \top$. Our reason for doing away with integrity constraints is purely for technical convenience. If need be they can be accommodated in our framework.

Definition 1 (Distance). *A pseudo-distance between states is a function $d : \mathcal{W} \times \mathcal{W} \mapsto \mathbb{N}$ such that for any $w, w' \in \mathcal{W}$, $d(w, w') = d(w', w)$ and $d(w, w') = 0$ iff $w = w'$. A pseudo-distance is a distance iff it also satisfies the triangle inequality: $d(v, x) \leq d(v, w) + d(w, x)$.*

Distances between states are used to define distances between states and consistent belief bases as follows: $d(w, K) = \min\{d(w, w') \mid w' \in \text{mod}(K)\}$. Thus, the distance between a state w and a consistent base K is the minimal distance between w and the models of K . The distance between a state w and a belief set E is defined in terms of an *aggregation function* which combines the distances between w and the bases in E in a principled way.

Definition 2 (Aggregation function). *An aggregation function is a total function $f : \bigcup(\mathbb{R}^{+n}) \mapsto \mathbb{R}^+$ such that for any $x_1, \dots, x_n, x, y \in \mathbb{R}^+$:*

1. if $x \leq y$ then $f(x_1, \dots, x, \dots, x_n) \leq f(x_1, \dots, y, \dots, x_n)$ (non-decreasingness)
2. $f(x_1, \dots, x_n) = 0$ iff $x_1 = \dots = x_n = 0$ (minimality)
3. $f(x) = x$ (identity)

An anonymous aggregation function is an aggregation function which additionally satisfies the anonymity property: $f(x_1, \dots, x_n) = f(x_{\pi(1)}, \dots, x_{\pi(n)})$ where π is any permutation of $\{1, \dots, n\}$.

We are interested in anonymous aggregation functions only. Given an anonymous aggregation function f , the distance between a state w and a belief set E is: $d_f(w, E) = f(d(w, K_1), \dots, d(w, K_N))$. The result of merging the bases in E is obtained by selecting as its models those states *closest* to E in terms of d_f .

Definition 3 (Merging operator). Let d be a distance between states and f be an anonymous aggregation function. Define the total preorder $\preceq_E$ on $\mathcal{W}$ as: $v \preceq_E w$ iff $d_f(v, E) \leq d_f(w, E)$. For every belief set E , $\Delta_{d,f}(E)$ is such that $\text{mod}(\Delta_{d,f}(E)) = \{v \in \mathcal{W} \mid \forall w \in \mathcal{W}, v \preceq_E w\}$.

Distance measures commonly used include the following. The *Hamming* (or *Dalal*) distance d_h [9] between states is defined by the number of atoms on which the states differ, i.e. $d_h(w, w') = \sum_{i=1}^n |l_i - l'_i|$, where $|X|$ denotes the absolute value function. The drastic distance $d_d(w, w')$ assigns a 0 when $w = w'$ and 1 otherwise [2, 10]. Two commonly used anonymous aggregation functions are

1. max: $f(x_1, \dots, x_n) = \max\{x_1, \dots, x_n\}$ [6, 1], and
2. Σ : $f(x_1, \dots, x_n) = \sum_{i=1}^n x_i$ [5, 7, 1].

The anonymous aggregation function max takes the distance between a state w and a belief set E to be the maximum of the distances between w and the belief bases in E , while the anonymous aggregation function Σ takes it to be the sum of the distances between w and the belief bases in E .

Example 1. Revesz [5] proposes the following scenario. A teacher asks three students which among three languages, SQL, O₂ and Datalog, they would like to learn. Let s , o and d be the propositional letters used to denote the desire to learn SQL, O₂ and Datalog, respectively. The first student only wants to learn SQL or O₂, the second wants to learn one of Datalog or O₂ but not both, and the third wants to learn all three languages. So we have $E = \{K_1, K_2, K_3\}$ with $K_1 = (s \vee o) \wedge \neg d$, $K_2 = (\neg s \wedge d \wedge \neg o) \vee (\neg s \wedge \neg d \wedge o)$, and $K_3 = s \wedge d \wedge o$. Table 1 gives a summary of the results obtained using the Hamming distance applied to the anonymous aggregation functions max and Σ . It follows that $\Delta_{d_h, \text{max}}(E) \equiv (s \wedge (d \leftrightarrow \neg o)) \vee (o \wedge (s \leftrightarrow \neg d))$, $\Delta_{d_h, \Sigma}(E) \equiv (s \wedge \neg d \wedge o) \vee (\neg s \wedge \neg d \wedge o)$. Table 2 gives a summary of the results obtained using the drastic distance applied to the anonymous aggregation functions max and Σ . And it follows that $\Delta_{d_d, \text{max}}(E) \equiv \top$ and $\Delta_{d_d, \Sigma}(E) \equiv \neg s \wedge \neg d \wedge o$.

Table 1. $\Delta_{h,\max}$ and $\Delta_{h,\Sigma}$ applied to Example 1

w	$d_h(w, K_1)$	$d_h(w, K_2)$	$d_h(w, K_3)$	$d_{\max}(w, E)$	$d_\Sigma(w, E)$
sdo	1	2	0	2	3
$sd\bar{o}$	1	1	1	1	3
$s\bar{d}o$	0	1	1	1	2
$s\bar{d}\bar{o}$	0	2	2	2	5
$\bar{s}do$	1	1	1	1	3
$\bar{s}d\bar{o}$	1	0	2	2	3
$\bar{s}\bar{d}o$	0	0	2	2	2
$\bar{s}\bar{d}\bar{o}$	1	1	3	3	5

Table 2. $\Delta_{d_d,\max}$ and $\Delta_{d_d,\Sigma}$ applied to Example 1

w	$d_d(w, K_1)$	$d_d(w, K_2)$	$d_d(w, K_3)$	$d_{\max}(w, E)$	$d_\Sigma(w, E)$
sdo	1	1	0	1	2
$sd\bar{o}$	1	1	1	1	3
$s\bar{d}o$	0	1	1	1	2
$s\bar{d}\bar{o}$	0	1	1	1	2
$\bar{s}do$	1	1	1	1	3
$\bar{s}d\bar{o}$	1	0	1	1	2
$\bar{s}\bar{d}o$	0	0	1	1	1
$\bar{s}\bar{d}\bar{o}$	1	1	1	1	3

3 Model-Based m -Merging Operators

In this section we propose a merging framework similar to that described in Section 2. But in our case the distances are based on entities referred to as m -states. m -states are generalisations of states in which the truth values of 0 or more atoms are forgotten or ignored.

Definition 4 (m -state). An m -state is a vector of the form $l_{j_1}, \dots, l_{j_m}$, where $m \leq n$, $1 \leq j_1 < \dots < j_m \leq n$ and $l_{j_i} \in \{p_{j_i}, \bar{p}_{j_i}\}$.

So, for $n = 3$ the following are examples of 2-states: $p_1\bar{p}_3$, $\bar{p}_2p_3$, and $\bar{p}_1\bar{p}_2$. Note that the n -states (where no atoms are ignored) are exactly the states in $\mathcal{W}$. The set of all the m -states is denoted by $\mathcal{W}^m$. For each m -state w^m there is an associated set of states defined in the obvious way (the “models” of the m -state), which we shall denote by $\text{mod}(w^m)$. This can be extended to sets of m -states in the obvious way as well (the union of the states corresponding to the elements of the set of m -states).

Example 2. For $n = 3$, $\mathcal{W} = \mathcal{W}^3 = \{\bar{p}_1\bar{p}_2\bar{p}_3, \bar{p}_1\bar{p}_2p_3, \bar{p}_1p_2\bar{p}_3, \bar{p}_1p_2p_3, p_1\bar{p}_2\bar{p}_3, p_1\bar{p}_2p_3, p_1p_2\bar{p}_3, p_1p_2p_3\}$, $\mathcal{W}^2 = \{\bar{p}_1\bar{p}_2, \bar{p}_1p_2, p_1\bar{p}_2, p_1p_2, \bar{p}_1\bar{p}_3, \bar{p}_1p_3, p_1\bar{p}_3, p_1p_3, \bar{p}_2\bar{p}_3, \bar{p}_2p_3, p_2\bar{p}_3, p_2p_3\}$, and $\mathcal{W}^1 = \{\bar{p}_1, p_1, \bar{p}_2, p_2, \bar{p}_3, p_3\}$. For the 2-state $w^2 = p_1\bar{p}_3$, we have $\text{mod}(w^2) = \{p_1p_2\bar{p}_3, p_1\bar{p}_2\bar{p}_3\}$. For the 1-state $w^1 = p_1$, we have $\text{mod}(w^1) = \{p_1\bar{p}_2\bar{p}_3, p_1\bar{p}_2p_3, p_1p_2\bar{p}_3, p_1p_2p_3\}$.

Our definition of m -merging is distance-based, just like the classical case discussed in Section 2, but we use distances based on m -states instead of states. Firstly, we need to generalise the definition of the distance between two states to one of a distance between an m -state and a state. To do this is easy. Since we can think of an m -state as a set of states, the distance between an m -state w^m and a state w is simply the smallest distance between w and any of the models of w^m . In fact, we can think of a distance measure between m -states and states as being *determined* by a distance measure between states. Once we have a distance measure between m -states and states, it is easy to obtain a distance measure between m -states and belief bases, and between m -states and belief sets.

Definition 5 (Distance). *Let d be a distance between states. The distance measure between m -states and states determined by d is: $d(w^m, w) = \min\{d(w', w) \mid w' \in \text{mod}(w^m)\}$. The distance measure between m -states and belief bases is $d(w^m, K) = \min\{d(w^m, w) \mid w \in \text{mod}(K)\}$. And, given an anonymous aggregation function f , the distance measure between m -states and belief sets is: $d_f(w^m, E) = f(d(w^m, K_1), \dots, d(w^m, K_N))$.*

And once we have a distance measure between m -states and belief sets, it can be used to define an m -merging operator in the obvious way.

Definition 6 (m -merging operator). *Let d be a distance between states and f be an anonymous aggregation function. Define the total preorder $\preceq_E$ on $\mathcal{W}^m$ as: $v^m \preceq_E w^m$ iff $d_f(v^m, E) \leq d_f(w^m, E)$. For every belief set E , $\Delta_{d,f}^m(E)$ is such that $\text{mod}(\Delta_{d,f}^m(E)) = \text{mod}\{v^m \in \mathcal{W}^m \mid \forall w^m \in \mathcal{W}^m, v^m \preceq_E w^m\}$.*

Observe that an m -merging operator is defined in terms of an anonymous aggregation function f and a distance measure d on *states* (not m -states). So every anonymous aggregation function f and distance measure d on states define a whole class of m -merging operators $\Delta_{d,f}^m$, one for every $m \in \{1, \dots, n\}$. Moreover, $\Delta_{d,f}^n$ is a classical merging operator as defined in Definition 3.

Example 3. We continue with Example 1, again considering the drastic and Hamming distances applied to the anonymous aggregation functions $\max$ and Σ , but now for m -merging operators. Since n -merging corresponds to classical merging, it follows that $\Delta_{d_h, \max}^3(E) \equiv (s \wedge (d \leftrightarrow \neg o)) \vee (o \wedge (s \leftrightarrow \neg d))$, $\Delta_{d_h, \Sigma}^3(E) \equiv (s \wedge \neg d \wedge o) \vee (\neg s \wedge \neg d \wedge o)$, $\Delta_{d_d, \max}^3(E) \equiv \top$ and $\Delta_{d_d, \Sigma}^3(E) \equiv \neg s \wedge \neg d \wedge o$. Next we consider m -merging for $m = 2$. Table 3 contains a summary of the results for 2-merging obtained from the Hamming distance applied to $\max$ and Σ . From this it follows that $\Delta_{d_h, \max}^2(E) \equiv s \vee d \vee (\neg d \wedge o)$ and $\Delta_{d_h, \Sigma}^2(E) \equiv o$. And Table 4 contains a summary of the results for 2-merging obtained from the drastic distance applied to $\max$ and Σ . From this it follows that $\Delta_{d_d, \max}^2(E) \equiv \top$ and $\Delta_{d_d, \Sigma}^2(E) \equiv o \vee (\neg s \wedge \neg d)$. Furthermore, Table 5 contains a summary of the results for 1-merging obtained from the drastic and Hamming distances applied to $\max$ and Σ . (In this case the results obtained from the Hamming and drastic distances are identical). From this it follows that $\Delta_{d_h, \max}^1(E)$, $\Delta_{d_d, \max}^1(E)$, $\Delta_{d_h, \Sigma}^1(E)$, and $\Delta_{d_d, \Sigma}^1(E)$ are all equivalent to o .

Table 3. $\Delta_{d_h, \max}^2$ and $\Delta_{d_h, \Sigma}^2$ applied to Example 3

w^2	$d_h(w^2, K_1)$	$d_h(w^2, K_2)$	$d_h(w^2, K_3)$	$d_{\max}(w^2, E)$	$d_{\Sigma}(w^2, E)$
sd	1	1	0	1	2
$s\bar{d}$	0	1	1	1	2
so	0	1	0	1	1
$s\bar{o}$	0	1	1	1	2
$\bar{s}d$	1	0	1	1	2
$\bar{s}\bar{d}$	0	0	2	2	2
$\bar{s}o$	0	0	1	1	1
$\bar{s}\bar{o}$	1	0	2	2	3
do	1	1	0	1	2
$d\bar{o}$	1	0	1	1	2
$\bar{d}o$	0	0	1	1	1
$\bar{d}\bar{o}$	0	1	2	2	3

Table 4. $\Delta_{d_d, \max}^2$ and $\Delta_{d_d, \Sigma}^2$ applied to Example 3

w^2	$d_d(w^2, K_1)$	$d_d(w^2, K_2)$	$d_d(w^2, K_3)$	$d_{\max}(w^2, E)$	$d_{\Sigma}(w^2, E)$
sd	1	1	0	1	2
$s\bar{d}$	0	1	1	1	2
so	0	1	0	1	1
$s\bar{o}$	0	1	1	1	2
$\bar{s}d$	1	0	1	1	2
$\bar{s}\bar{d}$	0	0	1	1	1
$\bar{s}o$	0	0	1	1	1
$\bar{s}\bar{o}$	1	0	1	1	2
do	1	1	0	1	2
$d\bar{o}$	1	0	1	1	2
$\bar{d}o$	0	0	1	1	1
$\bar{d}\bar{o}$	0	1	1	1	2

Table 5. $\Delta_{d, \max}^1$ and $\Delta_{d, \Sigma}^1$ applied to Example 3

w^1	$d(w^1, K_1)$	$d(w^1, K_2)$	$d(w^1, K_3)$	$d_{\max}(w^1, E)$	$d_{\Sigma}(w^1, E)$
s	0	1	0	1	1
$\bar{s}$	0	0	1	1	1
d	1	0	0	1	1
$\bar{d}$	0	0	1	1	1
o	0	0	0	0	0
$\bar{o}$	0	0	1	1	1

3.1 Properties of m -Merging

In [1, 8] Konieczny and Pino-Pérez proposed the basic properties below for merging operators, rephrased without reference to integrity constraints (Two belief

sets E_1 and E_2 are said to be equivalent, denoted by $E_1 \approx E_2$, iff there is a bijection g from E_1 to E_2 such that $K \equiv g(K)$ for every base K in E_1 .

- (A1) $\Delta(E) \not\models \perp$
- (A2) If $\bigwedge E \not\models \perp$ then $\bigwedge E \equiv \Delta(E)$
- (A3) If $E_1 \approx E_2$ then $\Delta(E_1) \equiv \Delta(E_2)$
- (A4) If $\Delta(\{K_1, K_2\}) \wedge K_1 \not\models \perp$ then $\Delta(\{K_1, K_2\}) \wedge K_2 \not\models \perp$
- (A5) $\Delta(E_1) \wedge \Delta(E_2) \models \Delta(E_1 \sqcup E_2)$
- (A6) If $\Delta(E_1) \wedge \Delta(E_2) \not\models \perp$ then $\Delta(E_1 \sqcup E_2) \models \Delta(E_1) \wedge \Delta(E_2)$

Clearly m -merging satisfies (A1), which simply requires of the result of merging to be consistent. In general, m -merging does not satisfy (A2), which requires that merging be equivalent to conjunction if the belief set is consistent. For example, if $K_1 \equiv p_1$ and $K_2 \equiv p_2$, it is easily shown that $\Delta_{d,f}^1 \equiv p_1 \vee p_2$ for any distance d and any anonymous aggregation function f , even though (A2) requires that the result be equivalent to $p_1 \wedge p_2$. In addition, the identity requirement $f(x) = x$ on aggregation functions suggests that merging ought to satisfy the special case of (A2) where E contains a single base: $\Delta(\{K\}) \equiv K$. But this condition also does not hold for m -merging. For example, for $n = 2$, $\Delta_{d,f}^1(\{p_1 \wedge p_2\}) = p_1 \vee p_2$ for any distance function d and anonymous aggregation function f . However, one half of (A2) can be shown to hold.

Proposition 1. *For any $\Delta_{d,f}^n$, $\bigwedge E \not\models \perp$ implies $\bigwedge E \models \Delta_{d,f}^n(E)$.*

Moreover, as will be shown in Section 4, our proposed application of m -merging is such that (A2) will be satisfied.

The next property, (A3), is a version of Dalal's principle of the Irrelevance of Syntax [9]. It is easily shown that m -merging satisfies it. m -merging also satisfies the property (A4), known as fairness.

Proposition 2. *For every distance function d and every anonymous aggregation function f , the m -merging operator $\Delta_{d,f}^m$ satisfies (A4).*

(A5) and (A6) establish connections between, on the one hand, the result obtained when merging each of two belief sets and then taking their conjunction, and on the other hand, the result obtained when first combining the two belief sets and then performing a single merge. Together the two properties require that these two results be equivalent, provided that the conjunction referred to is not inconsistent. In general, m -merging does not satisfy (A5). To show this, we need look no further than Example 3. From Table 5 it can be shown that, with d being either the Hamming or the drastic distance and f being either max or Σ , it follows that $\Delta_{d,f}^1(\{K_1, K_2\}) \equiv \top$ and that $\Delta_{d,f}^1(\{K_3\}) \equiv s \vee d \vee o$. So $\Delta_{d,f}^1(\{K_1, K_2\}) \wedge \Delta_{d,f}^1(\{K_3\}) \equiv s \vee d \vee o$, but, as we know from Example 3, $\Delta_{d,f}^1(\{K_1, K_2, K_3\}) \equiv o$, thus violating (A5). Also, m -merging does not satisfy (A6). It was shown in [1] that $\Delta_{d,\max}^n$ does not satisfy (A6). By turning to Example 3 we can show that $\Delta_{d,\Sigma}^n$ also does not satisfy (A6). From Table 4 we can determine that (with d_d being the drastic distance) it follows that $\Delta_{d_d,\Sigma}^2(\{K_1, K_2\}) \equiv (\neg s \wedge \neg d) \vee (\neg s \wedge o) \vee (\neg d \wedge o)$ and $\Delta_{d_d,\Sigma}^2(\{K_3\}) \equiv (s \wedge d) \vee$

$(s \wedge o) \vee (d \wedge o)$. So $\Delta_{d,\Sigma}^2(\{K_1, K_2\}) \wedge \Delta_{d,\Sigma}^2(\{K_3\}) \equiv o \wedge (s \leftrightarrow \neg d)$. Furthermore, we know from Example 3 that $\Delta_{d,\Sigma}^2(\{K_1, K_2, K_3\}) \equiv o \vee (\neg s \wedge \neg d)$, which does not entail $\Delta_{d,\Sigma}^2(\{K_1, K_2\}) \wedge \Delta_{d,\Sigma}^2(\{K_3\}) \equiv o \wedge (s \leftrightarrow \neg d)$.

While the results above show that m -merging satisfies some merging properties, they also indicate that it falsifies some properties that are basic to classical merging; properties such as (A2). But our intention is not to view m -merging as a class of classical merging operators, as will be outlined in the next section.

4 Mediation as m -Merging

In this section we show how the class of m -merging operators can be used to define logic-based agent *mediation*. Mediation, as we define it, is similar in spirit to the logic-based approaches to *negotiation* of [11, 12, 13, 14, 15, 16]. Essentially, the scenario is this. Each one of a finite number of agents has some demands, represented as sentences in a logic. The purpose of the exercise is to get all agents to agree on a consistent set of demands acceptable to all agents. But whereas negotiation-style approaches provide a mechanism where agents themselves weaken, or modify, their demands to reach an agreement, our approach postulates the existence of an independent mediator whose task it is to present viable alternatives to all the parties involved. What m -merging provides is a detailed description of the *process* of presenting these alternatives to the parties involved. Let us represent the demands of the agents in a belief set E . Furthermore, we assume that we have at our disposal a fixed method for performing m -merging. That is, we need to fix an anonymous aggregation function f and a distance measure d on states. As we have seen in Definition 6, such a d and f uniquely determines an m -merging operator $\Delta_{d,f}^m$ for $m \in \{1, \dots, n\}$.

The mediation process is started with the mediator presenting the classical merging result $\Delta_{d,f}^1(E)$ as the first alternative to the agents involved, and then determining to what extent each source is satisfied with the result. If all sources are sufficiently satisfied, the process terminates with $\Delta_{d,f}^n(E)$ as the result. If not, the same process is repeated, but now the mediator presents $\Delta_{d,f}^{n-1}(E)$ instead of $\Delta_{d,f}^n(E)$ as the proposed compromise. In general, whenever the agents involved are not sufficiently satisfied with $\Delta_{d,f}^m(E)$ as a solution, the mediator proposes $\Delta_{d,f}^{m-1}(E)$ as the next alternative. The process continues until all sources are sufficiently satisfied or the mediator has run out of viable alternatives. In the latter case the result defaults to the tautology $\top$.

Before defining the mediation framework formally, it is necessary to say more about what it means for agents to be satisfied with the result of the mediation. The level of satisfaction of an individual agent with a merged base $\Delta_{d,f}^m(E)$ relative to its own base K will be modelled in the style of [10]: as a real value in the interval $[0, 1]$, with 1 denoting total satisfaction and 0 total dissatisfaction. So, a *satisfaction index* is a function $i : \mathcal{L} \times \mathcal{L} \mapsto [0, 1]$, and $i(K, K_\Delta)$ indicates the level of satisfaction of an agent with the proposed outcome K_Δ , given its own demands K . The satisfaction indices considered in [10] are: 1) Weak drastic index: $i_w(K, K_\Delta) = 1$ if $K \wedge K_\Delta$ is consistent, and 0 otherwise; 2)

Strong drastic index: $i_s(K, K_\Delta) = 1$ if $K_\Delta \models K$, and 0 otherwise; and 3) Probabilistic index: $i_p(K, K_\Delta) = (\#(\text{mod}(K) \cap \text{mod}(K_\Delta)) / (\# \text{mod}(K_\Delta)))$, where $\#X$ indicates the cardinality of the set X (observe that K_Δ is intended to be the result of a merging operator and is therefore always consistent). Clearly there are numerous other acceptable satisfaction indices as well. For example, one could use the Hamming distance d_H between bases to define a Hamming satisfaction index as follows: $i_H(K, K_\Delta) = 1 - (d_H(K, K_\Delta)/n)$, where $d_H(K, K_\Delta) = \min\{d_h(v, w) \mid v \in \text{mod}(K) \ \& \ w \in \text{mod}(K_\Delta)\}$. An agent is judged to be sufficiently satisfied if its satisfaction index is at least as high as some *threshold* value v in the real interval $[0, 1]$.

Definition 7 (Belief Mediation Model (BMM)). A Belief Mediation Model for a belief set $E = \{K_1, \dots, K_N\}$ is a quadruple $\mathcal{N} = \langle d, f, I, V \rangle$ where d is a distance, f an anonymous aggregation function, $I = \{i_1, \dots, i_N\}$ is a set of satisfaction indices, one for each agent, and $V = \{v_1, \dots, v_N\}$ is a set of threshold values, one for each agent. The mediated solution $\mathcal{N}(E)$ to a belief set $E = \{K_1, \dots, K_N\}$ with respect to $\mathcal{N}$ is $\Delta_{d,f}^m(E)$ where $m = \max\{k \mid i_j(K_j, \Delta_{d,f}^k(E)) \geq v_j \ \forall j \in \{1, \dots, N\}\}$, if there is a k such that $i_j(K_j, \Delta_{d,f}^k(E)) \geq v_j \ \forall j \in \{1, \dots, N\}$, and $\top$ otherwise.

Let us now apply mediation to the case of the three students and the three programming languages discussed in Example 1.

Example 4. Let $\mathcal{N} = \langle d_h, \Sigma, I, V \rangle$ where every i in I is the Hamming satisfaction index defined above, every threshold value v in V is 1. In the first round $i(K_1, \Delta_{d_h, \Sigma}^3(E)) = i(K_2, \Delta_{d_h, \Sigma}^3(E)) = 1$, and $i(K_3, \Delta_{d_h, \Sigma}^3(E)) = \frac{1}{3}$. So, only two of the students, K_1 and K_2 , are satisfied with the merged base $\Delta_{d_h, \Sigma}^3(E) \equiv \neg s \wedge \neg d \wedge o$, and a second round is necessary. The proposal in the second round satisfies all three students, and the mediation process thus terminates with $\Delta_{d_h, \Sigma}^2(E) \equiv o$. Now let $\mathcal{N}' = \langle d_h, \max, I, V \rangle$ where every i in I is the strong drastic satisfaction index defined above, and every threshold value v in V is 1. In the first round $i(K_1, \Delta_{d_h, \max}^3(E)) = i(K_2, \Delta_{d_h, \max}^3(E)) = i(K_3, \Delta_{d_h, \max}^3(E)) = 0$, and so none of the students are satisfied with the proposed outcome $\Delta_{d_h, \max}^3(E) \equiv (s \wedge (d \leftrightarrow \neg o)) \vee (o \wedge (s \leftrightarrow \neg d))$. Now the automated mediator suggests $\Delta_{d_h, \max}^2(E) \equiv s \vee d \vee (\neg d \wedge o)$ as an alternative. Again, all three students have a satisfaction level of 0, prompting the mediator to propose $\Delta_{d_h, \max}^1(E) \equiv o$. Now K_3 is satisfied, but K_1 and K_2 are not. So, the mediator has run out of viable alternatives, and terminates the mediation process with the default proposal of $\top$.

Observe that mediation does not imply a sequence of monotonically weaker proposals. In the example above, the proposal $s \vee d \vee (\neg d \wedge o)$ was replaced by the logically incomparable proposal o .

Since m -merging satisfies the merging properties (A1), (A3) and (A4), it immediately follows that mediation does as well. And since classical model-based merging satisfies (A2), and n -merging corresponds to classical merging, it follows that mediation does as well. As a consequence mediation satisfies all four of the basic merging properties. And of course, this also means that mediation satisfies $\Delta(\{K\}) \equiv K$.

5 Related Work

Mediation, as we have defined it, bears some resemblance to the belief merging framework of Booth [11, 12] and Konieczny [13]. Unlike much of the work on merging their proposals focus on the process of *arriving* at the permissible outcomes. The basic idea is this: if the set of bases in a belief set is inconsistent, each one of a non-empty subset of selected sources is obliged to perform a non-trivial weakening of its belief base. This process is continued until the set of (possibly weakened) belief bases becomes consistent. The conjunction of these belief bases is taken to be the result of the merging process. There are a number of important differences between mediation and the Booth-Konieczny approach. Firstly, their framework requires of individual agents to weaken their demands to obtain a potentially acceptable solution. Contrast this with mediation where a proposed solution is constructed by an independent mediator. Secondly, unlike their framework where a proposal is acceptable whenever it is consistent, mediation *always* yields consistent proposals; their acceptability is determined by the satisfaction indices of the individual agents. Lastly, the Booth-Konieczny framework is such that the sequence of proposals becomes logically weaker. In contrast, mediation proposals might be logically incomparable, or even logically stronger, than previous proposals.

Since the Booth-Konieczny framework involves a series of successive weakenings to arrive at a merged base, it can be viewed as a type of *negotiation*, and their work can thus be seen as a definition of merging in terms of negotiation. This raises the question of whether there are connections between mediation and logic-based negotiation in the style of [14, 15, 16]. We focus on the approach detailed in [14, 15], which is closely related to that found in [16]. In this approach a finite number of agents, each with a set of demands represented as sentences in a logic, go through a process of negotiation to arrive at a mutually agreed upon settlement, or *deal*, also represented logically. Some of the basic requirements of negotiation correspond to the merging postulates (A1), (A2) and (A3), and are therefore also satisfied by mediation. Whenever there are conflicting demands, negotiated deals are partitioned into three classes: dominated deals, cooperative deals, and neutral deals. The dominated deals are all cases where the outcome is logically as strong as the demands of one of the agents. Since this conflicts with the fairness property (A4) which is satisfied by mediation, it follows that mediation will never produce a dominated deal as an outcome. Cooperative deals comprise all those cases where the outcome is logically no stronger than the disjunction of the demands of all agents, while neutral deals are all the cases where the outcome is inconsistent with this same disjunction. The first case in Example 4 shows that mediation may yield results that correspond neither to cooperative deals, nor to neutral deals. Observe that the disjunction of K_1 , K_2 and K_3 is equivalent to the sentence $\varphi = (\neg d \wedge (s \vee o)) \vee (d \wedge (s \leftrightarrow o))$, but that the outcome o of the mediation process does not entail φ , nor is it inconsistent with φ . In summary then, although mediation shares the basic properties associated with logic-based merging and negotiation, it can produce results that are quite different.

On a different level, the notion of an m -state is, to some extent, similar to *variable forgetting* [17, 18]. The idea underlying variable forgetting is to completely remove any influence of a selected subset of propositional atoms in the belief bases under consideration. For example, if one chooses to forget the atom p_1 , the belief base $\neg p_1 \wedge p_2$ will be modified to p_2 , while $p_1 \vee p_2$ will be modified to $\top$. When using m -states, however, it is not a particular set of atoms that is ignored. Instead a commitment is made to ignore a fixed *number* of atoms (with $n - m$ being this number), regardless of which atoms these are. If one were to use m -states to modify belief base, quite different results may be obtained. For example, the set of 1-states closest to the belief base $\neg p_1 \wedge p_2$ (i.e. with a distance of 0) is $\{\bar{p}_1, p_2\}$. And this set is associated with the modified belief base $\neg p_1 \vee p_2$ (in the sense that $\text{mod}(\{\bar{p}_1, p_2\}) = \text{mod}(\neg p_1 \vee p_2)$).

6 Conclusion

We have presented the class of distance-based m -merging operations based on m -states. m -states are generalisations of classical propositional interpretations. They bear some resemblance to the process of variable forgetting, but the results obtained using them are quite different.

We propose that m -merging be used to define a framework for logic-based agent mediation. An automated mediator goes through a process of suggesting compromises whenever the agents involved have conflicting demands. These compromises are instances of m -merging. The process terminates when all agents are sufficiently satisfied with the current proposed compromise. While m -merging fails to satisfy some of the basic properties of belief merging, the mediation framework we present satisfies basic properties of merging, as well as logic-based negotiation. However, we show that mediation is substantially different from negotiation, in terms of intent as well as outcomes.

We have shown that mediation satisfies some basic intuitive properties, and made some initial comparisons with classical merging and logic-based negotiation. But a formal characterisation of mediation, in terms of a set of postulates, still needs to be obtained. Such a characterisation will facilitate a more detailed comparison with merging and negotiation.

Acknowledgements

Pilar Pozos Parra's work is supported by the Australian Research Council. National ICT Australia is funded by the Australia Government's Department of Communications, Information and Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence program. It is supported by its members the Australian National University, University of NSW, ACT Government, NSW Government and affiliate partner University of Sydney. Thanks are due to Richard Booth, Norman Foo, Abhaya Nayak, and Maurice Pagnucco for some useful comments and suggestions.

References

1. Konieczny, S., Pino-Pérez, R.: On the logic of merging. In Cohn, A.G., Schubert, L., Shapiro, S.C., eds.: *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR '98)*, San Francisco, California, Morgan Kaufmann (1998) 488–498
2. Konieczny, S., Lang, J., Marquis, P.: Distance-based merging: a general framework and some complexity results. In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Eighth International Conference (KR '02)*. (2002) 97–108
3. Liberatore, P., Schaerf, M.: Arbitration (or How to Merge Knowledge Bases). *IEEE Transactions on Knowledge and Engineering* **10** (1998) 76–90
4. Konieczny, S., Lang, J., Marquis, P.: DA^2 Merging Operators. *Artificial Intelligence* **157** (2004) 49–79
5. Revesz, P.Z.: On the semantics of theory change: Arbitration between old and new information. In: *Proceedings PODS'93, 12th ACM SIGACT SIGMOD SIGART Symposium on the Principles of Database Systems*. (1993) 71–82
6. Revesz, P.Z.: On the semantics of arbitration. *International Journal of Algebra and Computation* **7** (1997) 133–160
7. Lin, J., Mendelzon, A.O.: Knowledge base merging by majority. In Pareschi, R., Fronhofer, B., eds.: *Dynamic Worlds: From the Frame Problem to Knowledge Management*. Kluwer (1999)
8. Konieczny, S., Pérez, R.P.: Merging information under constraints: a qualitative framework. *Journal of Logic and Computation* **5** (2002) 773–808
9. Dalal, M.: Investigations into a theory of knowledge base revision. In: *Proceedings of the 7th National Conference of the American Association for Artificial Intelligence*, Saint Paul, Minnesota. (1988) 475–479
10. Everaere, P., Konieczny, S., Marquis, P.: On Merging Strategy-Proofness. In: *Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning*. (2004)
11. Booth, R.: A negotiation-style framework for non-prioritised revision. In: *Proceedings of the Eighth Conference on Theoretical Aspects of Rationality and Knowledge*. (2001) 137–150
12. Booth, R.: Social contraction and belief negotiation. In: *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning*. (2002) 374–384
13. Konieczny, S.: Propositional Belief Merging and Belief Negotiation Model. In: *Tenth International Workshop on Non-Monotonic Reasoning*. (2004)
14. Meyer, T., Foo, N., Kwok, R., Zhang, D.: Logical foundations of negotiation: strategies and preferences. In: *Proceedings of KR'04*. (2004)
15. Meyer, T., Foo, N., Kwok, R., Zhang, D.: Logical foundations of negotiation: Outcome, concession and adaptation. In: *Proceedings of AAAI04*. (2004)
16. Zhang, D., Foo, N., Meyer, T., Kwok, R.: Negotiation as mutual belief revision. In: *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*. (2004)
17. Lin, F., Reiter, R.: Forget it! In: *Proceedings of the AAAI Falls Symposium on Relevance*, New Orleans (LA) (1994) 154–159
18. Lang, J., Marquis, P.: Complexity results for independence and definability. In: *Proceedings of the 6th International Conference on Knowledge Representation and Reasoning (KR '98)*, Trento (1998) 356–367

Combining Multiple Knowledge Bases by Negotiation: A Possibilistic Approach

Guilin Qi, Weiru Liu, and David A. Bell

School of Computer Science, Queen's University Belfast, Belfast, BT7 1NN, UK
{G.Qi, W.Liu, DA.Bell}@qub.ac.uk

Abstract. A negotiation model consists of two functions: a negotiation function and a weakening function. A negotiation function is defined to choose the weakest sources and these sources will weaken their point of view using a weakening function. However, the currently available belief negotiation models are based on classical logic, which make it difficult to define weakening functions. In this paper, we define a prioritized belief negotiation model in the framework of possibilistic logic. The priority between formulae provides us with important information to decide which beliefs should be discarded. The problem of merging uncertain information from different sources is then solved by two steps. First, beliefs in the original knowledge bases will be weakened to resolve inconsistencies among them. This step is based on a prioritized belief negotiation model. Second, the knowledge bases obtained by the first step are combined using a *conjunctive* operator or a *reinforcement* operator in possibilistic logic.

1 Introduction

In recent years, some belief merging methods based on belief negotiation models were proposed to make the merging process more “active” [6, 7, 12]. Belief negotiation models based methods deal with the merging problem by several rounds of negotiation or competition. In each round, some sources are chosen by a negotiation function, then these sources have to weaken their point of view using a weakening function. However, both Konieczny’s belief negotiation model and Booth’s belief negotiation model are defined in purely propositional logic systems. So it is difficult for them to define a weakening function.

The importance of priorities in handling inconsistencies has been addressed by many researchers in recent years, e.g. [3, 11, 13]. Priority between formulae provides us with important information to decide which formulae should be discarded. So it is helpful to consider priority when we define a belief negotiation model. Possibilistic logic [9] provides a good framework to express priorities and reason with uncertain information. In possibilistic logic, each classical first order formula is attached with a number or weight, denoting the *necessity degree* of the formula. The necessity degrees can be interpreted as the priorities of formulae.

In this paper, we propose a prioritized belief negotiation model, where priorities between formulae are handled in the framework of possibilistic logic. Each source of beliefs is represented as a *possibilistic belief base*. The procedure of

merging different sources of beliefs is carried out in two steps. The first step is called a negotiation step, beliefs in some of the original knowledge bases will be weakened to make it possible for them to be added together consistently (this step is called “social contraction” in [7]). Some negotiation functions and weakening functions will be defined by considering the priority in this step. The second step is called a combination step, the knowledge bases obtained by the first step are combined using a *conjunctive* operator or a *reinforcement* operator in possibilistic logic [2, 4].

This paper is organized as follows. We introduce Konieczny’s belief game model in Section 2. Then in Section 3, we give a brief review of possibilistic logic. Our prioritized belief negotiation model will be presented in Section 4. In Section 5, we give some particular negotiation functions and weakening functions. In Section 6, we instantiate the prioritized belief negotiation model and provide an example to illustrate the new merging methods. Finally, we conclude the paper in Section 7.

2 Preliminaries

In this paper, we will consider a propositional language $\mathcal{L}$ over a finite alphabet $\mathcal{P}$. $\mathcal{W}$ denotes the set of possible worlds, where each possible world is a function from $\mathcal{P}$ to $\{\top, \perp\}$ ($\top$ denotes truth value *true* and $\perp$ denotes the truth value *false*). A model of a formula ϕ is a possible world w which makes the formula *true*. We use $mod(\phi)$ to denote the set of models of formula ϕ , i.e., $mod(\phi) = \{w \in \mathcal{W} | w \models \phi\}$. Deduction in classical propositional logic is denoted by symbol $\vdash$ as usual. $\phi, \psi, \gamma, \dots$ represent classical formulae. Given two formulae ϕ and ψ , ϕ and ψ is equivalent, denoted as $\phi \equiv \psi$, if and only if $\phi \vdash \psi$ and $\psi \vdash \phi$.

A belief base φ is a consistent propositional formula (or, equivalently, a finite consistent set of propositional formulae). Let $\varphi_1, \dots, \varphi_n$ be n belief bases (not necessarily different). A *belief profile* is a multi-set Ψ consisting of those n belief bases: $\Psi = (\varphi_1, \dots, \varphi_n)$. The *conjunction* of the belief bases of Ψ is denoted as $\bigwedge \Psi$, i.e., $\bigwedge \Psi = \varphi_1 \wedge \dots \wedge \varphi_n$. $\sqcup$ and $\sqsubseteq$ are used to denote the *union* and *inclusion* of belief profiles respectively. Two belief profiles Ψ_1 and Ψ_2 are said to be equivalent ($\Psi_1 \equiv \Psi_2$) if and only if there is a bijection f between Ψ_1 and Ψ_2 such that $\forall \varphi \in \Psi_1, \varphi \equiv f(\varphi)$, where $f(\varphi)$ is the image of φ in Ψ_2 . $\mathcal{E}$ denotes the set of all finite non-empty belief profiles.

Belief Game Model: A belief game model [12] is developed from Booth’s belief negotiation model [7] which provides a framework for merging sources of beliefs incrementally. It consists of two functions. One is called a negotiation function, which selects from every belief profile in $\mathcal{E}$ a subset of belief bases. The other is called a weakening function, which aims to weaken the beliefs of a selected source.

Definition 1. A negotiation function is a function $g: \mathcal{E} \rightarrow \mathcal{E}$ such that:

- (n1) $g(\Psi) \sqsubseteq \Psi$, (n2) $g(\Psi) \neq \emptyset$, (n3) $\exists \varphi \in g(\Psi)$ s.t. $\varphi \neq \top$,
- (n4) If $\Psi \equiv \Psi'$, then $g(\Psi) \equiv g(\Psi')$

The first two conditions guarantee a non-empty subset is chosen from a belief profile to be weakened. The third condition states that at least one non-tautological base must be selected. The last condition is about irrelevance of syntax.

Definition 2. A weakening function is a function $\nabla : \mathcal{L} \rightarrow \mathcal{L}$ such that:

- (w1) $\varphi \vdash \nabla(\varphi)$, (w2) If $\varphi \equiv \nabla(\varphi)$, then $\varphi \equiv \top$,
- (w3) If $\varphi \equiv \varphi'$, then $\nabla(\varphi) \equiv \nabla(\varphi')$

The first two conditions ensure that a base will be replaced by a strictly weaker one unless the base is already a tautological one. The last condition is an irrelevance of syntax requirement, i.e., the result of weakening depends only on the information conveyed by a base, not on its syntactical form.

A weakening function can be extended as follows. Let Ψ' be a subset of Ψ , $\nabla_{\Psi'}(\Psi) = \cup_{\varphi \in \Psi'} \nabla(\varphi) \cup \cup_{\varphi \in \Psi \setminus \Psi'} \varphi$.

Definition 3. A Belief Game Model (BGM) is a pair $\mathcal{N} = \langle g, \nabla \rangle$ where g is a negotiation function and ∇ is a weakening function. The solution to a belief profile Ψ for a Belief Game Model $\mathcal{N} = \langle g, \nabla \rangle$, noted as $\mathcal{N}(\Psi)$, is the belief profile $\Psi_{\mathcal{N}}$, defined as:

- $\Psi_0 = \Psi$
- $\Psi_{i+1} = \nabla_{g(\Psi_i)}(\Psi_i)$
- $\Psi_{\mathcal{N}}$ is the first Ψ_i that is consistent

3 Possibilistic Logic

Possibilistic logic [9] is a weighted logic where each classical logic formula is associated with a level of priority. A possibilistic belief base (**PBB**) is a set of possibilistic formulae of the form $B = \{(\phi_i, \alpha_i) : i = 1, \dots, n\}$, where $\alpha_i \in [0, 1]$ and they are meant to be the necessity degrees of the ϕ_i . The classical base associated with B is denoted as B^* , namely $B^* = \{\phi_i | (\phi_i, \alpha_i) \in B\}$. A PBB B is consistent if and only if its classical base B^* is consistent. In possibilistic logic, a *possibility distribution*, denoted by π , is a mapping from a set of possible worlds $\mathcal{W}$ to the interval $[0, 1]$. $\pi(\omega)$ represents the possibility degree of the interpretation ω with regard to available beliefs. From a *possibility distribution* π , two measures defined on a set of propositional or first order formulae can be determined. One is the possibility degree of formula ϕ , denoted as $\Pi(\phi) = \max\{\pi(\omega) : \omega \models \phi\}$. The other is the necessity degree of formula ϕ , and is defined as $N(\phi) = 1 - \Pi(\neg\phi)$.

A possibilistic belief profile $\mathcal{KP}$ is a multi-set of PBBs, where these PBBs are not necessarily different. $\mathcal{KP} = (B_1, \dots, B_n)$ is consistent iff $B_1^* \cup \dots \cup B_n^*$ is consistent. We use $\mathcal{PE}$ to denote the set of all finite non-empty possibilistic belief profiles and $\mathcal{K}$ to denote the set of all the PBBs.

Definition 4. Let B be a PBB, and $\alpha \in [0, 1]$. The α -cut of B is $B_{\geq \alpha} = \{\phi \in B^* | (\phi, a) \in B \text{ and } a \geq \alpha\}$.

The *inconsistency degree* of B , which defines its level of inconsistency, is defined as: $Inc(B) = \max\{\alpha_i | B_{\geq \alpha_i} \text{ is inconsistent}\}$.

Let B and B' be two PBBs. B and B' are said to be equivalent, denoted by $B \equiv_s B'$, iff $\forall a \in [0, 1]$, $B_{\geq a} \equiv_s B'_{\geq a}$. Two possibilistic belief profiles $\mathcal{KP}_1$ and $\mathcal{KP}_2$ are said to be equivalent ($\mathcal{KP}_1 \equiv_s \mathcal{KP}_2$) if and only if there is a bijection between them such that each PBB of $\mathcal{KP}_1$ is equivalent to its image in $\mathcal{KP}_2$.

Definition 5. Let B be a PBB. Let (ϕ, α) be a piece of information with $\alpha > Inc(B)$. (ϕ, α) is said to be a consequence of B , denoted by $B \vdash_\pi (\phi, \alpha)$, iff $B_{\geq \alpha} \vdash \phi$.

Given a PBB B , a unique *possibility distribution*, denoted by π_B , can be obtained by the principle of minimum specificity. For all $\omega \in \Omega$,

$$\pi_B(\omega) = \begin{cases} 1 & \text{if } \forall (\phi_i, \alpha_i) \in B, \omega \models \phi_i, \\ 1 - \max\{\alpha_i | \omega \not\models \phi_i, (\phi_i, \alpha_i) \in B\} & \text{otherwise.} \end{cases} \quad (1)$$

Many combination rules for merging PBBs have been proposed [2, 4]. Let us first introduce some operators which combine possibility distributions.

Definition 6. [4] A conjunctive operator is a two place function $\oplus : [0, 1] \times [0, 1] \rightarrow [0, 1]$ such that $\forall a \in [0, 1]$, $a \oplus 1 = 1 \oplus a = a$.

Examples of conjunctive operators the minimum operator and the product operator.

Definition 7. [4] A reinforcement operator is a two place function $\oplus : [0, 1] \times [0, 1] \rightarrow [0, 1]$ such that $\forall a, b \neq 1$ and $a, b \neq 0$, $a \oplus b < \min(a, b)$.

Examples of reinforcement operator are the product operator and the *Lukasiewicz t-norm* $\max(0, a+b-1)$. It is clear a conjunctive operator may be a reinforcement operator.

Given two PBBs B_1 and B_2 , and a conjunctive operator or a reinforcement operator $\oplus$, a semantic combination rule combines the possibility distributions π_{B_1} and π_{B_2} using $\oplus$ as $\pi_{\oplus}(w) = \pi_{B_1}(w) \oplus \pi_{B_2}(w)$. Its syntactical counterpart is the following PBB [4]:

$$B_1 \oplus B_2 = \{(\phi_i, 1 - (1 - \alpha_i) \oplus 1) : (\phi_i, \alpha_i) \in B_1\} \cup \{(\psi_j, 1 - 1 \oplus (1 - b_j)) : (\psi_j, b_j) \in B_2\} \cup \{(\phi_i \vee \psi_j, 1 - (1 - \alpha_i) \oplus (1 - b_j)) : (\phi_i, \alpha_i) \in B_1 \text{ and } (\psi_j, b_j) \in B_2\}. \quad (2)$$

For example, when $\oplus = \min$, $B_1 \oplus B_2 = B_1 \cup B_2$. It is often assumed that an operator used to combine possibility distributions should be both commutative and associative, i.e., $a \oplus b = b \oplus a$ and $a \oplus (b \oplus c) = (a \oplus b) \oplus c$. In this case, the order of the combination will not influence the result of merging when multiple PBBs need to be combined. When the union of original PBBs is consistent, it is advisable to use a conjunctive operator based combination rule or a reinforcement operator based combination rule because all the formulae in these PBBs are kept in the resulting PBB and their necessity degrees are not decreased.

4 A Prioritized Belief Negotiation Model

In this section, we will propose a prioritized belief negotiation model to generalize the belief game model [12], where priorities between formulae are handled in the framework of possibilistic logic. Each source of beliefs is represented as a PBB. We assume that the original PBBs are self-consistent.

Definition 8. A negotiation function is a function $g: \mathcal{PE} \rightarrow \mathcal{PE}$ such that:

$$(N1) \ g(\mathcal{KP}) \sqsubseteq \mathcal{KP}, \quad (N2) \ g(\mathcal{KP}) \neq \emptyset, \\ (N3) \ \text{If } \exists B \in \mathcal{KP} \text{ s.t. } B^* \not\equiv \top, \text{ then } \forall B' \in g(\mathcal{KP}), (B^*)' \not\equiv \top.$$

Conditions N1 and N2 are directly generalized from Conditions n1 and n2 in BGM. Condition N3 states that the negotiation function will not select the PBB whose classical base is equivalent to the “tautology” if there is a PBB whose classical base is not equivalent to the “tautology”. That is, we do not choose the “tautology” to weaken if possible. Our negotiation function relies on the syntactical form of the PBBs, because every formula is attached a weight in a PBB, and we need to consider the syntax of the PBB.

Next we will give the definition of a weakening function.

Definition 9. A weakening function is a function $\nabla: \mathcal{K} \times \mathcal{PE} \times \mathcal{PE} \rightarrow \mathcal{K}$ such that: for each triple consisting of a PBB B and two possibilistic profiles $\mathcal{KP}$ and $\mathcal{KP}'$, if $\mathcal{KP}' \sqsubseteq \mathcal{KP}$ and $B \in \mathcal{KP}'$, then $\nabla_{\mathcal{KP}, \mathcal{KP}'}(B)$ should satisfy the conditions (W1) and (W2) below, otherwise $\nabla_{\mathcal{KP}, \mathcal{KP}'}(B) = B$.

$$(W1) \ B \vdash_{\pi} (\phi, a), \text{ for all } (\phi, a) \in \nabla_{\mathcal{KP}, \mathcal{KP}'}(B) \\ (W2) \ \text{If } B = \nabla_{\mathcal{KP}, \mathcal{KP}'}(B), \text{ then } B^* \equiv \top$$

Unlike the weakening function in BGM, our weakening function only weakens the PBBs in a subset of possibilistic belief profile and keeps other PBBs unchanged. The priority between formulae in a PBB makes the construction of weakening function easy.

We can extend a weakening function on belief profiles as follows: let $\mathcal{KP}'$ be a subset of $\mathcal{KP}$, $\nabla_{\mathcal{KP}, \mathcal{KP}'}(\mathcal{KP}) = \cup_{B \in \mathcal{KP}} \nabla_{\mathcal{KP}, \mathcal{KP}'}(B)$.

Definition 10. A prioritized belief negotiation model (relative to sources) is a pair $\mathcal{N} = \langle g, \nabla \rangle$ where g is a negotiation function and ∇ is a weakening function. The solution to a possibilistic belief profile $\mathcal{KP}$ for a belief negotiation model $\mathcal{N} = \langle g, \nabla \rangle$, noted as $\mathcal{N}(\mathcal{KP})$, is the belief profile $\mathcal{KP}_{\mathcal{N}}$ defined as:

- $\mathcal{KP}_0 = \mathcal{KP}$
- $\mathcal{KP}_{i+1} = \nabla_{\mathcal{KP}_i, g(\mathcal{KP}_i)}(\mathcal{KP}_i)$
- $\mathcal{KP}_{\mathcal{N}}$ is the first $\mathcal{KP}_i$ that is consistent.

The merging of PBBs based on a prioritized belief negotiation model is to obtain a set of consistent PBBs through negotiation and then apply a suitable combination operator (usually a conjunctive operator or a reinforcement operator) to merge them.

5 Negotiation and Weakening Functions

5.1 Negotiation Function

Distance-Based Negotiation Function. The first category of negotiation functions is based on a distance between two PBBs.

The following is the definition of a distance between two PBBs, which is a simple extension of the distance between two classical belief bases in [12].

Definition 11. A (pseudo) distance between two PBBs is a function $d: \mathcal{KP} \times \mathcal{KP} \rightarrow [0, 1]$ such that: $d(B, B') = 0$ iff $B^* \cup B'^* \not\models \perp$, and $d(B, B') = d(B', B)$.

Clearly, a very simple distance can be defined as follows: $d_D(B, B') = 0$ if $B^* \cup B'^* \not\models \perp$ and $d_D(B, B') = 1$ otherwise.

Now we will define a quantity of conflict between two PBBs based on *weighted prime implicants*. This can be used to define a distance between two PBBs.

An *implicant* of a belief base B is a conjunction of literals D such that $D \vdash B$ and D does not contain two complementary literals.

Definition 12. A prime implicant of a belief base B is an implicant D of B such that for every other implicant D' of B , $D \not\vdash D'$.

Prime implicants are often used in knowledge compilation to make the deduction tractable. Suppose $D_1, \dots, D_k$ are all the prime implicants of B , we have $B \vdash \phi$ iff for every prime implicant D_i , $D_i \vdash \phi$, for any ϕ .

Now we define the weighted prime implicant of a PBB. Let us first define the weighted prime implicant for PBB $B = \{(\phi_1, a_1), \dots, (\phi_n, a_n)\}$ where ϕ_i are clauses, and a clause is a disjunction of literals. For a more general PBB, we can decompose it as an equivalent PBB whose formulae are clauses by the min-decomposability of necessity measures, i.e., $N(\bigwedge_{i=1,k} \phi_i) \geq m \Leftrightarrow \forall i, N(\phi_i) \geq m$ [10]. That is, a possibilistic formula $(\phi_1 \wedge \dots \wedge \phi_k, a)$ can be equivalently decomposed as a set of possibilistic formulae $(\phi_1, a), \dots, (\phi_k, a)$.

Let $B = \{(\phi_1, a_1), \dots, (\phi_n, a_n)\}$ be a PBB where ϕ_i are clauses. A weighted implicant of B is $D = \{(\psi_1, b_1), \dots, (\psi_k, b_k)\}$, a PBB, such that $D \vdash_\pi B$, where ψ_i are literals. Let D and D' be two weighted implicants of B , D is said to be *subsumed* by D' iff $D \neq D'$, $D'^* \subseteq D^*$ and $\forall (\psi_i, a_i) \in D, \exists (\psi_i, b_i) \in D'$ with $b_i \leq a_i$ (b_i is 0 if $\psi_i \in D^*$ but $\psi_i \notin D'^*$).

Definition 13. Let $B = \{(\phi_1, a_1), \dots, (\phi_n, a_n)\}$ be a PBB where ϕ_i are clauses. A weighted prime implicant (**WPI**) of B is D such that

1. D is a weighted implicant of B
2. $\nexists D'$ of B such that D is subsumed by D' .

Let us look at an example to illustrate how to construct WPIs.

Example 1. Let $B = \{(p, 0.8), (q \vee r, 0.5), (q \vee \neg s, 0.6)\}$ be a PBB. The WPIs of B are $D_1 = \{(p, 0.8), (q, 0.6)\}$, $D_2 = \{(p, 0.8), (r, 0.5), (\neg s, 0.6)\}$, and $D_3 = \{(p, 0.8), (q, 0.5), (\neg s, 0.6)\}$.

The WPI generalizes the prime implicant.

Proposition 1. *Let $B = \{(\phi_1, 1), \dots, (\phi_n, 1)\}$ be a PBB where all the formulae have weight 1, i.e., B is a classical knowledge base. Then D is a WPI of B iff D is a prime implicant of B .*

However, given PBB B , if D is a WPI of B , then D^* is not necessary to be a prime implicant of B^* . A counterexample can be found in Example 1, where D_3 is a WPI, but $D_3^* = \{p, q, \neg s\}$ is not a prime implicant of B^* .

Definition 14. *Let B_1 and B_2 be two PBBs. Suppose C and D are WPIs of B_1 and B_2 respectively, then the quantity of conflict between C and D is defined as*

$$q_{Con}(C, D) = \Sigma_{(p,a) \in C \text{ and } (\neg p,b) \in D} \min(a, b). \quad (3)$$

When the weights associated with all the formulae are 1, $q_{Con}(C, D)$ is the cardinality of the set of atoms which are in conflict in $C \cup D$.

Definition 15. *Let B_1 and B_2 be two PBBs. Suppose $\mathcal{C}$ and $\mathcal{D}$ are the sets of weighted prime implicants of B_1 and B_2 respectively, then the quantity of conflict between B_1 and B_2 is defined as*

$$Q_{Con}(B_1, B_2) = \min\{q_{Con}(C, D) | C \in \mathcal{C}, D \in \mathcal{D}\}. \quad (4)$$

The quantity of conflict between B_1 and B_2 measures information that is in conflict between B_1 and B_2 . We have proved that the quantity of conflict between two classical belief bases are the Dalal distance between them [8] (We will not include the proof here due to the page limit.). So we can define a distance function d_C based on the quantity of conflict such that $d_C(B_1, B_2) = Q_{Con}(B_1, B_2)$ (it is easy to check that d_C satisfies the requirements of a distance function in Definition 11).

Definition 16. [12] *An aggregation function is a total function f associating a non-negative integer to every finite tuple of nonnegative integers and verifying the following conditions:*

- if $x \leq y$, then $f(x_1, \dots, x, \dots, x_n) \leq f(x_1, \dots, y, \dots, x_n)$. (non-decreasingness)
- $f(x_1, \dots, x_n) = 0$ iff $x_1 = \dots = x_n = 0$. (minimality)
- for every nonnegative integer x , $f(x) = x$. (identity)

Two most commonly used aggregation functions are the maximum and the sum Σ .

Now we can define the distance-based negotiation function.

Definition 17. *Let $\mathcal{KP} = \{B_1, \dots, B_n\}$ be a multi-set of PBBs. A distance-based negotiation function is defined as follows: for all $B \in \mathcal{KP}$,*

$$B \in g^{d,f}(\mathcal{KP}) \text{ iff } f(d(B, B_1), \dots, d(B, B_n)) \text{ is maximal,}$$

where f is an aggregation function, d is a distance function between two PBBs.

Therefore, those sources that are “furthest” from the group are weakened.

Conflict-Based Negotiation Function. Priority provides an easy way for us to deal with inconsistency. In belief revision and belief merging, an implicit or explicit priority is often assumed. The inconsistency of a PBB can be resolved by dropping those formulae that are in conflict with lowest priorities in a minimally inconsistent subbase [5, 11]. A natural negotiation function can be defined by selecting those PBBs which contain conflict formulae in the lowest level of the union of all the PBBs.

Definition 18. [3] A subbase C of PBB B is said to be minimally inconsistent if and only if it satisfies the following two requirements: (1) $C^* \models \perp$, (2) $\forall \phi \in C^*$, $C^* - \{\phi\} \not\models \perp$.

Definition 19. [3] A possibilistic formula (ϕ, α) is said to be in conflict in B iff it belongs to some minimally inconsistent subbase of B .

Definition 20. Let B be an inconsistent PBB. A possibilistic formula (ϕ, a) is said to be a weakest conflict formula in B iff it satisfies (1) ϕ is in conflict in B , (2) $\forall (\psi, b) \in B$, if $b < a$, then ψ is not in conflict in B

Definition 21. Let $\mathcal{KP} = \{B_1, \dots, B_n\}$ be a multi-set of PBBs. A weakest-conflict-based negotiation function is defined as follows:

$$g^{wc}(\mathcal{KP}) = \{B_i \in \mathcal{KP} \mid \exists \text{ a weakest conflict formula in } \cup(\mathcal{KP}) \text{ belonging to } B_i\}.$$

The weakest-conflict-based negotiation function is often used with the weakest-conflict-based weakening function that will be defined in the next subsection.

5.2 Weakening Function

The priority derived from the necessity degrees of possibilistic formulae allows us to define some syntax-based weakening functions. The first weakening function deletes the weakest conflict formulae in a belief base.

Definition 22. Let $B_1, \dots, B_n$ be PBBs and $\mathcal{KP} = \{B_1, \dots, B_n\}$ be a possibilistic belief profile. A possibilistic formula (ϕ, a) is said to be the weakest conflict formula of B in $\mathcal{KP}$ iff

- ϕ is in conflict in $\cup(\mathcal{KP})$
- $\forall (\psi, b) \in B$, if $b < a$, then ψ is not in conflict in $\cup(\mathcal{KP})$

Definition 23. Let $B_1, \dots, B_n$ be PBBs and $\mathcal{KP} = \{B_1, \dots, B_n\}$ be a possibilistic belief profile and $\mathcal{KP}'$ be a subset of $\mathcal{KP}$. Let $B \in \mathcal{KP}'$ and $C = \{\phi \in B \mid \phi \text{ is a weakest conflict formula of } B \text{ in } \cup(\mathcal{KP})\}$. The weakest-conflict-based (WC for short) weakening function is defined as:

$$\nabla_{\mathcal{KP}, \mathcal{KP}'}^{wc}(B) = B \setminus C.$$

The WC-weakening function deletes those formulae that are the weakest conflict formulae from a PBB which is selected by a negotiation function.

The weakening function defined above need to compute the conflict formulae, which is computationally too complex. In the following, we define a weakening function which does not need to compute conflict formulae.

Definition 24. Let $\mathcal{KP} = \{B_1, \dots, B_n\}$ be a possibilistic belief profile and $\mathcal{KP}'$ be an arbitrary subset of $\mathcal{KP}$. $B \in \mathcal{KP}'$. Let $\alpha = \min\{a \in (0, 1] : \exists \phi, (\phi, a) \in B\}$. The blind-optimized weakening function is defined as:

$$\nabla_{\mathcal{KP}, \mathcal{KP}'}^{bo}(B) = \{(\phi, a) \in B : a \neq \alpha\}.$$

The blind-optimized weakening function deletes formulae in the lowest layer. The weakening function applies when the agent does not know which formula is in conflict in the PBB, so it deletes those formulae that have the least priority.

6 Instantiating the Framework and Examples

6.1 Instantiation

Different combinations of the negotiation functions and the weakening functions will result in different prioritized belief negotiation models and then different belief merging methods. In the examples given below, we assume that after some PBBs are weakened, the combination operator is the minimum, i.e., the PBBs are conjoined.

- $\langle g^{wc}, \nabla^{wc} \rangle^1$: This merging method deletes the conflict formulae from the lower levels, i.e weights of formulae are lower. That is, the agents always choose the weakest information to discard. This idea can be found in [5].
- $\langle g^{d_D, f^{Max}}, \nabla^{wc} \rangle$: In this case, every PBB which is in conflict with any of other PBBs deletes their weakest conflict formulae in each round. This merging method usually deletes more formulae than the merging method based on $\langle g^{wc}, \nabla^{wc} \rangle$.
- $\langle g^{d_D, f^{\Sigma}}, \nabla^{wc} \rangle$: In this case, in each round of negotiation, those PBBs which have the greatest number of PBBs in conflict will be selected and have their weakest conflict formulae deleted.
- $\langle g^{Q_{Con}, f^{\Sigma}}, \nabla^{wc} \rangle$: In this case, in each round of negotiation, those PBBs which have more quantities of information in conflict with other PBBs will be selected and have their weakest conflict formulae deleted.
- $\langle g^{Q_{Con}, f^{\Sigma}}, \nabla^{bo} \rangle$: In this case, in each round of negotiation, those PBBs which have more quantities of information in conflict with other PBBs will be selected and have their lowest layers deleted. This merging method deletes more formulae than the merging method based on $\langle g^{Q_{Con}, f^{\Sigma}}, \nabla^{wc} \rangle$. However, it is computationally simpler.

¹ For simplicity, we will ignore the subscript of the weakening functions.

In the examples above, we require that the combination rule used in the second step of merging be the minimum. If we relax this restriction, we can get some more merging methods. For example, in the case of $\langle g^{wc}, \nabla^{wc} \rangle$, if we further assume that the combination operator is the product operator, then we can get a merging method which has a reinforcement effect.

Compared with merging methods in [1, 4], our methods are more *active*, i.e. agents resolve their conflicting information through the process of negotiation. Moreover, the merging results of our methods may retain more important information than those of methods in [1, 4]. For example, given two PBBs B_1 and B_2 , a merging method in [1] first merges them using a *t-norm* operator through Equation 1, then deletes any formulae whose necessity degrees are under the inconsistency level of the resulting PBB. If the inconsistency degree of $B_1 \cup B_2$ is very high (0.9, for example), then possibilistic formulae in B_1 and B_2 whose necessity degrees are lower than 0.9 will be deleted even if some of them are not involved in conflict. However, using our methods, for example, the merging method which is based on the pair $\langle g^{Q_{Con}, f^\Sigma}, \nabla^{bo} \rangle$, some possibilistic formulae with necessity degrees lower than 0.9 can also be kept after merging.

6.2 Illustrative Example

In this section, we will give an example to illustrate some prioritized belief negotiation model based merging methods, i.e., those based on $\langle g^{d_D, f^\Sigma}, \nabla^{wc} \rangle$ and $\langle g^{Q_{Con}, f^\Sigma}, \nabla^{wc} \rangle$.

Example 2. Three people are talking about origins of human beings and planets. Their opinions are summarized as weighted logical sentences in a possibilistic belief profile $\mathcal{KP} = \{A, B, C\}$, where

$$A = \{(p, 0.4), (q \rightarrow r, 1), (s, 0.8), (\neg s \rightarrow \neg r, 0.9)\}$$

$$B = \{(q, 0.8), (\neg s, 0.6), (e, 0.8)\}$$

$$C = \{(\neg p, 0.8), (\neg q, 0.6), (e \rightarrow r, 0.4)\}$$

- p represents “there were human beings in Mars before”
- q represents “scientists have detected some strange signals from outer space”
- r represents “there are aliens in other planets”
- s represents “the ancestors of human are gorillas”
- e represents “the earth was created by chance, not by a creator”.

In this example, C is quite sure that there were no human beings in Mars before and is unsure that if the earth was created by chance, then there are aliens in other planets too.

Now we will see how they can negotiate with each other to make their opinions coherent.

- Method 1: $\langle g^{d_D, f^\Sigma}, \nabla^{wc} \rangle$ and $\oplus = \text{Lukasiewicz } t\text{-norm}$:

Since A, B and C are in conflict, $g^{d_D, f^\Sigma}(\mathcal{KP}) = \mathcal{KP}$. So A is replaced by $\nabla^{wc}(A) = \{(q \rightarrow r, 1), (s, 0.8), (\neg s \rightarrow \neg r, 0.9)\}$,² B is replaced by $\nabla^{wc}(B) = \{(q, 0.8), (e, 0.8)\}$ and C is replaced by $\nabla^{wc}(C) = \{(\neg p, 0.8), (\neg q, 0.6)\}$. Now $\nabla^{wc}(B)$ and $\nabla^{wc}(C)$ are still in conflict, and they will have to weaken their beliefs in the second round. So $\nabla^{wc}(B) = \{(e, 0.8)\}$ and $\nabla^{wc}(C) = \{(\neg p, 0.8)\}$. In this case, we have reached a consistent possibilistic belief profile. By combining $\nabla^{wc}(A)$, $\nabla^{wc}(B)$ and $\nabla^{wc}(C)$ using *Lukasiewicz t-norm*, we have the following result of merging:

$$\begin{aligned} \mathcal{KP}_\oplus = \{ & (q \rightarrow r, 1), (s, 0.8), (\neg s \rightarrow \neg r, 0.9), (e, 0.8), (\neg p, 0.8), (e \vee \neg p, 1), \\ & (\neg q \vee r \vee e, 1), (s \vee e, 1), (\neg s \vee \neg r \vee e, 1), (\neg p \vee \neg q \vee r, 1), (\neg p \vee s, 1), \\ & (\neg p \vee \neg s \vee \neg r, 1), (\neg p \vee q \vee r \vee e, 1), (\neg p \vee s \vee e, 1), (\neg p \vee \neg s \vee \neg r \vee e, 1)\}. \end{aligned}$$

- Method 2: $\langle g^{Q_C, f^\Sigma}, \nabla^{wc} \rangle$ and $\oplus = \text{Lukasiewicz } t\text{-norm}$:

Since $\mathcal{KP}$ is not consistent, we need to compute the distance from each PBB to others using g^{Q_C, f^Σ} . $Q_C(A, B) = 0.6$, $Q_C(A, C) = 0.4$, $Q_C(B, C) = 0.6$. So $f_{\mathcal{KP}}^\Sigma(A) = 1$, $f_{\mathcal{KP}}^\Sigma(B) = 1.2$, $f_{\mathcal{KP}}^\Sigma(C) = 1$. In the first round, $g^{Q_C, f^\Sigma}(\mathcal{KP}) = \{B\}$. So B is replaced by $\nabla^{wc}(B) = \{(q, 0.8), (e, 0.8)\}$. The obtained belief profile is still inconsistent, we must then go to the second round. Now $Q_C(A, B) = 0$, $Q_C(A, C) = 0.4$, $Q_C(B, C) = 0.6$. So $f_{\mathcal{KP}}^\Sigma(A) = 0.4$, $f_{\mathcal{KP}}^\Sigma(B) = 0.6$, $f_{\mathcal{KP}}^\Sigma(C) = 1$. So $g^{Q_C, f^\Sigma}(\mathcal{KP}) = \{C\}$. C is then replaced by $\nabla^{wc}(C) = \{(\neg p, 0.8), (\neg q, 0.6)\}$. The obtained belief profile is inconsistent again, we must now go to the third round. $Q_C(A, B) = 0$, $Q_C(A, C) = 0.4$, $Q_C(B, C) = 0.6$. So $f_{\mathcal{KP}}^\Sigma(A) = 0.4$, $f_{\mathcal{KP}}^\Sigma(B) = 0.6$, $f_{\mathcal{KP}}^\Sigma(C) = 1$. So $g^{Q_C, f^\Sigma}(\mathcal{KP}) = \{C\}$. C is then replaced by $\nabla^{wc}(C) = \{(\neg p, 0.8)\}$. Since the obtained belief profile is still inconsistent, we must go to the fourth round. Now $Q_C(A, B) = 0$, $Q_C(A, C) = 0.4$, $Q_C(B, C) = 0.6$. So $f_{\mathcal{KP}}^\Sigma(A) = 0.4$, $f_{\mathcal{KP}}^\Sigma(B) = 0$, $f_{\mathcal{KP}}^\Sigma(C) = 0.4$, and $g^{Q_C, f^\Sigma}(\mathcal{KP}) = \{A, C\}$. A is then replaced by $\nabla^{wc}(A) = \{(q \rightarrow r, 1), (s, 0.8), (\neg s \rightarrow \neg r, 0.9)\}$ and C is replaced by $\nabla^{wc}(C) = \emptyset$. Finally C loses the game and gives up all the beliefs. The obtained belief profile is consistent, and the result of merging is

$$\begin{aligned} \mathcal{KP}_\oplus = \{ & (q \rightarrow r, 1), (s, 0.8), (\neg s \rightarrow \neg r, 0.9), (q, 0.8), (e, 0.8), (q \vee s, 1), (\neg q \vee e \vee r, 1), \\ & (e \vee s, 1), (e \vee \neg s \vee r, 1)\} \end{aligned}$$

It is clear that the negotiation process in the second method is more complex than that of the first one. However, in the second merging method, C loses the game and gives up all its beliefs.

² To make the notation simpler, we will ignore the subscript of the weakening functions. Moreover, we don't use subscripts to denote the different weakening steps of the bases.

7 Conclusions

In this paper, we proposed a prioritized belief negotiation model which generalizes Konieczny's belief game model [12]. We then presented a two-step scenario for merging PBBs based on the prioritized belief negotiation model. In the first step, original PBBs are weakened to make them consistent. Then in the second step, we combine the resulting PBBs using some combination rules in possibilistic logic [4]. Unlike the belief game model and Booth's belief negotiation model, our prioritized belief negotiation model takes into account the syntax of the PBBs and we have defined some particular negotiation functions and weakening functions by considering the priorities of formulae in each PBB.

Acknowledgements

The authors are grateful to Richard Booth for his valuable comments on the draft paper.

References

1. Benferhat, S., Dubois, D., and Prade, H.: From semantic to syntactic approaches to information combination in possibilistic logic. In Bouchon-Meunier, B. eds., *Aggregation and Fusion of Imperfect Information*, 141-151. Physica. Verlag.
2. Benferhat, S., Dubois, D., Prade, H., and Williams, M.A.: A Practical Approach to Fusing Prioritized Knowledge Bases. *Proc. of 9th Portuguese Conf. on Artificial Intelligence*, 223-236, 1999.
3. Benferhat S., Dubois D., Prade H. Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study. Part 2: The prioritized case. *Logic at work : essays dedicated to the memory of Helena Rasiowa / Ewa Orłowska*. - New York : Physica-Verlag, 473-511, 1998.
4. Benferhat, S., Dubois, D., Kaci, S., and Prade, H.: Possibilistic merging and distance-based fusion of propositional information. *Annals of Mathematics and Artificial Intelligence*, vol. 34, 217-252 , 2002.
5. Benferhat, S., Garcia, L.: Handling locally stratified inconsistent knowledge bases. *Studia Logica*, vol. 70, 77-104, 2002.
6. Booth R.: A Negotiation-style Framework for Non-prioritized Revision. *Proc. of TARK'01*, 137-150, 2001.
7. Booth R.: Social Contraction and Belief Negotiation. *Proc. of KR'02*, 374-384, 2002.
8. Dalal M.: Investigations into a theory of knowledge base revision: preliminary report. *Proc. of AAAI'88*, 475-479, 1988.
9. Dubois, D., Lang, J., and Prade, H. : Possibilistic logic. In *Handbook of logic in Artificial Intelligence and Logic Programming*, Volume 3. Oxford University Press, 439-513, 1994.

10. Dubois, D., Konieczny, S., and Prade, H.: Quasi-possibilistic logic and its measures of information and conflict. *Fundamenta Informaticae*, vol. 57(2-4), 101-125, 2003.
11. Gärdenfors P.: *Knowledge in Flux-Modeling the Dynamic of Epistemic States*. Mass.: MIT Press. 1988.
12. Konieczny, S.: Belief base merging as a game. *Journal of Applied Non-Classical Logics*, vol. 14(3), 275-294, 2004.
13. Lin, J. and Mendelzon, A.: Merging databases under constraints. *International Journal of Cooperative information Systems*, vol. 7(1), 55-76, 1998.

Conciliation and Consensus in Iterated Belief Merging

Olivier Gauwin, Sébastien Konieczny, and Pierre Marquis

CRIL – CNRS, Université d’Artois, 62300 Lens, France
{gauwin, konieczny, marquis}@cril.univ-artois.fr

Abstract. Two conciliation processes for intelligent agents based on an iterated merge-then-revise change function for belief profiles are introduced and studied. The first approach is skeptical in the sense that at any revision step, each agent considers that her current beliefs are more important than the current beliefs of the group, while the other case is considered in the second, credulous approach. Some key features of such conciliation processes are pointed out for several merging operators; especially, the convergence issue, the existence of consensus and the properties of the induced iterated merging operators are investigated.

1 Introduction

Belief merging is about the following question: given a set of agents whose belief bases are (typically) mutually inconsistent, how to define a belief base reflecting the beliefs of the group of agents? There are many different ways to address the belief merging issue in a propositional setting (see e.g.[11, 18, 16, 15, 2, 3, 13, 14]). The variety of approaches just reflects the various ways to deal with inconsistent beliefs.

The belief merging issue is not concerned with the way the result is exploited by the group. One possibility is to suppose that all the belief bases are replaced by the (agreed) merged base. This scenario is sensible with low-level agents that are used for distributed computation, or for applications with distributed information sources (like distributed databases). Once the merged base has been computed, all the agents participating to the merging process are equivalent in the sense that they share the same belief base. Such a drastic approach clearly leads to impoverish the beliefs of the system. Contrastingly, when high-level intelligent agents are considered, the previous scenario looks rather unlikely: it is not reasonable to assume that the agents are ready to completely discard their current beliefs and inconditionnally accept the merged base as a new belief base. It seems more adequate for them to incorporate the result of the merging process into their current belief base. Such an incorporation of new beliefs calls for belief revision [1, 7, 8]. In this perspective, two revision strategies can be considered. The first one consists in giving more priority to the previous beliefs; this is the strategy at work for skeptical agents. The second one, used by credulous agents, views the current beliefs of the group as more important than their own, current beliefs. Thus, given a revision strategy, every merging operator Δ induces what we call a conciliation operator which maps every belief profile E (i.e., the beliefs associated to each agent at start) to a new belief profile where the new beliefs of an agent are obtained by confronting her previous beliefs with the merged base given by E and Δ .

Obviously enough, it makes sense to iterate such a merge-then-revise process when the aim of agents is to reach an agreement (if possible): after a first merge-then-revise

round, each agent has possibly new beliefs, defined from her previous ones and the beliefs of the group; this may easily give rise to new beliefs for the group, which must be incorporated into the previous beliefs of agents, and so on. The purpose of this paper is to study the two conciliation processes induced by the two revision strategies for various merging operators under two simplifying assumptions: homogeneity (the same revision operator is used by all the agents) and compatibility (the revision operator used is the one induced by the merging operator under consideration). Some key issues are considered, including the convergence of the processes (i.e., the existence of a round from which no further evolution is possible), the existence of consensus (i.e., the joint consistency of all belief bases at some stage), and the logical properties of the iterated merging operator defined by the last merged base once a fixed point has been reached.

The rest of the paper is organized as follows. In the next section, some formal preliminaries are provided. Section 3 presents the main results of the paper: in Section 3.1 the conciliation processes are defined, in Section 3.2 the focus is laid on the skeptical ones and in Section 3.3 on the credulous ones. In Section 4 we investigate the connections between the conciliation processes and the merging operators they induce. Especially, we give some properties of the corresponding iterated merging operators. Section 5 is devoted to related work. Finally, Section 6 gives some perspectives.

2 Preliminaries

We consider a propositional language $\mathcal{L}$ over a finite alphabet $\mathcal{P}$ of propositional symbols. An interpretation is a function from $\mathcal{P}$ to $\{0, 1\}$. The set of all the interpretations is denoted $\mathcal{W}$. An interpretation ω is a model of a formula K , noted $\omega \models K$, if it makes it true in the usual classical truth functional way. Let K be a formula, $[K]$ denotes the set of models of K , i.e., $[K] = \{\omega \in \mathcal{W} \mid \omega \models K\}$.

A *belief base* K is a consistent propositional formula (or, equivalently, a finite consistent set of propositional formulas considered conjunctively), viewed up to logical equivalence.

Let $K_1, \dots, K_n$ be n belief bases (not necessarily pairwise different). We call *belief profile* the vector E consisting of those n belief bases in a specific order, $E = (K_1, \dots, K_n)$, so that the n^{th} base gathers the beliefs of agent n . We note $\bigwedge E$ the conjunction of the belief bases of E , i.e., $\bigwedge E = K_1 \wedge \dots \wedge K_n$. We say that a belief profile E is consistent if $\bigwedge E$ is consistent. The union operator for belief profiles (actually, of the associated multi-sets) will be noted $\sqcup$.

Let $\mathcal{E}$ be the set of all finite non-empty belief profiles. Two belief profiles E_1 and E_2 from $\mathcal{E}$ are said to be equivalent (noted $E_1 \equiv E_2$) if and only if there is a bijection between the profile E_1 and the profile E_2 s.t. each belief base of E_1 is logically equivalent to its image in E_2 . Note that the order given by the profile is not relevant for equivalence.

For every belief revision operator $*$, every profile $E = (K_1, \dots, K_n)$ and every belief base K , we define the revision of E by K (resp. the revision of K by E) as the belief profile given by $(K_1, \dots, K_n) * K = (K_1 * K, \dots, K_n * K)$ (resp. $K * (K_1, \dots, K_n) = (K * K_1, \dots, K * K_n)$). Since sequences of belief profiles will be considered, we use superscripts to denote belief profiles obtained at some stage, while subscripts are used (as before) to denote belief bases within a profile. For instance, E^i

denotes the belief profile obtained after i elementary evolution steps (in our framework, i merge-then-revise steps), and K_j^i the belief base associated the the j^{th} coordinate of vector E^i (i.e. the beliefs of agent j at step i).

2.1 IC Merging Operators

Some basic work in belief merging aims at determining sets of axiomatic properties valuable operators should exhibit [17, 18, 15, 12, 13, 14]. We focus here on the characterization of Integrity Constraints (IC) merging operators [13, 14].

The aim is to characterize the belief base $\Delta_\mu(E)$, that represents the merging of the profile E under the integrity constraints μ . μ is a formula that encodes some constraints on the result (such as physical constraints, laws, norms, etc...).

Definition 1. Δ is an IC merging operator if and only if it satisfies the following properties:

- (IC0) $\Delta_\mu(E) \models \mu$
- (IC1) If μ is consistent, then $\Delta_\mu(E)$ is consistent
- (IC2) If $\bigwedge E$ is consistent with μ , then $\Delta_\mu(E) \equiv \bigwedge E \wedge \mu$
- (IC3) If $E_1 \equiv E_2$ and $\mu_1 \equiv \mu_2$, then $\Delta_{\mu_1}(E_1) \equiv \Delta_{\mu_2}(E_2)$
- (IC4) If $K_1 \models \mu$ and $K_2 \models \mu$, then $\Delta_\mu(\{K_1, K_2\}) \wedge K_1$ is consistent if and only if $\Delta_\mu(\{K_1, K_2\}) \wedge K_2$ is consistent
- (IC5) $\Delta_\mu(E_1) \wedge \Delta_\mu(E_2) \models \Delta_\mu(E_1 \sqcup E_2)$
- (IC6) If $\Delta_\mu(E_1) \wedge \Delta_\mu(E_2)$ is consistent, then $\Delta_\mu(E_1 \sqcup E_2) \models \Delta_\mu(E_1) \wedge \Delta_\mu(E_2)$
- (IC7) $\Delta_{\mu_1}(E) \wedge \mu_2 \models \Delta_{\mu_1 \wedge \mu_2}(E)$
- (IC8) If $\Delta_{\mu_1}(E) \wedge \mu_2$ is consistent, then $\Delta_{\mu_1 \wedge \mu_2}(E) \models \Delta_{\mu_1}(E)$

For explanations on those properties see [14]. Two subclasses of IC merging operators have been defined. *IC majority operators* aim at resolving conflicts by adhering to the majority wishes, while *IC arbitration operators* exhibit a more consensual behaviour:

Definition 2. An IC majority operator is an IC merging operator which satisfies the following majority postulate:

$$(\mathbf{Maj}) \quad \exists n \quad \Delta_\mu(E_1 \sqcup \underbrace{E_2 \sqcup \dots \sqcup E_2}_n) \models \Delta_\mu(E_2).$$

An IC arbitration operator is an IC merging operator which satisfies the following arbitration postulate:

$$(\mathbf{Arb}) \quad \left. \begin{array}{l} \Delta_{\mu_1}(K_1) \equiv \Delta_{\mu_2}(K_2) \\ \Delta_{\mu_1 \leftrightarrow \neg \mu_2}(\{K_1, K_2\}) \equiv (\mu_1 \leftrightarrow \neg \mu_2) \\ \mu_1 \not\models \mu_2 \\ \mu_2 \not\models \mu_1 \end{array} \right\} \Rightarrow \Delta_{\mu_1 \vee \mu_2}(\{K_1, K_2\}) \equiv \Delta_{\mu_1}(K_1).$$

See [13, 14] for explanations about those two postulates and the behaviour of the two corresponding classes of merging operators. Let us now give some examples of IC merging operators.

Definition 3. A pseudo-distance d between interpretations is a total function $d : \mathcal{W} \times \mathcal{W} \mapsto \mathbb{R}^+$ such that for any $\omega, \omega', \omega'' \in \mathcal{W}$, $d(\omega, \omega') = d(\omega', \omega)$, and $d(\omega, \omega') = 0$ if and only if $\omega = \omega'$.

Two widely used pseudo-distances between interpretations are Dalal distance [6], denoted d_H , which is the Hamming distance between interpretations (i.e., the number of propositional variables on which the two interpretations differ); and the drastic distance, denoted d_D , which is the simplest pseudo-distance one can define: it gives 0 if the two interpretations are the same one, and 1 otherwise.

Definition 4. An aggregation function f is a total function associating a nonnegative real number to every finite tuple of nonnegative real numbers and s.t. for any $x_1, \dots, x_n, x, y \in \mathbb{R}^+$:

- if $x \leq y$, then $f(x_1, \dots, x, \dots, x_n) \leq f(x_1, \dots, y, \dots, x_n)$. (non-decreasingness)
- $f(x_1, \dots, x_n) = 0$ if and only if $x_1 = \dots = x_n = 0$. (minimality)
- $f(x) = x$. (identity)

Widely used functions are the max [18, 14], the sum Σ [18, 16, 13], or the leximax $GMax$ [13, 14]. Then, given a distance d and an aggregation function f , one can define a merging operator $\Delta^{d,f}$:

Definition 5. Let d be a pseudo-distance between interpretations and f be an aggregation function. The result $\Delta_\mu^{d,f}(E)$ of the merging of E given the integrity constraints μ is defined by:

- $d(\omega, K) = \min_{\omega' \models K} d(\omega, \omega')$.
- $d(\omega, E) = f_{\{K_i \in E\}}(d(\omega, K_i))$.
- $\omega \leq_E \omega'$ if and only if $d(\omega, E) \leq d(\omega', E)$.
- $[\Delta_\mu^{d,f}(E)] = \min([\mu], \leq_E)$.

2.2 Merging vs. Revision

Belief revision operators can be viewed as special cases of belief merging operators when applied to singleton profiles, as stated below.

Theorem 1 ([14]). If Δ is an IC merging operator (it satisfies (IC0-IC8)), then the operator $*_\Delta$, defined as $K *_\Delta \mu = \Delta_\mu(K)$, is an AGM revision operator (it satisfies (R1-R6)) [8]. This operator is called the revision operator associated to the merging operator Δ .

3 Conciliation Operators

Conciliation operators aim at reflecting the evolution of belief profiles, typically towards the achievement of some agreements between agents. It can be viewed as a simple form of negotiation, where the way beliefs may evolve is uniform.

3.1 Definitions

Let us first give the following, very general, definition of conciliation operators:

Definition 6. A conciliation operator is a function from the set of belief profiles to the set of belief profiles.

This definition does not impose any strong constraints on the result, except that each resulting belief profile is solely defined from the given one. This does not prevent conciliation operators from taking advantage of additional information as parameters. For instance, integrity constraints representing norms or laws of nature can be taken into account. There are several ways to do it; if one assumes that agents must obey such laws, one can discard from the profile any agent who does not satisfy this requirement; one can also ask each agent to revise her own beliefs by the integrity constraints as a preliminary step so as to ensure it. In the following we adhere to a more liberal attitude and require integrity constraints to be satisfied at the group level, i.e. we do not ask that each agent satisfies the constraints. This relaxation is all the more important when conciliation is about preferences (i.e., goals): each agent is about to change her preferences in the light of the preferences of other agents, in the objective of achieving some agreements; each agent is free to have her own preferences, even if they are unfeasible. Nevertheless, the most preferred alternatives at the group level have to be feasible.

Clearly, pointing out the desirable properties for such conciliation operators is an interesting issue. We let this for future work, but one can note that the *social contraction* functions introduced by Booth [5] are very close to this idea.

In this paper we focus on a particular family of conciliation operators: conciliation operators induced by an iterated merge-then-revise process. The idea is to compute the belief merging from the profile, to revise the beliefs of each source by the result of the merging, and to repeat this process until a fixed point is reached. When such a fixed point exists, the conciliation operator is defined and the resulting profile is the image of the original profile by this operator.

When a fixed point has been reached, incorporating the beliefs of the group has no further impact on the own beliefs of each agent; in some sense, each agent did its best w.r.t. the group, given its revision function. Then there are two possibilities: either a consensus has been obtained, or no consensus can be obtained that way:

Definition 7. *There is a consensus for a belief profile E if and only if E is consistent (with the integrity constraints).*

The existence of a consensus for a belief profile just means that the associated agents agree on at least one possible world. When this is the case, the models of the corresponding merged base w.r.t. any IC merging operator reduce to such possible worlds ((IC2) ensures it). Interestingly, it can be shown that the existence of a consensus at some stage of the merge-then-revise process is sufficient to ensure the existence of a fixed point, hence the termination of the process. Let us now consider two additional properties on conciliation operators in order to keep the framework simple enough: homogeneity and compatibility.

Definition 8. *Let Δ be a revision operator, and let $*_1, \dots, *_n$ be n revision operators. An iterated merging conciliation operator is a function from the set of belief profiles to the set of belief profiles, where the evolution of a profile is characterized by a merge-then-revise approach. It is:*

- *homogeneous if all the agents use the same revision operator $*_1 = \dots = *_n = *$,*
- *compatible if the revision operator is associated to the merging operator $* = *\Delta$.*

In this work, we focus on compatible homogeneous iterated merging conciliation operators (CHIMC in short). Under the compatibility and homogeneity assumptions,

defining a CHIMC operator just requires to make precise the belief merging operator under use and the revision strategy (skeptical or credulous):

Definition 9. Let $\triangle$ be an IC merging operator, and $*$ its associated revision operator (i.e., $\varphi * \mu = \triangle_\mu(\{\varphi\})$). Let E be any belief profile. We define the sequence $(E_s^i)_i$ (depending on both $\triangle$ and E) by:

- $E_s^0 = E$,
- $E_s^{i+1} = \triangle_\mu(E_s^i) * E_s^i$

The skeptical CHIMC operator induced by $\triangle$ is defined by $\triangle_\mu^*(E) = E_s^k$, where k is the lowest rank i such that $E_s^i = E_s^{i+1}$, and $\triangle_\mu^*(E)$ is undefined otherwise. We note $E_s^* = E_s^k$ the resulting profile.

Definition 10. Let $\triangle$ be an IC merging operator, and $*$ its associated revision operator. Let E be any belief profile. We define the sequence $(E_c^i)_i$ by:

- $E_c^0 = E$,
- $E_c^{i+1} = E_c^i * \triangle_\mu(E_c^i)$

The credulous CHIMC operator induced by $\triangle$ is defined by ${}^*\triangle_\mu(E) = E_c^k$, where k is the lowest rank i such that $E_c^i = E_c^{i+1}$, and ${}^*\triangle_\mu(E)$ is undefined otherwise. We note $E_c^* = E_c^k$ the resulting profile.

Every CHIMC operator induces a merging operator: the operator that associates to each profile the merged base of the resulting profile. Formally:

Definition 11. Let $\triangle$ be an IC merging operator, and $*$ its associated revision operator:

- The skeptical CHIM operator induced by $\triangle$ is the function that maps every profile E to $\triangle_\mu(\triangle_\mu^*(E))$.
- The credulous CHIM operator induced by $\triangle$ is the function that maps every profile E to $\triangle_\mu({}^*\triangle_\mu(E))$.

Let us now study the key features of the two sequences $(E_s^i)_i$ and $(E_c^i)_i$ and the properties of the corresponding iterated merging operators, based on various IC merging operators.

3.2 Skeptical Operators

We start with skeptical CHIMC operators. Let us first give an important monotony property, which states that the conciliation process given by any IC merging operator $\triangle$ may only lead to strengthen the beliefs of each agent:

Theorem 2. Let K_j^i denote the belief base corresponding to agent j in the belief profile E_s^i characterized by the initial belief profile E and the IC merging operator $\triangle$. For every i, j , we have $K_j^{i+1} \models K_j^i$.

On this ground, it is easy to prove that the sequence $(E_s^i)_i$ is stationary at some stage, for every profile E and every IC merging operator $\triangle$. Accordingly, the induced skeptical conciliation operator and the induced skeptical iterated merging operator are defined for every E :

Theorem 3. *For every belief profile E and every IC merging operator Δ , the stationarity of $(E_s^i)_i$ is reached at a rank bounded by $(\sum_{K \in E} \#([K]) - \#(E))$. Therefore, the CHIMC operator Δ^* and the CHIM operator $\Delta(\Delta^*)$ are total functions.*

The bound on the number of iterations is easily obtained from the monotony property. Another interesting property is that the sequence of profiles and the corresponding sequence of merged bases are equivalent with respect to stationarity:

Theorem 4. *Let E be a belief profile and Δ be an IC merging operator. Let μ be any integrity constraint. The sequence $(E_s^i)_i$ is stationary from some stage if and only if the sequence $(\Delta_\mu(E_s^i))_i$ is stationary from some stage.*

The number of iterations needed to reach the fixed point of $(E_s^i)_i$ is one for the IC merging operators defined from the drastic distance. More precisely, the skeptical CHIM operator induced by any IC merging operator Δ defined from the drastic distance coincides with Δ .

Theorem 5. *Let $E = (K_1, \dots, K_n)$ be a profile. If the IC merging operator Δ is among $\Delta^{d_D, \text{Max}}$, $\Delta^{d_D, \Sigma}$, $\Delta^{d_D, \text{GMax}}$, then for every j , the base K_j^* from the resulting profile $E^* = \Delta_\mu^*(E)$ can be characterized by:*

$$K_j^* = \begin{cases} \mu \wedge \Delta_\mu(E) & \text{if consistent, else} \\ \Delta_\mu(E) & \text{otherwise.} \end{cases}$$

Furthermore, the resulting profile is obtained after at most one iteration (i.e., for every $i > 0$, $E^i = E^{i+1}$).

We have no direct (i.e., non-iterative) definition for any skeptical CHIM operator based on an IC merging operator defined from Dalal distance. Let us give an example of such an operator.

Example 1. Let us consider the profile $E = (K_1, K_2, K_3)$ with $[K_1] = \{(0, 0, 0), (0, 0, 1), (0, 1, 0)\}$, $[K_2] = \{(0, 1, 1), (1, 1, 0), (1, 1, 1)\}$, $[K_3] = \{(0, 0, 0), (1, 0, 0), (1, 0, 1), (1, 1, 1)\}$, no integrity constraints ($\mu \equiv \top$), and the skeptical CHIMC operator defined from the $\Delta^{d_H, \text{GMax}}$ operator. The complete process is represented in Table 1. The first three columns show the Dalal distance between each interpretation and the corresponding source. The last column shows the distance between each interpretation and the profile according to the aggregation function. So the selected interpretations for the corresponding operators are the ones with minimal aggregated distance. As there are several (three in that case) iterations, we sum up the three tables (corresponding to the three merging steps) in the same one. So, for example in column $d(\omega, K_1^i)$, the first number denotes the distance between the interpretation ω and K_1^1 , the second one the distance between ω and K_1^2 , etc.

Let us explain the full process in details. The first profile is $E^0 = E$. The first merging iteration gives as result $[\Delta^{d_H, \text{GMax}}(E^0)] = \{(0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0)\}$. Then, every source revises the result of the merging with its old beliefs, i.e., $K_i^1 = \Delta^{d_H, \text{GMax}}(E^0) * K_i^0$, so $[K_1^1] = \{(0, 0, 1), (0, 1, 0)\}$, $[K_2^1] = \{(0, 1, 1), (1, 1, 0)\}$ and $[K_3^1] = \{(1, 0, 0), (1, 0, 1)\}$. Since each of the three bases is consistent with the merged base, the new base of each agent is just the conjunction of her previous base with the merged base (in accordance to revision postulates). Then, the second

Table 1. $\Delta_{\mu}^{* d_H, GMax}$

ω	$d(\omega, K_1^i)$	$d(\omega, K_2^i)$	$d(\omega, K_3^i)$	$d_{GMax}(\omega, E^i)_{d(\omega, \Delta_{\mu}(E^i))}$
(0,0,0)	0,1,1	2,2,2	0,1,1	$(2, 0, 0)_1, (2, 1, 1)_1, (2, 1, 1)_1$
(0,0,1)	0,0,0	1,1,3	1,1,1	$(1, 1, 0)_0, (1, 1, 0)_0, (3, 1, 0)_1$
(0,1,0)	0,0,2	1,1,1	1,2,2	$(1, 1, 0)_0, (2, 1, 0)_1, (2, 2, 1)_2$
(0,1,1)	1,1,1	0,0,2	1,2,2	$(1, 1, 0)_0, (2, 1, 0)_1, (2, 2, 1)_2$
(1,0,0)	1,2,2	1,1,1	0,0,0	$(1, 1, 0)_0, (2, 1, 0)_1, (2, 1, 0)_0$
(1,0,1)	1,1,1	1,2,2	0,0,0	$(1, 1, 0)_0, (2, 1, 0)_1, (2, 1, 0)_0$
(1,1,0)	1,1,3	0,0,0	1,1,1	$(1, 1, 0)_0, (1, 1, 0)_0, (3, 1, 0)_1$
(1,1,1)	2,2,2	0,1,1	0,1,1	$(2, 0, 0)_1, (2, 1, 1)_1, (2, 1, 1)_1$

merging iteration gives $[\Delta^{d_H, GMax}(E^1)] = \{(0, 0, 1), (1, 1, 0)\}$, and the revision of each base gives $[K_1^2] = \{(0, 0, 1)\}$, $[K_2^2] = \{(1, 1, 0)\}$, and $[K_3^2] = \{(1, 0, 0), (1, 0, 1)\}$. The third iteration step gives $[\Delta^{d_H, GMax}(E^2)] = \{(1, 0, 0), (1, 0, 1)\}$, and the revision step does not change any belief base, i.e., $E^2 \equiv E^3$, so a stationary point is reached and the process stops on this profile.

As to skeptical operators, the conciliation process cannot lead to a consensus, unless a consensus already exists at start:

Theorem 6. *Let E be a belief profile and Δ be an IC merging operator. There exists a rank i s.t. a consensus exists for E_s^i if and only if $i = 0$ and there is a consensus for E .*

3.3 Credulous Operators

Let us now turn to credulous CHIMC operators. Let us first give some general properties about credulous operators.

Theorem 7. *Let K_j^i now denote the belief base corresponding to agent j in the belief profile E_c^i characterized by the initial belief profile E and the IC merging operator Δ .*

- $\forall i, j \ K_j^{i+1} \models \Delta_{\mu}(E_c^i)$,
- $\forall i > 0 \ \forall j \ K_j^i \models \mu$,
- $\forall i, j$, if $K_j^i \wedge \Delta_{\mu}(E_c^i)$ is consistent, then $K_j^{i+1} \equiv K_j^i \wedge \Delta_{\mu}(E_c^i)$.

The first item states that, during the evolution process, each base implies the previous merged base. The second item states that from the first iteration, all the bases implies the integrity constraints. The last one is simply a consequence of a revision property: if, at a given step, a base is consistent with the result of the merging, then the base at the next step will be that conjunction.

Unfortunately, the monotony property as reported in Theorem 2 does not hold in the credulous case. At that point, we can just conjecture that our credulous CHIMC operators (and the corresponding iterated merging operators) are defined for every profile:

Conjecture 1. For every belief profile E and every merging operator Δ using the aggregation function Max , $GMax$ or Σ , the sequence $(E_c^i)_i$ is stationary from some rank.

This claim is supported by some empirical evidence. We have conducted exhaustive tests for profiles containing up to three bases, when the set of propositional symbols contains up to three variables. The following IC merging operators have been considered: $\Delta^{d_H, \text{Max}}$, $\Delta^{d_H, \text{GMax}}$ and $\Delta^{d_H, \Sigma}$. We have also conducted non-exhaustive tests when four propositional symbols are considered in the language (this leads to billions of tests). All the tested instances support the claim (stationarity is reached in less than five iterations when up to three symbols are considered, and less than ten iterations when four symbols are used).

We can nevertheless prove the stationarity of $(E_c^i)_i$ for every belief profile E when some specific IC merging operators Δ are considered. In particular, for IC merging operators defined from the drastic distance, it is possible to find out a non-iterative definition of the corresponding CHIMC operator, and to prove that it is defined for every profile.

Theorem 8. *Let $E = (K_1, \dots, K_n)$ be a profile. If the IC merging operator is $\Delta^{d_D, \text{Max}}$, then for every j , the base K_j^* from the resulting profile $E^* = *\Delta_\mu^{d_D, \text{Max}}(E)$ can be characterized by:*

$$K_j^* = \begin{cases} \mu \wedge \bigwedge_{K_i: K_i \wedge \mu \neq \perp} K_i & \text{if consistent, else} \\ \mu \wedge K_j & \text{if consistent, else} \\ \mu & \text{otherwise.} \end{cases}$$

Furthermore, the resulting profile is obtained after at most two iterations (i.e., for every $i > 1$, $E^i = E^{i+1}$).

Theorem 9. *Let $E = (K_1, \dots, K_n)$ be a profile. If the IC merging operator is $\Delta^{d_D, \text{GMax}}$ or $\Delta^{d_D, \Sigma}$, then for every j , the base K_j^* from the resulting profile $E^* = *\Delta_\mu^{d_D, \text{GMax}}(E) = *\Delta_\mu^{d_D, \Sigma}(E)$ can be characterized by:*

$$K_j^* = \begin{cases} K_j \wedge \Delta_\mu^{d_D, \text{GMax}}(E) & \text{if consistent, else} \\ \Delta_\mu^{d_D, \text{GMax}}(E) & \text{otherwise.} \end{cases}$$

Furthermore, the resulting profile is obtained after at most one iteration (i.e., for every $i > 0$, $E^i = E^{i+1}$).

Finally, like for the skeptical case, the sequence of profiles and the corresponding sequence of merged bases are equivalent w.r.t. stationarity in the credulous case:

Definition 12. *Let E be a belief profile and Δ be an IC merging operator. Let μ be any integrity constraint. The sequence $(E_s^i)_i$ is stationary from some stage if and only if the sequence $(\Delta_\mu(E_s^i))_i$ is stationary from some stage.*

Let us consider an example of credulous operator at work.

Example 2. Consider the profile $E = (K_1, K_2, K_3, K_4)$, with $[K_1] = \{(0, 0, 0), (0, 0, 1), (0, 1, 0)\}$, $[K_2] = \{(1, 0, 0), (1, 0, 1), (1, 1, 1)\}$, $[K_3] = \{(0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 1, 0)\}$, $[K_4] = \{(0, 1, 1), (1, 0, 0), (1, 1, 0), (1, 1, 1)\}$. There is no integrity constraint $\mu \equiv \top$. Let us consider the credulous CHIMC operator defined from the merging

Table 2. $*\Delta_{\mu}^{d_H, \Sigma}$

ω	$d(\omega, K_1^i)$	$d(\omega, K_2^i)$	$d(\omega, K_3^i)$	$d(\omega, K_4^i)$	$d_{\Sigma}(\omega, E^i)$
(0,0,0)	0,1,1	1,1,1	1,1,1	1,1,1	3,4,4
(0,0,1)	0,0,0	1,2,2	0,0,0	1,1,2	2,3,4
(0,1,0)	0,2,2	2,2,2	0,1,2	1,1,2	3,6,8
(0,1,1)	1,1,1	1,3,3	0,0,1	0,0,3	2,4,8
(1,0,0)	1,2,2	0,0,0	1,1,2	0,0,0	2,3,4
(1,0,1)	1,1,1	0,1,1	1,1,1	1,1,1	3,4,4
(1,1,0)	1,3,3	1,1,1	0,0,3	0,0,1	2,4,8
(1,1,1)	2,2,2	0,2,2	1,1,2	0,1,2	3,6,8

operator $\Delta^{d_H, \Sigma}$. The computations are summarized in Table 2. The resulting profile is $[K_1^2] = \{(0, 0, 1)\}$, $[K_2^2] = \{(1, 0, 0)\}$, $[K_3^2] = \{(0, 0, 1)\}$ and $[K_4^2] = \{(1, 0, 0)\}$. And the corresponding CHIM operator gives as result a base whose models are $\{(0, 0, 0), (0, 0, 1), (1, 0, 0), (1, 0, 1)\}$, that is different from the result of the merging of E by the IC merging operator $[\Delta^{d_H, \Sigma}(E)] = \{(0, 0, 1), (0, 1, 1), (1, 0, 0), (1, 1, 0)\}$.

4 Iterated Merging Operators

An interesting question is to investigate the properties of the CHIM operators. A first important question is whether such operators are IC merging operators. The answer is negative in general, only some basic postulates are guaranteed to hold:

Theorem 10. *Credulous and skeptical CHIM operators satisfy (IC0)-(IC3), (IC7) and (IC8).*

Thus, some important properties of IC merging operators are usually lost through the merge-then-revise process. We claim that this is not so dramatic since the main purpose of conciliation processes is not exactly the one of belief merging. Furthermore, specific iterated merging operators (i.e., those induced by some specific merging operators Δ) may easily satisfy additional postulates:

Theorem 11. *The credulous iterated merging operator associated to $*\Delta_{\mu}^{d_D, Max}$ satisfies (IC0)-(IC5), (IC7)-(IC8) and (Arb). It satisfies neither (IC6) nor (Maj).*

In fact, the CHIM operator defined from $*\Delta_{\mu}^{d_D, Max}$ can be defined as follows (this is a straightforward consequence of Theorem 8):

$$\Delta_{\mu}^{d_D, Max}(*\Delta_{\mu}^{d_D, Max}(E)) = \begin{cases} \mu \wedge \bigwedge_{K_i: K_i \wedge \mu \not\models \perp} K_i & \text{if consistent, else} \\ \mu & \text{otherwise.} \end{cases}$$

Theorem 12. *The credulous iterated operator associated to $*\Delta_{\mu}^{d_D, GMax} = *\Delta_{\mu}^{d_D, \Sigma}$ satisfies (IC0)-(IC8), (Arb) and (Maj).*

This result easily comes from the fact that this credulous CHIM operator actually coincides with the IC merging operator $\Delta_{\mu}^{d_D, GMax} = \Delta_{\mu}^{d_D, \Sigma}$ it is based on.

Thus, as for skeptical operators (see Theorem 5), each CHIM operator based on the drastic distance coincides with the underlying IC merging operator, so it satisfies exactly the same properties (see [14]).

As to the operators based on Dalal distance, things are less easy. Up to now, we did not find an equivalent, non-iterative, definition for any of them. We group the following results on credulous/skeptical operators since they satisfy the same properties, but the proofs of the results are different for the two kinds of operators. Furthermore, since stationarity is only conjectured for credulous operators (cf. Conjecture 1), we do not have a proof that the corresponding CHIM operators are total functions. So the two following results on credulous operators are guaranteed under the conjecture of stationarity, only.

Theorem 13. *The credulous (resp. skeptical) CHIM operator associated to $^*\Delta_{\mu}^{d_H, \Sigma}$ (resp. $\Delta_{\mu}^{*d_H, \Sigma}$) satisfies (IC0)-(IC3), (IC7)-(IC8) and (Maj), but does not satisfy (IC5)-(IC6) and (Arb). The satisfaction of (IC4) is an open issue.*

Theorem 14. *The credulous (resp. skeptical) CHIM operators associated to $^*\Delta_{\mu}^{d_H, Max}$ and $^*\Delta_{\mu}^{d_H, GMax}$ (resp. $\Delta_{\mu}^{*d_H, Max}$ and $\Delta_{\mu}^{*d_H, GMax}$) satisfy (IC0)-(IC3), (IC7)-(IC8), but satisfy none of (IC5)-(IC6), (Maj) and (Arb). The satisfaction of (IC4) is an open issue.*

5 Related Work

In [5, 4] Richard Booth presents what he calls *Belief Negotiation Models*. Such negotiation models can be formalized as games between sources: until a coherent set of sources is reached, at each round a contest is organized to find out the weakest sources, then those sources have to be logically weakened. This idea leads to numerous new interesting operators (depending of the exact meaning of “weakest” and “weaken”, which correspond to the two parameters for this family). Booth is interested at the same time in the evolution of the profile (in connection to what he calls *social contraction*), and to the resulting merged base (the result of the Belief Negotiation Model).

In [10, 9] a systematic study of a subclass of those operators, called *Belief Game Models*, is achieved. This subclass contains operators closer to merging ones than the general class which also allows negotiation-like operators.

All those operators are close in spirit to the CHIMC/CHIM operators defined in this work. A main difference is that in the work presented in this paper, the evolution of a profile does not always lead to a consensus. Scenarios where agents disagree at a final stage are allowed. Whereas in the former work, the evolution process leads to consensus (in fact consensus is the halting condition of the iterative definition). So CHIMC operators seem more adequate to formalize interaction between agents’ beliefs. Thus, they are closer to negotiation processes, since the agents’ beliefs change due to the interaction with other agents’ beliefs, but this interaction can be stopped when the agents have achieved the best possible compromise.

6 Conclusion and Perspectives

In this paper, we have introduced two conciliation processes based on an iterated merge-then-revise change function for the beliefs of agents. On this ground, a family of conciliation operators and an associated family of iterated merging operators have been defined and studied.

This work calls for several perspectives. One of them concerns the stationarity conjecture related to credulous CHIMC operators. A second perspective is about rationality postulates for conciliation operators; such postulates should reflect the fact that at the end of the conciliation process, the disagreement between the agents participating to the conciliation process is expected not to be more important than before; a difficulty is that it does not necessarily mean that this must be the case at each step of a conciliation process.

Furthermore, when a consensus is reached for those conciliation operators, one can use the number of steps needed to reach the consensus as a measure of conflict of the profiles. Such a measure could be used to compare several profiles and to determine what are the less conflictual ones.

Another perspective is to enrich our framework in several directions; one of them consists in relaxing the homogeneity assumption; in some situations, it can prove sensible to consider that an agent is free to reject a negotiation step, would it lead her to a belief state “too far” from its original one; another direction is to study less drastic revision behaviours, for example obtained through non-prioritized belief revision operators.

Acknowledgements

We would like to thank the anonymous reviewers for many helpful comments. The authors have been supported by the Université d’Artois, the Région Nord/Pas-de-Calais, the IRCICA consortium, and by the European Community FEDER Program.

References

1. C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.
2. C. Baral, S. Kraus, and J. Minker. Combining multiple knowledge bases. *IEEE Transactions on Knowledge and Data Engineering*, 3(2):208–220, 1991.
3. C. Baral, S. Kraus, J. Minker, and V. S. Subrahmanian. Combining knowledge bases consisting of first-order theories. *Computational Intelligence*, 8(1):45–71, 1992.
4. R. Booth. A negotiation-style framework for non-prioritised revision. In *Proceedings of the Eighth Conference on Theoretical Aspects of Rationality and Knowledge (TARK’01)*, pages 137–150, 2001.
5. R. Booth. Social contraction and belief negotiation. In *Proceedings of the Eighth Conference on Principles of Knowledge Representation and Reasoning (KR’02)*, pages 374–384, 2002.
6. M. Dalal. Investigations into a theory of knowledge base revision: preliminary report. In *Proceedings of the Seventh American National Conference on Artificial Intelligence (AAAI’88)*, pages 475–479, 1988.

7. P. Gärdenfors. *Knowledge in flux*. MIT Press, 1988.
8. H. Katsuno and A. O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52:263–294, 1991.
9. S. Konieczny. Belief base merging as a game. *Journal of Applied Non-Classical Logics*, 14(3):275–294, 2004.
10. S. Konieczny. Propositional belief merging and belief negotiation model. In *Proceedings of the International Workshop on Non-Monotonic Reasoning (NMR'04)*, pages 249–257, 2004.
11. S. Konieczny, J. Lang, and P. Marquis. DA² merging operators. *Artificial Intelligence*, 157(1-2):49–79, 2004.
12. S. Konieczny and R. Pino Pérez. On the logic of merging. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 488–498, 1998.
13. S. Konieczny and R. Pino Pérez. Merging with integrity constraints. In *Proceedings of the Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'99)*, LNAI 1638, pages 233–244, 1999.
14. S. Konieczny and R. Pino Pérez. Merging information under constraints: a qualitative framework. *Journal of Logic and Computation*, 12(5):773–808, 2002.
15. P. Liberatore and M. Schaerf. Arbitration (or how to merge knowledge bases). *IEEE Transactions on Knowledge and Data Engineering*, 10(1):76–90, 1998.
16. J. Lin and A. O. Mendelzon. Knowledge base merging by majority. In *Dynamic Worlds: From the Frame Problem to Knowledge Management*. Kluwer, 1999.
17. P. Z. Revesz. On the semantics of theory change: arbitration between old and new information. In *Proceedings of the Twelfth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Databases*, pages 71–92, 1993.
18. P. Z. Revesz. On the semantics of arbitration. *International Journal of Algebra and Computation*, 7(2):133–160, 1997.

An Argumentation Framework for Merging Conflicting Knowledge Bases: The Prioritized Case

Leila Amgoud¹ and Souhila Kaci²

¹ Institut de Recherche en Informatique de Toulouse (I.R.I.T.)–C.N.R.S.,
Université Paul Sabatier, 118 route de Narbonne,
31062 Toulouse Cedex 4, France

² Centre de Recherche en Informatique de Lens (C.R.I.L.)–C.N.R.S.,
Rue de l'Université SP 16, 62307 Lens Cedex, France

Abstract. An important problem in the management of knowledge-based systems is the handling of inconsistency. Inconsistency may appear because the knowledge may come from different sources of information. To solve this problem, two kinds of approaches have been proposed. The first category *merges* the different bases into a unique base, and the second category of approaches, such as argumentation, accepts inconsistency and copes with it.

Recently, a “powerful” approach [7, 8, 13] has been proposed to merge *prioritized* propositional bases encoded in possibilistic logic. This approach consists of combining prioritized knowledge bases into a new prioritized knowledge base, and then to infer from this.

In this paper, we present a *particular* argumentation framework for handling inconsistency arising from the presence of multiple sources of information. Then, we will show that this framework retrieves the results of the merging operator defined in [7, 8, 13]. Moreover, we will show that an argumentation-based approach palliates the limits, due to the *drowning* problem, of the merging operator.

Keywords: Argumentation, Information merging, Possibilistic logic.

1 Introduction

In many areas such as cooperative information systems, multi-databases, multi-agents reasoning systems, GroupWare, distributed expert systems, information comes from multiple sources. The multiplicity of sources providing information makes that information is often contradictory and the use of priorities is crucial to solve conflicts.

We distinguish two approaches to deal with contradictory information coming from multiple sources:

- The first approach consists of merging these items of information and constructing a consistent set of information which represents the result of merging [10, 14, 16, 17, 20, 21]. In other words, starting from different bases $B_1, \dots, B_n$ which are conflicting, these works return a *unique consistent base*. Several approaches have been proposed for merging multiple sources of information where priorities are either implicitly [14, 16, 17, 20, 21] or explicitly expressed [7, 8, 13]. Possibilistic

logic [11, 15] is a suitable framework for modeling explicit priorities. It is an extension of classical logic which allows to model prioritized information encoded by means of weighted propositional formulas. Possibilistic logic has a syntactic inference which is sound and complete w.r.t. semantics based on the notion of possibility distributions [11]. Merging prioritized information in this framework turns out to build from sets of prioritized information a new set of prioritized information, from which inferences are drawn.

- The second approach consists of solving the conflicts without merging the bases. Argumentation is one of the most promising of these approaches. It is based on the construction of arguments and counter-arguments (defeaters) and the selection of the most acceptable of these arguments.

The present paper completes the results presented in [4] where the *relationship* between information merging, when priorities are implicitly expressed, and argumentation theory has been established.

In this paper, we consider the case of priorities expressed *explicitly* in a possibilistic logic framework. We will show that the results of the merging operator defined in [7, 8, 13] are retrieved in a *particular* argumentation framework. In that framework, the arguments are built from the different bases, and each argument has an *intrinsic force* based on the certainty level of the information used in that argument. Moreover, we will show that an argumentation-based approach palliates the limits, due to the *drowning* problem, of the merging operator. All the proofs of the results given in this paper can be found in [3].

The paper is organized as follows: section 2 recalls briefly the basics of possibilistic logic. Section 3 introduces a merging operator based on possibilistic logic. In section 4 a general preference-based argumentation framework is presented. Section 5 connects argumentation theory with the merging operator defined in section 3. Section 6 is devoted to some concluding remarks and perspectives.

2 Brief Refresher on Possibilistic Logic

Let us consider a propositional language $\mathcal{L}$ over a finite alphabet $\mathcal{P}$ of atoms. Ω denotes the set of all the interpretations. Logical equivalence is denoted by $\equiv$ and classical conjunction and disjunction are respectively denoted by $\wedge, \vee$. $\vdash$ denotes classical inference. The notation $\omega \models \phi$ means that the interpretation ω is a model of (or satisfies) the formula ϕ .

At the semantic level, possibilistic logic is based on the notion of a possibility distribution [22], denoted by π , which is a mapping from Ω to $[0,1]$ representing the available information. $\pi(\omega)$ represents the degree of compatibility of the interpretation ω with the available beliefs about the real world if we are representing uncertain pieces of knowledge (or the degree of satisfaction of reaching a state ω if we are modeling preferences). By convention, $\pi(\omega) = 1$ means that it is totally possible for ω to be the real world (or that ω is fully satisfactory), $1 > \pi(\omega) > 0$ means that ω is only somewhat possible (or satisfactory), while $\pi(\omega) = 0$ means that ω is certainly not the real world (or not satisfactory at all). Associated with a possibility distribution π is the necessity degree of any formula ϕ : $N(\phi) = 1 - \Pi(\neg\phi)$ which evaluates to what extent ϕ is entailed by

the available beliefs, and defined from the consistency degree of a formula ϕ w.r.t. the available information, $\Pi(\phi) = \max\{\pi(\omega) : \omega \models \Omega \text{ and } \omega \models \phi\}$.

Note that the mapping N reverses the scale on which π is ranging, and that $N(\phi) = 1$ means that ϕ is a totally certain piece of knowledge or a compulsory goal, while $N(\phi) = 0$ expresses the complete lack of knowledge or of priority about ϕ , but does not mean that ϕ is or should be false. Moreover, the duality equation $N(\phi) = 1 - \Pi(\neg\phi)$ extends the existing one in classical logic, where a formula is entailed from a set of classical formulas if and only if its negation is consistent with this set.

At the syntactic level, prioritized items of information are represented by means of a *possibilistic knowledge base* (or a *possibilistic base* for short) which is a set of weighted formulas of the form $B = \{(\phi_i, a_i) : i = 1, \dots, n\}$, where ϕ_i is a propositional formula and a_i belongs to a totally ordered scale such as $[0, 1]$. The pair (ϕ_i, a_i) means that the certainty degree of ϕ_i is at least equal to a_i ($N(\phi_i) \geq a_i$). We denote by B^* the propositional base associated with B , namely the base obtained from B by forgetting the weights of formulas. A possibilistic base B is consistent if and only if its associated propositional base B^* is consistent.

Given a possibilistic base B , we can generate a unique possibility distribution, denoted by π_B , such that all the interpretations satisfying all the formulas in B will have the highest possibility degree, namely 1, and the other interpretations will be ranked w.r.t. the highest formula that they falsify, namely we get [11]:

Definition 1. $\forall \omega \in \Omega$,

$$\pi_B(\omega) = \begin{cases} 1 & \text{if } \forall (\phi_i, a_i) \in B, \omega \models \phi_i \\ 1 - \max\{a_i : (\phi_i, a_i) \in B \text{ and } \omega \not\models \phi_i\} & \text{otherwise.} \end{cases}$$

Example 1. Let $B = \{(\neg p \vee \neg q, .7); (p, .6)\}$ be a knowledge base. Its associated possibility distribution is: $\pi_B(p \neg q) = 1$; $\pi_B(\neg p \neg q) = \pi_B(\neg pq) = .4$ and $\pi_B(pq) = .3$. The interpretation $p \neg q$ is the most preferred since it satisfies all the formulas in B . The interpretations $\neg p \neg q$ and $\neg pq$ are more preferred than pq since the highest formula falsified by $\neg p \neg q$ and $\neg pq$ (i.e., $(p, .6)$) is less certain (or less prioritized) than the highest formula falsified by pq (i.e., $(\neg p \vee \neg q, .7)$).

In the following, we give some definitions useful for the rest of the paper:

Definition 2. Let B_1 and B_2 be two possibilistic bases. B_1 and B_2 are said to be equivalent, denoted by $B_1 \equiv_s B_2$, iff $\pi_{B_1} = \pi_{B_2}$.

Definition 3 (a-cut and strict a-cut). Let B be a possibilistic knowledge base, and $\mathbf{a} \in [0, 1]$. We call the **a-cut** (resp. **strict a-cut**) of B , denoted by $B_{\geq \mathbf{a}}$ (resp. $B_{> \mathbf{a}}$), the set of propositional formulas in B having a certainty degree at least equal to $\mathbf{a}$ (resp. strictly greater than $\mathbf{a}$).

Definition 4 (Inconsistency degree). The inconsistency degree of a possibilistic base B is:

$$Inc(B) = \max\{a_i : B_{\geq a_i} \text{ is inconsistent}\},$$

with $Inc(B) = 0$ when B is consistent.

Definition 5 (Subsumption). Let (ϕ, a) be a formula in B . (ϕ, a) is said to be subsumed in B if:

$$(B - \{(\phi, a)\})_{\geq a} \vdash \phi.$$

And (ϕ, a) is said to be strictly subsumed in B if $B_{>a} \vdash \phi$.

Subsumed formulas are in some sense redundant formulas as it is shown by the following lemma [8]:

Lemma 1. Let (ϕ, a) be a subsumed formula in B . Then B and $B' = B - \{(\phi, a)\}$ are equivalent.

Lastly, weights are propagated out in the inference process in the following way:

Definition 6 (Plausible inference). Let B be a possibilistic base. The formula ϕ is a plausible consequence of B iff

$$B_{>Inc(B)} \vdash \phi.$$

Definition 7 (Possibilistic inference). Let B be a possibilistic base. The formula (ϕ, a) is a possibilistic consequence of B , denoted $B \vdash_{\pi} (\phi, a)$, iff

- $B_{>Inc(B)} \vdash \phi$,
- $a > Inc(B)$ and $\forall b > a, B_{>b} \not\vdash \phi$.

3 Merging Prioritized Information in Possibilistic Logic Framework

Merging prioritized information in possibilistic logic is a two step process:

1. From a set of possibilistic bases¹, computing a new possibilistic base, called the *aggregated base*, which is generally inconsistent [8].
2. Inferring conclusions from the new base.

A possibilistic merging operator, denoted by $\oplus$, is a function from $[0, 1]^n$ to $[0, 1]$. $\oplus$ is used to aggregate the certainty degrees associated with pieces of information provided by different sources. Formally, let $\mathcal{B} = \{B_1, \dots, B_n\}$ be a set of n (possibly inconsistent) possibilistic bases. The result of merging the bases of $\mathcal{B}$ using $\oplus$, denoted by $\mathcal{B}_{\oplus}$, is defined as follows [7]:

Definition 8 (Aggregated base). Let $\mathcal{B} = \{B_1, \dots, B_n\}$ be a set of possibilistic bases and $\oplus$ a merging operator. The result of merging $\mathcal{B}$ with $\oplus$ is defined by:

$$\mathcal{B}_{\oplus} = \{(D_j, \oplus(x_1, \dots, x_n)) : j = 1, \dots, n\},$$

where D_j are disjunctions of size j between formulas taken from different B_i 's ($i = 1, \dots, n$) and x_i is either equal to a_i or to 0 depending respectively on whether ϕ_i belongs to D_j or not.

¹ These bases may be individually inconsistent.

Two properties for $\oplus$ are assumed in this definition [9, 8]:

1. $\oplus(0, \dots, 0) = 0$,
2. If $a_i \geq b_i$ for all $i = 1, \dots, n$ then $\oplus(a_1, \dots, a_n) \geq \oplus(b_1, \dots, b_n)$.

The first property says that if a formula doesn't explicitly appear in any base, then it should not appear explicitly in the result of merging. The second property is simply the unanimity property (called also monotonicity property) which means that if all the sources say that a formula ϕ is more plausible than (or preferred to) another formula ψ , then the result of merging should confirm this preference.

Example 2. Let $B_1 = \{(\phi \vee \psi, .9), (\neg\phi, .8), (\xi, .1)\}$ and $B_2 = \{(\neg\psi, .7), (\phi, .6)\}$. Let $\oplus$ be the probabilistic sum defined by $\oplus(a, b) = a + b - ab$. Following Definition 8, we get:

$\mathcal{B}_{\oplus} = \{(\phi \vee \psi, .9), (\neg\phi, .8), (\xi, .1)\} \cup \{(\neg\psi, .7), (\phi, .6)\} \cup \{(\phi \vee \psi, .96), (\neg\phi \vee \neg\psi, .94), (\xi \vee \neg\psi, .73), (\xi \vee \phi, .64)\}$ which is equivalent to $\{(\phi \vee \psi, .96), (\neg\phi \vee \neg\psi, .94), (\neg\phi, .8), (\xi \vee \neg\psi, .73), (\neg\psi, .7), (\xi \vee \phi, .64), (\phi, .6), (\xi, .1)\}$.

Lemma 2 gives a rewriting of $\mathcal{B}_{\oplus}$ given in Definition 8 which will be useful in the rest of the paper, but first let us give the following definition:

Definition 9 (Existential consequence). Let B be a possibilistic base. The formula (ϕ, a) is an existential consequence of B , denoted by $B \Vdash (\phi, a)$, iff:

1. $\exists B' \subseteq B$ s.t. $B' \vdash_{\pi} (\phi, a)$,
2. B' is consistent,
3. $a = \min\{a_i : (\phi_i, a_i) \in B'\}$,
4. B' is a minimal for set inclusion,
5. $\nexists B'' \subseteq B$ satisfying the above conditions with $B'' \vdash_{\pi} (\phi, b)$ and $b > a$.

This definition focuses on the subbases containing the most prioritized formulas.

Example 3. Let $B = \{(\phi \vee \psi, .9), (\neg\phi, .7), (\xi \vee \psi, .6), (\neg\xi, .5)\}$. Then $B \Vdash (\phi \vee \psi, .9)$, $B \Vdash (\neg\phi, .7)$ and $B \Vdash (\psi, .7)$ however $B \nVdash (\neg\psi, 0)$.

Lemma 2. Let $\mathcal{B}_{\oplus}$ be the result of merging $\mathcal{B} = \{B_1, \dots, B_n\}$ with $\oplus$. Then, $\mathcal{B}_{\oplus}$ is equivalent to

$$\{(\phi, \oplus(a_1, \dots, a_n)) : \phi \in \mathcal{L} \text{ and } B_i \Vdash (\phi, a_i)\}.$$

Now that the base $\mathcal{B}_{\oplus}$ is defined, we are ready to define the result of merging. This corresponds to the possibilistic consequences of $\mathcal{B}_{\oplus}$. Formally:

Definition 10 (Useful result of merging). Let $\mathcal{B} = \{B_1, \dots, B_n\}$ be a set of n possibilistic bases, $\oplus$ be a merging operator and $\mathcal{B}_{\oplus}$ be the result of merging $\mathcal{B}$ with $\oplus$. The useful result of merging is:

$$\mathcal{T} = \{(\phi_i, a_i) \mid \mathcal{B}_{\oplus} \vdash_{\pi} (\phi_i, a_i)\}.$$

4 Basic Argumentation Framework

Argumentation is a reasoning model based on the construction and the comparison of arguments. Argumentation frameworks have been developed for decision making under uncertainty [5], and for handling inconsistency in knowledge bases where each conclusion is justified by arguments [1, 18]. Arguments represent the reasons to believe in a fact. In what follows, we present the general framework proposed in [2] which is an extension of the famous framework presented by Dung in [12].

Definition 11 (Argumentation framework). *An argumentation framework (AF) is a triplet $\langle \mathcal{A}, \mathcal{R}, \succeq \rangle$, where $\mathcal{A}$ is a set arguments, $\mathcal{R}$ is a binary relation representing defeat relationship between arguments and $\succeq$ is a (partial or complete) pre-ordering on $\mathcal{A} \times \mathcal{A}$. The strict ordering associated with $\succeq$ is denoted $\succ$.*

Since arguments are conflicting, it is important to define the acceptable ones (i.e the “good” ones). Different semantics have been introduced in [12]. In what follows, we will focus only on one of them, the so-called *grounded extension*.

The preference order between arguments makes it possible to distinguish different types of relations between arguments:

Definition 12. *Let A, B be two arguments of $\mathcal{A}$.*

- *B attacks A iff $B \mathcal{R} A$ and it is not the case that $A \succ B$.*
- *If $B \mathcal{R} A$ then A defends itself against B iff $A \succ B$.*
- *A set of arguments $\mathcal{S}$ defends A if there is some argument in $\mathcal{S}$ which attacks every argument B where B attacks A .*

Henceforth, $C_{\mathcal{R}, \succeq}$ will gather all non-defeated arguments and arguments defending themselves against all their defeaters. In [2], it has been shown that the set $\underline{\mathcal{S}}$ of acceptable arguments of the argumentation framework $\langle \mathcal{A}, \mathcal{R}, \succeq \rangle$ is the least fixpoint of a function $\mathcal{F}$:

$$\begin{aligned} \mathcal{S} &\subseteq \mathcal{A} \\ \mathcal{F}(\mathcal{S}) &= \{A \in \mathcal{A} \mid A \text{ is defended by } \mathcal{S}\}. \end{aligned}$$

Definition 13. *The set of acceptable arguments for an argumentation framework $\langle \mathcal{A}, \mathcal{R}, \succeq \rangle$ is:*

$$\begin{aligned} \underline{\mathcal{S}} &= \bigcup \mathcal{F}_{i \geq 0}(\emptyset) \\ &= C_{\mathcal{R}, \succeq} \cup \left[\bigcup \mathcal{F}_{i \geq 1}(C_{\mathcal{R}, \succeq}) \right]. \end{aligned}$$

An argument is acceptable if it is a member of the acceptable set.

5 Relating Merging in Possibilistic Logic with Argumentation

In section 4, we have introduced a general argumentation framework. In that framework, the structure and the origin of arguments are not defined. Similarly, the defeasibility and the preference relations between arguments are not given too. In what follows, we will give an instantiation of the above framework for handling inconsistency in knowledge bases, especially when the inconsistency occurs because of the presence of different and conflicting sources of information (let's say, $B_1, \dots, B_n$). We will then show that the obtained system retrieves the results of the merging operator introduced in section 3.

Let's first recall some concepts. Let $B_1, \dots, B_n$ be different possibilistic bases. $Disj$ will denote the set of all disjunctions of different size that can be formed from formulas of the n bases. $Conj$ will denote the set of formulas of $B_1, \dots, B_n$ with possibly new weights. Weights of formulas in $Disj$ and $Conj$ are aggregated using an operator $\otimes$. For instance, if the formula (ϕ, a) is in B_1 and (ψ, b) is in B_2 , then the formula $(\phi \vee \psi, \otimes(a, b))$ will be in $Disj$ and the formulas $(\phi, \otimes(a, 0))$ and $(\psi, \otimes(0, b))$ will be in $Conj$, with $\otimes(x, y)$ is $\max(x, y)$ or $\min(x, y)$ etc. In what follows, $\mathcal{B} = Conj \cup Disj$. In fact, it can be shown that if the aggregation operator $\otimes$ is exactly the operator $\oplus$, then the two bases $\mathcal{B}$ and $\mathcal{B}_\oplus$ are equivalent.

Proposition 1. *Let $B_1, \dots, B_n$ be different possibilistic bases. If $\otimes = \oplus$, then the bases $\mathcal{B}$ and $\mathcal{B}_\oplus$ are equivalent.*

Let's start now by defining the notion of argument. An argument has a deductive form and takes the form of an explanation. Each argument is constructed from formulas of $B_1, \dots, B_n$ and disjunctions between formulas of these bases.

Definition 14 (Argument). *An argument is a pair $\langle H, h \rangle$, where h is a formula of the language $\mathcal{L}$ and H a subset of $\mathcal{B}$ satisfying:*

1. $H \subseteq \mathcal{B}^*$,
2. $H \vdash h$,
3. H is consistent and minimal (no strict subset of H satisfies 1 and 2).

H is called the support and h the conclusion of the argument.

$\mathcal{A}(\mathcal{B})$ will denote the set of all arguments that can be built from $\mathcal{B}$.

Note that it is not necessary to construct the bases $Disj$ and $Conj$ in order to define the arguments. Fragments of these bases are constructed only when needed i.e., when building arguments.

The most appropriate defeat relation which will capture all the different kinds of conflicts which may exist between arguments is the following relation “undercut”.

Definition 15 (Undercut relation). *Let $\langle H, h \rangle$ and $\langle H', h' \rangle$ be two arguments of $\mathcal{A}(\mathcal{B})$. $\langle H, h \rangle$ undercuts $\langle H', h' \rangle$ iff for some $k \in H'$, $h \equiv \neg k$. An argument is undercut if there exists at least one argument against one element of its support.*

In [1], it has been argued that arguments may have forces of various strengths. These forces allow an agent to compare different arguments in order to select the ‘best’ ones.

When explicit priorities are given between the beliefs, such as certainty degrees, the arguments using more certain beliefs are found stronger than arguments using less certain beliefs. The force of an argument corresponds to the *certainty degree* of the less entrenched belief involved in the argument.

Definition 16 (Force of an argument). Let $A = \langle H, h \rangle$ be an argument. The force of A , denoted by $force(A)$, is

$$force(A) = \min\{a_i : \phi_i \in H \text{ and } (\phi_i, a_i) \in \mathcal{B}\}.$$

The following proposition shows that an argument and its force can be constructed from $\mathcal{B}$ without computing explicitly the base $Disj$.

Proposition 2. Let $B_1, \dots, B_n$ be n possibilistic bases. Let $A = \langle H, \phi \rangle$ be an argument in $\mathcal{A}(\mathcal{B})$. It holds that:

- $\forall \phi_j \in H, B_i \Vdash (\phi_j, a_{ji}), i=1, \dots, n.$
- $force(A) = \min\{\otimes(a_{j1}, \dots, a_{jn}) \text{ with } \text{and } a_j = \otimes(a_{j1}, \dots, a_{jn})\}.$

Example 4. Let's compute an argument for $\phi \vee \psi$ from $\mathcal{B}_\oplus$. We get $A_1 = \langle \{\phi \vee \psi\}, \phi \vee \psi \rangle$ and $A_2 = \langle \{\phi\}, \phi \vee \psi \rangle$.

A_1 is stronger than A_2 since $force(A_1) = .96$ whereas $force(A_2) = .6$.

Now $B_1 \Vdash (\phi \vee \psi, .9)$ and $B_2 \Vdash (\phi \vee \psi, .6)$. Then, $force(A_1) = \min\{\oplus(.9, .6)\} = .96$.

The forces of arguments make it possible to compare pairs of arguments as follows:

Definition 17 (Preference relation). Let A and A' be two arguments in $\mathcal{A}(\mathcal{B})$. A is preferred to A' , denoted by $A \succ A'$, iff $force(A) > force(A')$.

Example 5. Let us consider again the possibilistic base given in Example 3: $B = \{(\phi \vee \psi, .9), (\neg\phi, .7), (\xi \vee \psi, .6), (\neg\xi, .5)\}$. There are two arguments in favor of ψ :

- $A_1 = \langle \{\phi \vee \psi, \neg\phi\}, \psi \rangle,$
- $A_2 = \langle \{\xi \vee \psi, \neg\xi\}, \psi \rangle.$

However, it is clear that A_1 is preferred to A_2 since $force(A_1) = .7$ whereas $force(A_2) = .5$.

Definition 18 (Acceptable arguments). Let $\langle \mathcal{A}(\mathcal{B}), Undercut, \succ \rangle$ be an argumentation framework. Its set of acceptable arguments is:

$$\begin{aligned} \underline{\mathcal{S}} &= \bigcup \mathcal{F}_{i \geq 0}(\emptyset) \\ &= C_{Undercut, \succ} \cup \left[\bigcup \mathcal{F}_{i \geq 1}(C_{Undercut, \succ}) \right] \end{aligned}$$

An important result states that the obtained set of acceptable arguments is not conflicting. Moreover, the set of formulas that constitute that set of acceptable arguments is consistent.

Definition 19. Let $T \subseteq \mathcal{A}(\mathcal{B})$. $\text{Supp}(T) = \cup H_i$ such that $\langle H_i, h_i \rangle \in T$.

Proposition 3. Let $\langle \mathcal{A}(\mathcal{B}), \text{Undercut}, \succ \rangle$ be an argumentation framework.

1. $\nexists A, B \in \underline{\mathcal{S}}$ such that A undercuts B .
2. $\text{Supp}(\underline{\mathcal{S}})$ is consistent.

We can show easily that any plausible consequence of a given possibilistic base B_i is supported by an acceptable argument, if we consider only the arguments $\mathcal{A}(B_i)$ built from that base B_i .

Proposition 4. Let B_i be a possibilistic base, and let $\langle \mathcal{A}(B_i), \text{Undercut}, \succ \rangle$ be an argumentation framework and $\underline{\mathcal{S}}$ its set of acceptable arguments. If ϕ is a plausible consequence of B_i , then $\exists A = \langle H, \phi \rangle \in \underline{\mathcal{S}}$.

Another interesting result states that any possibilistic consequence (ϕ, a) of a given possibilistic base B_i is supported by an acceptable argument A whose force is equal to a . Moreover, A is the strongest argument w.r.t $\succ$ in favor of ϕ . This means that the degree a of a possibilistic consequence ϕ corresponds to the force of the best argument in favor of ϕ .

Proposition 5. Let B_i be a possibilistic base, and let $\langle \mathcal{A}(B_i), \text{Undercut}, \succ \rangle$ be an argumentation framework and $\underline{\mathcal{S}}$ its set of acceptable arguments.

If (ϕ, a) is a possibilistic consequence of B_i , then $\exists A = \langle H, \phi \rangle \in \underline{\mathcal{S}}$ with $\text{force}(A) = a$, and $\forall A' = \langle H', \phi \rangle \in \underline{\mathcal{S}}, A \succ A'$.

An important concept in possibilistic logic is that of *inconsistency degree* of a possibilistic base B_i . In what follows, we will show that that inconsistency degree can be computed from the forces of the conflicting arguments as follows:

Proposition 6. Let B be a possibilistic base, and let $\langle \mathcal{A}(B), \text{Undercut}, \succ \rangle$ be an argumentation framework.

$$\text{Inc}(B) = \max\{\min(\text{force}(A_i), \text{force}(A_j)) \mid A_i \text{ undercuts } A_j\}.$$

Example 6. Let's consider the base $\mathcal{B}_\oplus$ constructed in Example 2: $\mathcal{B}_\oplus = \{(\phi \vee \psi, .96), (\neg\phi \vee \neg\psi, .94), (\neg\phi, .8), (\xi \vee \neg\psi, .73), (\neg\psi, .7), (\xi \vee \phi, .64), (\phi, .6), (\xi, .1)\}$.

Table 1 summarizes the different arguments which can be constructed from $\mathcal{B}_\oplus$ and their force. As we mentioned before, note that we only focus on the best arguments (i.e., having the highest force) in favor of formulas. For example, there is an argument $A = \langle \{\phi\}, \phi \vee \psi \rangle$, with a force equal to .6, in favor of $\phi \vee \psi$ however it is not considered since there is another argument A_1 in favor of $\phi \vee \psi$ with a higher force.

We have $\text{Undercut} = \{(A_6, A_3), (A_6, A_4), (A_7, A_5), (A_7, A_6), (A_6, A_7)\}$. Then, $\max\{\min(.7, .8), \min(.7, .73), \min(.8, .7), \min(.8, .7), \min(.7, .8)\} = .7$. It can be checked that the inconsistency degree of $\mathcal{B}_\oplus$ is .7.

Table 1

Argument	Force
$A_1 = \langle \{\phi \vee \psi\}, \phi \vee \psi \rangle$	.96
$A_2 = \langle \{\neg\phi \vee \neg\psi\}, \neg\phi \vee \neg\psi \rangle$	.94
$A_3 = \langle \{\neg\phi\}, \neg\phi \rangle$	.8
$A_4 = \langle \{\xi \vee \neg\psi, \neg\phi, \phi \vee \psi\}, \xi \rangle$	.73
$A_5 = \langle \{\neg\psi\}, \neg\psi \rangle$	.7
$A_6 = \langle \{\phi \vee \psi, \neg\psi\}, \phi \rangle$	.7
$A_7 = \langle \{\neg\phi, \phi \vee \psi\}, \psi \rangle$	.8

Indeed we have the following result:

Proposition 7. *Let B be a possibilistic base.*

1. *A formula ϕ is a plausible consequence of B iff $\exists A = \langle H, \phi \rangle$ in $\mathcal{A}(B)$ s.t. $\text{force}(A) > \text{Inc}(B)$.*
2. *A formula (ϕ, a) is a possibilistic consequence of B iff $\exists A = \langle H, \phi \rangle$ in $\mathcal{A}(B)$ s.t. $\text{force}(A) > \text{Inc}(B)$ and $\text{force}(A) = a$.*

Example 7. *Let's consider the different arguments of Example 6. Only the arguments having a weight strictly greater than .7 are considered. Namely A_1, A_2, A_3, A_4 and A_7 . Thus, the plausible consequences of $\mathcal{B}_\oplus$ are $\phi \vee \psi, \neg\phi \vee \neg\psi, \neg\phi, \xi$ and ψ . The possibilistic consequences of $\mathcal{B}_\oplus$ are $(\phi \vee \psi, .96), (\neg\phi \vee \neg\psi, .94), (\neg\phi, .8), (\xi, .73)$ and $(\psi, .8)$.*

From the previous propositions, it can be shown that the result of merging is captured in argumentation framework. Formally:

Theorem 1. *Let $B_1, \dots, B_n$ different possibilistic bases, and $\langle \mathcal{A}, \text{Undercut}, \succeq \rangle$ be an argumentation framework. If $\oplus = \otimes$ then the following result holds:*

$$\mathcal{T} \subseteq \text{Supp}(\underline{\mathcal{S}}),$$

where $\mathcal{T}$ is given in Definition 10.

The above result shows that an argumentation framework is “stronger” than the merging operator defined in section 3 in the sense that it may return more results. The reason is that possibilistic logic suffers from the so-called *drowning* problem. A drowning problem means that some information which are not responsible of conflicts may be ignored [6]. More precisely, formulas at the level and below the inconsistency degree are ignored.

Example 8. *Let us consider again the bases B_1 and B_2 given in Example 2. Let $\oplus$ be the max operator. Then,*

$$\mathcal{B}_\oplus = B_1 \cup B_2 = \{(\phi \vee \psi, .9), (\neg\phi, .8), (\neg\psi, .7), (\phi, .6), (\xi, .1)\}.$$

Using the inference in possibilistic logic, plausible consequences are $\phi \vee \psi, \neg\phi$ and ψ while the argumentation-based inference gives $\{\phi \vee \psi, \neg\phi, \psi, \xi\}$.

6 Conclusion

We presented in this paper an argumentation-based framework for resolving conflicts between knowledge bases in a prioritized case where priorities are represented in possibilistic logic framework. The proposed approach is different from the classical way used in the literature to deal with conflicting multiple sources information.

The classical existing approaches consist of first merging individual bases into a new base from which conclusions are drawn. The new base is composed of the most prioritized consistent formulas. The drawback of this approach is that it may ignore formulas which are not responsible for the conflicts.

The argumentation-based approach proposed here builds arguments from the separate bases, evaluates them and lastly computes a set of acceptable arguments from which conclusions are drawn.

The main result of the work presented in this paper is that the argumentation framework captures the result of the merging operator defined in [7, 8, 13] without merging the different bases. This is of great importance since merging the bases is computationally very costly. Moreover, it is not always interesting to merge the bases as it is the case in a multi-agent system. In such a system, each agent has its own base which may conflict with the bases of the other agents.

We have shown also that the argumentation-based framework solves the drowning problem. Consequently, it returns more formulas than the approach which merges the bases.

An extension of this work would be to study the behavior of the argumentation-based approach proposed in this paper from a postulate point of view inspired from the description of possibilistic merging operators from postulate point of view given in [9]. We are also planning to investigate how argumentation framework can capture the result of merging when multiple-operators are used as in [19]. In that work, two merging operators are used for consistent and conflicting formulas respectively. Another extension consists of comparing the argumentation-based approach and the merging-based approach from a complexity in space and time point of view.

References

1. L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *International Journal of Automated Reasoning*, Volume 29, N2:125–169, 2002.
2. L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34:197–216, 2002.
3. L. Amgoud and S. Kaci. An argumentation framework for merging conflicting knowledge bases: The prioritized case. In *Technical report. Artois University, CRIL.*, 2005.
4. L. Amgoud and S.D. Parsons. An argumentation framework for meging conflicting knowledge bases. In *Proceedings of International Conference on Logics in Artificial Intelligence*, pages 27–37, 2002.
5. L. Amgoud and H. Prade. Using arguments for making decisions. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 10–17, 2004.
6. S. Benferhat, D. Dubois, C. Cayrol, J. Lang, and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *13th International Joint Conference on Artificial Intelligence IJCAI'93*, pages 640–645, 1993.

7. S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Possibilistic merging and distance-based fusion of propositional information. In *Annals of Mathematics and Artificial Intelligence*, volume 34(1-3), pages 217–252, 2002.
8. S. Benferhat, D. Dubois, H. Prade, and M. Williams. A practical approach to fusing and revising prioritized belief bases. In *Proceedings of EPIA 99, LNAI n° 1695*, Springer Verlag, pages 222–236, 1999.
9. S. Benferhat and S. Kaci. Fusion of possibilistic knowledge bases from a postulate point of view. *International Journal on Approximate Reasoning*, 33:255–285, 2003.
10. L. Cholvy. Reasoning about merging information. *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, 3:233–263, 1998.
11. D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, D. Gabbay et al., eds, 3, Oxford University Press:pages 439–513, 1994.
12. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games. *Artificial Intelligence*, 77:321–357, 1995.
13. S. Kaci. Connaissances et Préférences: Représentation et fusion en logique possibiliste. In *Thèse de doctorat. Université Paul Sabatier. Toulouse*, 2002.
14. S. Konieczny and R. Pino Pérez. On the logic of merging. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, Trento, pages 488–498, 1998.
15. J. Lang. Possibilistic logic: Complexity and algorithms. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, 5:179–220, 2000.
16. J. Lin. Integration of weighted knowledge bases. *Artificial Intelligence*, 83:363–378, 1996.
17. J. Lin and A. Mendelzon. Merging databases under constraints. *International Journal of Cooperative Information Systems*, 7(1):55–76, 1998.
18. H. Prakken and G. Sartor. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics*, 7:25–75, 1997.
19. G. Qi, W. Liu, and D.H. Glass. Combining individually inconsistent prioritized knowledge bases. In *Proceedings of the international workshop on non-monotonic reasoning*, 2004.
20. N. Rescher and R. Manor. On inference from inconsistent premises. *Theory and Decision*, 1:179–219, 1970.
21. P. Z. Revesz. On the semantics of theory change: arbitration between old and new information. In *12th ACM SIGACT-SIGMOD-SIGART symposium on Principles of Databases*, pages 71–92, 1993.
22. L. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

Probabilistic Transformations of Belief Functions

Milan Daniel*

Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod vodárenskou věží 2, CZ - 182 07, Prague 8, Czech Republic
`milan.daniel@cs.cas.cz`

Abstract. Alternative approaches to the widely known pignistic transformation of belief functions are presented and analyzed. A series of various probabilistic transformations is examined namely from the point of view of their consistency with rules for belief function combination and their consistency with probabilistic upper and lower bounds. A new definition of general probabilistic transformation is introduced and a discussion of their applicability is included.

Keywords: Belief function, Dempster-Shafer theory, Probabilistic transformation, Pignistic probability, Combination consistency, *ulb*-consistency.

1 Introduction

Belief functions are formalisms widely used for uncertainty representation and processing. For combination of the beliefs the Dempster's rule of combination is used in the Dempster-Shafer theory. Besides, series of modifications of the Dempster's rule were suggested and alternative approaches were created: e. g. Transferable Belief Model (TBM) using the so called non-normalized Dempster's rule [28], combination 'per elements' [5] with its special case — minC combination, see [6], and others. Subsequently, numerous practical applications were suggested and implemented in a wide range of domains.

What is common for their applications? It is an aim to transform the resulting evidence representation by a general belief function to representation by probability for the purpose of easier decision making, resulting beliefs comparison and ordering. Such a probability should be consistent with the original belief function. In fact, we can consider it as a belief function of a special type, so called Bayesian belief function. We call such a transformation as a

Frequently only a special case of probabilistic transformation – Pignistic transformation — is used. In the last years several papers on alternative probabilistic transformations have been published [2, 3, 10, 11, 31, 32], and a new justification of pignistic transformation has appeared [29, 30].

This paper summarizes and completes the study of probabilistic transformations presented in [10, 11, 13]. Besides the new original results, Baroni & Vicigs's

* Partial support by the COST action 274 TARSKI acknowledged.

results from [2] and Cobb & Shenoy's results [3], the present study includes also Sudano's transformations [31, 32] and Smets' new results [29, 30].

Basic notions, both general, and those from [10] and [11] are introduced in Section 2. Section 3 presents a series of probabilistic transformations from various sources and it shows that some of them are equivalent to other one(s). Section 4 brings a summary of consistencies of the transformations. A new definition of the general probabilistic transformation based on their analysis and a justification of two main alternatives to pignistic transformation is presented in Section 5. A discussion about which transformation should be applied in applications concludes the paper.

2 Preliminaries

2.1 Basic Notions

Let us first recall some basic notions from the theory of belief functions. Let us consider an n -element frame of discernment¹ $\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}$. A

() is a mapping $m : \mathcal{P}(\Omega) \longrightarrow [0, 1]$ such that $\sum_{A \subseteq \Omega} m(A) = 1$; the values of the bba are called (). If $m(\emptyset) = 0$, we speak about (BF) is a mapping $bel : \mathcal{P}(\Omega) \longrightarrow [0, 1]$,

$bel(A) = \sum_{\emptyset \neq X \subseteq A} m(X)$. $\mathcal{P}(\Omega)$ is often denoted by 2^Ω . Let us further recall a $Pl(A) = \sum_{\emptyset \neq X \subseteq A} m(X)$.

A is a subset X of the frame of discernment, such that $m(X) > 0$. If all the focal elements are (i.e. one-element subsets of Ω), then we speak about a B , it is a probability distribution on Ω in fact. If all the focal elements are either singletons or whole Ω (i.e. $|X| = 1$ or $|X| = |\Omega|$), then we speak about a $-B$, it is something like 'non-normalized probability distribution'.

To underline the cardinality of a frame of discernment, we use the left lower indices, e.g. $n_D bel(X)$, $3_D m(X)$, etc., and we speak about nD BF bel , 3D bba m , etc. Let $2_D 0 = (0, 0)$ and $n_D 0 = (0, \dots, 0)$ denote special BF's bel_0 such that $m_0(\Omega) = 1$, $2_D 0' = (\frac{1}{2}, \frac{1}{2})$ and $n_D 0' = (\frac{1}{n}, \dots, \frac{1}{n}, 0, \dots, 0)$ denote special BF's $bel_{0'}$ such that $m_{0'}(X) = \frac{1}{n}$ for $|X| = 1$.

$T \quad D$, () is given as $(m_1 \oplus m_2)(A) = \sum_{X \cap Y = A} K m_1(X) m_2(Y)$ for $A \neq \emptyset$, where $K = 1 / (1 - \sum_{X \cap Y = \emptyset} m_1(X) m_2(Y)) = \frac{1}{1 - \kappa}$, and $m(\emptyset) = 0$, see [26]; putting $K = 1$ and $m(\emptyset) = \sum_{X \cap Y = \emptyset} m_1(X) m_2(Y) = \kappa$ we obtain the $\ominus$, see e.g. [28].

T is given by the formula $(m_1 \odot m_2)(A) = \sum_{X \cup Y = A} m_1(X) m_2(Y)$, see [19]. B is defined as a normalized point-wise multiplication of probabilities of singletons.

$$(P_1 \otimes P_2)(x) = \frac{P_1(x) P_2(x)}{\sum_{y \in \Omega} P_1(y) P_2(y)}.$$

¹ We use the classical Shaferian terminology. Besides, it is also possible to use the new more user-friendly simplification of the terminology suggested by Dempster, see e.g. [15], using a notion *state space* instead of a frame of discernment, and similarly.

2.2 General Definition of Probabilistic Transformations

Let us consider the following very general definition now ². A (or briefly a) is a mapping $T : Bel_{\Omega} \longrightarrow ProbDistr_{\Omega}$. Thus the probabilistic transformation assigns a Bayesian belief function (i.e. probability distribution) to every general one. It is a reason why the transformations of belief functions to probability distributions are sometimes called also Bayesian transformations, see e.g. [33]. As we suppose finite frames of discernments, we can compute $(T(bel))(X) = \sum_{A \in X} (T(bel))(A)$ for any $X \subseteq \Omega$.

The fundamental well know example of a probabilistic transformation is the pignistic transformation $BetT$ and its resulting pignistic probability $BetP$ ³ introduced by Smets. We do not use the name pignistic transformation for the other ones, and we use the general name probabilistic transformation, in accordance with Philippe Smets' wish not to mix new alternatives together with his classical pignistic transformation. Moreover, it allows us to use a more general definition with less assumptions.

2.3 ulb-Consistency and p-Consistency

Probabilistic transformation PT is () if its resulting transformed probability TP satisfies the following consistency condition: $Bel(X) \leq TP(X) \leq Pl(X) = 1 - Bel(\overline{X})$. Probabilistic transformation PT is () if $PT(m) = m$ for any Bayesian bba m . In other words Bayesian BFs are fix points of p-consistent PTs. p-consistency is in fact ulb-consistency on Bayesian BFs (i.e. weakening of ulb-consistency) because $bel(X) = Pl(X)$ for Bayesian BFs.

2.4 Combination Consistencies

A of a PT is based on commutation of a combination rule $\star$ with PT, i.e. we obtain the same results if we combine beliefs bel_1 and bel_2 using the combination rule $\star$ and perform PT after it as in the case, where we first compute probabilistic transformations of the both input beliefs bel_1 and bel_2 and combine them with the combination rule $\star$ after.

Probabilistic transformation PT is $\oplus$ - if it commutes with the Dempster's rule (with $\oplus$ combination). Analogically ⁴ PT is $\odot$ -consistent if it commutes with $\odot \circ u$. Where u stands for the nD generalization of the original 2D homomorphism $u: {}_{2D}u(a, b) = (a, b) \odot (\frac{1}{a+b}, \frac{1}{a+b}) = (\frac{a}{a+b}, \frac{b}{a+b})$, and its

² For precision of the definition see Section 5.

³ We denote all transformations with suffix T and related probabilities with P .

⁴ It is possible to define analogically other combination consistencies w.r.t. to other combination rules, see e.g. $\odot$ -consistency [11]. Due to the limitation of applicability of the *consensus operator* $\odot$ [8, 24] to quasi-Bayesian BFs only [9], we omit a presentation of $\odot$ -consistency in this text.

nD generalization $u(x_1, \dots, x_n, x_{n+1}, \dots, x_{2^n-1}) = (\frac{x_1}{\sum_{i=1}^n x_i}, \dots, \frac{x_n}{\sum_{i=1}^n x_i}, 0, \dots, 0)$, see [7, 12].

3 Probabilistic Transformations

3.1 Pignistic Transformation

The *BetT* distributes $m(X)$ equally among all elements of X . It was named and justified by Smets in [27] for Transferable Belief Model (TBM), see [27, 28] in 1990. Nevertheless, the transformation based on the same principle was used by Dubois & Prade [18] as "equidistribution of the values of bba" and by Williams [34] in 1982 already.

The pignistic transformation *BetT* projects BF *bel* given bba m to probability *BetP* defined on the frame of discernment Ω as follows:

$$BetP(A) = \sum_{A \in X \subseteq \Omega} \frac{1}{|X|} \frac{m(X)}{1 - m(\emptyset)}.$$

It includes normalization and division of bbms assigned to focal elements by their cardinality, non-normalized beliefs used in TBM are admissible.

The justification of the pignistic transformation is based on the assumption of the so called *linearity property*, see e. g. [29, 30], i. e. on commutation of the transformation with a convex combination of beliefs: $T(\alpha m_1 + (1 - \alpha)m_2) = \alpha T(m_1) + (1 - \alpha)T(m_2)$.⁵ This property was originally derived from the so called *α -consistency*, see [27]. In correspondence with the definition of combination consistencies we can call the linearity property assumption as *α -consistency*. No justification of the transformation has been presented by Dubois & Prade or by Williams.

From the definition and justification of the pignistic transformation, we can immediately see that it is *ulb-consistent* and *α -consistent*. *BetT* is neither $\oplus$ -consistent nor $\odot$ -consistent.

3.2 Plausibility or Cautious Probabilistic Transformation

Let us introduce three different definitions of the main alternative to pignistic probability in this subsection.

Widely known it the following one. The *Pl.T*, see e.g. [2] or [3], is defined as a normalized plausibility of singletons⁶. Hence we have

$$Pl.P(A) = \frac{Pl(A)}{\sum_{B \in \Omega} Pl(B)} = \frac{\sum_{A \in X \subseteq \Omega} m(X)}{\sum_{B \in \Omega} \sum_{B \in X \subseteq \Omega} m(X)}.$$

⁵ The special case of a convex combination of bbas for $\alpha = \frac{1}{2}$ was mentioned as averaging of bbas in [11].

⁶ Despite of the fact that, Cobb and Shenoy introduce it as a new method [3] in 2003, and Sudano also introduces it as *PrNPl* in 2003, it was known already in 1991 [1].

This transformation is called 'the pignistic transformation' (*PrNPl*) by Sudano in [32].⁷

The combination of a belief *bel* with $0'$: $CautT(bel) = bel \oplus 0'$. It is a generalization of homomorphism *h*, which corresponds to Hájek & Valdés results on 2D belief functions [21, 22]: ${}_2DCautP(A) = \frac{1-m(B)}{2-m(A)-m(B)}$.⁸ In the nD case we have:

$$CautP(A) = \frac{\sum_{A \in X} m(X)}{n - \sum_{B \in \Omega, X \subseteq \Omega, B \notin X} m(X)}.$$

V (or B) (VBT)⁹ published in 1989, see [2] and [33], is given by

$$VBP(A) = \frac{\sum_{A \in X} m(X)}{\sum_{Y \subseteq \Omega} (m(Y) \cdot |Y|)}.$$

Theorem 1. *T* (or *B*) (VBT) is a belief transformation. $CautP(A) = Pl_P(A) = VBP(A)$.

For equality $CautT \equiv Pl_T$ see [13], and for equality $Pl_T \equiv VBT$ see [2].

Pl_T is $\oplus$ -consistent. It is neither $\odot$ -consistent nor α -consistent. Pl_T is neither *ulb*-consistent in general. It is *ulb*-consistent for quasi-Bayesian BF's only; it implies *p*-consistency in general on nD and *ulb*-consistency on 2D BF's.

3.3 Belief or Disjunctive Probabilistic Transformation

In [10], the transformation $DisjT$ has been presented which has been defined on 2D frames so that it commutes with $\odot \circ u$, $DisjP(\{A\}) = \frac{m(\{A\})}{m(\{A\}) + m(\Omega - \{A\})}$. Its nD generalization [13] is given by the following formula:

$$DisjP(\{A\}) = \frac{m(A)}{\sum_{X \in \Omega} m(X)}.$$

A (or B) Bel.T [11] is defined as a normalization of beliefs of singletons (bbms of singletons), i.e. by the same

⁷ This name does not correspond to Smets' wish of using the name of the pignistic transformation, besides it does not satisfy all assumptions required from Smets' pignistic transformation, either the original [27, 28] or the recent ones [29, 30]. For this reason we eliminate the word 'pignistic' from the name of the transformation and add a letter *T* (or *P*) to abbreviation of the transformation (or resulting probability) to obtain *PrNPlT* (or *PrNPlP*) to be consistent with the other names. The same holds also for the other Sudano's transformations, see [31, 32].

⁸ This 2D transformation was used already in the Expert System Shell EQUANT-PC in late 80's, see [20].

⁹ Voorbraak proposed VBT not for decision making, but for approximation of BF's.

formula. Thus it is evident that $Bel\_T \equiv DisjT$. We have to note that Bel_T is not defined if $\sum_{X \in \Omega} m(X) = 0$; we can complete its definition analogically to the proportional transformation, see later, but such a definition breaks the $\odot$ -consistency which was a motivation for definition of $DisjT$. Further, we have to note that Bel_T is significantly sensitive to the bbms of singletons because it ignores completely the bbms of non-singleton focal elements.

Bel_T is $\odot$ -consistent, it is not $\oplus$ -consistent. It is neither α -consistent nor ulb -consistent in general. It is ulb -consistent only for quasi-Bayesian BF's; it implies p -consistency in general on nD and ulb -consistency on 2D BF's.

3.4 Proportional Probabilistic Transformations

Proportional transformations take bbm $m(A)$ of a singleton A and add to it proportional parts of $m(X)$ for all its supersets $A \subset X$. From this assumption it is obvious that these $Prop_{Bel}T$ are ulb -consistent.

If the proportionalization is computed with respect to the beliefs of singletons, we speak about the $Prop_{Bel}T$, see [11, 13]:

$$Prop_{Bel}P(A) = \sum_{A \in X \subseteq \Omega} \frac{m(A)}{\sum_{B \in X} m(B)} \cdot m(X).$$

If $\sum_{B \in X} m(B) = 0$, then $|X|$ is used instead of it and thus $m(X)$ is relocated per the same portions among all elements of X in such a case.

The equivalent $PrBlT$, see [31, 32], is based on the same idea as $Prop_{Bel}T$, also the formula for computing of $PrBlP$ corresponds to that for computing $Prop_{Bel}P$. Hence $PrBlT \equiv Prop_{Bel}T$.

In order to correct a statement from [11], we have to note that the equivalence $Bel\_T \equiv Prop_{Bel}T$ holds on 2D and nD quasi-Bayesian BF's only, see [14].

$Prop_{Bel}P(A)$ is defined for all BF's, but similarly to Bel_T it is also significantly sensitive to the bbms of singletons. To improve it, the $StProp_{Bel}T$ or simply

$StBel\_T$ has been defined in [11]. Bbms $m^{(i-1)}(X)$ for $|X| = (n + 1 - i)$ are proportionally relocated in the i -th step among $m^{(i)}(Y)$ for $Y \subset X$, $|Y| = (n - i)$. $m^{(i)}(Z) = m^{(i-1)}(Z) = m(Z)$ for $|Z| < n - i$, and $m^{(i)}(Z) = 0$ for $|Z| > n - i$. If $\sum_{Y \subset X, |Y|=|X|} m(Y) = 0$ then $|X|$ is used instead of it, thus $m(X)$ is relocated per the same portions among all Y in such a case.

If the proportionalization is computed with respect to the plausibilities of singletons, we speak about the $Prop_{Pl}T$, see [11], which is defined by

$$Prop_{Pl}P(A) = \sum_{A \in X \subseteq \Omega} \frac{Pl(A)}{\sum_{B \in X} Pl(B)} \cdot m(X).$$

The equivalent $PrPlT$ [31, 32] is based on the same idea as $Prop_{Pl}T$, also the formula for computing of $PrPl$ corresponds to that for computing $Prop_{Pl}P$. Hence $PrPlT \equiv Prop_{Pl}T$.

Two other probabilistic proportional transformations are defined by Sudano in [31], see also [32]. P $PraPlT$ and iterative $PrScT$.

$$PraPlP(A) = m(A) + \frac{1 - \sum_{B \in \Omega} m(B)}{\sum_{B \in \Omega} Pl(B)} \cdot Pl(A).$$

$PraPlT$ is equal to $PrPlT$ and $PropPlT$ on 2D and on nD qBBFs, but it does not satisfy our introductive assumption of proportional probabilistic transformations. Moreover, it is not *ulb*-consistent in general, even if its *ulb*-consistency is assumed and claimed in [31]¹⁰. Nevertheless, $PraPlT$ satisfies the weaker *p*-consistency.

$$PrScP(A) = \sum_{A \in X} \frac{PrScP(A)}{\sum_{B \in X} PrScP(B)} \cdot m(X).$$

$PrScT$ transformation satisfies our assumption, thus it is really *ulb*-consistent. Sudano's $PrHybT$ [32] is also *ulb*-consistent.

$$PrHybP(A) = \sum_{A \in X} \frac{PraPlP(A)}{\sum_{B \in X} PraPlP(B)} \cdot m(X).$$

Analogically to starting a proportional transformation from the bbms or the beliefs of singletons $m(a) = bel(A)$ and adding some proportions of $m(X)$ to it for $A \in X$, we can start from $Pl(A)$ and remove some proportions of $m(X)$ from it, see [11, 14].

4 Summary of Consistencies of Probabilistic Transformations

The reason of defining the new transformations in [11] was an endeavour to find a probabilistic transformation which is both $\oplus$ -consistent and *ulb*-consistent or $\ominus$ -consistent and *ulb*-consistent. This endeavour was unsuccessful, on contrary it is possible to prove the following theorem.

Theorem 2. (.) PlT $\oplus$ -
(.) $BelT$ $\ominus$ - PT p -
(.) $BetT$ α - PT p -
 S , A I , S 5
[30].

¹⁰ A counter-example: $m(\{a\}) = m(\{b\}) = m(\{c\}) = 0.1$, $m(\{a, b\}) = 0.7$, we obtain $PrPl(\{a\}) = PrPl(\{b\}) = 0.4294$ and $PrPl(\{c\}) = 0.1412 > 0.1 = Pl(\{c\})$.

For proofs of (i) and (ii) see [14], (iii) follows Smets' necessity of pignistic transformation [30]. From Theorem 2 the following corollary immediately follows.

Corollary 1. $(\oplus-)T$ $ulb-$ PlT T $-B$

$(\oplus-)T$ $ulb-$ $BelT$ T $-B$

$(\oplus-)T$ S $\oplus-$ $\ominus-$ D

$(\oplus-)T$ $\oplus-$ $\ominus-$ (W)

Hence there is no need to look for another new probabilistic transformation. We can summarize consistencies of probabilistic transformations in Table 1.

Table 1. Consistencies of probabilistic transformations

	$\oplus$ -consistency	$\ominus$ -consistency	α -consistency	ulb -consistency	p -consistency
PlT	$\oplus$ -consistent	no	no	2D BFs nD qBBFs	yes
$BelT^*$	no	$\ominus_I$ -consistent	no	2D BFs nD qBBFs	yes
$BetT$	no	no	α -consistent	ulb -consistent	yes
$Prop_{Bel}T$	no	2D BFs - $(0, 0)$ nD qBBFs - $n_D 0$	no	ulb -consistent	yes
$StBelT$	no	2D BFs - $(0, 0)$ nD qBBFs - $n_D 0$	no	ulb -consistent	yes
$Prop_{Pl}T$	no	no	no	ulb -consistent	yes
$PraPlT$	no	no	no	2D BFs nD qBBFs	yes

* $BelT$ is not defined for BFs such that $\sum_{A \in \Omega} m(A) = 0$.
qBBFs stands for quasi Bayesian belief functions.
All these transformations are $\oplus-$, $\ominus-$, and α -consistent on nD Bayesian BFs.

We have to recall the following equivalencies: $PlT \equiv CautT \equiv VBT \equiv PrNPlT$, $BelT \equiv DisjT$, $Prop_{Bel}T \equiv PrBlT$, and $Prop_{Pl}T \equiv PrPlT$. On 2D BFs and on nD quasi-Bayesian BFs (qBBFs) it holds further $BelT \equiv Prop_{Bel}T \equiv StBelT$, and $Prop_{Pl}T \equiv PrPlT \equiv PraPlT$. The equivalency $\oplus \equiv \ominus \circ u \equiv \otimes$ holds on general nD Bayesian BFs, see [12].

5 Justification of Probabilistic Transformations

The recent justification of pignistic transformation is presented in [29, 30]. Let us make a general justification of the probabilistic transformations, which have been studied in this text.

Let us assume that a general probabilistic transformation PT is a function from the set of all belief functions to the Bayesian ones, i. e. to the set of probabilistic distributions on Ω . $PT(m) = P$, where $P(X) = PT(m)(X) = m'(X)$. It includes Smets' assumption of $C \rightarrow P \rightarrow L$, see Proposition 3.1 in [30]. Smets' assumption of E , see Proposition 4.1 in [30], also holds because $P(\Omega) = \sum_{A \in \Omega} P(A) = \sum_{A \in \Omega} m'(A) = bel'(\Omega) = 1$. All the studied transformations are p -consistent, thus we can, without loss of generality, assume this very natural assumption which requires that Bayesian BF's are transformed back to themselves. It corresponds to the Smets' P assumption, see Proposition 3.2 from [30].

All our probabilistic transformations satisfy also the Smets' assumption of A , i.e. independence of the result of transformation on permutation of elements of Ω , see Proposition 4.2 in [30], and the assumption of I

requiring probability of an impossible event equal to zero, see Proposition 4.3 in [30].

The L , see Proposition 1.1 in [30], i.e. α -consistency in our terminology, is the only Smets' assumption that we do not include in our general assumptions. We can summarize our assumptions to the following definition.

Definition 1. $A \rightarrow P \rightarrow L$ probabilistic transformation $B \rightarrow P$ is a function $PT: B \rightarrow P$ such that:

- (i) p -consistency, $\dots$ $PT(bel) = bel$ $B \rightarrow BF\ bel$,
- (ii) $PT(bel)(X) = 0$ $X, \dots$ X $Pl(X) = 0$,
- (iii) $\dots$ $TP(bel^*)(R(X)) = P^*(R(X)) = P(X) = TP(bel)(X)$,
 $R \quad \Omega \quad BF\ bel^* \quad m^*(R(X)) = m(X)$.

Theorem 3. $L \rightarrow D \rightarrow 1. T \rightarrow$

(i) $I \rightarrow \dots$ $(-)$ α - $\dots$,
 $BetT$.
(ii) $I \rightarrow \dots$ $(-)$ $\oplus$ - $\dots$,
 PlT .
(iii) $I \rightarrow \dots$ $(-)$ $\odot$ - $\dots$,
 $BelT$.

The proofs of the statements immediately follow Definition 1, Theorem 2, and properties of the transformations. Note that both Cobb & Shenoy's I and I [3] follow the assumption (iv-c) of $\oplus$ -consistency.

The addition of an assumption of the *ulb*-consistency does not justify any unique probabilistic transformation. On the other hand, it excludes *Pl.T* and *Bel.T*, hence we do not assume any *ulb*-consistency in our new definition of probabilistic transformations.

6 Applicability of Probabilistic Transformations

Several probabilistic transformations have been presented and compared in this text. None of them is the best of all in general. Thus a natural question arises: Which probabilistic transformation should be used in our applications? As the answer is not unique, we will discuss it in this section.

The answer depends on the reason why we want to compute the probabilistic transformation and how we want to use it: Whether our goal is only to find the most prospective element of the frame of discernment or whether we have some specific assumptions to the result, and what operations we want to perform with the resulting probability.

Let us assume that we have all our evidence represented with BFs, i.e. that there is no other explicit nor implicit information about bbms assigned to multi-element focal elements. If we want to use a transformed probability for betting, we have to follow the Smets' necessity of pignistic transformation and compute pignistic probabilities. Nevertheless, we have to use them strictly on the pignistic level and to keep in mind that we cannot handle pignistic probabilities like the Bayesian BFs and combine them with the conjunctive or disjunctive rule of combination and similarly.

If we assume that the belief corresponds to lower probability and the plausibility to upper probability, we have to use some of the *ulb*-consistent probabilistic transformations. Similarly as before, we have to keep in mind that we have left the credal level and that we cannot handle probabilities as Bayesian BFs. If we, moreover, assume the α -consistency, then it is the only possibility of the pignistic probability again.

If we assume or want to be prepared for a combination of the resulting probabilities with the conjunctive combination, we have to use $\oplus$ -consistent transformation, i.e. *Pl.T*. It is just the case of Cobb & Shenoy's assumptions. Similarly, if we assume disjunctive or α -combination of the resulting probabilities we have to use $\ominus$ - or α -consistent transformation, i.e. *Bel.T* or *Bet.T* respectively.

If we are interested in selection of the most plausible element we have to use normalized plausibility transformation *Pl.T*. For determining the most believable element we have to use normalized belief *Bel.T* or preferably its stepwise version *StBel.T*. In the case where $\ominus$ rule and *Bel.T* are used, we can handle probability as a Bayesian belief and combine it with $\ominus$. While in the case *StBel.T* we have to keep in mind that the credal level was left.

In the case of general looking for the most prospective element of the frame of discernment (without any other assumption) we can select a transformation with regard to its interpretation, see [10, 13].

If we have some other information on the domain, on the belief functions which are transformed or some special requirements to the resulting probabilities, we can use some special probabilistic transformation.

We assume that the evidence about application domain is represented with belief functions. It is called the *pignistic transformation* by Smets. By applying the pignistic transformation we leave this level and move us to the *probabilistic level*. In the case that we do not assume α -consistency and do not use the pignistic transformations, we cannot speak longer about the pignistic level than about the *probabilistic level*. or, more generally, about the *probabilistic level* of a representation and a solution of the decisional task.

7 Conclusion

A series of probabilistic transformations of belief functions have been analyzed and compared in this text, namely from the point of view of combination consistencies. They have different pros and cons. It has been shown that there does not exist a probabilistic transformation which is the best in general.

A new definition of probabilistic transformations which covers all the investigated transformations has been presented.

A particular discussion about which transformation should be applied in applications concludes the paper. It has been shown that both the Smets' approach of the necessity of the pignistic transformation and the Cobb & Shenoy's necessity of the normalized plausibility transformation are right within their assumptions which are mutually different. Besides, the other assumptions tend to other alternative solutions.

References

1. Appriou, A.: Probabilités et Incertitude en Fusion de Données Multisenseurs. *Revue Scientifique et Technique de la Défense* **11** (1991) 27–40.
2. Baroni, P., Vici, P.: Transformations from Imprecise to Precise Probabilities. In: Nielsen, T. D., Zhang, N. L. (eds.): *Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2003); Lecture Notes in Artificial Intelligence* 2711, Springer-Verlag (2003) 37–49.
3. Cobb, B. R., Shenoy, P. P.: A Comparison of Methods for Transforming Belief Functions Models to Probability Models. In: Nielsen, T. D., Zhang, N. L. (eds.): *Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2003); Lecture Notes in Artificial Intelligence* 2711, Springer-Verlag (2003) 255–266.
4. Cohen, M. S.: An expert system framework for non-monotonic reasoning about probabilistic assumptions. In: Kanal, L.N., Lemmer, J.F. (eds.): *Uncertainty in Artificial Intelligence 1*. North-Holland (1986) 279–293.
5. Daniel, M.: Associativity and Contradiction in Combination of Belief Functions. In: *Proceedings Eight International Conference Information Processing and Management of Uncertainty, IPMU (2000)* 133–140.
6. Daniel, M.: Associativity in combination of belief functions; a derivation of minC combination. *Soft Computing* **7** (2003) 288–296.

7. Daniel, M.: Algebraic Structures Related to the Combination of Belief Functions. *Scientiae Mathematicae Japonicae* **60**/ 2 (2004) 245–255. *Scientiae Mathematicae Japonicae Online* **10** (2004) 501–511.
8. Daniel, M.: Algebraic Structures Related to the Consensus Operator for Combining of Beliefs. In: Nielsen, T. D., Zhang, N. L. (eds.): *Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2003)*; LNAI 2711, Springer-Verlag (2003) 332–344.
9. Daniel, M.: Combination of Belief Functions on Two-element and on General n -element Frame of Discernment. Ramík, J. (ed.): *Proceedings of 6th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty* (2003).
10. Daniel, M.: Transformations of Belief Functions to Probabilities. In: Vejnarová, J. (ed.): *Proceedings of 6th Workshop on Uncertainty Processing (WUPES 2003)*; VŠE - Oeconomica Publishers (2003) 77–90.
11. Daniel, M.: Consistency of Probabilistic Transformations of Belief Functions. In: *Proceedings of the Tenth International conference IPMU* (2004) 1135–1142.
12. Daniel, M.: A Relation of Conjunctive and Disjunctive Rules of Combination on Bayesian Belief Functions. In: Noguchi, H., Ishii, H., Inuiguchi, M. (eds.): *Proceedings of 7th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty* (2004) 179–184.
13. Daniel, M.: Transformations of Belief Functions to Probabilities. *International Journal of Intelligent Systems*. (in print).
14. Daniel, M.: On Probabilistic Transformations of Belief Functions. Tech. Rep. V-934, Inst. of Comp. Sci., Academy of Sci. of the Czech Rep., Prague (2005).
15. Dempster, A. P.: How to "sell" the Dempster-Shafer theory. Oral presentation in WUPES 2003, Hejnice, Czech republic, September 24–27, 2003.
16. Dezert, J.: Foundations for a New Theory of Plausible and Paradoxical Reasoning. *Information and Security Journal* **9** (2002).
17. Dezert, J., Smarandache, F., Daniel, M.: The Generalized Pignistic Transformation. In: Svensson, P., Schubert, J. (eds.): *Proceedings of the Seventh International Conference on Information Fusion, FUSION 2004* (2004) 384–391.
18. Dubois, D., Prade, H.: On several representations of an uncertain body of evidence. In: Gupta, M. M., Sanchez, E. (eds.): *Fuzzy Information and Decision Processes*. North-Holland, Amsterdam (1982) 167–181.
19. Dubois, D., Prade, H.: Consonant Approximations of Belief Functions. *International Journal of Approximate Reasoning* **4** (1990) 419–449.
20. Hájek, P., Hájková, M., Havránek, T., Daniel, M.: The Expert System Shell EQUANT-PC: Brief information. *Kybernetika* 1–3 **25** (1989) suppl. 4–9.
21. Hájek, P., Havránek, T., Jiroušek, R.: *Uncertain Information Processing in Expert Systems*. CRC Press, Boca Raton, Florida (1992).
22. Hájek, P., Valdés, J. J.: Generalized algebraic foundations of uncertainty processing in rule-based expert systems (dempsteroids). *Computers and Artificial Intelligence* **10** (1991) 29–42.
23. Chateaufneuf, A., Jaffray, J.-Y.: Some Characterizations of lower probabilities and other monotone capacities through the use of Moebius Inversion. *Mathematical Social Sciences* **17** (1989) 263–283.
24. Josang, A.: The Consensus Operator for Combining Beliefs. *Artificial Intelligence Journal* **141**/1–2 (2002) 157–170.
25. Lefevre, E., Colot, O., Vannoorenberghe, P.: Belief Functions Combination and Conflict Management. *Information Fusion* **3**/2 (2002) 149–162.
26. Shafer, G.: *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, New Jersey (1976).

27. Smets, Ph.: Constructing the Pignistic Probability Function in a Context of Uncertainty. Henrion, M., Schachter, R. D., Kanal, L. N., Lemmer, J. F. (eds.): Uncertainty in Artificial Intelligence 5. North Holland, Amsterdam (1990) 29–39.
28. Smets, Ph., Kennes, R.: The transferable belief model. *Artificial Intelligence* **66** (1994) 191–234.
29. Smets, Ph.: Decision Making in a Context where Uncertainty is Represented by Belief Functions. In: Srivastava, R. P., Mock, T. J. (eds.): *Belief Functions in Business Decision*. Physica-Verlag, Heidelberg, Germany, (2002) 17–61.
30. Smets, Ph.: Decision Making in the TBM: the Necessity of the Pignistic Transformation. *International Journal of Approximative Reasoning* **38** (2005) 133–147.
31. Sudano, J. J.: Pignistic Probability transforms for Mixes of Low- and High-Probability Events. In: *Proc. of the 4th Int. Conf. on Information Fusion (Fusion 2001)*, Montreal, Canada (2001) TUB3 23–27.
32. Sudano, J. J.: Equivalence Between Belief Theories and Naïve Bayesian Fusion for Systems with Independent Evidential Data: Part II, the Example. In: *Proc. of the 6th Int. Conf. on Information Fusion (Fusion 2003)*, Cairns, Australia, (2003) 1357–1364.
33. Voorbraak, F.: A Computationally Efficient Approximation of Dempster-Shafer Theory. *International Journal of Man-Machine Studies* **30** (1989) 525–536.
34. Williams, P. M.: Discussion of Shafer G. "Belief Functions and Parametric Models". *Journal of Royal Statistical Society* **B44** (1982) 342 et seq.

Contextual Discounting of Belief Functions

David Mercier^{1,2}, Benjamin Quost¹, and Thierry Denœux¹

¹ Université de Technologie de Compiègne, UMR CNRS 6599 Heudiasyc,
BP20529, F-60205 Compiègne Cedex, France
{dmercier, bquost, tdenoeux}@hds.utc.fr

² SOLYSTIC, 14 avenue Raspail,
F-94257 Gentilly Cedex, France

Abstract. The Transferable Belief Model is a general framework for managing imprecise and uncertain information using belief functions. In this framework, the *discounting* operation allows to combine information provided by a source (in the form of a belief function) with metaknowledge regarding the reliability of that source, to compute a “weakened”, less informative belief function. In this article, an extension of the discounting operation is proposed, allowing to make use of more detailed information regarding the reliability of the source in different contexts, a context being defined as a subset of the frame of discernment. Some properties of this *contextual discounting* operation are studied, and its relationship with classical discounted is explained.

1 Introduction

In the past years, the need to manipulate various forms of imperfect information and partial knowledge has led to study new uncertainty management frameworks. One of them, the *theory of evidence* [6] or *theory of belief functions*, has been declined into several approaches, among which the *Transferable Belief Model* (TBM) [8, 11]. This model, on which we will focus in this article, constitutes a powerful and flexible framework, well suited for information fusion [2, 5, 9].

In information fusion applications, it is usually important to take into account the reliability of the different sources in the evidence aggregation process. In the TBM, this is achieved by the discounting operation, which transforms each belief function provided by a source into a less informative one, based on a degree of confidence in the reliability of the source [6, 7]. In certain applications, however, it is possible to assess the reliability of the source in different contexts [1]. The *contextual discounting* operation presented in this paper extends the classical discounting so as to exploit such information.

This paper is organized as follows. Background material on the TBM will first be recalled in Section 2. Contextual discounting will then be introduced in Section 3, and an example will be analyzed in Section 4. Section 5 will conclude the paper.

2 The Transferable Belief Model

2.1 Basic Concepts

Let x be a variable taking values in a finite set $\Omega = \{\omega_1, \dots, \omega_K\}$, called the *frame of discernment* (or *frame*). The knowledge held by a rational agent Y , regarding the actual value ω_0 taken by x , given an evidential corpus EC , can be quantified by *basic belief assignment* (bba) $m_Y^\Omega[EC]$ defined as a function from 2^Ω to $[0, 1]$ verifying :

$$\sum_{A \subseteq \Omega} m_Y^\Omega[EC](A) = 1$$

When there is now ambiguity, the full notation $m_Y^\Omega[EC]$ will be simplified to m_Y^Ω , m^Ω , or even m . The vacuous bba, defined by $m(\Omega) = 1$, represents complete ignorance.

Two distinct pieces of evidence, quantified by bbas m_1 and m_2 , may be combined using the *conjunctive rule of combination* (CRC) or the *disjunctive rule of combination* (DRC), defined, respectively, as :

$$\begin{aligned} m_1 \odot m_2(A) &= \sum_{B \cap C = A} m_1(B) m_2(C), \\ m_1 \oplus m_2(A) &= \sum_{B \cup C = A} m_1(B) m_2(C), \quad \forall A \subseteq \Omega. \end{aligned}$$

The CRC applies when both sources are known to be reliable, whereas the DRC corresponds the hypothesis that at least one of the two sources is reliable [7].

2.2 Marginalization and Vacuous Extension

A bba defined on a product space $\Omega \times \Theta$ may be marginalized on Ω , by transferring each mass $m^{\Omega \times \Theta}(B)$ for $B \subseteq \Omega \times \Theta$ to its projection on Ω :

$$m^{\Omega \times \Theta \downarrow \Omega}(A) = \sum_{\{B \subseteq \Omega \times \Theta \mid \text{Proj}(B \downarrow \Omega) = A\}} m^{\Omega \times \Theta}(B), \quad \forall A \subseteq \Omega \quad (1)$$

where $\text{Proj}(B \downarrow \Omega)$ denotes the projection of B onto Ω .

It is usually not possible to retrieve the original bba $m^{\Omega \times \Theta}$ from its marginalization $m^{\Omega \times \Theta \downarrow \Omega}$ on Ω . However, the *least committed bba* [7] such that its projection on Ω is $m^{\Omega \times \Theta \downarrow \Omega}$ may be computed; this *vacuous extension* of a bba m^Ω on the product space $\Omega \times \Theta$ is given by:

$$m^{\Omega \uparrow \Omega \times \Theta}(B) = \begin{cases} m^\Omega(A) & \text{if } B = A \times \Theta, A \subseteq \Omega \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Marginalization and vacuous extension are both illustrated in Figure 1.

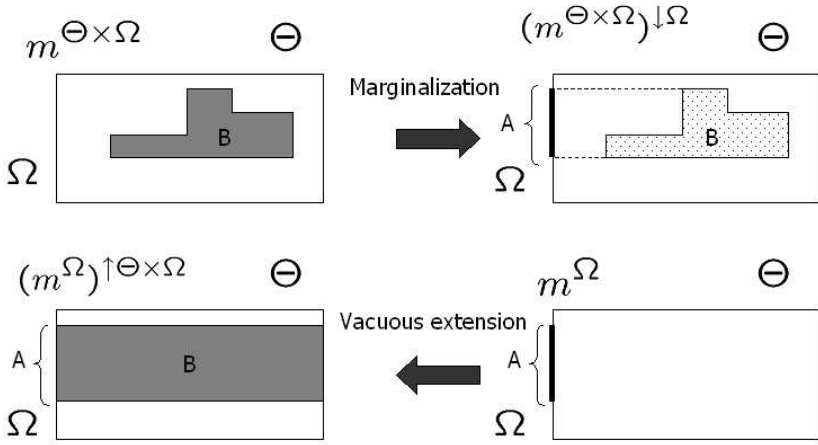


Fig. 1. Marginalization (above) and vacuous extension (below) of a bba in the case of a product space

2.3 Conditioning and Ballooning Extension

Conditional beliefs represent knowledge which is valid provided that an hypothesis is satisfied. Let m be a bba, $B \subseteq \Omega$ an hypothesis and m_B such as $m_B(B) = 1$; the conditional belief function $m[B]$ is:

$$m[B] = m \odot m_B. \quad (3)$$

If $m^{\Omega \times \Theta}$ is defined on the product space $\Omega \times \Theta$, and θ_0 is a subset of Θ , the conditional bba $m^{\Omega}[\theta_0]$ is defined by combining $m^{\Omega \times \Theta}$ with $m_{\theta_0}^{\Theta \uparrow \Omega \times \Theta}$, with $m_{\theta_0}^{\Theta}(\theta_0) = 1$, and marginalizing the result on Ω :

$$m^{\Omega}[\theta_0] = \left(m^{\Omega \times \Theta} \odot m_{\theta_0}^{\Theta \uparrow \Omega \times \Theta} \right)^{\downarrow \Omega} \quad (4)$$

Assume now that $m^{\Omega}[\theta_0]$ represents your beliefs on Ω conditionally on θ_0 , i.e., in a context where θ_0 holds. There are usually many bbas on $\Omega \times \Theta$, whose conditioning on θ_0 yields $m^{\Omega}[\theta_0]$. Among these, the least committed one is the ballooning extension defined by:

$$m^{\Omega}[\theta_0]^{\uparrow \Omega \times \Theta}(A \times \theta_0 \cup \Omega \times \overline{\theta_0}) = m^{\Omega}[\theta_0](A), \quad \forall A \subseteq \Omega. \quad (5)$$

Conditioning and ballooning extension are both presented in Figure 2.

2.4 Discounting

Let us assume that Y receives a bba m_S^{Ω} from a source S , describing the source's beliefs regarding the actual value ω_0 . Moreover, Y has some knowledge about the *reliability* of S , quantified by a bba $m_Y^{\mathcal{R}}$ on the space $\mathcal{R} = \{R, NR\}$, where

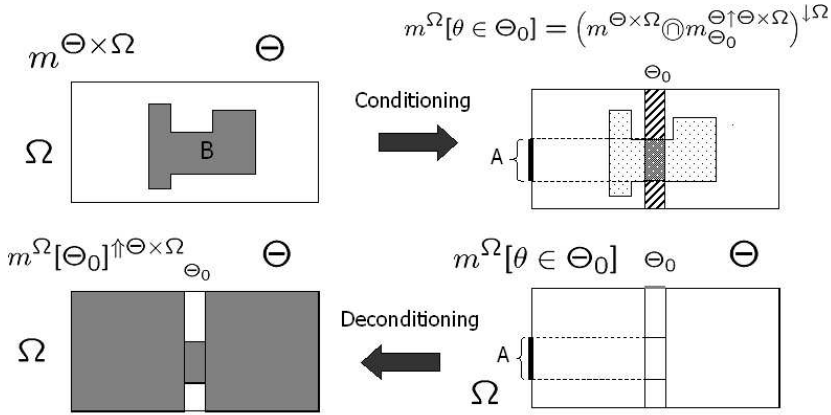


Fig. 2. Conditioning (above) and deconditioning (below) of a bba in the case of a product space

R stands for “the source is reliable”, and NR for “the source is not reliable” [7]. Let us assume that $m_Y^{\mathcal{R}}$ has the following form:

$$\begin{cases} m_Y^{\mathcal{R}}(\{R\}) &= 1 - \alpha \\ m_Y^{\mathcal{R}}(\{R, NR\}) &= \alpha, \end{cases} \quad (6)$$

for some $\alpha \in [0, 1]$.

If S is reliable, the information provided by S becomes Y ’s knowledge:

$$m_Y^{\Omega}[\{R\}] = m_S^{\Omega}. \quad (7)$$

If S is not reliable, the information provided by S cannot be taken into account, and Y ’s knowledge is vacuous:

$$m_Y^{\Omega}[\{NR\}](\Omega) = 1. \quad (8)$$

Therefore, we have two non-vacuous pieces of evidence, $m_Y^{\mathcal{R}}$ and $m_Y^{\Omega}[\{R\}]$. Assuming that they are distinct, they can be combined by vacuously extending $m_Y^{\mathcal{R}}$ to $\Omega \times \mathcal{R}$, computing the ballooning extension of $m_Y^{\Omega}[\{R\}]$ in the same space, applying the CRC, and marginalizing the result on Ω :

$$m_Y^{\Omega}[m_S^{\Omega}, m_Y^{\mathcal{R}}] = \left(m_Y^{\Omega}[\{R\}] \uparrow^{\Omega \times \mathcal{R}} \oslash m_Y^{\mathcal{R}} \uparrow^{\Omega \times \mathcal{R}} \right) \downarrow^{\Omega}. \quad (9)$$

The resulting bba $m_Y^{\Omega}[m_S^{\Omega}, m_Y^{\mathcal{R}}]$ (where the brackets $[]$ indicate the evidential corpus) only depends on m_S^{Ω} and α . Let us denote it by ${}^{\alpha}m_Y^{\Omega}$. It is equal to

$$\begin{cases} {}^{\alpha}m_Y^{\Omega}(A) = (1 - \alpha)m_S^{\Omega}(A), & \forall A \subset \Omega, \\ {}^{\alpha}m_Y^{\Omega}(\Omega) = (1 - \alpha)m_S^{\Omega}(\Omega) + \alpha. \end{cases} \quad (10)$$

This operation was called discounting by Shafer [6], who introduced it on intuitive grounds. The justification presented in this section was proposed by Smets [7].

Remark 1. If $m_Y^{\mathcal{R}}$ is Bayesian:

$$\begin{cases} m_Y^{\mathcal{R}}(\{R\}) &= 1 - \alpha, \\ m_Y^{\mathcal{R}}(\{NR\}) &= \alpha, \end{cases} \quad (11)$$

the result of the discounting is the same [7].

Remark 2. We can see ${}^\alpha m_Y^\Omega$ as the disjunctive combination of m_S^Ω with m_0^Ω defined by $m_0^\Omega(\emptyset) = 1 - \alpha$ and $m_0^\Omega(\Omega) = \alpha$.

Remark 3. Alternatively, ${}^\alpha m_Y^\Omega$ can be computed as

$${}^\alpha m_Y^\Omega(A) = \sum_{B \subseteq \Omega} G(A, B) m_S^\Omega(B) \quad (12)$$

with

$$G(A, B) = \begin{cases} 1 - \alpha & \text{if } A = B \neq \Omega, \\ \alpha & \text{if } A = \Omega \text{ and } B \subset A, \\ 1 & \text{if } A = B = \Omega \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

$G(A, B)$ is equal to the fraction on m_S^Ω transferred to A , for each $A \supseteq B$. The whole set of such coefficients define a generalization matrix [10].

3 Contextual Discounting

3.1 Basic Assumptions

Let us now assume that we have evidence regarding the reliability of S , conditionally on each $\omega_k \in \Omega$. We thus have K conditional bbas $m_Y^{\mathcal{R}}[\{\omega_k\}]$, $k = 1, \dots, K$, instead of the single unconditional bba in (6). Assume that they are defined as

$$\begin{cases} m_Y^{\mathcal{R}}[\{\omega_k\}](\{R\}) &= \beta_k, \\ m_Y^{\mathcal{R}}[\{\omega_k\}](\{R, NR\}) &= \alpha_k, \end{cases} \quad (14)$$

with $\beta_k = 1 - \alpha_k$.

Each of these bbas is conditional to a context ω_k : their combination with m_S^Ω will define a *contextual discounting* $m_Y^\Omega [m_S^\Omega, m_Y^{\mathcal{R}}[\{\omega_1\}], \dots, m_Y^{\mathcal{R}}[\{\omega_K\}]]$. As the classical discounting, characterized by a scalar α , is written ${}^\alpha m$, the contextual discounting is defined by a vector $(\alpha_1, \dots, \alpha_K)$, and it will be written ${}^{(\alpha)} m_Y^\Omega$.

3.2 Computation of $(\alpha)m_Y^\Omega$

Ballooning Extension and Combination of the $m_Y^\mathcal{R}[\{\omega_k\}]$. The ballooning extension of $m_Y^\mathcal{R}[\{\omega_k\}]$ is defined as:

$$m_Y^{\mathcal{R}\uparrow\Omega\times\mathcal{R}}(\{\omega_k\} \times \{R\} \cup \overline{\{\omega_k\}} \times \mathcal{R}) = \beta_k, \quad (15)$$

$$m_Y^{\mathcal{R}\uparrow\Omega\times\mathcal{R}}(\Omega \times \mathcal{R}) = \alpha_k. \quad (16)$$

Let $m_r^{\Omega\times\mathcal{R}}$ be the conjunctive combination of the $m_Y^\mathcal{R}[\{\omega_k\}]^{\uparrow\Omega\times\mathcal{R}}$. Using the following equality, for any $k \neq l$:

$$(\{\omega_k\} \times \{R\} \cup \overline{\{\omega_k\}} \times \mathcal{R}) \cap (\{\omega_l\} \times \{R\} \cup \overline{\{\omega_l\}} \times \mathcal{R}) = \{\omega_k, \omega_l\} \times \{R\} \cup \overline{\{\omega_k, \omega_l\}} \times \mathcal{R},$$

we easily obtain the expression of $m_r^{\Omega\times\mathcal{R}}$ as:

$$m_r^{\Omega\times\mathcal{R}}(\overline{C} \times \{R\} \cup C \times \mathcal{R}) = \begin{cases} \prod_{\omega_i \in C} \alpha_i \prod_{\omega_j \in \overline{C}} \beta_j & \text{if } C \neq \emptyset \text{ and } C \neq \Omega, \\ \prod_{\omega_i \in \Omega} \alpha_i & \text{if } C = \Omega, \\ \prod_{\omega_j \in \Omega} \beta_j & \text{if } C = \emptyset. \end{cases} \quad (17)$$

In the following, we simply note:

$$m_r^{\Omega\times\mathcal{R}}(\overline{C} \times R \cup C \times \mathcal{R}) = \prod_{\omega_i \in C} \alpha_i \prod_{\omega_j \in \overline{C}} \beta_j \quad (18)$$

with the convention that a product of terms vanishes when the index set is empty.

It can be checked that the initial conditional bbas are retrieved by conditioning $m_r^{\Omega\times\mathcal{R}}$ on each ω_k :

$$m_r^{\Omega\times\mathcal{R}}[\{\omega_k\}] = \beta_k = m_Y^\mathcal{R}[\{\omega_k\}], \quad k = 1, \dots, K. \quad (19)$$

Combination with m_S^Ω . The contextual discounting can be obtained from the bbas $m_Y^\mathcal{R}[\{R\}]^{\uparrow\Omega\times\mathcal{R}}$ and $m_r^{\Omega\times\mathcal{R}}$:

$$(\alpha)m_Y^\Omega = (m_Y^\mathcal{R}[\{R\}]^{\uparrow\Omega\times\mathcal{R}} \odot m_r^{\Omega\times\mathcal{R}})^{\downarrow\Omega} \quad (20)$$

The bbas $m_Y^\mathcal{R}[\{R\}]^{\uparrow\Omega\times\mathcal{R}}$ and $m_r^{\Omega\times\mathcal{R}}$ have focal sets of the form $B \times \{R\} \cup \Omega \times \{NR\}$ and $\overline{C} \times \{R\} \cup C \times \mathcal{R}$, respectively, with $B, C \subseteq \Omega$. The intersection of two such focal sets is:

$$(\overline{C} \times \{R\} \cup C \times \mathcal{R}) \cap (B \times \{R\} \cup \Omega \times \{NR\}) = B \times \{R\} \cup C \times \{NR\},$$

and it can be obtained only for a particular choice of B and C . Then:

$$m_Y^{\uparrow\Omega\times\mathcal{R}} \odot m_r^{\Omega\times\mathcal{R}}(B \times \{R\} \cup C \times \{NR\}) = \left[\prod_{\omega_i \in C} \alpha_i \prod_{\omega_j \in \overline{C}} \beta_j \right] m_S^\Omega(B). \quad (21)$$

Marginalizing this bba on Ω gives:

$$^{(\alpha)}m^\Omega(A) = \sum_{B \cup C = A} \left[\prod_{\omega_i \in C} \alpha_i \prod_{\omega_j \in \overline{C}} \beta_j \right] m_S^\Omega(B), \quad \forall A \subseteq \Omega, \quad (22)$$

which can also be written as:

$$^{(\alpha)}m^\Omega(A) = \sum_{B \subseteq A} G(A, B) m_S^\Omega(B), \quad \forall A \subseteq \Omega, \quad (23)$$

with:

$$G(A, B) = \sum_{C: B \cup C = A} \prod_{\omega_i \in C} \alpha_i \prod_{\omega_j \in \overline{C}} \beta_j, \quad \forall B \subseteq A \subseteq \Omega. \quad (24)$$

Coefficients $G(A, B)$ for all $A, B \subseteq \Omega$ define a generalization matrix [10]: $G(A, B)$ is equal to the fraction of $m_S^\Omega(B)$ transferred to $^{(\alpha)}m^\Omega(A)$, for $A \supseteq B$.

Proposition 1. *A simpler form of the generalization matrix in (24) is*

$$G(A, B) = \prod_{\omega_i \in A \setminus B} \alpha_i \prod_{\omega_j \in \overline{A}} \beta_j, \quad \forall B \subseteq A \subseteq \Omega, \quad (25)$$

Proof: We have

$$B \cup C = A \Leftrightarrow \exists D \subseteq B : C = A \setminus B \cup D \Leftrightarrow \exists D \subseteq B : \overline{C} = \overline{A} \cup B \setminus D,$$

and therefore:

$$\begin{aligned} G(A, B) &= \sum_{D \subseteq B} \prod_{\omega_i \in A \setminus B \cup D} \alpha_i \prod_{\omega_j \in \overline{A} \cup B \setminus D} \beta_j \\ &= \prod_{\omega_i \in A \setminus B} \alpha_i \prod_{\omega_j \in \overline{A}} \beta_j \underbrace{\sum_{D \subseteq B} \prod_{\omega_i \in B \setminus D} \beta_i \prod_{\omega_j \in D} \alpha_j}_{=1}. \end{aligned}$$

Remark 4. It can be seen from Equation (22) that $^{(\alpha)}m^\Omega$ is the disjunctive combination of m_S^Ω with a bba m_0^Ω defined by $m_0^\Omega(C) = \prod_{\omega_i \in C} \alpha_i \prod_{\omega_j \in \overline{C}} \beta_j$, for all $C \subseteq \Omega$.

Remark 5. Contextual discounting as defined in this section does *not* generalize the classical discounting recalled in Section 2.4. In particular, the solution obtained by discounting m_S^Ω with rates $\alpha_i = \alpha$, $i = 1, \dots, K$ is different, in general, from the one obtained using the classical discounting operation with a single rate α . Both classical and contextual discounting appear in fact to be two instances of a more general concept, which is introduced in the next section.

3.3 Θ -Contextual Discounting

Contextual discounting defined above may be generalized by assuming that the available evidence allows to assess the reliability of S in more general contexts $\theta_l \subseteq \Omega$, $l = 1, \dots, L$, where $\theta_1, \dots, \theta_L$ form a partition of Ω . The set $\Theta = \{\theta_1, \dots, \theta_L\}$ then constitutes a coarsening of Ω .

In such a case, information regarding the reliability of the source takes the form of L conditional bbas

$$\begin{cases} m_Y^{\mathcal{R}}[\theta_l](\{R\}) &= \beta_l, \\ m_Y^{\mathcal{R}}[\theta_l](\{R, NR\}) &= \alpha_l, \quad l = 1, \dots, L. \end{cases} \quad (26)$$

A similar line of reasoning as performed in Section 3.2 yields

$$m_r^{\Omega \times \mathcal{R}}(\overline{C} \times R \cup C \times \mathcal{R}) = \prod_{\theta_i: \cup_i \theta_i = C} \alpha_i \prod_{\theta_j: \cup_j \theta_j = \overline{C}} \beta_j, \quad (27)$$

which is the equivalent of (18) in the previous case, but where C now ranges in the set $\mathcal{C}$ of subsets of Ω which are the union of some θ_i 's:

$$C \in \mathcal{C} = \{A \subseteq \Omega \mid \exists I \subseteq \{1, \dots, L\}, A = \bigcup_{i \in I} \theta_i\}.$$

After marginalizing on Ω , we finally obtain:

$$\begin{aligned} {}^{(\alpha)}m^{\Omega}(A) &= \sum_{B \cup C = A} m_r^{\Omega \times \mathcal{R}}(\overline{C} \times R \cup C \times \mathcal{R}) m_S^{\Omega}(B), \quad \forall A \subseteq \Omega \\ &= \sum_{B \cup C = A} \left[\prod_{\theta_i: \cup_i \theta_i = C} \alpha_i \prod_{\theta_j: \cup_j \theta_j = \overline{C}} \beta_j \right] m_S^{\Omega}(B), \quad \forall A \subseteq \Omega \quad (28) \\ &= \sum_{B \subseteq A} G(A, B) m_S^{\Omega}(B), \quad \forall A \subseteq \Omega, \end{aligned}$$

where $G(A, B)$ denote again the coefficients of the generalization matrix associated with the contextual discounting:

$$G(A, B) = \sum_{B \cup C = A} \prod_{\theta_i: \cup_i \theta_i = C} \alpha_i \prod_{\theta_j: \cup_j \theta_j = \overline{C}} \beta_j, \quad \forall B \subseteq A \subseteq \Omega. \quad (29)$$

The operation defined by Equation (28) will be called Θ -contextual discounting, with discount rates $\alpha_1, \dots, \alpha_L$. The contextual discounting defined in Section 3.2 corresponds to the special case where $\theta_i = \{\omega_i\}$, $i = 1, \dots, L$. It will be called Ω -contextual discounting for short.

Remark 6. As before, it can be seen from (28) that ${}^{(\alpha)}m^{\Omega}$ is the disjunctive combination of m_S^{Ω} with a bba m_0^{Ω} defined by $m_0^{\Omega}(C) = \prod_{\theta_i: \cup_i \theta_i = C} \alpha_i \prod_{\theta_j: \cup_j \theta_j = \overline{C}} \beta_j$ if $C \in \mathcal{C}$, and $m_0^{\Omega}(C) = 0$ otherwise.

Remark 7. Assume that Θ is composed of a single element $\theta = \Omega$. Then, from Remark 6, $^{(\alpha)}m^\Omega$ is the disjunctive combination of m_S^Ω with a bba m_0^Ω defined by $m_0^\Omega(\emptyset) = 1 - \alpha$ and $m_0^\Omega(\Omega) = \alpha$. Hence, from Remark 2, $^{(\alpha)}m^\Omega$ is equal to the classical discounting of m_S^Ω : classical discounting is thus Θ -contextual discounting with $\Theta = \{\Omega\}$.

Remark 8. It can be shown that the same results are obtained if knowledge about the reliability of S is expressed as

$$\begin{cases} m_Y^{\mathcal{R}}[\theta_k](\{R\}) = \beta_k, \\ m_Y^{\mathcal{R}}[\theta_k](\{NR\}) = \alpha_k. \end{cases} \quad (30)$$

4 Examples

Example 1. Let $\Omega = \{\omega_1, \omega_2, \omega_3\}$, un m a bba on Ω . The Ω -contextual discounting of m with rates $(\alpha) = (\alpha_1, \alpha_2, \alpha_3)$ yields

$$\begin{aligned} ^{(\alpha)}m(\emptyset) &= \beta_1\beta_2\beta_3 m(\emptyset) \\ ^{(\alpha)}m(\{\omega_1\}) &= \beta_2\beta_3 [m(\omega_1) + \alpha_1 m(\emptyset)] \\ ^{(\alpha)}m(\{\omega_2\}) &= \beta_1\beta_3 [m(\omega_2) + \alpha_2 m(\emptyset)] \\ ^{(\alpha)}m(\{\omega_3\}) &= \beta_1\beta_2 [m(\omega_3) + \alpha_3 m(\emptyset)] \\ ^{(\alpha)}m(\{\omega_1, \omega_2\}) &= \beta_3 [m(\{\omega_1, \omega_2\}) + \alpha_1 m(\{\omega_2\}) + \alpha_2 m(\{\omega_1\}) + \alpha_1\alpha_2 m(\emptyset)] \\ ^{(\alpha)}m(\{\omega_1, \omega_3\}) &= \beta_2 [m(\{\omega_1, \omega_3\}) + \alpha_1 m(\{\omega_3\}) + \alpha_3 m(\{\omega_1\}) + \alpha_1\alpha_3 m(\emptyset)] \\ ^{(\alpha)}m(\{\omega_2, \omega_3\}) &= \beta_1 [m(\{\omega_2, \omega_3\}) + \alpha_2 m(\{\omega_3\}) + \alpha_3 m(\{\omega_2\}) + \alpha_2\alpha_3 m(\emptyset)] \\ ^{(\alpha)}m(\Omega) &= m(\Omega) + \alpha_1 m(\{\omega_2, \omega_3\}) + \alpha_2 m(\{\omega_1, \omega_3\}) + \alpha_3 m(\{\omega_1, \omega_2\}) \\ &\quad + \alpha_1\alpha_2 m(\{\omega_3\}) + \alpha_2\alpha_3 m(\{\omega_1\}) + \alpha_1\alpha_3 m(\{\omega_2\}) + \\ &\quad \alpha_1\alpha_2\alpha_3 m(\emptyset). \end{aligned}$$

The corresponding generalization matrix is show in Table 1.

Table 1. Generalization matrix associated to the Ω -contextual discounting of m

	$\emptyset$	$\{\omega_1\}$	$\{\omega_2\}$	$\{\omega_1, \omega_2\}$	$\{\omega_3\}$	$\{\omega_1, \omega_3\}$	$\{\omega_2, \omega_3\}$	$\{\omega_1, \omega_2, \omega_3\}$
$\emptyset$	$\beta_1\beta_2\beta_3$							
$\{\omega_1\}$	$\alpha_1\beta_2\beta_3$	$\beta_2\beta_3$						
$\{\omega_2\}$	$\beta_1\alpha_2\beta_3$		$\beta_1\beta_3$					
$\{\omega_1, \omega_2\}$	$\alpha_1\alpha_2\beta_3$	$\alpha_2\beta_3$	$\alpha_1\beta_3$	β_3				
$\{\omega_3\}$	$\beta_1\beta_2\alpha_3$				$\beta_1\beta_2$			
$\{\omega_1, \omega_3\}$	$\alpha_1\beta_2\alpha_3$	$\beta_2\alpha_3$			$\alpha_1\beta_2$	β_2		
$\{\omega_2, \omega_3\}$	$\beta_1\alpha_2\alpha_3$		$\beta_1\alpha_3$		$\beta_1\alpha_2$		β_1	
$\{\omega_1, \omega_2, \omega_3\}$	$\alpha_1\alpha_2\alpha_3$	$\alpha_2\alpha_3$	$\alpha_1\alpha_3$	α_3	$\alpha_1\alpha_2$	α_2	α_1	1

With $\alpha_1 = 1$, $\alpha_2 = \alpha_3 = 0$, we obtain:

$$\begin{aligned} {}^{(\alpha)}m(\emptyset) &= {}^{(\alpha)}m(\{\omega_2\}) = {}^{(\alpha)}m(\{\omega_3\}) = {}^{(\alpha)}m(\{\omega_2, \omega_3\}) = 0 \\ {}^{(\alpha)}m(\{\omega_1\}) &= m(\omega_1) + m(\emptyset) \\ {}^{(\alpha)}m(\{\omega_1, \omega_2\}) &= m(\{\omega_1, \omega_2\}) + m(\{\omega_2\}) \\ {}^{(\alpha)}m(\{\omega_1, \omega_3\}) &= m(\{\omega_1, \omega_3\}) + m(\{\omega_3\}) \\ {}^{(\alpha)}m(\{\omega_1, \omega_2, \omega_3\}) &= m(\{\omega_1, \omega_2, \omega_3\}) + m(\{\omega_2, \omega_3\}). \end{aligned}$$

The belief given to $\{\omega_1\}$ is unchanged (the others elements are perfectly recognized). The source being reliable when identifying $\{\omega_2\}$ and $\{\omega_3\}$, the belief given to each element A containing those latter is transferred on $A \cup \{\omega_1\}$: the ability of the source to recognize this element is indeed unknown.

Example 2. Consider now the Θ -contextual discounting of m from the previous example, for $\Theta = \{\theta_1, \theta_2\}$ with $\theta_1 = \{\omega_1\}$, $\theta_2 = \{\omega_2, \omega_3\}$, associated with α_1 and α_2 respectively. The generalization matrix is shown in Table 2.

Table 2. Generalization matrix associated to the Θ -contextual discounting of m , with $\Theta = \{\{\omega_1\}, \{\omega_2, \omega_3\}\}$

	$\emptyset$	$\{\omega_1\}$	$\{\omega_2\}$	$\{\omega_1, \omega_2\}$	$\{\omega_3\}$	$\{\omega_1, \omega_3\}$	$\{\omega_2, \omega_3\}$	$\{\omega_1, \omega_2, \omega_3\}$
$\emptyset$	$\beta_1\beta_2$							
$\{\omega_1\}$	$\alpha_1\beta_2$	β_2						
$\{\omega_2\}$			$\beta_1\beta_2$					
$\{\omega_1, \omega_2\}$			$\alpha_1\beta_2$	β_2				
$\{\omega_3\}$					$\beta_1\beta_2$			
$\{\omega_1, \omega_3\}$					$\alpha_1\beta_2$	β_2		
$\{\omega_2, \omega_3\}$	$\beta_1\alpha_2$		$\beta_1\alpha_2$		$\beta_1\alpha_2$		β_1	
$\{\omega_1, \omega_2, \omega_3\}$	$\alpha_1\alpha_2$	α_2	$\alpha_1\alpha_2$	α_2	$\alpha_1\alpha_2$	α_2	α_1	1

Remark that, with $\alpha_1 = 1$, $\alpha_2 = 0$, the result is the same as the one obtained previously, which is not true in the general case.

5 Conclusion

We defined in this article a contextual discounting. This concept allows to model accurately the reliability of a source; it is shown to generalize the classical discounting introduced by Shafer [6]. It seems to provide an adequate tool to tackle, e.g., sensor fusion applications, in which the reliability of sensors depends on the context.

It seems interesting to learn the reliability of the source from a training set, instead of having it assessed by an expert. In the case of classical discounting, an

approach has already been proposed in [4], where the discounting coefficients α for each source are computed such that they minimize a measure of discrepancy between observations and sensor outputs. In the case of the contextual discounting, both the partition Θ of Ω and the set of coefficients have to be determined. This is left for future research.

References

1. Appriou, A.: Uncertain data aggregation in classification and tracking. In B. Bouchon-Meunier, editor, *Aggregation and Fusion of imperfect information* (1998) 231-260.
2. Bloch, I.: *Fusion d'informations en traitement du signal et des images*. Hermès (2003).
3. Dubois, D., Prade, H.: *Possibility Theory*. Plenum Press, New-York (1988).
4. Elouedi, Z., Mellouli, K., Smets, Ph.: Assessing sensor reliability for multisensor data fusion with the transferable belief model. *IEEE Transactions on Systems, Man and Cybernetics* **B34** (2004) 782-787.
5. Milisavljevic, N., Bloch, I., van den Broek, S., Acheroy, M.: Improving mine recognition through processing and Dempster-Shafer fusion of ground-penetrating radar data. *Pattern Recognition* **36** (2003) 1233-1250.
6. Shafer, G.: *A mathematical theory of evidence*. Princeton University Press (1976).
7. Smets, Ph.: Belief functions: the disjunctive rule of combination and the generalized bayesian theorem. *International Journal of Approximate Reasoning* **9** (1993) 1-35.
8. Smets, Ph.: The transferable belief model for quantified belief representation. *Handbook of Defeasible Reasoning and Uncertainty Management Systems* **1** (1998) 267-301.
9. Smets, Ph.: Data fusion in the transferable belief model. *Proceedings of the 3rd International Conference on Information Fusion, FUSION 2000* (2000) 21-33.
10. Smets, Ph.: The application of the matrix calculus to belief functions. *International Journal of Approximate Reasoning* **31** (2002) 1-30.
11. Smets, Ph., Kennes, R.: The transferable belief model. *Artificial Intelligence* **66** (1994) 191-234.

Bilattice-Based Squares and Triangles

Ofer Arieli¹, Chris Cornelis², Glad Deschrijver², and Etienne Kerre²

¹ Department of Computer Science, The Academic College of Tel-Aviv, Israel

`oarieli@mta.ac.il`

² Fuzziness and Uncertainty Modelling Research Unit,

Department of Mathematics and Computer Science, Ghent University, Belgium

`{chris.cornelis, glad.deschrijver, etienne.kerre}@UGent.be`

Abstract. In this paper, Ginsberg's/Fitting's theory of bilattices is invoked as a natural accommodation and powerful generalization to both intuitionistic fuzzy sets (IFSs) and interval-valued fuzzy sets (IVFSs), serving on one hand to clarify the exact nature of the relationship between these two common extensions of fuzzy sets, and on the other hand providing a general and intuitively attractive framework for the representation of uncertain and potentially conflicting information.

1 Introduction

Bilattices are algebraic structures that were introduced by Ginsberg [19] as a general and uniform framework for a diversity of applications in artificial intelligence. In a series of papers it was then shown that these structures may serve as a foundation of many areas, such as logic programming [15], computational linguistics [23], distributed knowledge processing [22], and reasoning with imprecise information [1, 2, 18]. The usefulness of bilattices in the context of fuzzy set theory was recently made explicit in [3], where we demonstrated that so-called bilattice-based 'squares' and 'triangles' provide an elegant framework for bridging between intuitionistic fuzzy sets (IFSs) and interval-valued fuzzy sets (IVFSs), thus also shedding a clear light on the syntactical equivalence of these two commonly encountered extensions of Zadeh's fuzzy sets.

The present work is an elaboration on the latter observation. Starting from a complete lattice, we study the corresponding bilattice-based squares and triangles, compare and relate them to various extensions of IFSs and/or IVFSs that have been proposed in the literature, and equip them with suitable logical connectives. In this sense, this paper can also be viewed as a generalization of other papers [10, 11, 12] that refer to particular forms of 'triangle' and 'square', in which the underlying structure is the unit interval.

2 Preliminaries

2.1 Intuitionistic Fuzzy Sets and Interval-Valued Fuzzy Sets

Intuitionistic fuzzy set (IFS) theory [4] is an extension of fuzzy set theory in which any element u in a universe U is assigned not only a membership degree,

$\mu_A(u)$, but also a non-membership degree $\nu_A(u)$, where both degrees are drawn from the unit interval $[0, 1]$. While in Zadeh's fuzzy set theory [27] always $\nu_A(u) = 1 - \mu_A(u)$, in IFS theory a weaker constraint is imposed: $\nu_A(u) \leq 1 - \mu_A(u)$.¹ IFSs can also be regarded as a particular kind of Goguen's $\mathcal{L}$ -fuzzy sets [20], i.e., as mappings from a universe U into the complete lattice $\mathcal{L}^*$, defined as follows:

Definition 1. [13] $\mathcal{L}^* = (L^*, \leq_{L^*})$, where $L^* = \{(x_1, x_2) \in [0, 1]^2 \mid x_1 + x_2 \leq 1\}$ and $(x_1, x_2) \leq_{L^*} (y_1, y_2)$ iff $x_1 \leq y_1$ and $x_2 \geq y_2$.

Interval-valued fuzzy set (IVFS) theory is an alternative method of extending fuzzy set theory, motivated by the need to replace crisp, $[0, 1]$ -valued membership degrees by intervals in $[0, 1]$ that approximate the (unknown) membership degrees. Interval-valued fuzzy sets are also $\mathcal{L}$ -fuzzy sets, for which the corresponding lattice is $\mathcal{L}^I$, defined as follows:

Definition 2. [11] $\mathcal{L}^I = (L^I, \leq_{L^I})$, where $L^I = \{[x_1, x_2] \mid (x_1, x_2) \in [0, 1]^2, x_1 \leq x_2\}$ and $[x_1, x_2] \leq_{L^I} [y_1, y_2]$ iff $x_1 \leq y_1$ and $x_2 \leq y_2$.

2.2 Bilattices

As noted above, bilattices are the mathematical structures used here for relating IFSs and IVFSs. We first review some basic definitions that pertain to bilattices.

Definition 3. [16] A *pre-bilattice* is a structure $\mathcal{B} = (B, \leq_t, \leq_k)$, such that B is a set containing at least two elements, and $(B, \leq_t)$, $(B, \leq_k)$ are complete lattices.

Definition 4. Let $\mathcal{B} = (B, \leq_t, \leq_k)$ be a pre-bilattice.

A *negation* of $\mathcal{B}$ is a unary operation $\neg$ on B satisfying the following properties:

- (1) $\neg\neg x = x$ (2) if $x \leq_t y$ then $\neg x \geq_t \neg y$, (3) if $x \leq_k y$ then $\neg x \leq_k \neg y$.

A *conflation* of $\mathcal{B}$ is a unary operation $-$ on B satisfying the following properties:

- (1) $--x = x$ (2) if $x \leq_k y$ then $-x \geq_k -y$, (3) if $x \leq_t y$ then $-x \leq_t -y$.

Definition 5. [19] A *bilattice* is a structure $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$, such that $(B, \leq_t, \leq_k)$ is a pre-bilattice with a negation $\neg$.

In the sequel, following the usual notations for the basic bilattice operations, we shall denote by $\wedge$ (respectively, by $\vee$) the $\leq_t$ -meet (the $\leq_t$ -join) and by $\otimes$ (respectively, by $\oplus$) the $\leq_k$ -meet (the $\leq_k$ -join) of a bilattice $\mathcal{B}$. f and t denote the $\leq_t$ -extreme elements, and $\perp$, $\top$ denote the $\leq_k$ -extreme elements. Intuitively, these elements can be perceived as 'false', 'true', 'unknown' (i.e., neither true nor false) and 'contradictory' (both true and false), respectively. The two partial orders $\leq_t$ and $\leq_k$ are taken to represent differences in the degree of truth and in the amount of information (respectively), conveyed by a given assertion.

¹ The 'intuitionistic' characterizations of this approach should be understood here in a 'broad' sense, as it loosely denies the law of excluded middle. This approach bears no relationship to the conservative extension of intuitionistic logic [24].

Proposition 1. Let $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ be a bilattice.

a) [19] For every x, y in B :

$$\neg(x \wedge y) = \neg x \vee \neg y, \neg(x \vee y) = \neg x \wedge \neg y, \neg(x \otimes y) = \neg x \otimes \neg y, \neg(x \oplus y) = \neg x \oplus \neg y.$$

$$\neg f = t, \neg t = f, \neg \perp = \perp, \neg \top = \top.$$

b) [16] If $\mathcal{B}$ has conflation $-$, then, for every x, y in B :

$$-(x \wedge y) = -x \wedge -y, -(x \vee y) = -x \vee -y, -(x \otimes y) = -x \oplus -y, -(x \oplus y) = -x \otimes -y.$$

$$-f = f, -t = t, -\perp = \top, -\top = \perp.$$

Definition 6. A bilattice $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ is *distributive* [19] if all the (twelve) possible distributive laws concerning $\wedge, \vee, \otimes$, and $\oplus$ hold.

Following Fitting [14], we consider a special kind of distributive bilattices.

Definition 7. A distributive bilattice $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ with a conflation $-$ that commutes with $\neg$ is called *classical*, if $x \vee -\neg x = t$ for every x in B .

3 Relating IFSs and IVFSs Through Bilattices

In this section, we introduce a general context featuring a number of bilattice-based structures to relate and generalize the IFS/IVFS constructs $\mathcal{L}^*$ and $\mathcal{L}^I$, as well as some of their extensions.

Definition 8. [19] Let $\mathcal{L} = (L, \leq_L)$ be a complete lattice. A (bilattice-based) *square* is a structure $\mathcal{L}^2 = (L \times L, \leq_t, \leq_k, \neg)^2$ where $\neg(x_1, x_2) = (x_2, x_1)$, and

- (1) $(x_1, x_2) \leq_t (y_1, y_2) \Leftrightarrow x_1 \leq_L y_1 \text{ and } x_2 \geq_L y_2$,
- (2) $(x_1, x_2) \leq_k (y_1, y_2) \Leftrightarrow x_1 \leq_L y_1 \text{ and } x_2 \leq_L y_2$.

An element (x_1, x_2) of a square $\mathcal{L}^2$ may intuitively be understood such that x_1 represents the amount of belief *for* some assertion, and x_2 is the amount of belief *against* it. This corresponds to Atanassov's idea [4] of distinguishing between a membership component $\mu_A(u)$ and a non-membership component $\nu_A(u)$, with the amendment that in the case of a square no restriction like $\mu_A(u) + \nu_A(u) \leq 1$ for every u in U is imposed. Note also that the $\leq_t$ -ordering of $\mathcal{L}^2$ is completely in line with the partial order of $\mathcal{L}^*$; the $\leq_k$ -ordering additionally discriminates couples in L^2 according to the amount of information they carry³.

Denoting the join and meet operations of the complete lattice $\mathcal{L}$ by $\sqcap$ and $\sqcup$, respectively, we have, for $(x_1, x_2), (y_1, y_2)$ in L^2 ,

$$(x_1, x_2) \wedge (y_1, y_2) = (x_1 \sqcap y_1, x_2 \sqcup y_2), (x_1, x_2) \vee (y_1, y_2) = (x_1 \sqcup y_1, x_2 \sqcap y_2)$$

$$(x_1, x_2) \otimes (y_1, y_2) = (x_1 \sqcap y_1, x_2 \sqcap y_2), (x_1, x_2) \oplus (y_1, y_2) = (x_1 \sqcup y_1, x_2 \sqcup y_2)$$

² Incidentally, Ginsberg considered slightly more general structures defined on the cartesian product of two not necessarily equal complete lattices.

³ Note also that the $\leq_k$ -order of a square appears to correspond to the partial order of $\mathcal{L}^I$.

Moreover, denoting $0_{\mathcal{L}} = \inf L$ and $1_{\mathcal{L}} = \sup L$, it holds that $\perp_{\mathcal{L}^2} = (0_{\mathcal{L}}, 0_{\mathcal{L}})$, $\top_{\mathcal{L}^2} = (1_{\mathcal{L}}, 1_{\mathcal{L}})$, $t_{\mathcal{L}^2} = (1_{\mathcal{L}}, 0_{\mathcal{L}})$, and $f_{\mathcal{L}^2} = (0_{\mathcal{L}}, 1_{\mathcal{L}})$. When $\mathcal{N}$ is an involution of $\mathcal{L}$, then $-\mathcal{N}(x_1, x_2) = (\mathcal{N}(x_2), \mathcal{N}(x_1))$ is a conflation of $\mathcal{L}^2$.

It is easy to verify that every square $\mathcal{L}^2$ is distributive when $\mathcal{L}$ is distributive.

Example 1. Let $\mathcal{L}_2 = (\{0, 1\}, \leq)$ and $\mathcal{L}_3 = (\{0, \frac{1}{2}, 1\}, \leq)$, with $\leq$ in each case the usual ordering. The bilattices $\mathcal{L}_2^2$ and $\mathcal{L}_3^2$ are shown in Figure 1. In the literature, these structures are commonly referred to as *FOUR* (after Belnap's [6, 7] original four-valued logic) and *NINE* (see, e.g., [1, 2]), respectively. Both these bilattices are distributive, and *FOUR* is also classical, while *NINE* is not. An example of a square with an infinite amount of elements is, for instance, $([0, 1], \leq)^2$.

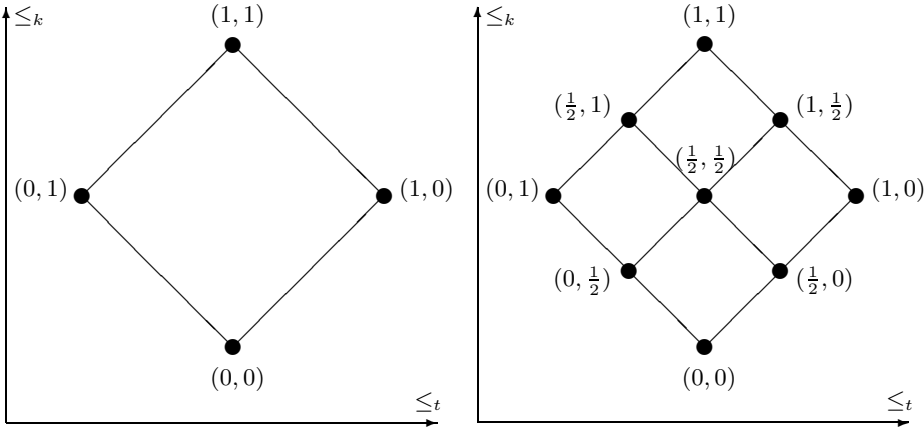


Fig. 1. The bilattices $\mathcal{L}_2^2$ and $\mathcal{L}_3^2$

The second bilattice-based structure investigated here is due to Fitting [16].

Definition 9. Let $\mathcal{L} = (L, \leq_L)$ be a complete lattice, and $I(L) = \{[x_1, x_2] \mid (x_1, x_2) \in L^2, x_1 \leq_L x_2\}$. A (bilattice-based) *triangle* is a structure $\mathcal{I}(\mathcal{L}) = (I(L), \leq_t, \leq_k)$, where

- (1) $[x_1, x_2] \leq_t [y_1, y_2] \Leftrightarrow x_1 \leq_L y_1 \text{ and } x_2 \leq_L y_2$,
- (2) $[x_1, x_2] \leq_k [y_1, y_2] \Leftrightarrow x_1 \leq_L y_1 \text{ and } x_2 \geq_L y_2$.

Note that a triangle $\mathcal{I}(\mathcal{L})$ is in fact not a (pre-)bilattice, since the substructure $(I(L), \leq_k)$ is not a lattice (the supremum of any two elements does not necessarily exist). Still, triangles are very much in the same spirit as bilattices, since the $\leq_k$ -ordering also represents differences in the amount of information that each interval exhibits.

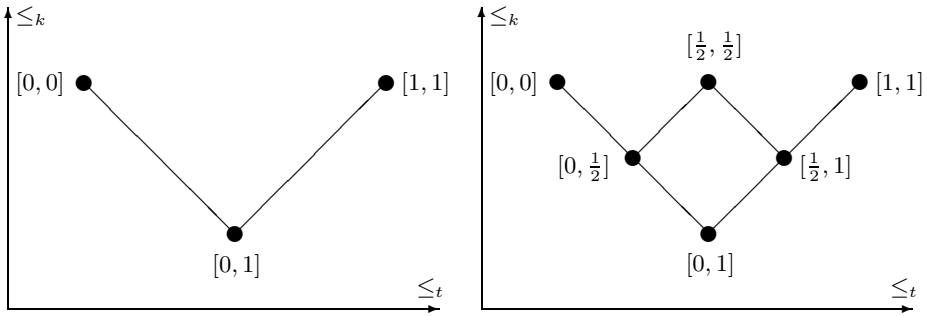


Fig. 2. The triangles $\mathcal{I}(\mathcal{L}_2)$ and $\mathcal{I}(\mathcal{L}_3)$

Example 2. The triangles $\mathcal{I}(\mathcal{L}_2)$ and $\mathcal{I}(\mathcal{L}_3)$ are shown in Figure 2. When $\mathcal{L}$ is the unit interval with the usual ordering, $\mathcal{I}(\mathcal{L})$ is a structure that extends the lattice $\mathcal{L}^I$ in the sense that $\mathcal{L}^I$ is exactly $(I([0, 1]), \leq_t)$. Moreover, $\mathcal{I}(\mathcal{L})$ also contains the partially ordered set $(I([0, 1]), \leq_k)$ that orders intervals according to their exactness.

Definition 10. [16] Let $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ be a bilattice with a conflation $\neg$. An element x in B is called *exact* with respect to this conflation if $x = \neg x$; it is *consistent* if $x \leq_k \neg x$.

Intuitively, exact elements exhibit precise information, while the consistent ones endorse non-contradictory evidence about their assertions.

Definition 11. Let $\neg$ be a conflation of a bilattice $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$. Denote by $\mathcal{C}^-(\mathcal{B})$ the substructure of $\mathcal{B}$ with the consistent elements (w.r.t. $\neg$) of B .

The following proposition relates squares and triangles:

Proposition 2. [16] Suppose that $\mathcal{L}$ is a complete lattice with an involution $\mathcal{N}$. Then $\mathcal{I}(\mathcal{L})$ is isomorphic to $\mathcal{C}^{-\mathcal{N}}(\mathcal{L}^2)$.

The isomorphism f between $\mathcal{I}(\mathcal{L})$ and $\mathcal{C}^{-\mathcal{N}}(\mathcal{L}^2)$ for Proposition 2 is given by $f([x_1, x_2]) = (x_1, \mathcal{N}(x_2))$. If L is the unit interval, $f([x_1, x_2]) = (x_1, 1 - x_2)$, which is the transformation considered in [10] for switching between IVFSs and IFSs. The above result shows that the same transformation is useful not only for relating $\mathcal{L}^I$ and $\mathcal{L}^*$ (i.e., when the underlying lattice is the unit interval), but also for *any* complete lattice with an involution.

The result above may also serve as a clarification of Atanassov's decision to consider only the elements (x_1, x_2) in $[0, 1]^2$ s.t. $x_1 + x_2 \leq 1$: these are exactly the consistent elements of $[0, 1]^2$, when the conflation is defined in $[0, 1]^2$, by $\neg(x_1, x_2) = (1 - x_2, 1 - x_1)$. The fact that we consider super-lattices of $\mathcal{L}^*$ (i.e., *all* the elements in $[0, 1]^2$ are considered) allows us to introduce elements such as $(x_1, x_2) = (1, 1)$, in which the membership degree (x_1) and the non-membership

degree (x_2) are both maximal, so we have a totally inconsistent belief in this case. As an important aspect of fuzzy set theory is reasoning with uncertainty, such values should not be ruled out.

Note 1. In [5], Atanassov introduced a pair of bijections between L^* and $[0, 1]^2$, which at first glance seems to shatter the remarks made above that the latter is a more expressive structure. It was shown in [11], however, that these bijections do not preserve order and hence they do *not* constitute an isomorphism between $\mathcal{L}^*$ and $([0, 1]^2, \leq_t)$. The following proposition generalizes this result to *any* complete lattice $\mathcal{L}$ with an involution $\mathcal{N}$.

Proposition 3. *For a complete lattice $\mathcal{L} = (L, \leq_L)$ with an involution $\mathcal{N}$, the structures $\mathcal{L}^2$ and $\mathcal{C}^{-\mathcal{N}}(\mathcal{L}^2)$ are not isomorphic.*

4 Graded (Bilattice-Based) Logical Connectives

In this section we recall some common extensions to $\mathcal{L}$ -fuzzy set theory of the main connectives of classical logic, and show how they can be related to bilattices. In what follows $\mathcal{L} = (L, \leq_L)$ is a complete lattice, $0_{\mathcal{L}} = \inf L$ and $1_{\mathcal{L}} = \sup L$.

4.1 Negation

Definition 12. A *negator* on $\mathcal{L}$ is any $\leq_L$ -decreasing mapping $\mathcal{N} : L \rightarrow L$ satisfying $\mathcal{N}(0_{\mathcal{L}}) = 1_{\mathcal{L}}$ and $\mathcal{N}(1_{\mathcal{L}}) = 0_{\mathcal{L}}$. If, for every x in L , $\mathcal{N}(\mathcal{N}(x)) = x$, then $\mathcal{N}$ is called an *involution negator* on $\mathcal{L}$.

The operation $\neg$ in Definition 5 is an involutive negator on the lattice $(B, \leq_t)$. Therefore, the operation $\neg$, defined on the square $\mathcal{L}^2$ by $\neg(x_1, x_2) = (x_2, x_1)$, is an involutive negator on $(L^2, \leq_t)$. If a bilattice $\mathcal{B}$ has a conflation $-$, then by its definition this operation is an involutive negator on the lattice $(B, \leq_k)$.

Suppose now that $\mathcal{N}$ is an involutive negator on $\mathcal{L}$. Then, as we have shown above, a conflation $-\mathcal{N}$ of $\mathcal{L}^2$ may be defined by $-\mathcal{N}(x_1, x_2) = (\mathcal{N}(x_2), \mathcal{N}(x_1))$. In this case, another natural negator $\sim_{\mathcal{N}}$ on $(L^2, \leq_t)$ is obtained by combining $\neg$ and $-\mathcal{N}$ as follows: $\sim_{\mathcal{N}}(x_1, x_2) = \neg -\mathcal{N}(x_1, x_2) = (\mathcal{N}(x_1), \mathcal{N}(x_2))$.

One might wonder if there exist other ‘interesting’ negators apart from the prototypical ones described above. In [12] it was shown however that for the particular structure $([0, 1]^2, \leq_t)$ all involutive negators can be generated by simple transformations of the two basic choices $\neg$ and $\sim_{\mathcal{N}}$. The next proposition is a generalization of that result to squares.

Definition 13. For $x = (x_1, x_2)$ in L^2 , denote: $pr_1(x) = x_1$ and $pr_2(x) = x_2$.

Proposition 4. *Let $\mathcal{L} = (L, \leq_L)$ be a chain. An operation $\mathfrak{N}$ is an involutive negator on $(L^2, \leq_t)$ iff either*

$$\mathfrak{N}(x_1, x_2) = (\mathcal{N}_1(x_1), \mathcal{N}_2(x_2)) \tag{1}$$

where $\mathcal{N}_1$ and $\mathcal{N}_2$ are two involutive negators on $\mathcal{L}$ such that $\mathcal{N}_1(x) = pr_1\mathfrak{N}(x, 0_{\mathcal{L}})$ and $\mathcal{N}_2(x) = pr_2\mathfrak{N}(0_{\mathcal{L}}, x)$, or

$$\mathfrak{N}(x_1, x_2) = (\varphi(x_2), \varphi^{-1}(x_1)) \quad (2)$$

where φ is an increasing permutation of $\mathcal{L}$ such that $\varphi(x) = pr_1\mathfrak{N}(0_{\mathcal{L}}, x)$.

Clearly, $\sim_{\mathcal{N}}$ is obtained from Expression (1) where $\mathcal{N} = \mathcal{N}_1 = \mathcal{N}_2$, and $\neg$ is obtained from Expression (2) where φ is the identity permutation of L .

One of the advantages of $\sim_{\mathcal{N}}$ is that it preserves the following weakened version of the law of the excluded middle.

Definition 14. An involutive negator $\mathcal{N}$ on $\mathcal{L}$ is called *Kleene negator*, if for all x, y in L , $x \wedge_L \mathcal{N}(x) \leq_L y \vee_L \mathcal{N}(y)$.

The intuition here is that even when the excluded middle or contradiction law do not hold, ‘intended’ contradictions should not surpass ‘intended’ tautologies.

Proposition 5. If $\mathcal{N}$ is a Kleene negator on $\mathcal{L}$, then $\sim_{\mathcal{N}}$ is a Kleene negator on $(L^2, \leq_t)$.

Unlike $\sim_{\mathcal{N}}$, the negator $\neg$ never satisfies Kleene’s condition (to see this, consider, for instance, $(x_1, x_2) = (1_{\mathcal{L}}, 1_{\mathcal{L}})$ and $(y_1, y_2) = (0_{\mathcal{L}}, 0_{\mathcal{L}})$). On the other hand, $\sim_{\mathcal{N}}$ also has some disadvantages. For instance, it cannot serve as a (bilattice) negation on $\mathcal{L}^2$ in the sense of Definition 5, since it does not preserve the $\leq_k$ -order (In $\mathcal{L}_2^2$, for example, although $(1, 0) \leq_k (1, 1)$, still $\sim_{\mathcal{N}}(1, 0) \not\leq_k \sim_{\mathcal{N}}(1, 1)$).

Consider now negators in triangles $\mathcal{I}(\mathcal{L})$, or — equivalently — the substructure $\mathcal{C}^{-\mathcal{N}}(\mathcal{L})$. By the following proposition, it is rather straightforward to find an analogous definition of $\neg$ for these structures, while for $\sim_{\mathcal{N}}$ this is not possible.

Proposition 6. Let $\mathcal{L}$ be a complete lattice with an involutive negator $\mathcal{N}$. Then $\mathcal{C}^{-\mathcal{N}}(\mathcal{L})$ is closed under $\neg$ but not under $\sim_{\mathcal{N}}$.

Thus, for the negator $\neg$, a corresponding triangle operation may be obtained by applying the isomorphism $f([x_1, x_2]) = (x_1, \mathcal{N}(x_2))$, used in the context of Proposition 2, to obtain an operation $\mathfrak{N}$ defined, for every $[x_1, x_2]$ in $I(L)$, by

$$\mathfrak{N}([x_1, x_2]) = [\mathcal{N}(x_2), \mathcal{N}(x_1)]. \quad (3)$$

As $[\mathcal{N}(x_2), \mathcal{N}(x_1)]$ is an interval, $\mathfrak{N}$ is an involutive negator on $(I(L), \leq_t)$.

Next we show, as we did for squares (cf. Proposition 4), that Expression (3) is a characterization of involutive negators in many common triangles:

Definition 15. For $x = [x_1, x_2] \in I(L)$, denote: $l(x) = x_1$ and $r(x) = x_2$.

Proposition 7. Let $\mathcal{L} = (L, \leq_L)$ be a chain with at least four elements. An operation $\mathfrak{N}$ is an involutive negator on $(I(L), \leq_t)$ iff $\mathfrak{N}([x_1, x_2]) = [\mathcal{N}(x_2), \mathcal{N}(x_1)]$, where $\mathcal{N}$ is an involutive negator on $\mathcal{L}$ such that $\mathcal{N}(x) = r(\mathfrak{N}[x, 1_{\mathcal{L}}]) = l(\mathfrak{N}[0_{\mathcal{L}}, x])$.

Proposition 7 is not true unless the chain $\mathcal{L}$ has at least four elements:

Example 3. Consider a mapping $\mathcal{N}$ on $(I(L_3), \leq_t)$, defined as follows:

$$\mathcal{N}([x_1, x_2]) = \begin{cases} [\frac{1}{2}, \frac{1}{2}] & \text{if } [x_1, x_2] = [0, 1] \\ [0, 1] & \text{if } [x_1, x_2] = [\frac{1}{2}, \frac{1}{2}] \\ [1 - x_2, 1 - x_1] & \text{otherwise} \end{cases}$$

It is easy to check that this is an involutive negator on $(I(L_3), \leq_t)$, which is not of the form of Expression (3) (thus it is not generated as described in Proposition 7).

In [12] it is shown that there does not exist a Kleene negator on $\mathcal{I}([0, 1], \leq_t)$. The following example shows that this does not hold in general for *any* triangle.

Example 4.

- a) The operation $\mathcal{N}$, defined by $\mathcal{N}([0, 0]) = [1, 1]$, $\mathcal{N}([1, 1]) = [0, 0]$ and $\mathcal{N}([0, 1]) = [0, 1]$ is a Kleene negator on $(I(L_2), \leq_t)$.
- b) The mapping $\mathcal{N}$ of Example 3 is a Kleene negator on $(I(L_3), \leq_t)$.

Proposition 8. Let $\mathcal{L} = (L, \leq_L)$ be a chain with at least four elements. Then there does not exist a Kleene negator on $(I(L), \leq_t)$.

4.2 Conjunction and Disjunction

Definition 16. A *triangular norm* (a *t-norm*, for short) on $\mathcal{L} = (L, \leq_L)$ is a mapping $\mathcal{T} : L \times L \rightarrow L$ that is $\leq_L$ -increasing in both arguments, commutative, associative, and satisfies, for every x in L , $\mathcal{T}(1_{\mathcal{L}}, x) = x$.

Definition 17. A *triangular conorm* (a *t-conorm*, for short) on $\mathcal{L} = (L, \leq_L)$ is a mapping $\mathcal{S} : L \times L \rightarrow L$ that is $\leq_L$ -increasing in both arguments, commutative, associative, and satisfies, for every x in L , $\mathcal{S}(0_{\mathcal{L}}, x) = x$.

Given a pre-bilattice $\mathcal{B} = (B, \leq_t, \leq_k)$, its $\leq_t$ -meet $\wedge$ and $\leq_k$ -meet $\otimes$ are clearly t-norms on $(B, \leq_t)$ and $(B, \leq_k)$, respectively. Also, the $\leq_t$ -join $\vee$ and the $\leq_k$ -join $\oplus$ of $\mathcal{B}$ are t-conorms on $(B, \leq_t)$, and $(B, \leq_k)$, respectively. This implies that for a complete lattice $\mathcal{L} = (L, \leq)$ with a meet $\wedge_L$ and a join $\vee_L$, $\mathcal{T}_{\leq_t}((x_1, x_2), (y_1, y_2)) = (x_1 \wedge_L y_1, x_2 \vee_L y_2)$ is a t-norm on $(L^2, \leq_t)$ and $\mathcal{T}_{\leq_k}((x_1, x_2), (y_1, y_2)) = (x_1 \wedge_L y_1, x_2 \wedge_L y_2)$ is a t-norm on $(L^2, \leq_k)$. Similarly, $\mathcal{S}_{\leq_t}((x_1, x_2), (y_1, y_2)) = (x_1 \vee_L y_1, x_2 \wedge_L y_2)$ is a t-conorm on $(L^2, \leq_t)$ and $\mathcal{S}_{\leq_k}((x_1, x_2), (y_1, y_2)) = (x_1 \vee_L y_1, x_2 \vee_L y_2)$ is a t-conorm on $(L^2, \leq_k)$. Also, $\mathcal{T}_{\leq_t}$ is the $\leq_t$ -greatest t-norm of $(L^2, \leq_t)$ and $\mathcal{T}_{\leq_k}$ is the $\leq_k$ -greatest t-norm of $(L^2, \leq_k)$. Similarly, $\mathcal{S}_{\leq_t}$ and $\mathcal{S}_{\leq_k}$ are, respectively, the $\leq_t$ -smallest t-conorm of $(L^2, \leq_t)$ and the $\leq_k$ -smallest t-conorm of $(L^2, \leq_k)$.

The definition of $\mathcal{T}_{\leq t}$, $\mathcal{S}_{\leq t}$, $\mathcal{T}_{\leq k}$, and $\mathcal{S}_{\leq k}$ is an example of an effective way of generating t-(co)norms on (substructures of) squares $\mathcal{L}^2$ by taking advantage of existing connectives on the underlying lattice $\mathcal{L}$. This leads us to define the notion of $\mathcal{L}$ -representability.

Definition 18. Let $\mathcal{L} = (L, \leq_L)$ be a complete lattice. A t-norm $\mathfrak{T}$ on $(L^2, \leq_t)$ (respectively, a t-conorm $\mathfrak{S}$) is called $\mathcal{L}$ -representable, if there exist a t-norm $\mathcal{T}$ and a t-conorm $\mathcal{S}$ on $\mathcal{L}$ (respectively, a t-conorm $\mathcal{S}'$ and a t-norm $\mathcal{T}'$ on $\mathcal{L}$) such that, for every $(x_1, x_2), (y_1, y_2)$ in L^2 ,

$$\mathfrak{T}((x_1, x_2), (y_1, y_2)) = (\mathcal{T}(x_1, y_1), \mathcal{S}(x_2, y_2)) \quad (4)$$

$$\mathfrak{S}((x_1, x_2), (y_1, y_2)) = (\mathcal{S}'(x_1, y_1), \mathcal{T}'(x_2, y_2)) \quad (5)$$

$\mathcal{T}$ and $\mathcal{S}$ (resp. $\mathcal{S}'$ and $\mathcal{T}'$) are called the *representants* of $\mathfrak{T}$ (resp. $\mathfrak{S}$).

Analogously, $\mathcal{L}$ -representable t-(co)norms on $(L^2, \leq_k)$ can be defined in the obvious way.

The definition above allows a straightforward construction of t-(co)norms by operations that meet Definitions 16 and 17; it suffices to take *any* t-norm $\mathcal{T}$ and t-conorm $\mathcal{S}$ on $\mathcal{L}$, and to use them as representants in formulas (4) and (5) above. The converse, however, is not true; not any t-(co)norm on $\mathcal{L}^2$ can be obtained by a representation. For instance, in [21] it is shown that the mapping $\mathcal{T} : [0, 1]^2 \rightarrow [0, 1]$, given by:

$$\mathcal{T}((x_1, x_2), (y_1, y_2)) = (\min(x_1, y_1, \max(x_2, y_2)), \min(x_2, y_2)) \quad (6)$$

is indeed a t-norm on $([0, 1]^2, \leq_k)$, but clearly it is not $\mathcal{L}$ -representable, since its first component also depends on x_2 and y_2 .

Next we relate t-norms and t-conorms by appropriate negators. A natural way of doing so is to impose de Morgan's laws.

Definition 19. Let $\mathcal{T}$ be a t-norm on $\mathcal{L}$, and let $\mathcal{N}$ be an involutive negator on $\mathcal{L}$. A t-conorm $\mathcal{S}$ on $\mathcal{L}$ defined by $\mathcal{S}(x, y) = \mathcal{N}(\mathcal{T}(\mathcal{N}(x), \mathcal{N}(y)))$ is called the $\mathcal{N}$ -dual of $\mathcal{T}$. A t-norm on $\mathcal{L}$ that is the $\mathcal{N}$ -dual of a given t-conorm, is defined on $\mathcal{L}$ in a similar way.

For $\mathcal{L}$ -representable t-norms with $\mathcal{N}$ -dual representants on e.g. $(L^2, \leq_t)$, the choice of the negator $\sim_{\mathcal{N}}$ or $\neg$ does not affect the identity of the dual t-conorm.

Proposition 9. Suppose that $\mathfrak{T}$ is an $\mathcal{L}$ -representable t-norm on $(L^2, \leq_t)$ with representants $(\mathcal{T}, \mathcal{S})$, such that $\mathcal{T}$ is the $\mathcal{N}$ -dual t-norm of $\mathcal{S}$ and $\mathcal{N}$ is an involutive negator on $\mathcal{L}$. Then the $(\sim_{\mathcal{N}})$ -dual and the $(\neg)$ -dual of $\mathfrak{T}$ are the same.

A similar discussion applies also to the definitions of t-(co)norms on (substructures of) $\mathcal{I}(\mathcal{L})$, with the caveat that (1) there are no t-norms and t-conorms on $(\mathcal{I}(L), \leq_k)$, and (2) in the choice of representants on $\mathcal{L}$ it must be assured that the resulting composite operation always yields an element of $\mathcal{I}(L)$.

4.3 Implication

Definition 20. An *implicator* on a lattice $\mathcal{L}$ is a mapping $\mathcal{I} : L \times L \rightarrow L$, $\leq_L$ -increasing in its first component and $\leq_L$ -decreasing in its second component, such that $\mathcal{I}(0_{\mathcal{L}}, 0_{\mathcal{L}}) = 1_{\mathcal{L}}$, $\mathcal{I}(1_{\mathcal{L}}, 0_{\mathcal{L}}) = 0_{\mathcal{L}}$, $\mathcal{I}(0_{\mathcal{L}}, 1_{\mathcal{L}}) = 1_{\mathcal{L}}$, and $\mathcal{I}(1_{\mathcal{L}}, 1_{\mathcal{L}}) = 1_{\mathcal{L}}$.

Given a t-norm $\mathcal{T}$ and an implicator $\mathcal{I}$ on $\mathcal{L}$, it is usual to require the following condition, known as the *residuation principle*.

$$\mathcal{T}(x, z) \leq_L y \Leftrightarrow z \leq_L \mathcal{I}(x, y). \quad (7)$$

This leads to the following class of implicators:

Definition 21. Let $\mathcal{T}$ be a t-norm on $\mathcal{L}$. An $\mathcal{R}$ -*implicator* $\mathcal{I}_{\mathcal{T}}$ (the *residuum* of $\mathcal{T}$) is defined, for every x, y in L , by $\mathcal{I}_{\mathcal{T}}(x, y) = \sup \{z \in L \mid \mathcal{T}(x, z) \leq_L y\}$.

Another definition of a family of implicators is motivated by the classical definition of the material implication $x \rightarrow y$ as $\neg x \vee y$.

Definition 22. Let $\mathcal{S}$ be a t-conorm and $\mathcal{N}$ a negator on $\mathcal{L}$. The $\mathcal{S}$ -*implicator* $\mathcal{I}_{\mathcal{S}, \mathcal{N}}$ (generated by $\mathcal{S}$ and $\mathcal{N}$) is defined by $\mathcal{I}_{\mathcal{S}, \mathcal{N}}(x, y) = \mathcal{S}(\mathcal{N}(x), y)$.

Clearly, each $\mathcal{R}$ -implicator and $\mathcal{S}$ -implicator is in particular an implicator. Moreover, these definitions reveal that very often implicators are linked to ‘simpler’ connectives. Also, we can exploit the classical equivalence between the formulas $x \rightarrow f$ and $\neg x$, to define the following special kind of negator on $\mathcal{L}$.

Definition 23. Let $\mathcal{L} = (L, \leq)$ be a complete lattice with an implicator $\mathcal{I}$. Then $\mathcal{N}_{\mathcal{I}}$ defined by $\mathcal{N}_{\mathcal{I}}(x) = \mathcal{I}(x, 0_{\mathcal{L}})$, is called the *induced negator* of $\mathcal{I}$.

Examples of all the above operations on bilattice-based squares and triangles are thus easy to generate using the constructs introduced in the previous sections.

Proposition 10. Let $\mathcal{B} = (B, \leq_t, \leq_k, \neg)$ be a classical bilattice with conflation $-$. Then $\mathcal{I}_{\wedge} = \mathcal{I}_{(\vee, \sim)}$, thus $\mathcal{I}_{\wedge}$ is an $\mathcal{S}$ -implicator and an $\mathcal{R}$ -implicator on $(B, \leq_t)$.

Proposition 10 substantiates the claim, hinted at by Definition 7, that in classical bilattices the combination ‘ $- \neg$ ’ is the one that really plays the role of Boolean negation, and $x \vee -\neg x$ are the analogies of classical tautologies.

Definition 24. Let $\mathcal{L} = (L, \leq_L)$ be a complete lattice. Let $\mathcal{T}$ be a t-norm on $\mathcal{L}$ for which Equation (7) holds together with $\mathcal{I} = \mathcal{I}_{\mathcal{T}}$, and for every x, y in L ,

$$\mathcal{I}_{\mathcal{T}}(\mathcal{I}_{\mathcal{T}}(x, y), y) = \mathcal{I}_{\mathcal{T}}(\mathcal{I}_{\mathcal{T}}(y, x), x) = x \vee_L y. \quad (8)$$

Then $(L, \leq_L, \mathcal{I}_{\mathcal{T}})$ is called an *MV-algebra*.⁴

⁴ This definition is not a reproduction of the original, lengthy one, given in [8], but is rather a minimal characterization in terms of required properties (see e.g. [26]).

Proposition 10 follows from the following observation and the facts that if $(B, \leq_t, \leq_k, \neg)$ is a classical bilattice then $(B, \wedge, \vee, \neg)$ is a Boolean algebra, and any Boolean algebra $(B, \wedge_B, \vee_B, \neg_B)$ is an MV-algebra, where $\mathcal{T} = \wedge_B$ (see [26]).

Proposition 11. In an MV-algebra $(L, \leq_L, \mathcal{T})$, the mappings $\mathcal{I}_{\mathcal{T}}$ and $\mathcal{I}_{\mathcal{S}, \mathcal{N}}$, where $\mathcal{N} = \mathcal{N}_{\mathcal{I}_{\mathcal{T}}}$ and $\mathcal{S}$ is the $\mathcal{N}$ -dual of $\mathcal{T}$, are identical.

Now we investigate what happens in squares that correspond to non-classical bilattices. The following proposition presents the general picture.

Proposition 12. Let $\mathcal{L} = (L, \leq_L)$ be a complete lattice, and let $\mathfrak{T}$ be a t-norm on $(L^2, \leq_t)$. Then $(L^2, \leq_t, \mathfrak{T})$ is an MV-algebra if and only if there exist t-norms $\mathcal{T}$ and $\mathcal{T}'$ on $\mathcal{L}$ such that $(L, \leq_L, \mathcal{T})$ and $(L, \leq_L, \mathcal{T}')$ are both MV-algebras, and such that $\mathfrak{T}$ is $\mathcal{L}$ -representable with representants $\mathcal{T}$ and $\mathcal{S}$, where $\mathcal{S}$ is the $\mathcal{N}$ -dual t-conorm of a t-norm $\mathcal{T}'$ for some involutive negator $\mathcal{N}$.

We turn now to triangles. In these structures the situation is complicated by the fact that there need not exist a Kleene negator on $(I(L), \leq_t)$, while this is a prerequisite of an MV-algebra [9].⁵ The following example summarizes previous findings when $\mathcal{L}$ is the unit interval.

Example 5. Consider the lattice $(I([0, 1]), \leq_t)$. The mapping $\mathcal{T}([x_1, x_2], [y_1, y_2]) = [\max(0, x_1 + y_1 - 1), \max(0, x_2 - 1 + y_1, y_2 - 1 + x_1)]$ is a non-representable t-norm and it satisfies the residuation principle (7) together with $\mathcal{I} = \mathcal{I}_{\mathcal{T}}$. Moreover, here $\mathcal{I}_{\mathcal{T}} = \mathcal{I}_{\mathcal{S}, \neg}$, where $\mathcal{S}$ is the $\neg$ -dual of $\mathcal{T}$. Nevertheless, since there is no Kleene negator on $(I([0, 1]), \leq_t)$, the triple $(I([0, 1]), \leq_t, \mathcal{T})$ is *not* an MV-algebra.

Example 5 thus shows that the property of having coinciding $\mathcal{R}$ - and $\mathcal{S}$ -implicators is not unique to MV-algebras. Conversely, one might also wonder if substructures of bilattice-based triangles can ever be MV-algebras; the following example answers this question in the affirmative.

Example 6. Consider the triangle $\mathcal{I}(\mathcal{L}_2)$ from Example 2. As we have seen, the mapping $\mathcal{N}$ defined in Example 4 is a Kleene negator on $(I(L_2), \leq_t)$. Consider the following truth tables that define a t-norm $\mathcal{T}$ and an implicator $\mathcal{I}$ on $(I(L_2), \leq_t)$:

$\mathcal{T}$	$[0, 0]$	$[1, 1]$	$[0, 1]$	$\mathcal{I}$	$[0, 0]$	$[1, 1]$	$[0, 1]$
$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[1, 1]$	$[1, 1]$	$[1, 1]$
$[1, 1]$	$[0, 0]$	$[1, 1]$	$[0, 1]$	$[1, 1]$	$[0, 0]$	$[1, 1]$	$[0, 1]$
$[0, 1]$	$[0, 0]$	$[0, 1]$	$[0, 0]$	$[0, 1]$	$[0, 1]$	$[1, 1]$	$[1, 1]$

Then $\mathcal{I} = \mathcal{I}_{\mathcal{T}}$, the residuation principle is satisfied in this case, and as it is easily verified (e.g., by checking the truth tables), $(I(L_2), \leq_t, \mathcal{T})$ is an MV-algebra.

⁵ Indeed, if $(\mathcal{L}, \leq_L, \mathcal{T})$ is an MV-algebra, then $\mathcal{N}_{\mathcal{I}_{\mathcal{T}}}$ is a Kleene negator on $\mathcal{L}$; see [9–Theorem 2.31].

We note, finally, that even if a Kleene negator exists on $(I(L), \leq_t)$, it might happen that there is no t-norm $\mathcal{T}$ on this triangle such that $(I(L), \leq_t, \mathcal{T})$ is an MV-algebra. An example of this situation is the triangle $\mathcal{I}(\mathcal{L}_3)$ from Example 2. As we have shown (Example 4-b), there exists a Kleene-negator on $(I(L_3), \leq_t)$. Nevertheless, there is no t-norm on $(I(L_3), \leq_t)$ satisfying Conditions (7) and (8).

5 Conclusion

In this paper we have described an ongoing work that identifies bilattices, and in particular the constructs of bilattice-based squares and triangles, as appropriate structures for relating IVFSs and IFSs within one uniform and general framework. By not constraining ourselves to consistent elements only, a natural setting to represent and handle contradictions emerges.

We have shown that the definition and representation of suitable logical connectives within this setting can benefit a lot from bringing together results from both bilattice and $\mathcal{L}$ -fuzzy set theory, and – moreover – it raises many non-trivial questions regarding the inter-relationships among the various alternatives. In a forthcoming paper we illustrate the application potential and the intuitive appeal of our framework in the context of preference modeling, showing that our approach clarifies and simplifies exiting works in this area (e.g., [17] and [25]).

Acknowledgements

This paper was prepared during the second author's visit to the Academic College of Tel-Aviv, sponsored by the National Science Foundation–Flanders.

References

1. O. Arieli and A. Avron. Reasoning with logical bilattices. *Journal of Logic, Language, and Information*, 5(1):25–63, 1996.
2. O. Arieli and A. Avron. Bilattices and paraconsistency. In *Frontiers of Paraconsistent Logic*, pages 11–27. Research Studies Press, 2000.
3. O. Arieli, C. Cornelis, G. Deschrijver, and E. Kerre. Relating intuitionistic fuzzy sets and interval-valued fuzzy sets through bilattices. In *Applied Computational Intelligence*, pages 57–64. World Scientific, 2004.
4. K. T. Atanassov. Intuitionistic fuzzy sets, 1983. VII ITKR's Session, Sofia (deposed in Central Sci.-Technical Library of Bulg. Acad. of Sci., 1697/84) (in Bulgarian).
5. K. T. Atanassov. Remark on a property of the intuitionistic fuzzy interpretation triangle. *Notes on Intuitionistic Fuzzy Sets*, 8:8–37, 2002.
6. N. D. Belnap. How a computer should think. In G. Ryle, editor, *Contemporary Aspects of Philosophy*, pages 30–56. Oriel Press, 1977.
7. N. D. Belnap. A useful four-valued logic. In *Modern Uses of Multiple-Valued Logic*, pages 7–37. Reidel Publishing Company, 1977.
8. C. Chang. Algebraic analysis of many valued logics. *Trans. AMS*, 93:74–80, 1958.

9. C. Cornelis. Two-sidedness in the representation and processing of imprecise information, 2004. Ph. D. Thesis, Ghent University.
10. C. Cornelis, K. T. Atanassov, and E. Kerre. Intuitionistic fuzzy sets and interval-valued fuzzy sets: a comparison. In *Proc. EUSFLAT'03*, pages 159–163, 2003.
11. C. Cornelis, G. Deschrijver, and E. Kerre. Implication in intuitionistic and interval-valued fuzzy set theory: Construction, classification, application. *International Journal of Approximate Reasoning*, 35(1):55–95, 2004.
12. G. Deschrijver, C. Cornelis, and E. Kerre. Square and triangle: a comparison. In *Proc. IPMU'04*, pages 1389–1396, 2004.
13. G. Deschrijver and E. Kerre. On the relationship between some extensions of fuzzy set theory. *Fuzzy Sets and Systems*, 133(2):227–235, 2003.
14. M. Fitting. Personal communication.
15. M. Fitting. Bilattices and the semantics of logic programming. *Journal of Logic Programming*, 11(2):91–116, 1991.
16. M. Fitting. Kleene's logic, generalized. *Logic and Computation*, 1:797–810, 1992.
17. P. Fortemps and R. Słowiński. A graded quadrivalent logic for ordinal preference modelling. *Fuzzy Optimization and Decision Making*, 1:93–111, 2002.
18. G. Gargov. Knowledge, uncertainty and ignorance in logic: bilattices and beyond. *Journal of Applied Non-Classical Logics*, 9(2–3):195–283, 1999.
19. M. L. Ginsberg. Multi-valued logics: A uniform approach to reasoning in artificial intelligence. *Computer Intelligence*, 4:256–316, 1988.
20. J. Goguen. L-fuzzy sets. *Journal Math. Anal. Appl.*, 18:145–174, 1967.
21. S. Jenei and B. De Baets. On the direct decomposability of t-norms on product lattices. *Fuzzy Sets and Systems*, 139(3):699–707, 2003.
22. B. Messing. Combining knowledge with many-valued logics. *Data and Knowledge Engineering*, 23:297–315, 1997.
23. R. Nelken and N. Francez. Bilattices and the semantics of natural language questions. *Linguistic and Philosophy*, 25(1):37–64, 2002.
24. G. Takeuti and S. Titani. Intuitionistic fuzzy logic and intuitionistic fuzzy sets theory. *Journal of Symbolic Logic*, 49:851–866, 1984.
25. A. Tsoukiàs and P. Vincke. Extended preference structures in mcda. In J. Clímaco, editor, *Multi-criteria Analysis*, pages 37–50. Springer-Verlag, 1997.
26. E. Turunen. *Mathematics behind fuzzy logic*. Advances in Soft Computing, 1999.
27. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

A New Algorithm to Compute Low T-Transitive Approximation of a Fuzzy Relation Preserving Symmetry. Comparisons with the T-Transitive Closure

Luis Garmendia¹ and Adela Salvador²

¹ Facultad de Informática, Dpto. de Lenguajes y Sistemas Informáticos,
Universidad Complutense of Madrid, 28040 Madrid, Spain
lgarmend@fdi.ucm.es

² E.T.S.I. Caminos Canales y Puertos, Dpto. de Matemática Aplicada ,
Technical University of Madrid, 28040 Madrid, Spain
ma09@caminos.upm.es

Abstract. It is given a new algorithm to compute a lower T-transitive approximation of a fuzzy relation that preserves symmetry. Given a reflexive and symmetric fuzzy relation, the new algorithm computes a T-indistinguishability that is contained in the fuzzy relation. It has been developed a C++ program that generates random symmetric fuzzy relations or random symmetric and reflexive fuzzy relations and computes their T-transitive closure and the new low T-transitive approximation. Average distances of the fuzzy relation with the T-transitive closure are similar than the average distances with the low T-transitive approximation.

1 Introduction

Fuzzy relations have many applications to make fuzzy inference in many branches of Artificial Intelligence with uncertainty, imprecision or lack of knowledge. Reflexive and T-transitive fuzzy relation (called T-preorders, for any continuous t-norm T) make Tarski consequences when using the composite rule of inference, obtaining all the consequences of a few premises in just one S-T-composition. Reflexive symmetric and T-transitive fuzzy relations (called T-indistinguishabilities) have been very useful in many classification and clustering methods, allowing to represent the knowledge to distinguish objects.

A new method to T-transitivize fuzzy relations [Garmendia & Salvador; 2000] can be used to measure of T-transitivity of fuzzy relations and to build T-transitive low approximations of a given fuzzy relations. That algorithm preserves all the diagonal values, so it preserves the α -reflexivity, however it doesn't preserve the symmetry property, so we have developed a different version of the algorithm that keeps the symmetry property.

Fuzzy relations on a finite set can also represent labeled directed graphs. The T-transitive closure generalize the transitive closure of a directed graph, and lower T-transitive approximations are T-transitive subgraphs. Symmetric fuzzy relations can represent non directed graphs, where a generalized transitive property could be studied or inferred.

The new algorithm is implemented in a C++ program that generate random symmetric fuzzy relations or random reflexive and symmetric fuzzy relations of a given dimension and computes their Min-transitive closure, Prod-transitive closure and W-transitive closure, and compares them with their Min-transitive, Prod-transitive and W-transitive low symmetric approximations using the new proposed algorithm.

It is computed the measure of low T-transitivity of fuzzy relations measuring the difference between the transitive low approximations and the original fuzzy relation, using several distances as the absolute value of the difference, euclidean distances or normalized distances. Those distances are also measured between the same random fuzzy relations and their T-transitive closures, resulting to be higher than the average distances with the T-transitive low approximations for all dimensions computed.

2 Preliminaries

2.1 The Importance of the Transitivity Property

The T-transitive property is held by T-indistinguishabilities and T-preorders, and it is important when making fuzzy inference to have *Tarski* consequences. The similarities and T-indistinguishabilities generalize the classical equivalence relations, and are useful to classify or to make fuzzy partitions of a set. T-indistinguishability relations generalize the classical equivalence relations and they are useful to define degrees of ‘similarities’ or generalized distances.

Even though not all the fuzzy inference in control needs transitivity, it looks important to know whether the fuzzy relation is T-transitive in order to make fuzzy inference, and if a relation is not T-Transitive it is possible to find another T-transitive fuzzy relation as close as possible with the initial fuzzy relation.

2.2 Transitive Closure

The T-transitive closure R^T of a fuzzy relation R is the lowest relation that contains R and is T-transitive. There are many proposed algorithms to compute the T-transitive closure [Naessens, De Meyer, De Baets; 2002].

An algorithm used to compute the transitive closure is the following:

- 1) $R' = R \cup_{\text{Max}} (R \circ_{\text{Sup-T}} R)$
- 2) If $R' \neq R$ then $R := R'$ and go back to 1), otherwise stop and $R^T := R'$.

2.3 A New T-Transitivization Algorithm

At ‘On a new method to T-transitivize fuzzy relations’ [Garmendia & Salvador; 2000] it is proposed a new algorithm to compute low T-transitive approximations of fuzzy relations, obtaining a fuzzy T-transitive relation ‘as close as possible’ from the initial fuzzy relation. If the initial relation is T-transitive then it is equal to the T-transitivized relation.

The transitivized relation keeps important properties as the μ -T-conditionality property and reflexivity that also preserves the transitive closure, but it also keeps some more properties as the invariance of the relation degree of every element with himself (or diagonal), and so it preserves α -reflexivity. The transitivity closure does not preserve α -reflexivity, but preserves symmetry.

2.4 Previous Concepts

Let $E = \{a_1, \dots, a_n\}$ be a finite set.

A fuzzy relation $R: E \times E \rightarrow [0, 1]$ is a T-indistinguishability when it is reflexive, symmetric and T-transitive.

A T-indistinguishability is called a similarity when T is the minimum t-norm.

Definition 1: Let T be a triangular t-norm [Schweizer & Sklar; 1983]. A fuzzy relation $R: E \times E \rightarrow [0, 1]$ is **T-transitive** if $T(R(a,b), R(b,c)) \leq R(a,c)$ for all a, b, c in E .

Given a fuzzy relation R it is called element $a_{i,j}$ to the relation degree in $[0, 1]$ between the elements a_i and a_j in E . So $a_{i,j} = R(a_i, a_j)$.

Definition 2: An element $a_{i,j}$ is called **T-transitive element** if $T(a_{i,k}, a_{k,j}) \leq a_{i,j}$ for all k from 1 to n .

Algorithm: The proposed algorithm transform a fuzzy relation R^0 into another T-transitive relation R_T contained in R^0 in n^2-1 steps. In each step can be reduced some degrees so $R = R^0 \supseteq R^1 \supseteq \dots \supseteq R^m \supseteq \dots \supseteq R^{n^2-1} = R_T$.

The idea of this method is to get profit of the fact that each step makes sure that an element $a_{i,j}$ will be T-transitive for all further steps, and so it will be T-transitive in the final relation R_T . In summary, each step $m+1$ T-transitivize an element $a_{i,j}^m$ in R^m reducing other elements $a_{i,k}^m$ or $a_{k,j}^m$, when it is necessary, resulting that $a_{i,j}^r$ is T-transitive in R^r for all $r \geq m$. To achieve this, it is important to choose in each step the minimum non T-transitivized element as the candidate to transitivize (reducing other elements). When choosing to transitivize the minimum $a_{i,j}^m$ in R^m it is sure that $a_{i,j}^m = a_{i,j}^r$ for all $r \geq m$ (it will not change in further steps), because the reduction of other elements will not make it intransitive anymore and because $a_{i,j}^m$ is lower or equal further transitivized elements, it will not cause intransitivity and it will not be reduced.

Let τ be a set of pairs (i, j) where i, j are integers from 1 to n .

Definition 3: τ^m is a subset of τ defined by:

- 1) $\tau^0 = \emptyset$
- 2) $\tau^{m+1} = \tau^m \cup (i, j)$ if $a_{i,j}^m$ is the element in R^m chosen to be T-transitivized in the $m+1$ step.

So τ^m is the set of pairs (i, j) corresponding the T-transitivized elements in R^m and $(\tau^m)'$ is the set of n^2-m pairs (i, j) corresponding the not yet transitivized elements.

Building R^{m+1} from R^m : Let $a_{i,j}^m$ be the element in R^m that is going to be transitivized at step $m+1$ ($a_{i,j}^m = \text{Min}\{a_{v,w}^m \text{ such that } (v, w) \in (\tau^m)'\}$).

It is defined $a_{r,s}^{m+1}$ as

$$\begin{cases} J^T(a_{s,j}^m, a_{i,j}^m) & \text{if } r=i, T(a_{r,s}^m, a_{s,j}^m) > a_{i,j}^m \text{ and } a_{i,s}^m \leq a_{s,j}^m \\ J^T(a_{i,r}^m, a_{i,j}^m) & \text{if } s=j, T(a_{i,r}^m, a_{r,s}^m) > a_{i,j}^m \text{ and } a_{i,r}^m \geq a_{r,s}^m \\ a_{r,s}^m & \text{otherwise} \end{cases} \quad (1)$$

where J^T is the residual operator of the t-norm T , defined by $J^T(x, y) = \sup\{z/ T(x, z) \leq y\}$.

If $T(a_{i,k}^m, a_{k,j}^m) > a_{i,j}^m$ for some k , either $a_{i,k}^m$ or $a_{k,j}^m$ will reduce its degree (it could be chosen the minimum of both) to achieve that $T(a_{i,k}^{m+1}, a_{k,j}^{m+1}) \leq a_{i,j}^{m+1} = a_{i,j}^m$.

When choosing the minimum between $a_{i,k}^m$ and $a_{k,j}^m$ to reduce, if it is chosen the minimum one, the difference between R^m and R^{m+1} is lower, so if $a_{i,k}^m \leq a_{k,j}^m$ then $a_{i,k}^{m+1} = J^T(a_{k,j}^m, a_{i,j}^m)$ and if $a_{i,k}^m > a_{k,j}^m$ then $a_{k,j}^{m+1} = J^T(a_{i,k}^m, a_{i,j}^m)$. The degree of the rest of elements remains invariant ($a_{r,s}^{m+1} = a_{r,s}^m$).

3 A New Algorithm to Compute low T-Transitive Approximation of a Fuzzy Relation Preserving Symmetry

Algorithm 2.3 can be used to compute low T-transitive approximations of any fuzzy relations. However, the algorithm can be modified to take profit of the knowledge that the input is going to be a symmetric fuzzy relation

The idea is that when a relation degree $a_{i,j}$ is T-transitivised, we can use the calculations to T-transitivized the symmetric degree $a_{j,i}$ at the same time. So the new algorithm will need half of the steps.

The final algorithm that preserves symmetry is similar to 2.3, but computing $a_{r,s}^{m+1}$ at the same time than $a_{s,r}^{m+1}$

Let E be a set of n elements and let $R^0 : E \times E \rightarrow [0,1]$ be a symmetric fuzzy relation.

Algorithm

The proposed algorithm transform a fuzzy relation R^0 into another T-transitive relation R_T contained in R^0 in $\lceil n^2/2 \rceil$ steps. In each step can be reduced some degrees so R

$$= R^0 \supseteq R^1 \supseteq \dots \supseteq R^m \supseteq \dots \supseteq R^{\lceil \frac{n^2}{2} \rceil} = R_T.$$

Let τ be a set of pairs (i, j) where i, j are integers from 1 to n .

- 1) $\tau^0 = \emptyset$
- 2) $\tau^{m+1} = \tau^m \cup (i, j) \cup (j, i)$ if $a_{i,j}^m$ and is the element in R^m chosen to be T-transitivized at step $m+1$.

Building R^{m+1} from R^m : Let $a_{i,j}^m$ be the element in R^m that is going to be transitivized at step $m+1$ ($a_{i,j}^m = \text{Min}\{a_{v,w}^m \text{ such that } (v, w) \in (\tau^m)'\}$).

It is defined $a_{r,s}^{m+1} := a_{s,r}^{m+1} :=$

$$\begin{cases} J^T(a_{s,j}^m, a_{i,j}^m) & \text{if } r=i, T(a_{r,s}^m, a_{s,j}^m) > a_{i,j}^m \text{ and } a_{i,s}^m \leq a_{s,j}^m \\ J^T(a_{i,r}^m, a_{i,j}^m) & \text{if } s=j, T(a_{i,r}^m, a_{r,s}^m) > a_{i,j}^m \text{ and } a_{i,r}^m \geq a_{r,s}^m \\ a_{r,s}^m & \text{otherwise} \end{cases}$$

where J^T is the residual operator of the t-norm T , defined by $J^T(x, y) = \sup\{z / T(x, z) \leq y\}$.

Example 3.1

Let R be a symmetric fuzzy relation on a set $E = \{a_1, a_2, a_3\}$ defined by the matrix

$$R^0 = \begin{pmatrix} 0,4 & 1 & 0,7 \\ 1 & 0,3 & 0,4 \\ 0,7 & 0,4 & 0,2 \end{pmatrix}$$

To compute the low Min-transitive approximation, the first step is to Min-transitivize the lower relation degree, which is $R(a_3, a_3) = a_{3,3} = 0,2$ using the residuated operator of the Min t-norm on values $a_{3,1}$, $a_{1,3}$ and $a_{3,2}$, $a_{2,3}$, so

$$R^1 = \begin{pmatrix} 0,4 & 1 & 0,2 \\ 1 & 0,3 & 0,2 \\ 0,2 & 0,2 & 0,2 \end{pmatrix}$$

As $a_{3,1}$ and $a_{3,2}$ are Min-transitive (and then their symmetric values), no values are reduced in the next two steps, and $R_2 = R_3 = R_1$.

The lower non Min-transitivized value is $a_{2,2} = 0,3$, that is not Min-transitive. Then

$$R^4 = \begin{pmatrix} 0,4 & 0,3 & 0,2 \\ 0,3 & 0,3 & 0,2 \\ 0,2 & 0,2 & 0,2 \end{pmatrix}$$

$R_T = R^4$ is a low Min-transitive approximation of R

The Min-transitive closure of R is $R^T = \begin{pmatrix} 1 & 1 & 0,7 \\ 1 & 1 & 0,7 \\ 0,7 & 0,7 & 0,7 \end{pmatrix}$, which does not pre-

serve the diagonal values.

4 The Program

The most important continuous t-norms that generalize the AND logical values are the Minimum, Product, and the Lukasiewicz t-norm, $W(x, y) = \max\{0, x+y-1\}$.

4.1 Program Description

It has been developed a program in C++ that generates a random symmetric fuzzy relation (shown at the top of the figure) or a random reflexive and symmetric fuzzy relations and computes the Min-transitive closure, Prod-transitive closure and W-transitive closure, measuring the absolute value distance and euclidean distance with the initially generated fuzzy relation. It also computes the Min-transitive, Prod-transitive and W-transitive low approximations (second row of relations in the figure 1), and also measures their distances with the same original fuzzy relation.

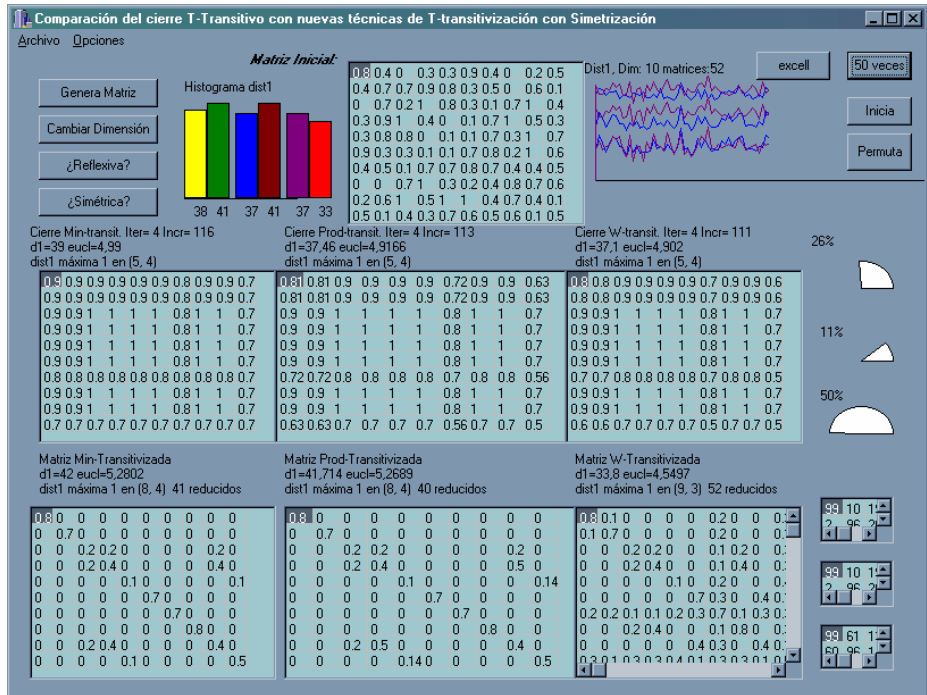


Fig. 1. General front-end of the program

As an example, the program generates the following random symmetric fuzzy relation:

0.8	0.4	0	0.3	0.3	0.9	0.4	0	0.2	0.5
0.4	0.7	0.7	0.9	0.8	0.3	0.5	0	0.6	0.1
0	0.7	0.2	1	0.8	0.3	0.1	0.7	1	0.4
0.3	0.9	1	0.4	0	0.1	0.7	1	0.5	0.3
0.3	0.8	0.8	0	0.1	0.1	0.7	0.3	1	0.7
0.9	0.3	0.3	0.1	0.1	0.7	0.8	0.2	1	0.6
0.4	0.5	0.1	0.7	0.7	0.8	0.7	0.4	0.4	0.5
0	0	0.7	1	0.3	0.2	0.4	0.8	0.7	0.6
0.2	0.6	1	0.5	1	0.4	0.7	0.4	0.1	0.5
0.5	0.1	0.4	0.3	0.7	0.6	0.5	0.6	0.1	0.5

Fig. 2. Example of generated symmetric random fuzzy relation

The program computes the Min-transitive closure, Prod-transitive closure and W-transitive closure measuring the absolute value distance and euclidean distance with the initial fuzzy relation:

Cierre Min-transit. Iter= 4 Incr= 116 d1=39 euc1=4,99 dist1 máxima 1 en {5, 4}	Cierre Prod-transit. Iter= 4 Incr= 113 d1=37,46 euc1=4,9166 dist1 máxima 1 en {5, 4}	Cierre W-transit. Iter= 4 Incr= 111 d1=37,1 euc1=4,902 dist1 máxima 1 en {5, 4}
0.9 0.9 0.9 0.9 0.9 0.9 0.8 0.9 0.9 0.7 0.9 0.9 0.9 0.9 0.9 0.9 0.8 0.9 0.9 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7	0.8 0.8 0.9 0.9 0.9 0.9 0.7 0.9 0.9 0.6 0.8 0.8 0.9 0.9 0.9 0.9 0.7 0.9 0.9 0.6 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.7 0.7 0.8 0.8 0.8 0.8 0.7 0.8 0.8 0.56 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.63 0.63 0.7 0.7 0.7 0.7 0.56 0.7 0.7 0.5	0.8 0.8 0.9 0.9 0.9 0.9 0.7 0.9 0.9 0.6 0.8 0.8 0.9 0.9 0.9 0.9 0.7 0.9 0.9 0.6 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.7 0.7 0.8 0.8 0.8 0.8 0.7 0.8 0.8 0.5 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.9 0.9 1 1 1 1 0.8 1 1 0.7 0.6 0.6 0.7 0.7 0.7 0.7 0.5 0.7 0.7 0.5

Fig. 3. Example of Min-Transitive closure, Prod-transitive closure and W-transitive closure of the random fuzzy relation of Fig. 2, measuring the absolute value distance and euclidean distance with the initial fuzzy relation

It also computes the Min-transitive, Prod-transitive and W-transitive low approximations (second row of relations in the figure) using the algorithm that preserves symmetry, and also measures their distances with the same original fuzzy relation:

Matriz Min-Transitivizada d1=42 euc1=5,2802 dist1 máxima 1 en {8, 4} 41 reducidos	Matriz Prod-Transitivizada d1=41,714 euc1=5,2689 dist1 máxima 1 en {8, 4} 40 reducidos	Matriz W-Transitivizada d1=33,8 euc1=4,5497 dist1 máxima 1 en {9, 3} 52 reducidos
0.8 0 0 0 0 0 0 0 0 0 0 0.7 0 0 0 0 0 0 0 0 0 0 0.2 0.2 0 0 0 0 0.2 0 0 0 0.2 0.4 0 0 0 0 0.4 0 0 0 0 0 0.1 0 0 0 0 0.1 0 0 0 0 0 0.7 0 0 0 0 0 0 0 0 0 0 0.7 0 0 0 0 0 0 0 0 0 0 0.8 0 0 0 0 0.2 0.4 0 0 0 0 0.4 0 0 0 0 0 0.1 0 0 0 0 0.5	0.8 0 0 0 0 0 0 0 0 0 0 0.7 0 0 0 0 0 0 0 0 0 0 0.2 0.2 0 0 0 0 0.2 0 0 0 0.2 0.4 0 0 0 0 0.5 0 0 0 0 0 0.1 0 0 0 0 0.14 0 0 0 0 0 0.7 0 0 0 0 0 0 0 0 0 0 0.7 0 0 0 0 0 0 0 0 0 0 0.8 0 0 0 0 0.2 0.5 0 0 0 0 0.4 0 0 0 0 0 0.14 0 0 0 0 0.5	0.8 0.1 0 0 0 0 0.2 0 0 0 0.1 0.7 0 0 0 0 0.2 0 0 0 0 0 0.2 0.2 0 0 0.1 0.2 0 0 0 0 0.2 0.4 0 0 0.1 0.4 0 0 0 0 0 0 0.1 0 0.2 0 0 0 0 0 0 0 0 0.7 0.3 0.4 0 0 0.2 0.2 0.1 0.1 0.2 0.3 0.7 0.1 0.3 0 0 0 0.2 0.4 0 0 0.1 0.8 0 0 0 0 0 0 0 0.4 0.3 0 0.4 0 0.2 0.1 0.3 0.3 0.4 0.1 0.3 0.3 0.1 0

Fig. 4. Example of Min-transitive, Prod-transitive and W-transitive symmetric low approximation of the random fuzzy relation of Fig. 2, measuring the absolute value distance and euclidean distance with the initial fuzzy relation

When choosing to generate reflexive and symmetric random fuzzy relations their computed T-transitive closures will be generated T-indistinguishabilities.

The histogram shows the absolute value distance of the last random generated fuzzy relation with the (in this order from the left to the right) Min-transitive closure, the Min-transitive low approximation, the Prod-transitive closure, the Prod-transitive low approximation, the W-transitive closure and the W-transitive low approximation. The graphic at the right of the picture compares the absolute value distances of both T-transitivization methods for the t-norms (in this order, from the upper to the lower

graphs) minimum, product and Lukasiewicz for the last hundred of generated random symmetric fuzzy relations.

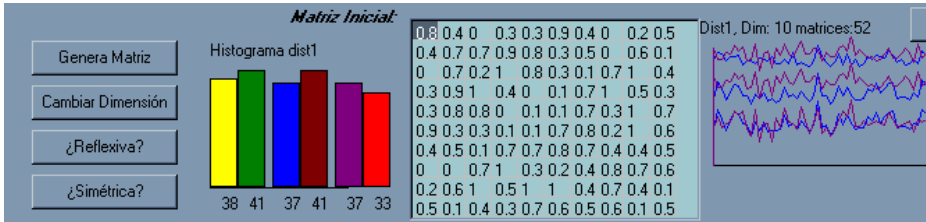


Fig. 5. The histogram shows the absolute value distance of the last random generated fuzzy relation with the Min-transitive closure, the Min-transitive low approximation, the Prod-transitive closure, the Prod-transitive low approximation, the W-transitive closure and the W-transitive low approximation. The graph at the right of the picture compares the absolute value distances of both T-transitivization methods for the t-norms minimum, product and Lukasiewicz for the last hundred of random fuzzy relations

The dimension can be changed. The results for the relation of example 3.1 are in the following figure:

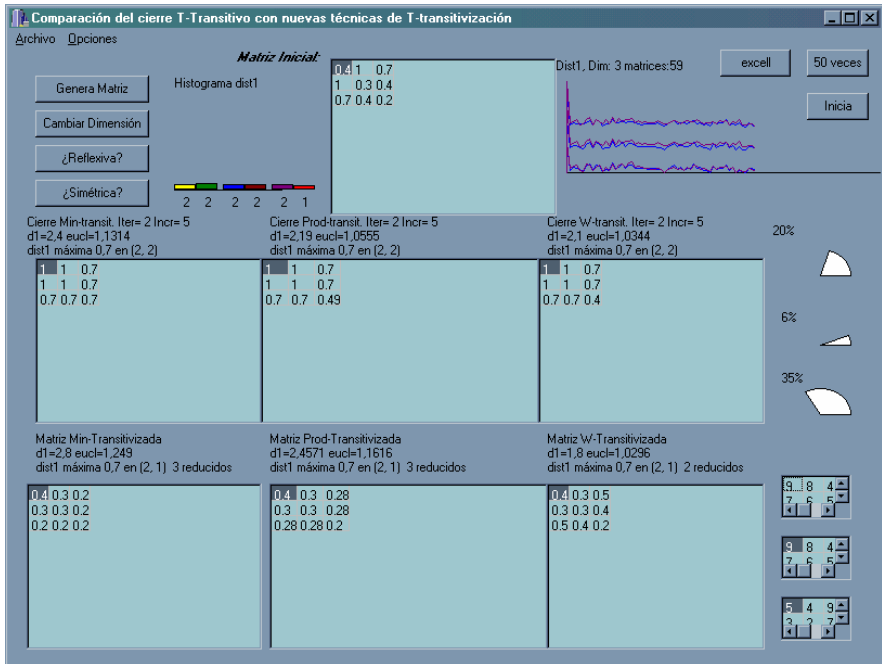


Fig. 6. Program output for example 3.1

The program has been scheduled to generate one hundred of random fuzzy relations for each dimension from two to one hundred. The average distances for each dimension have been saved in an Excel document.

5 Comparing Low Symmetric T-Transitive Approximations with T-Transitive Closures of Random Reflexive and Symmetric Fuzzy Relation

It has been run the program one hundred times for each dimension from two to one hundred, it is, the program has generated 9900 random fuzzy reflexive and symmetric relations, computing their T-transitive closures and their T-transitivized relations for different t-norms, and computing their average distance of absolute value and euclidean for each dimension.

The function in the graphic below represents, for each dimension, the average absolute value distance with their W-transitive closure (the line of higher distances) and the W-transitivized relation. The aspect of the results could change when using other distances, but it is got the same looking for the three t-norms used.

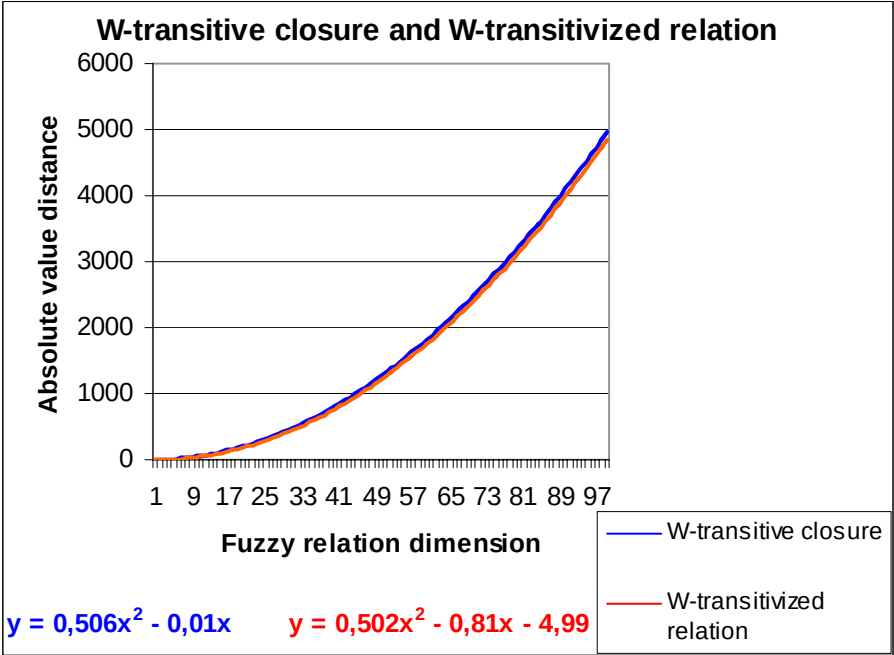


Fig. 7. Average of the absolute value distances of 100 random reflexive and symmetric fuzzy relations with their W-transitive closure and W-transitive low approximation for each dimension from two to one hundred

Table 1. Interpolation function of the average absolute value distance and euclidean distance of the T-transitive closure and T- transitive low approximation of one hundred random fuzzy relations for each dimension from two to one hundred

Absolute value distances	Min	Prod	W
Transitive Closure	$y=0,5x^2+1,2x-16,3$	$y=0,59x^2-1,27x+5,9$	$y=0,506x^2-0,01x$
Transitivized relation	$y=0,46x^2-1,27x+5,9$	$y=0,50x^2-0,83x+2,5$	$y=0,502x^2-0,8x-4,9$
Euclidean distances	Min	Prod	W
Transitive Closure	$y=0,61x-0,42$	$y=0,61x-0,63$	$y=0,61x-0,68$
Transitivized relation	$y=0,56x-0,76$	$y=0,56x-0,77$	$y=0,56x-1,19$

6 Results Analysis

After generating 100 random fuzzy relations for each dimensions from 2 to 100, and compute their average distance with the T-transitive closure and with the T-transitivized relation, we have seen for any distance, for any t-norm and for any dimension that the T-transitive low approximation is similar to the initial relations than the T-transitive closure.

7 Conclusions

The T-transitivization algorithm that keeps symmetry, applies to reflexive and symmetric fuzzy relations, computes T-transitive low approximations with similar distances than the T-transitive closure for any dimension and any t-norm. They are also different, because computes T-transitive relations contained in the initial relation.

The T-transitive closure is uniquely defined, however we can find several maximal T-transitive relations contained in the initial relation.

It is proven [Garmendia & Salvador; 2000] that the T-transitivization algorithm keeps the reflexivity and α -reflexivity. The new algorithm version also preserves symmetry, so produce T-indistinguishabilities from reflexive and symmetric relations. However the T-transitive closure keeps reflexivity, but not α -reflexivity.

Acknowledgment

This research is partially supported by the Spanish MCyT project BFM2002-00281.

References

1. Garmendia, L., Campo, C., Cubillo, S., Salvador, A. A Method to Make Some Fuzzy Relations T-Transitive. *International Journal of Intelligence Systems*. Vol. 14, N° 9, (1999) 873 – 882.
2. Garmendia, L., Salvador, A. On a new method to T-transitivize fuzzy relations, *Information Processing and Management of Uncertainty in Knowledge - based Systems*, IPMU 2000. (2000) 864 – 869.
3. Garmendia, L., Salvador, A. On a new method to T-transitivize fuzzy relations, in *Technologies for Constructing Intelligent Systems 2*, Springer. Edited by Bouchon-Meunier, B., Gutierrez-Rios, J., Magdalena, L., Yager, R. R, (2000) 251 – 260.
4. Klir, G. J., Yuan, B. *Fuzzy Sets and Fuzzy Logic. Theory and Applications*, Prentice Hall, New Jersey, (1995).
5. Hashimoto, H. Transitivity of generalized fuzzy matrices, *Fuzzy Sets and Systems*. Vol. 17, no. 1, (1985) 83-90.
6. Montero, F., Tejada, J. On fuzzy transitivity, *Instituto Nacional de Estadística*, 111, (1986) 49-58.
7. Naessens, H., De Meyer, H., De Baets, B., Algorithms for the Computation of T-Transitive Closures, *IEEE Trans Fuzzy Systems* 10:4 (2002) 541-551.
8. Ovchinnikov, S. Representations of Transitive Fuzzy Relations, in *Aspects of Vagueness*, H. J. Skala, S. Termini y E. Trillas (Eds.), Reidel Pubs. (1984) 105-118.
9. Schweizer, B., Sklar A. *Probabilistic Metric Spaces*, North-Holland, New York, (1983).
10. Trillas, E., Alsina, C., Terricabras, J. M., *Introducción a la lógica borrosa*, Ariel Matemática, (1995).
11. Lee, H.-S. An optimal algorithm for computing the max–min transitive closure of a fuzzy similarity matrix , *Fuzzy Sets and Systems* 123 (2001) 129–136.
12. Xian Xiao, An algorithm for calculating fuzzy transitive closure, *Fuzzy Math.* 5 (4) (1985) 71–73.
13. L.A. Zadeh, Fuzzy sets, *Inform. and Control* 8 (1965) 338–353.
14. Zhen Zhao, An improved algorithm for fuzzy classification analysis, *J. Lanzhou Univ.* 19 (3) (1983) 160–163.
15. Potoczny, H.B., On similarity relations in fuzzy relational databases, *Fuzzy Sets and Systems* 12 (3) (1984) 231–235.
16. Jacas, J., Recasens, J., Fuzzy T-transitive relations: eigenvectors and generators, *Fuzzy Sets and Systems* 72 (1995) 147–154.
17. Jacas, J., Recasens, J., Decomposable indistinguishability operators, *Proceedings of the Sixth IFSA Congress*, Sao Paulo, 1995.
18. Valverde, L., On the structure of F-indistinguishability operators, *Fuzzy Sets and Systems* 17 (1985) 313–328.
19. Zadeh, L., A., Similarity relations and fuzzy orderings, *Inform. Sci.* 3 (1971) 177–200.
20. Boixader, D., On the relationship between T-transitivity and approximate equality Fuzzy Sets and Systems 33 (2003) 6 – 69
21. Wagenknecht, M., On transitive solutions of fuzzy equations, inequalities and lower approximation of fuzzy relations. *Fuzzy Sets and Systems* 75 (1995) 229-240.

Computing a Transitive Opening of a Reflexive and Symmetric Fuzzy Relation

Luis Garmendia¹ and Adela Salvador²

¹ Facultad de Informática, Dpto. de Lenguajes y Sistemas Informáticos,
Universidad Complutense of Madrid, 28040 Madrid, Spain
lgarmend@fdi.ucm.es

² E.T.S.I. Caminos Canales y Puertos, Dpto. de Matemática Aplicada,
Technical University of Madrid, 28040 Madrid, Spain
ma09@caminos.upm.es

Abstract. There are fast algorithms to compute the transitive closure of a fuzzy relation, but there are only a few different algorithms that compute transitive openings from a given fuzzy relation. In this paper a method to compute a transitive opening of a reflexive and symmetric fuzzy relation is given. Even though there is not a unique transitive opening of a fuzzy relation, it is proved that the computed transitive opening closure is maximal.

1 Introduction

The transitivity property of fuzzy relations can be understood as a threshold of a degree of relation (for example, a degree of equality) between two elements, when a degree of relation between those elements with a third one in a universe of discourse is known. The classical concept of transitivity is generalised in fuzzy logic by the T-transitivity property of fuzzy relations.

Fuzzy relations are useful to represent degrees of relations between elements of a universe, and can be used to obtain consequences from a set of premises by the use of the fuzzy compositional rule of inference. Some properties of fuzzy relations give a lot of information of how the consequences are going to be. For example, when an inference is done making a fuzzy composition of a fuzzy set with a reflexive and T-transitive fuzzy relation (called T-preorder), the output contains all the inferable information. The consequences $C(A)$ drawn by making fuzzy inference from a fuzzy set A with T-preorders are Tarski consequences that verify the fuzzy inclusion, so $A \subseteq C(A)$, monotony, so if $A \subseteq B$ then $C(A) \subseteq C(B)$, and idempotence, so $C(C(A)) = C(A)$.

Similarities can be used to represent the concept of equality, neighbourhood, generalising the classical equivalence relations. In fact, the α -cut of a similarity is a classical equivalence relation for any value α . Some applications of similarities can be found in some classification and clusterization methods to distinguish and ‘classify’ objects.

Fuzzy relations on a finite set can also represent labelled directed graphs. Symmetric fuzzy relations can represent weighted complete undirected graphs where the set

of nodes is the universe of discourse and the weights of the edges are the relationship degrees.

Given a fuzzy relation, it is well known that a unique transitive closure exists. Some proposed algorithms to compute the transitive closure of a fuzzy relation are given in Dunn [1974], Kander and Yelowitz [1974], Larsen and Yager [1990], Guoyao Fu [1992], Lee [2001], Naessens, De Meyer and De Baets [2002].

An algorithms to compute T-transitive openings of fuzzy relations for any t-norm T and any fuzzy relation is given by Garmendia and Salvador [2000]. Other algorithms are given by Baets [2003] and Dawyndt [2003].

There are transitive opening of a fuzzy relation, but in general the highest transitive opening cannot be found. This paper puts forward the existence of a maximal Min-transitive opening from a reflexive and symmetric fuzzy relations, which is not unique, but there is not a transitive fuzzy relation that contains the opening and is contained in the fuzzy relation. It is given an algorithm to compute it and it is proved that such transitive opening is maximal.

2 Preliminaries

Let $E = \{a_1, \dots, a_n\}$ be a finite set.

Given a fuzzy relation $R: E \times E \rightarrow [0, 1]$, let a_{ij} be the value of the relation degree of the elements a_i and a_j in E . So $a_{ij} = R(a_i, a_j)$.

A fuzzy relation R is **reflexive** if $a_{ii} = 1$ for all $1 \leq i \leq n$.

The relation R is **symmetric** if $a_{ij} = a_{ji}$ for all $1 \leq i, j \leq n$.

Definition 2.1. A fuzzy relation $R: E \times E \rightarrow [0, 1]$ is **transitive** (or Min-transitive) if $\text{Min}(R(a, b), R(b, c)) \leq R(a, c)$ for all a, b, c in E . So $\text{Min}(a_{ik}, a_{kj}) \leq a_{ij}$ for all $1 \leq i, j \leq n$.

Definition 2.2. A reflexive and symmetric fuzzy relation is called a **proximity** relation.

A **similarity** is a reflexive, symmetric and min-transitive fuzzy relation.

Definition 2.3. The relation A **includes** the relation B ($A \supseteq B$) if $a_{ij} \geq b_{ij}$ for all $1 \leq i, j \leq n$.

Definition 2.4. Given a t-norm T and a fuzzy relation B on a finite universe there exists a unique fuzzy T-transitive relation A , called the **T-transitive closure** of B , that includes B , and if a fuzzy T-transitive relation includes B then it also includes A .

Definition 5. Given a reflexive and symmetric fuzzy relation A on a finite universe, the **transitive opening** of A is a fuzzy similarity relation B satisfying:

B is included in A ($B \subseteq A$)

If any fuzzy similarity relation H includes B and is included in A then it is B . (If $\exists H; B \subseteq H \subseteq A$ then $H = B$).

Note that it can be several maximal transitive openings of a fuzzy relations, as it is shown in figure 1:

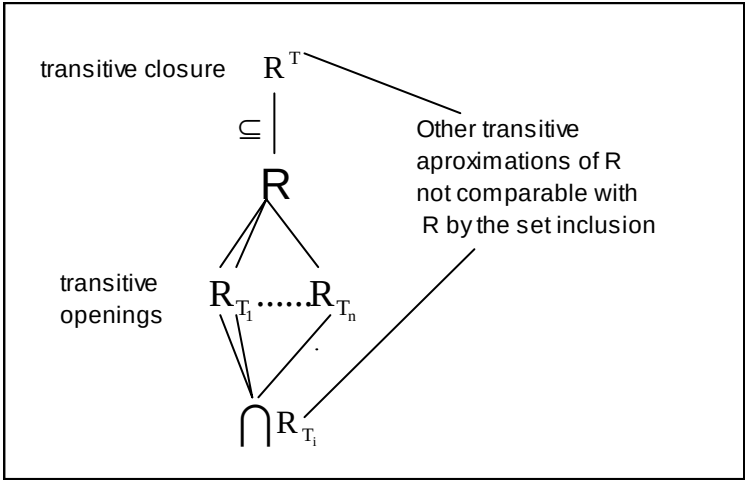


Fig. 1. Relation of the T-transitive closure, T-transitive openings and other T-transitive approximations not comparable by $\subseteq$

In this paper it is proven that in the case of reflexive and symmetric fuzzy relations and t-norm minimum, there exists at least a maximal transitive opening. It also provides an algorithm to compute the maximal transitive opening of a reflexive and symmetric fuzzy relation.

Lemma 2.1. Let π be a permutation on E . If A is a similarity then the fuzzy relation $P_\pi(A)$ is also a fuzzy similarity.

Proof. It is obvious. $P_\pi(A)$ is reflexive and symmetric. If $a_{ij} \geq \text{Min}\{a_{ik}, a_{kj}\}$ for all i, j, k then $a_{rs} = a_{\pi(i)\pi(j)} \geq \text{Min}\{a_{\pi(i)k}, a_{k\pi(j)}\} = T\{a_{rk}, a_{ks}\}$ for all r, s, k .

Example 1. Let π be the permutation $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix}$. An example of a similarity A and its permuted similarity $P_\pi(A)$ is the following:

$$A = \begin{pmatrix} 1 & 0,9 & 0,5 & 0,6 \\ 0,9 & 1 & 0,5 & 0,6 \\ 0,5 & 0,5 & 1 & 0,5 \\ 0,6 & 0,6 & 0,5 & 1 \end{pmatrix}, P_\pi(A) = \begin{pmatrix} 1 & 0,9 & 0,6 & 0,5 \\ 0,9 & 1 & 0,6 & 0,5 \\ 0,6 & 0,6 & 1 & 0,5 \\ 0,5 & 0,5 & 0,5 & 1 \end{pmatrix}$$

A method to build a similarity of lower dimension is given. This method allows to have an easier understanding of the algorithm to compute a transitive closure given at the end of this paper.

As the permutations of similarities are also similarities, it is possible to sort the elements of the universe of discourse E to decompose a similarity in boxes of sub-similarities.

3 Construction of a Fuzzy Similarity from Subsimilarities of Lower Dimension

Let C and D be two similarities with $\dim(C) = n_1$ and $\dim(D) = n_2$.

A similarity relation $R(F; C, D)$ of dimension $n_1 + n_2$ can be constructed with the following form:

$$R(F; C, D) = \begin{pmatrix} \boxed{C} & \boxed{F^T} \\ \boxed{F} & \boxed{D} \end{pmatrix}$$

A method for giving the bridging values e_{ij} in F , (when $j \leq n_1 < i$) is the assignation of a unique value, f , in all the $n_1 \times n_2$ values of F . This value must be chosen in an interval $[0, a]$ where $a = \min\{\min(C), \min(D)\}$. The values in F^T are the symmetric values f of the computed F .

So the computed values in F are equal and satisfy that $f = e_{ij} \leq \min\{\min(C), \min(D)\}$.

Lemma 3.1. If C and D are fuzzy similarities, then $R(f; C, D)$ is also a fuzzy similarity, $\forall f \in [0, \min(\min(C), \min(D))]$.

Proof. The proof is in Lee [2001].

Example 2. The similarity given in example 1 is constructed from other subsimilarities.

Let $T = \text{Min}$, let $C = \begin{pmatrix} 1 & 0,9 \\ 0,9 & 1 \end{pmatrix}$ and $D = (1)$ be two similarities. The construction

of $R(F; C, D)$ is given by assigning equal values to a_{31} and a_{32} in the interval $[0, 0,9]$. Those values can be, for example, $a_{31} = a_{32} = 0,6 = f$. Then

$$R(f; C, D) = \begin{pmatrix} 1 & 0,9 & 0,6 \\ 0,9 & 1 & 0,6 \\ 0,6 & 0,6 & 1 \end{pmatrix}$$

Now let $C_2 = \begin{pmatrix} 1 & 0,9 & 0,6 \\ 0,9 & 1 & 0,6 \\ 0,6 & 0,6 & 1 \end{pmatrix}$ and $D = (1)$, then the new values in F must be chosen equal in the interval $[0, 0,6]$. If $a_{41} = a_{42} = a_{43} = 0,5 = f$ is chosen then

$$R_2(f; C_2, D) = \begin{pmatrix} 1 & 0,9 & 0,6 & 0,5 \\ 0,9 & 1 & 0,6 & 0,5 \\ 0,6 & 0,6 & 1 & 0,5 \\ 0,5 & 0,5 & 0,5 & 1 \end{pmatrix}, \text{ which is the similarity given in example 1.}$$

The lowest similarity constructed from C and D is the following:

$$R3(f; C, D) = \begin{pmatrix} 1 & 0,9 & 0 \\ 0,9 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } R4(f; R3, D) = \begin{pmatrix} 1 & 0,9 & 0 & 0 \\ 0,9 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The highest similarity constructed from C and D is the following:

$$R5(f; C, D) = \begin{pmatrix} 1 & 0,9 & 0,9 \\ 0,9 & 1 & 0,9 \\ 0,9 & 0,9 & 1 \end{pmatrix} \text{ and } R6(f; R5, D) = \begin{pmatrix} 1 & 0,9 & 0,9 & 0,9 \\ 0,9 & 1 & 0,9 & 0,9 \\ 0,9 & 0,9 & 1 & 0,9 \\ 0,9 & 0,9 & 0,9 & 1 \end{pmatrix}$$

Lemma 3.2. If $B_{n \times n}$ is a fuzzy similarity, then there exists a decomposition such that $B_{n \times n} = P_{\pi}(R(f; C_{n1 \times n1}, D_{n2 \times n2}))$.

Proof. The proof is in Lee [2001].

4 Generation and Decomposition of a Given Similarity

A similarity with dimension greater than one can be generated from two subsimilarities. If those subsimilarities are also of dimension greater than one, it is possible to decompose them on other subsimilarities, and so on.

The following method of reasoning generates a given similarity using this decomposability concept in a reverse order. A first similarity of dimension two is created by using the greatest non-diagonal values and more similarities keep adding in such a way that the desired similarity is obtained.

Method to Generate a Given Similarity

Let A be a similarity on an universe E . Let $U(A)$ be the set of the upper triangular values of A sorted in a decreasing order. The method gives sub similarities B_k on the elements of E with the highest values in A .

First step:

Let a_{ij}^1 be the highest value in the list $U(A)$. The first dimension 2 similarity A_1 is created on the Cartesian Product of $\{e_i, e_j\}$, forcing it to be reflexive and symmetric.

$$\text{So } A_1 = B_1 = \begin{pmatrix} 1 & a_{ij}^1 \\ a_{ij}^1 & 1 \end{pmatrix}$$

Step k :

Let a_{ij}^k be the highest value in $U(A)$ not already computed. Then a similarity B_{k+1} and other similarity A_k whose dimension n_k depends on the position (i, j) of a_{ij}^k are created from B_k . Such position defines a partition of the subset of natural numbers

$E_k = \{1, 2, \dots, n_1+n_2+\dots+n_k\}$ into two disjoint sets I and I' in a way that the elements a_{ij} in B_{k-1} verify that $(i, j) \in I \times I$, the elements of the new similarity box A_k verify that $(i, j) \in I' \times I'$, and so the elements in the bridging box F are $f_{ij} = b_{ij}$, where $(i, j) \in I \times I'$.

The step 2 makes a similarity B_2 with the shape $\begin{pmatrix} B_1 & F^T \\ F & A_2 \end{pmatrix}$ or with the shape

$$\begin{pmatrix} B_1 & \\ & A_2 \end{pmatrix}$$

And step k makes a similarity with the shape $\begin{pmatrix} B_{k-1} & F^T \\ F & A_k \end{pmatrix}$ or with the shape

$$\begin{pmatrix} B_{k-1} & \\ & A_k \end{pmatrix}$$

The sets I and I' are defined from the indexes (i, j) of the chosen highest not computed a_{ij}^k in $U(A)$ as follows:

$I = \{j; b_{rj} \text{ is computed in } B_{k-1}\}$ and $I' = \{i; b_{is} \text{ is computed in } B_{k-1}\}$. As the generated similarities B_k must be always reflexive, it must be considered that all the values of the diagonal of B_k are already computed.

Then the elements of F and F^T can be computed in every step as follows:

Set $b_{ij} = b_{ji} = \min\{a_{ij}, \text{ where } i \in I \text{ and } j \in I'\}$, for all $(i, j) \in I \times I'$.

Example 3. Let A be the similarity given by the following matrix:

$$A = \begin{pmatrix} 1 & 0,9 & 0,4 & 0,4 & 0,3 \\ 0,9 & 1 & 0,4 & 0,4 & 0,3 \\ 0,4 & 0,4 & 1 & 0,7 & 0,3 \\ 0,4 & 0,4 & 0,7 & 1 & 0,3 \\ 0,3 & 0,3 & 0,3 & 0,3 & 1 \end{pmatrix}$$

To generate A (or to decompose A in similarities) the greatest element in $U(A)$,

which is $0,9 = a_{21}$ is chosen, so $A_1 = \begin{pmatrix} 1 & 0,9 \\ 0,9 & 1 \end{pmatrix} = B_1$.

In the second step the second greatest element in $U(A)$, which is $0,7 = a_{34}$ is chosen, so $I = \{j; b_{3j} \text{ is computed in the new } B\} = \{3\}$ and $I' = \{i; b_{i4} \text{ is computed in the new } B\} = \{4\}$.

Then $b_{34} = b_{43} = \min\{a_{ij}, \text{ where } i \in I \text{ and } j \in I'\} = 0,7$ for all $(i, j) \in I \times I'$. So

$$B_2 = \begin{pmatrix} \begin{pmatrix} 1 & 0,9 \\ 0,9 & 1 \end{pmatrix} & \\ & \begin{pmatrix} 1 & 0,7 \\ 0,7 & 1 \end{pmatrix} \end{pmatrix}$$

In the next step the next greatest element in $U(A)$, which is $0,4 = a_{13}$ is chosen, so $I = \{j; b_{1j} \text{ is computed in } B_2\} = \{1, 2\}$ and $I' = \{i; b_{i3} \text{ is not computed in } B_2\} = \{3, 4\}$.

Then $b_{ij} = b_{ji} = \min \{a_{ij}, \text{ where } i \in I \text{ and } j \in I'\} = 0,4 \text{ for all } (i, j) \in I \times I'$. So

$$B_3 = \begin{pmatrix} \begin{pmatrix} 1 & 0,9 \\ 0,9 & 1 \end{pmatrix} & \begin{pmatrix} 0,4 & 0,4 \\ 0,4 & 0,4 \end{pmatrix} \\ \begin{pmatrix} 0,4 & 0,4 \\ 0,4 & 0,4 \end{pmatrix} & \begin{pmatrix} 1 & 0,7 \\ 0,7 & 1 \end{pmatrix} \end{pmatrix}$$

In the next step, the greatest element in $U(A)$ is $0,3 = a_{51}$, so $I = \{5\}$ and $I' = \{1, 2, 3, 4\}$

$$\text{So } B_4 = \begin{pmatrix} \begin{pmatrix} \begin{pmatrix} 1 & 0,9 \\ 0,9 & 1 \end{pmatrix} & \begin{pmatrix} 0,4 & 0,4 \\ 0,4 & 0,4 \end{pmatrix} \\ \begin{pmatrix} 0,4 & 0,4 \\ 0,4 & 0,4 \end{pmatrix} & \begin{pmatrix} 1 & 0,7 \\ 0,7 & 1 \end{pmatrix} \end{pmatrix} & \begin{pmatrix} F^T \\ 1 \end{pmatrix} \\ \begin{pmatrix} \begin{pmatrix} 1 & 0,9 \\ 0,9 & 1 \end{pmatrix} & \begin{pmatrix} 0,4 & 0,4 \\ 0,4 & 0,4 \end{pmatrix} \\ \begin{pmatrix} 0,4 & 0,4 \\ 0,4 & 0,4 \end{pmatrix} & \begin{pmatrix} 1 & 0,7 \\ 0,7 & 1 \end{pmatrix} \end{pmatrix} & F \end{pmatrix} =$$

$$\begin{pmatrix} \begin{pmatrix} 1 & 0,9 \\ 0,9 & 1 \end{pmatrix} & \begin{pmatrix} 0,4 & 0,4 \\ 0,4 & 0,4 \end{pmatrix} & \begin{pmatrix} 0,3 \\ 0,3 \\ 0,3 \\ 0,3 \end{pmatrix} \\ \begin{pmatrix} 0,4 & 0,4 \\ 0,4 & 0,4 \end{pmatrix} & \begin{pmatrix} 1 & 0,7 \\ 0,7 & 1 \end{pmatrix} & \begin{pmatrix} 0,3 \\ 0,3 \\ 0,3 \\ 0,3 \end{pmatrix} \\ \begin{pmatrix} 0,3 & 0,3 \\ 0,3 & 0,3 \end{pmatrix} & \begin{pmatrix} 0,3 & 0,3 \\ 0,3 & 0,3 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \end{pmatrix} = A.$$

5 Algorithm to Compute a Maximal Transitive Opening of a Reflexive and Symmetric Fuzzy Relation

Input: a reflexive and symmetric fuzzy relation $A = [a_{ij}]$

Output: a similarity B that is a transitive opening of A

Step 1. Set B to be initially blank.

Step 2. Sort the elements of $U(A)$ in descendent order.

Step 3. Set $b_{ii} = 1$ for i from 1 to n .

Step 4. While there is a blank in B do

Let a_{rs} be the highest value of the list $U(A)$.

If b_{rs} is blank,

Let $I = \{j; b_{rj} \text{ is not blank}\}$ and $I' = \{i; b_{is} \text{ is not blank}\}$.

Let $f = \min\{a_{ij}, i \in I, j \in I'\}$.

Set $b_{ij} = b_{ji} = f$ where $i \in I$ and $j \in I'$.

Delete the highest value from $U(A)$.

Example 4. Given the following proximity fuzzy relation:

$$A = \begin{pmatrix} 1 & 0,7 & 0,8 & 0,9 \\ 0,7 & 1 & 0,2 & 0,3 \\ 0,8 & 0,2 & 1 & 0,7 \\ 0,9 & 0,3 & 0,7 & 1 \end{pmatrix}$$

The algorithm is applied to compute a transitive opening B as follows.

Step 1: Set B to be blank

Step 2: Let $U(A)$ be the set of elements of the upper triangular matrix of A sorted in descending order. $U(A) = \{0,9; 0,8; 0,7; 0,7; 0,3; 0,2\}$.

Step 3: Set $b_{ii} = 1$ for all i.

Step 4:

The greatest value of $U(A)$, $a_{14} = 0,9$, is taken. Let $I = \{j; b_{1j} \text{ that are not blank values in matrix B}\} = \{1\}$ and let $I' = \{i; b_{i4} \text{ that are not blank in matrix B}\} = \{4\}$. The values $b_{41} = b_{14} = a_{14} = 0,9$ are computed in B.

$$B = \begin{pmatrix} 1 & & & 0,9 \\ & 1 & & \\ & & 1 & \\ 0,9 & & & 1 \end{pmatrix}$$

The next highest element in $U(A)$ is $0,8 = a_{13}$. $I = \{j; b_{1j} \text{ are not blank in B}\} = \{1, 4\}$ and $I' = \{i; b_{i3} \text{ is not blank in B}\} = \{3\}$ are defined and the values b_{13} , b_{43} and its symmetric values, having $b_{13} = b_{43} = \text{Min}\{a_{ij}, i \in I, j \in I'\} = \text{Min}\{0,8; 0,7\} = 0,7$ are computed in B.

$$B = \begin{pmatrix} 1 & & 0,7 & 0,9 \\ & 1 & & \\ 0,7 & & 1 & 0,7 \\ 0,9 & & 0,7 & 1 \end{pmatrix}$$

The next non-blank highest element in $U(A)$ is $0,7 = a_{12}$. $I = \{j; b_{1j} \text{ are not blank in B}\} = \{1, 3, 4\}$ and $I' = \{i; b_{i2} \text{ is not blank in B}\} = \{2\}$ are defined and the values b_{12} , b_{32} , b_{42} and its symmetric values, having $b_{12} = b_{32} = b_{42} = \text{Min}\{a_{ij}, i \in I, j \in I'\} = \text{Min}\{0,7; 0,3; 0,2\} = 0,2$ are computed in B.

$$\text{So } B = \begin{pmatrix} 1 & 0,2 & 0,7 & 0,9 \\ 0,2 & 1 & 0,2 & 0,2 \\ 0,7 & 0,2 & 1 & 0,7 \\ 0,9 & 0,2 & 0,7 & 1 \end{pmatrix} \text{ is a transitive opening of A.}$$

An easier aspect of the similarity is shown making boxes of subsimilarities after

applying the permutation $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix}$, then $P_{\pi}(A) = \begin{pmatrix} 1 & 0,9 & 0,8 & 0,7 \\ 0,9 & 1 & 0,7 & 0,3 \\ 0,8 & 0,7 & 1 & 0,2 \\ 0,7 & 0,3 & 0,2 & 1 \end{pmatrix}$

and a maximal transitive opening is

$$P_{\pi}(B) = \begin{pmatrix} \begin{pmatrix} 1 & 0,9 \\ 0,9 & 1 \end{pmatrix} & 0,7 \\ \begin{pmatrix} 0,7 & 0,7 \\ 0,2 & 0,2 \end{pmatrix} & 0,2 \end{pmatrix}$$

Example 5. Given the following proximity fuzzy relation:

$$A = \begin{pmatrix} 1 & 0,1 & 0,2 & 0,5 \\ 0,1 & 1 & 0,4 & 0,1 \\ 0,2 & 0,4 & 1 & 0,1 \\ 0,5 & 0,1 & 0,1 & 1 \end{pmatrix}$$

The algorithm is applied to compute a transitive opening B.

Step 1: Set B to be blank

Step 2: Let $U(A)$ be the set of elements of the upper triangular matrix of A, sorted in descending order. $U(A) = \{0,5; 0,4; 0,2; 0,1; 0,1; 0,1\}$.

Step 3: Set $b_{ii} = 1$ for all i.

Step 4:

The highest value of $U(A)$, $a_{14} = 0,5$ is taken t. Let $I = \{j; b_{1j} \text{ that are not blank values in matrix B} = \{1\}\}$ and let $I' = \{i; b_{i4} \text{ that are not blank in matrix B} = \{4\}\}$. The values $b_{41} = b_{14} = a_{14} = 0,5$ are computed in B.

$$B = \begin{pmatrix} 1 & & & 0,5 \\ & 1 & & \\ & & 1 & \\ 0,5 & & & 1 \end{pmatrix}$$

The following highest element in $U(A)$ is $0,4 = a_{23}$. $I = \{j; b_{2j} \text{ are not blank in B} = \{2\}\}$ and $I' = \{i; b_{i3} \text{ is not blank in B} = \{3\}\}$ are defined. It is computed in B the value b_{23} and its symmetric value, having $b_{23} = a_{23} = 0,4$.

$$B = \begin{pmatrix} 1 & & & 0,5 \\ & 1 & 0,4 & \\ & 0,4 & 1 & \\ 0,5 & & & 1 \end{pmatrix}$$

The following non-blank highest element in $U(A)$ is $0,2 = a_{13}$. $I = \{j; b_{1j} \text{ are not blank in } B\} = \{1, 4\}$ and $I' = \{i; b_{i3} \text{ is not blank in } B\} = \{2, 3\}$ are defined. The values $b_{12}, b_{13}, b_{42}, b_{43}$ and its symmetric values, having $b_{12} = b_{13} = b_{42} = b_{43} = \text{Min}\{a_{ij}, i \in I, j \in I'\} = \text{Min}\{0,1; 0,2; 0,1; 0,1\} = 0,1$ are computed in B .

$$B = \begin{pmatrix} 1 & 0,1 & 0,1 & 0,5 \\ 0,1 & 1 & 0,4 & 0,1 \\ 0,1 & 0,4 & 1 & 0,1 \\ 0,5 & 0,1 & 0,1 & 1 \end{pmatrix}$$

An easier view of the similarity is shown after applying the permutation $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix}$, then $P_\pi(A) = \begin{pmatrix} 1 & 0,5 & 0,2 & 0,1 \\ 0,5 & 1 & 0,1 & 0,1 \\ 0,2 & 0,1 & 1 & 0,4 \\ 0,1 & 0,1 & 0,4 & 1 \end{pmatrix}$ and a maximal transitive opening is

$$P_\pi(B) = \begin{pmatrix} \begin{pmatrix} 1 & 0,5 \\ 0,5 & 1 \end{pmatrix} & \begin{pmatrix} 0,1 & 0,1 \\ 0,1 & 0,1 \end{pmatrix} \\ \begin{pmatrix} 0,1 & 0,1 \\ 0,1 & 0,1 \end{pmatrix} & \begin{pmatrix} 1 & 0,4 \\ 0,4 & 1 \end{pmatrix} \end{pmatrix}$$

The following lemmas show that the previous algorithm gives a maximal transitive opening of a reflexive and symmetric fuzzy relation.

Lemma 5.1. The output of the Algorithm 1 applied to a reflexive and symmetric fuzzy relation is a fuzzy similarity relation

Proof. The proof is trivial from lemma 3.1 and lemma 3.2.

Lemma 5.2. Let A be a reflexive and symmetric fuzzy relation, and let B be the output of the previous algorithm applied to A .

If any fuzzy similarity relation H includes B and is included in A then it is B (for all similarity H , if $B \subseteq H \subseteq A$ then $H = B$).

Proof. Let $H = (h_{ij})$ be a fuzzy similarity relation such that $B \subseteq H \subseteq A$. So $b_{ij} \leq h_{ij} \leq a_{ij}$ for all i, j .

If $B \neq H$ then

$$\exists (r, s) \text{ such that } b_{rs} < h_{rs} \leq a_{rs}. \quad (2)$$

Let I, I' be the set of indexes given by the algorithm in the step in which b_{rs} is generated. Then $(r, s) \in I \times I'$.

As B is computed by the algorithm, b_{rs} is generated from the value of some a_{kl} .

$$\exists (k, l) \in I \times I' \text{ such that } b_{kl} = a_{kl} = f = \text{Min}\{a_{ij}, i \in I, j \in I'\} = b_{rs}. \quad (3)$$

H is transitive, so $h_{kl} \geq \max\{\min\{h_{kl}, h_{ll}\}, \dots, \min\{h_{kn}, h_{nl}\}\}$

$B \subseteq H \subseteq A$, and the values in the indexes $I \times I'$ are always lower than the values in $I \times I'$ and $I' \times I$, so

$$h_{kl} \geq \max\{\min\{h_{kl}, h_{ll}\}, \dots, \min\{h_{kn}, h_{nl}\}\} = \max\{h_{kj}, h_{il}\} \quad \forall i \in I, \forall j \in I'$$

For $i = r$ it is held that $h_{kl} \geq h_{rl}$, so

$$h_{rl} \geq \max\{\min\{h_{rl}, h_{ll}\}, \dots, \min\{h_{rm}, h_{ml}\}\} = \max\{h_{rj}, h_{il}\} \quad \forall i \in I, \forall j \in I'$$

In particular, for $j = s$, it is held that $h_{kl} \geq h_{rl} \geq h_{rs}$.

But $H \subseteq A$, so $h_{rs} \leq h_{kl} \leq a_{kl} = b_{kl} = b_{rs}$.

This is contradictory to (2).

Thus, any fuzzy similarity H such that $B \subseteq H \subseteq A$ verifies that $H = B$.

Therefore, the algorithm computes a maximal similarity from a reflexive and symmetric fuzzy relation, which is a transitive opening.

Lemma 5.3. A maximal transitive opening of a reflexive and symmetric fuzzy relation can be computed in $O(n^2 \log n)$ time in the worst case.

Proof. The computational complexity of the time consumed by the given algorithm is analysed as follows:

Step 2 sorts $\frac{n^2}{2} - n$ values, so it takes $O(n^2 \log n)$ time.

The loop in step 4 iterates at most $n-1$ times, so it iterates $O(n)$ times in the worst case.

When a dimension 1 box is added in an iteration, the maximum number of computed bridging elements is $n-1$, so the computation of the bridging elements takes $O(n)$ time in the worst case. Hence the total time spent in step 4 when all new boxes are of dimension one is $O(n^2)$ time in the worst case. When the new box has dimension k , then the computation is on $(n-1) \cdot k$ values in the worst case, which have the same computational complexity than computing sorting at most $(n-1)$ values k times, so the total computational complexity of step 4 is $O(n^2)$ time in the worst case.

Therefore, the complexity of the time consumed by the algorithm is $O(n^2 \log n)$ time.

6 Conclusions

Giving a proximity relation on a finite universe, there exists a unique transitive similarity (called transitive closure) that contains it and that is contained in any similarity that contains the proximity relation. It is well known how to compute the transitive closure of a fuzzy relation, but there exists several maximal similarities that are contained in the original proximity relation, that are called transitive openings.

An open problem is the computation of transitive openings, but in general there is not a unique solution, and so in general it is not possible to find the highest transitive opening a given fuzzy relation.

In this paper it is proven that in the case of reflexive and symmetric fuzzy relations (proximity relation) that there exists at least a transitive opening (a maximal similarity relation). An $O(n^2 \log n)$ time algorithm to compute a transitive opening of a prox-

imity relation is given, and it is proven that the output is maximal, showing that there are not transitive solutions in between the initial reflexive and symmetric fuzzy relation and the computed similarity.

Acknowledgment

This research is partially supported by the Spanish MCyT project BFM2002-00281.

References

1. Alsina, C., Trillas, E., Valverde, L. On some logical connectives for fuzzy set theory, *J. Math. Ann. Appl.* 93 (1983) 15-26.
2. De Baets, B. and De Meyer, H, Transitive approximation of fuzzy relations by alternating closures and openings, *Soft Computing* 7 (2003) 210-219.
3. Esteva, F, Garcia, P., Godo, L., Rodriguez, R. O., Fuzzy approximation relations, modal structures and possibilistic logic, *Mathware and Soft Computing* 5 (2-3) (1998) 151-166.
4. Dawyndt, P., De Meyer, H., De Baets, B. The complete linkage clustering algorithm revisited, *Soft Computing*, in press (available on-line).
5. Di Nola, A., Kolodziejczyk, W., Sessa, S. Transitive solutions of relational equations on finite sets and linear lattices. *Lecture Notes in Computer Science*, Vol.521 Springer, Berlin (1991) 173-182.
6. Dunn, J. C. Some recent investigations of a new fuzzy partitioning algorithm and its application to pattern classification problems, *J. Cybernet.* 4 (1974) – 5.
7. Garmendia, L., Campo, C., Cubillo, S., Salvador, A. A Method to Make Some Fuzzy Relations T-Transitive. *International Journal of Intelligence Systems*. Vol. 14, Nº 9, (1999) 873 – 882.
8. Garmendia, L., Salvador, A. On a new method to T-transitivize fuzzy relations, *Information Processing and Management of Uncertainty in Knowledge - based Systems*, IPMU 2000. (2000) 864 – 869.
9. Garmendia, L., Salvador, A. On a new method to T-transitivize fuzzy relations, in *Technologies for Constructing Intelligent Systems 2*, Springer. Edited by Bouchon-Meunier, B., Gutierrez-Rios, J., Magdalena, L., Yager, R. R, (2000) 251 – 260.
10. Guoyao Fu, An algorithm for computing the transitive closure of a fuzzy similarity matrix, *Fuzzy Sets and Systems* 5 (1992) 89– 94.
11. Hashimoto, H. Transitivity of generalised fuzzy matrices, *Fuzzy Sets and Systems*. Vol. 17, no. 1, (1985) 83-90.
12. Jacas, J., Similarity relations. The calculation of minimal generating families. *Fuzzy Sets and Systems* 35 (1990) 151-162.
13. Kandel, L. Yelowitz, Fuzzy chains, *IEEE Trans. Systems Man Cybernet.* 4 (1974) 472– 475.
14. Larsen H., R. Yager, “Efficient computation of transitive closures,” *Fuzzy Sets Syst.*, vol. 38 (1990) 81–90.
15. Lee, H.-S. An optimal algorithm for computing the max–min transitive closure of a fuzzy similarity matrix, *Fuzzy Sets and Systems* 123 (2001) 129–136.
16. Naessens, H., De Meyer, H., De Baets, B., Algorithms for the Computation of T-Transitive Closures, *IEEE Trans Fuzzy Systems* 10:4 (2002) 541-551.
17. Ovchinnikov, S. Representations of Transitive Fuzzy Relations, in *Aspects of Vagueness*, H. J. Skala, S. Termini and E. Trillas (Eds.), Reidel Pubs. (1984) 105-118.

18. Rodriguez, R. O., Esteva, F, Garcia, P., Godo, L. On implicative closure operators in approximate reasoning. *International Journal of Approximate Reasoning* 33 (2003) 159-184.
19. Wagenknecht, M., On pseudo-transitive approximations of fuzzy relations. *Fuzzy Sets and Systems* 44 (1991) 45-55.
20. Wagenknecht, M., On transitive solutions of fuzzy equations, inequalities and lower approximation of fuzzy relations. *Fuzzy Sets and Systems* 75 (1995) 229-240.
21. Xian Xiao, An algorithm for calculating fuzzy transitive closure, *Fuzzy Math.* 5 (4) (1985) 71-73.
22. Zadeh, L. A., Similarity relations and fuzzy orderings, *Inform. Sci.* 3 (1971) 177-200.

Generating Fuzzy Models from Deep Knowledge: Robustness and Interpretability Issues

Raffaella Guglielmann¹ and Liliana Ironi²

¹ Department of Mathematics - University of Pavia,
via Ferrata 1, 27100 Pavia, Italy

² IMATI - CNR, via Ferrata 1, 27100 Pavia, Italy

Abstract. The most problematic and challenging issues in fuzzy modeling of nonlinear system dynamics deal with robustness and interpretability. Traditional data-driven approaches, especially when the data set is not adequate, may lead to a model that results to be either unable to reproduce the system dynamics or numerically unstable or unintelligible. This paper demonstrates that Qualitative Reasoning plays a crucial role to significantly improve both robustness and interpretability. In the modeling framework we propose both fuzzy partition of input-output variables and the fuzzy rule base are built on the available deep knowledge represented through qualitative models. This leads to a clear and neat model structure that does describe the system dynamics, and the parameters of which have a physically significant meaning. Moreover, it allows us to properly constrain the parameter optimization problem, with a consequent gain in numerical stability. The obtained substantial improvement of model robustness and interpretability in “actual” physical terms lays the groundwork for new application perspectives of fuzzy models.

1 Introduction

System dynamics modeling goes through two main stages, namely structure identification and parameter optimization, that heavily account for robustness and interpretability. As for input-output approaches, structure identification deals with the reconstruction of functional relationships $f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ between the input-output variables from the available data samples only¹. The modeling problem is usually solved by first selecting an appropriate functional form for $f(\cdot)$ in a space known to hold good approximation properties; then, by identifying its parameters.

Fuzzy Systems (FS) have been proved to be excellent candidates for identification purposes [1, 2]: (i) they hold the universal approximation property, (ii) they are able to exploit the qualitative and uncertain a priori knowledge on the system dynamics, expressed by inferential linguistic information in the form of IF-THEN rules, and (iii) they are able to handle data samples. Structure identification requires to determine the fuzzy partition of input-output variables, and the set of rules to be used to generate $f(\cdot)$. Its parameters, which are tuned on the experimental data through optimization

¹ Without loss of generality, we consider here Multiple Input - Single Output systems.

procedures, are associated with the membership functions of input-output variables or, in other words, with the locations of their fuzzy partition. In theory, both partitions and inference rules can be derived by the expert knowledge, but such information may be very poor, irregular, and unstructured, and then, in practice, it prevents from defining the optimal form of $f(\cdot)$, where by optimal we mean that $f(\cdot)$ is of minimal complexity, but able to capture all of the significant features of the system dynamics. For these reasons, the research efforts turned to the definition of learning methods that automatically generate the fuzzy systems from the data samples only [3, 4]. Although these methods have been successfully applied to a variety of domains, they are affected by two serious drawbacks: the resulting nonlinear function is not understandable from a physical viewpoint, and it does not guarantee that the generalization property holds unless a large amount of samples is employed. Even when the resulting $f(\cdot)$ is abstracted from the expert knowledge, the same problems may occur since it is mostly built on empirical rather than deep knowledge, and then it may not capture important pieces of information about the system dynamics.

For a great deal of dynamical systems from different domains the incompleteness of available deep knowledge does prevent from formulating a quantitative differential model, but does not prevent from formulating a qualitative one. This consideration motivated our work aiming at the definition of a new approach, called FS-QM [5, 6]. Its novelty consists in the way the FS is built: both the fuzzy partition and rule base are defined on deep knowledge. FS-QM is applicable whenever the incompleteness of a priori knowledge is such that it allows us (i) to write a QSIM model [7], and (ii) to bound the uncertainty on landmark values to confidence intervals. In outline, the whole range of possible system dynamics, represented and simulated within the QSIM modeling framework, is automatically translated into the fuzzy formalism. Each variable domain is automatically partitioned into fuzzy sets in accordance with its associated quantity space, and with the prior information on landmark numerical bounds. The cardinality of the fuzzy partition of a variable and its membership function locations are defined by the cardinality of the set of qualitative values the variable may assume, and by the confidence intervals on its landmarks, respectively. Given a landmark-based fuzzy partition and the simulated behavior set, the Fuzzy Rule Base (FRB) is straightforward generated by mapping each behavior of the input/output variables into a set of rules, where each rule describes a transition from a qualitative state to the next one. The mathematical interpretation of the FRB explicitly initializes $f(\cdot)$ which is then refined through parameter estimation from data. Let us remark that the integration of qualitative and fuzzy methods has been explored by other authors [8, 9, 10] that address the problem of enriching a qualitative model by a fuzzy description of either quantities or functional relations with the purpose of model simulation rather than model identification.

Herein, we discuss a revised version of FS-QM that aims to cope with robustness and interpretability problems. To this end, we report results obtained by the application of FS-QM to model the intracellular Thiamine kinetics in the intestine tissue. The classical differential approach is here inapplicable for the incompleteness of the available knowledge and for the difficulty of gathering an adequate number of experimental data. The latter being the cause of the failure of traditional fuzzy approaches [6].

2 The Fuzzy Modeling Problem

The reconstruction of nonlinear system dynamics from data may be seen as a problem of modeling nonlinear discrete-time dynamical systems. Among the possible schemes to describe the dynamics of the output variable y [11], let us consider the following one:

$$y_k = f(\underline{x}_{k-1}, \underline{\theta}) + \epsilon_k \quad (1)$$

where the output y measured at time k is a function of a n -dimensional vector $\underline{x}$, which includes both output and input variables, measured at time $k - 1$. The function $f(\cdot)$ is unknown and expresses the functional relationship between the output and the input vector, $\underline{\theta}$ is the parameter vector, and the terms ϵ_k 's, independent, zero mean random variables, account for the measurement errors. Then, our problem consists in finding a continuous function approximator $\tilde{f}$ of f within a proper class of FS's. Herein, we consider the Mamdani's approach [12] as suitable to encode qualitative descriptions of input-output relations expressed through IF-THEN rules:

IF x_1 is F_1 and ... and x_n is F_n THEN y is F_y .

The antecedents x_i are the input vector components, the consequent y is the output, F_i and F_y are fuzzy sets characterized by a membership function $\mu_F : \mathbb{R} \rightarrow [0, 1]$. In accordance with the interpretation of fuzzy operators in [4], M rules are mathematically interpreted by the following fuzzy system:

$$\tilde{f}(\underline{x}, \underline{\theta}) = \frac{\sum_{j=1}^M \hat{y}_j [\prod_{i=1}^n \mu_i^j(x_i, \hat{\theta}_i^j)]}{\sum_{j=1}^M [\prod_{i=1}^n \mu_i^j(x_i, \hat{\theta}_i^j)]} \quad (2)$$

where the parameter $\hat{y}_j$ is the center of the μ that characterizes F_y in the j -th rule; μ_i^j , that depends on the parameter vector $\hat{\theta}_i^j$, characterizes F_i associated with x_i in the j -th rule; the vector $\underline{\theta}$ includes all of the parameters. In general, the problem of constructing $\tilde{f}$ goes through two sub-problems that should be separately solved to make the modeled system behavior easily interpretable and transparent:

1. Structural identification:

- For each variable, define its fuzzy partition, i.e. the μ 's that define the fuzzy values it may assume. The locations of the μ 's initialize $\underline{\theta}$;
- define both the optimal number M of rules and the rules;
- mathematically interpret the rules.

2. Parameter estimation: seek for

$$\underline{\theta}^* = \arg \min_{\underline{\theta}} \|\underline{y} - \tilde{\underline{y}}\| \quad (3)$$

where $\|\cdot\|$ is a proper norm, $\underline{y}$ and $\tilde{\underline{y}}$ are N -dimensional vectors of the measured data and of the computed values according to the model $\tilde{f}(\underline{x}, \underline{\theta})$, respectively.

2.1 Data-Driven Approaches

Recently, data-driven approaches have received more and more attention [1, 2, 3, 4, 13, 14, 15]. They mainly differ from each other in the way they perform parameter initialization and rule base generation. But, to define $\tilde{f}(\cdot)$, all of them follow, in outline, the flow given in Fig. 1: structure identification and parameter estimation are mutually related, and are performed within the same loop. The procedure loops on increasing model complexity till the obtained model meets a given criterion, such as a prespecified target accuracy or a model evaluation index. The initial model complexity may fix either the number L of partitions of each variable domain, often performed in accordance with clustering techniques [16, 17], or the number M of rules. In the former case, the domain is splitted into L regions, to which a μ is assigned; in the latter one, the μ 's are initialized around M data by directly locating their centers on the data themselves. The rules are generated by determining either all the possible combinations of the μ 's [1] or the combinations of those μ 's that identify regions where the data pairs get the maximum degrees [4]. Given a fuzzy structure, the parameter vector is optimized through a nonlinear estimation procedure.

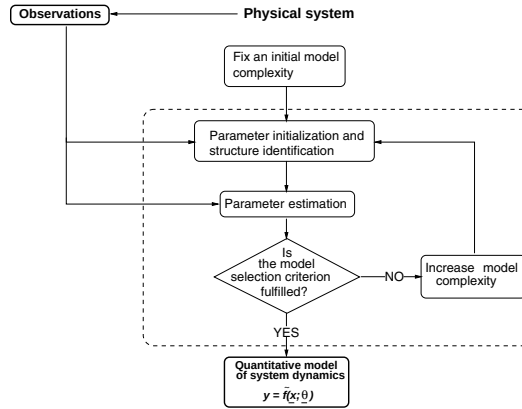


Fig. 1. Main steps in data-driven approaches

A - Robustness. The performance of these approaches strictly depends on both the number and the quality of data. As the data may be scattered and noisy, the completeness of the FRB might fail, and the model built on it, although optimal with respect to the fixed criterion, might not be able to capture the true underlying system dynamics. Moreover, if the input dimension or the number of rules are increased, the model complexity grows exponentially, and parameter estimation may become numerically untractable. This could lead to a sub-optimal model structure, and consequently to a model incapable of generalization. Moreover, if the parameter number is higher than the number of samples, overfitting phenomena may occur with a consequent loss of the ability to generalize [6]. In any case, the solution of the nonlinear parameter estimation problem (3) is critical: (i) the problem is ill-posed [18], (ii) the nonlinear optimization methods find

a local minimum that converges to the optimal value $\underline{\theta}^*$ when a “good” initialization $\underline{\theta}_0$ is given. Thus, the application of regularization techniques, that consist of constraining the model search space, and the definition of “good” partitions of input-output variables are a must to get both a stable and reasonable solution.

B - Interpretability. The way the fuzzy partitions and the rule base are built strongly influences interpretability. On the one hand, the variable partitions should be complete and distinguishable so that each of their subsets may be associated with a clear physical meaning. On the other hand, the rule base should be consistent, and made up of a reasonable number of rules: either contradictory rules or their combinatorial explosion, due to a too high dimension of input vector, makes the model very hard to be understood. Data-driven approaches may lead to incomplete partitions, inconsistent rules, and to an exponential growth of the number of rules and parameters [19]. But, even when these phenomena are suitably controlled, and consequently the initial model is interpretable, such a model feature may vanish after parameter adjustment. During the learning process, the parameters of the μ 's may be adjusted so drastically that the resulting fuzzy partition is not complete and distinguishable any more. Let us underline that, from the physical point of view, the interpretability potential of these approaches is, in general, rather weak, even when the identified model meets the conditions for it. As a matter of fact, the model parameters identify regions that do not necessarily correspond to physically significant descriptions of the system states.

3 FS-QM: A Qualitative Model-Based Approach

The method we propose, sketched in Fig. 2, clearly separates the structure identification phase from the parameter optimization one. FS-QM strongly exploits QSIM [7] to drive almost all modeling phases. QSIM provides both a formalism to formulate qualitative models of dynamical systems and a qualitative simulation algorithm. Qualitative simulation derives the set $\mathcal{B} = \{B_1, \dots, B_m\}$ of all the possible behaviors of a dynamical system modeled by a Qualitative Differential Equation (QDE) from an initial state $QS(t_0)$. A QDE describes a system in the same terms as an ordinary differential equation does, except that (i) the values of variables are qualitatively expressed, and (ii) the functional relationships between variables are described in terms of regions of monotonicity. The qualitative values of each system variable x are represented through landmark values: the real values the variable assumes are mapped into a finite ordered set Q_{L_x} , called *qualitative quantity-space*, whose elements are landmarks, l_k , and open intervals, (l_k, l_{k+1}) , bounded by two adjacent landmarks. A landmark is a symbolic name for a particular real number, whose value may be either unknown or uncertain, and defines regions where qualitative system properties hold. The landmark-based representation allows us to express incomplete knowledge about values of x as they are defined only by their order relations with the elements of Q_{L_x} . The set $\mathcal{B}$ is finite, and represented by a tree rooted in $QS(t_0)$. Each B_i is a finite sequence of qualitative states, linked by successor relations, that alternatively represents states in time-points $QS(t_k)$ and in time-interval $QS(t_k, t_{k+1})$:

$$B_i =: QS(t_0) \rightarrow QS(t_0, t_1) \rightarrow QS(t_1) \rightarrow \dots \rightarrow QS(t_{n-1}, t_n) \rightarrow QS(t_n) \quad (4)$$

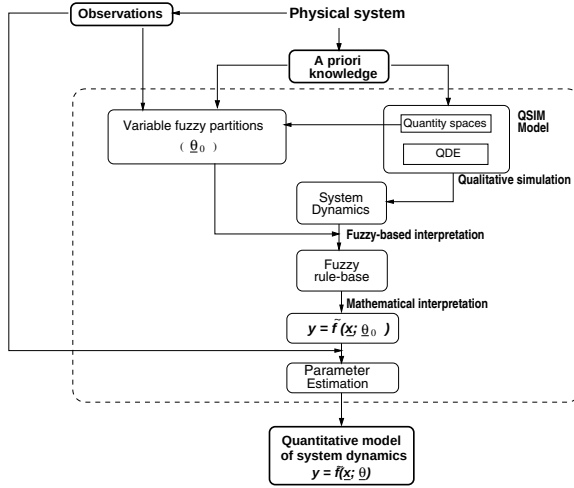


Fig. 2. Main steps in FS-QM

FS-QM builds the rules the model equation (2) is grounded on by encoding the qualitative value and state descriptions of the system dynamics, captured by each B_i defined in (4), into the fuzzy formalism. A crucial issue deals with the proper fuzzy representation of landmarks and intervals between them since it determines the variable fuzzy partition, or equivalently the initial value $\underline{\theta}_0$ of $\underline{\theta}$.

Variable Fuzzy Partition. Given the *qualitative quantity-space* Q_{L_x} of a generic variable x , let us assume that prior knowledge on the confidence interval $[a_k, b_k]$ on each $l_k \in Q_{L_x}$ is given. Let us call *fuzzy quantity-space* the finite ordered set Q_{F_x} , whose elements define the fuzzy partition of the domain of x . We define Q_{F_x} as image of a bijective mapping ν of Q_{L_x} . More precisely:

- $\nu(l_k) = F_{2k-1}$, characterized by $\mu_{F_{2k-1}}(x)$ with support (a_k, b_k) , and such that $\mu_{F_{2k-1}}(x) = 1$ in $x = c_k = (a_k + b_k)/2$;
- $\nu((l_k, l_{k+1})) = F_{2k}$, characterized by $\mu_{F_{2k}}(x)$ with support (c_k, c_{k+1}) , and such that $\mu_{F_{2k}}(x) = 1$ in $[b_k, a_{k+1}]$.

Figure 3 exemplifies how $Q_{L_x} = \{l_1, (l_1, l_2), l_2, (l_2, l_3), l_3\}$ is mapped into Q_{F_x} .

The fuzzy quantity-space Q_{F_x} , built for each variable, is associated with a parameter vector $\hat{\underline{\theta}}_0^{(l)} \in \mathbb{R}^{n_l}$, whose elements are the locations $\{a_k, c_k, b_k\}$ associated with all μ 's in Q_{F_x} . Then, the system parameter vector $\underline{\theta}_0$ is made up of $(n+1)$ vectors, i.e. $\underline{\theta}_0 = (\hat{\underline{\theta}}_0^{(1)}, \dots, \hat{\underline{\theta}}_0^{(n+1)})$ where n vectors are defined as above, and $\hat{\underline{\theta}}_0^{(n+1)}$ is made up of the centers of the μ 's of the output variable y .

The mapping ν defines a complete and consistent fuzzy partition, and states a one-to-one correspondence between the landmark-based and the fuzzy-based representation of real values. By definition, the μ 's have bounded supports: in this implementation, we have respectively chosen triangular and trapezoidal μ 's to represent landmarks and in-

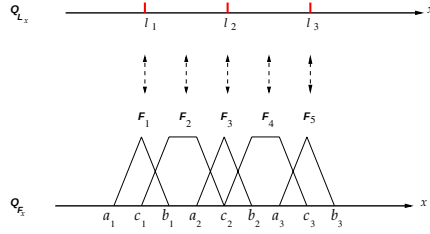


Fig. 3. Mapping Q_{Lx} into Q_{Fx}

tervals between them. But, due to the universal approximation theorem which holds for the considered class of FS's, other shapes could be chosen to represent the μ 's without affecting the approximation capabilities of the resulting fuzzy model.

Fuzzy rule generation. On the basis of the mapping ν , we can automatically translate the finite set of qualitative behaviors $\mathcal{B}$ generated by QSIM into fuzzy rules. First, the behavior tree is conveniently analyzed and preprocessed to filter out possible spurious behaviors, and to identify only behaviors representative of significant distinctions. The remaining Admissible Behaviors (AB) are translated into rules. In outline the algorithm, detailed in [6], maps each AB into a set of rules. In each rule, the antecedents and the consequent are the fuzzy representation of the qualitative value of all x_i at the current time, and of y at the next time, respectively. Thus, each rule gives a measure of the possible transition from one state to the next one. In this way, the entire range of possible system dynamics is embedded into the rule base. As it may happen that identical and/or conflicting rules are generated, filtering procedures are applied, and the final rule base results to be complete and consistent.

3.1 Robustness of FS-QM Models

Scattered and noisy data do not hamper the completeness of the rule base as it is generated from the QSIM model only. For the same reason, the rule base includes all of the possible state transitions. Then, we can surely assert that it is not sub-optimal. However, it may seldom happen to be redundant as we cannot prove that all spurious behaviors have been eliminated. But, a spurious rule is never instantiated by the data samples, and then its rule degree is equal to zero for all data pairs. Thus, it can be proved that such rules influence neither the approximation nor the generalization capabilities of the resulting model, although they may slightly reduce the computational efficiency. The number of parameters, independent of the number of rules and initialized on the basis of prior knowledge, grows linearly with the number of qualitative values, or equivalently with the number of variable partitions. This is an important feature of FS-QM as, rule number being equal, it builds a model with a significantly much smaller number of parameters than data-driven approaches. The reduced number of parameters together with a good initialization of both structure and parameter vector on deep knowledge, and not learned from data, results in its outperformance as for computational efficiency and generalization capabilities. This is still valid also when the number of available samples is not large: overfitting phenomena may very unlikely occur as demonstrated

in [6]. Although the model equation and the initial guess $\underline{\theta}_0$ have been built on structural knowledge, parameter estimation from data remains an ill-posed problem, and numerical instability may occur unless we further restrict the model search space. This can be done by imposing prior knowledge on the solution, namely by constraining either the function $f(\cdot)$ or the parameter vector $\underline{\theta}^*$ to belong, respectively, to a specific functional space or a specific trust region. Under the assumption that the prior knowledge used to define the initial estimate $\underline{\theta}_0$ is correct, we remove ill-posedness by constraining $\underline{\theta}^*$ to be in the neighborhood of $\underline{\theta}_0$. In practice, for each variable we constrain its associated parameter vector $\hat{\underline{\theta}}^{(l)}$ to belong to a “sufficiently small” region centred on $\hat{\underline{\theta}}_0^{(l)}$: $\mathcal{R} = \{\hat{\underline{\theta}}^{(i)} \in \mathbb{R}^{n_i}, \|\hat{\underline{\theta}}^{(i)} - \hat{\underline{\theta}}_0^{(i)}\| \leq \delta_i\}$; the δ_i ’s give a measure of the degree of confidence on the initial values of the parameters, i.e. on the prior knowledge about landmark values. The more certain the available knowledge on the initial locations of the μ ’s is, the smaller the region defined in the constraint is chosen. Then, the optimization problem (3) is reformulated as follows:

$$\underline{\theta}^* = \arg \min_{\underline{\theta} \in \mathcal{R}} \|\underline{y} - \tilde{\underline{y}}\| \quad (5)$$

We solve problem (5) by means of a classical optimization algorithm, namely Sequential Quadratic Programming [20]. The solution of the constrained problem is actually made stable, in the sense that “small” perturbations on data do not affect significantly the approximation properties of the identified model, and the estimated values of the parameters. To support this, we show the identification results obtained with two data sets, the second of which obtained from a perturbation of the first one, in the case of both unconstrained (Fig. 4) and constrained (Fig. 5) optimization. The perturbation, zero mean normally distributed random noise, is really a small quantity of the order of 10^{-7} .

3.2 Interpretability of FS-QM Models

To obtain an interpretable model two requirements have to be met: (1) the initial model, namely initial fuzzy partition and rule base, must be interpretable; (2) the model must remain interpretable after parameter estimation. For the way fuzzy partitions are defined by the mapping ν , i.e. (i) complete covering of the variable domain, and (ii) μ ’s

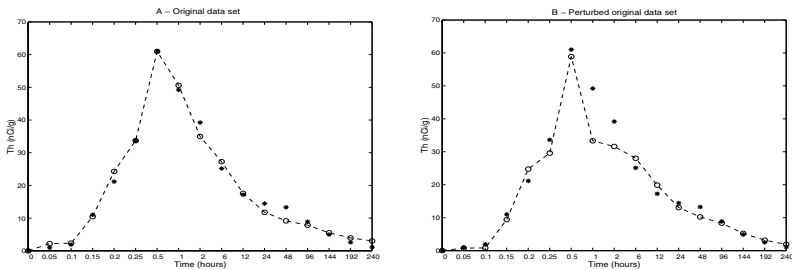


Fig. 4. Model identification with unconstrained parameter optimization: A - Original data; B - Original perturbed data

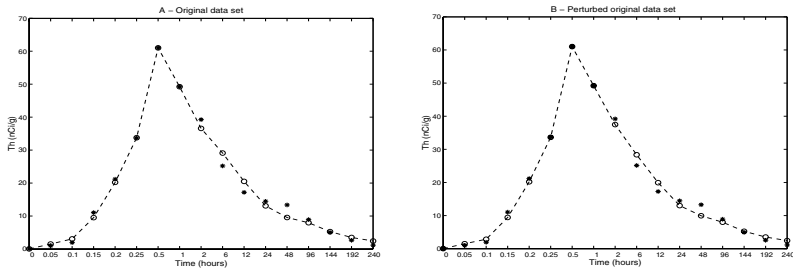


Fig. 5. Model identification with constrained parameter optimization: A - Original data; B - Original perturbed data

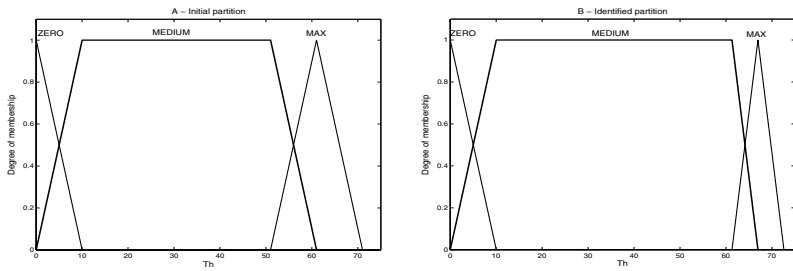


Fig. 6. Triangular/trapezoidal μ 's: A - Initial; B - Identified

with bounded supports, the conditions for interpretability, namely partition completeness and distinguishability, are guaranteed. Moreover, as the fuzzy partition of each variable domain is landmark-based, a sound physical meaning is associated with each fuzzy set and its parameters. The rule base results to be complete, because it embeds all the significant features of the system dynamics captured by the qualitative behaviors, and fully intelligible, as each rule expresses the transition of the system from a state to its successor. The rules are also consistent: in fact, possible conflicts between them, namely rules with the same IF-part and different consequent, are solved on the basis of the degree of each rule calculated on the data samples. Finally, let us remind that the rule base generation is grounded on the variable state transitions captured by the admissible behavior set, i.e., a physically significant subset of the simulated behavior tree. Then, the number of rules is linear with the number of such variable state transitions.

The parameter estimation procedure refines the numerical ranges to which landmark values are initially bounded. Interpretability of fuzzy partitions is preserved after parameter optimization thanks to the constraint defined in (5): as a matter of fact, such a constraint is able to keep the physical meaning of the parameters, and the consistency and separability properties initially ensured by the mapping ν (Fig. 6).

Let us emphasize that strong consistency and separability of the initial partition are necessary to guarantee the distinguishability of the identified μ 's. In fact, the regularization technique fails to keep weak consistency [21] as shown by Fig. 7-B. Figure 7-A

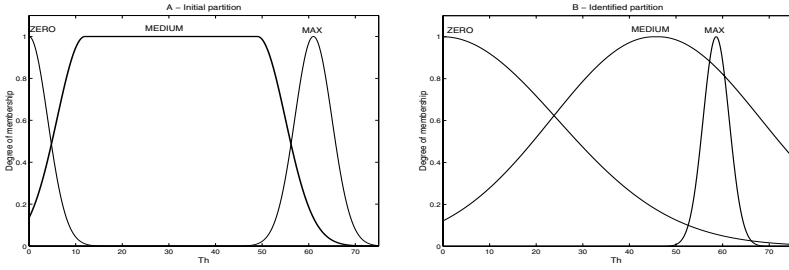


Fig. 7. Gaussian/double-gaussian μ 's: A - Initial; B - Identified

shows the initial partition of the same variable in Fig. 6 where gaussian/double gaussian μ 's are exploited to represent the knowledge about its landmarks. From Fig. 7-B it is evident that interpretability is definitely lost.

3.3 New Application Perspectives

The exploitation of deep knowledge rather than either shallow or empirical one makes a FS-QM model adequate to be applied in different contexts ranging from behavior prediction or simulation to automated reasoning tasks in knowledge-based systems. For example, we could exploit it in a diagnostic context either to classify observed system behavior or to test different hypotheses. The first task can be performed thanks to the clear physical meaning of the parameters of the μ 's, whereas hypotheses testing is more concerned with the rule base generation. In outline, the diagnostic classification problem may be tackled as follows. Let the initial model structure describing the system under normal conditions be the *nominal model*. When a new data set is available, parameter estimation is performed under the same conditions (same initial μ 's and width of the region $\mathcal{R}$ in (5)) used to identify the nominal model. If the optimization procedure succeeds in identifying an accurate model where the identified centers of the μ 's fall into their initial supports, we can infer that the data set is related to a normal situation. Otherwise, we can slightly relax the constraints on the parameters: if the optimization procedure succeeds, and again the centers of the identified μ 's belong to their initial supports, we can conclude that the data set refers to “quasi-normal” conditions. A further unsuccessful result means that the data set actually refers to a system corrupted by a fault or a disease, and prior knowledge on landmark values has to be updated accordingly, i.e. the initial fuzzy partitions have to be changed.

As exemplification, let us consider as nominal model that one related to Thiamine kinetics in normal patients [6]. We are given three different sets of data measured in as many different experimental settings. The first one, related to Thiamine kinetics in normal patients, has been already used in the paper to refine the parameter values of the nominal model (Fig. 6); the second and the third ones refer to insulin-treated and untreated diabetic subjects, respectively. Let us observe that the identified centers of the μ 's in Fig. 6-B do belong to their initial supports, as we expected. As for the second set of data, the optimization procedure fails to identify an accurate model under the same

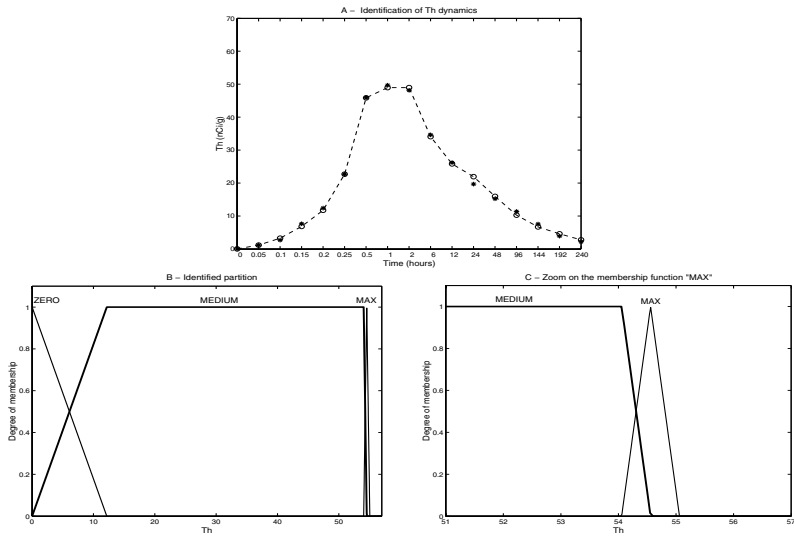


Fig. 8. Thiamine dynamics in treated diabetic subjects: A - Identification results; B - Identified partition; C - Zoom of identified partition

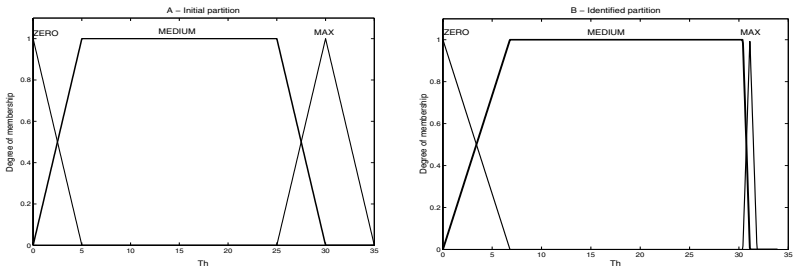


Fig. 9. Untreated diabetic subjects: Thiamine fuzzy partition. A - Initial; B - Identified

initial conditions, unless we enlarge the region $\mathcal{R}$. The results obtained are really satisfactory in terms of approximation accuracy (Fig. 8-A). The optimized μ 's in Fig. 8-B mainly differ from those in Fig. 6-B as for the fuzzy set labelled "MAX" zoomed in Fig. 8-C: we can observe that, even if the center moves away from its nominal value, it is still within the initial support of μ_{max} (Fig. 6-A). Also with the third data set FS-QM fails to approximate the system dynamics under the same initial conditions, and a significant enlargement of the region $\mathcal{R}$ reveals to be unsuccessful. Only a drastic change of the initial partitions (Fig. 9) allows us to get good results. Actually, this is due to the inadequacy of the prior knowledge on landmark values related to the physiological system to represent properly the pathological situation.

4 Conclusion

To achieve a robust and interpretable fuzzy model, FS-QM effectively employs all the available structural prior knowledge, represented in the QSIM formalism, and empirical data. The embedment of deep prior knowledge into the FS makes the identification problem better posed, since it properly delimits the model search space. In addition, the prior knowledge allows us to define a good initial estimate $\underline{\theta}_0$, and, then, to define a trust region where $\underline{\theta}^*$ is supposed to belong to. If the prior knowledge is correct, this will lead to a model that has good generalization and interpretability properties also in data-poor contexts. In FS-QM, the gained parameter interpretability from the physical point of view represents an added value that stands chances for fuzzy models to be used to perform a larger spectrum of tasks. FS-QM models might be conveniently applied, for example, in a diagnostic context. On the one hand, diagnostic hypotheses that explain the observed behaviors could be tested by introducing structural variations into the underlying qualitative model, and by validating the fuzzy model built on the basis of the newly generated rule base. On the other hand, diagnostic hypotheses could be drawn from the analysis of the deviations of the estimated values of parameters from the nominal ones. Future work will thoroughly explore the diagnostic potential of FS-QM models. A drawback of FS-QM deals with the possible, although infrequent, generation of a small number of spurious rules that might slightly reduce its computational efficiency, and interpretability of the rule base from a strictly physical point of view. The definition of a sufficient condition for a rule to be spurious is under investigation to further improve the overall performance of the method.

References

1. Jang, J.: Anfis: Adaptive network based fuzzy inference system. *IEEE Trans. on Systems, Man and Cybernetics* **23** (1993) 665–685
2. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on Systems, Man and Cybernetics* **15** (1985) 116–132
3. Wang, L., Mendel, J.: Generating fuzzy rules by learning from examples. *IEEE Trans. on Systems, Man and Cybernetics* **22** (1992) 1414–1427
4. Wang, L.: *Adaptive Fuzzy Systems and Control: design and stability analysis*. Englewood Cliff, NJ:Prentice–Hall, University of California at Berkeley (1994)
5. Bellazzi, R., Guglielmann, R., Ironi, L.: How to improve fuzzy-neural system modeling by means of qualitative simulation. *IEEE Trans. on Neural Network* **11** (2000) 249–253
6. Bellazzi, R., Guglielmann, R., Ironi, L., Patrini, C.: A hybrid input-output approach to model metabolic systems: An application to intracellular thiamine kinetics. *Journal of Biomedical Informatics* **34** (2001) 221–248
7. Kuipers, B.J.: *Qualitative Reasoning: modeling and simulation with incomplete knowledge*. MIT Press, Cambridge MA (1994)
8. D'Ambrosio, B.: *Qualitative process theory using linguistic variables*. Springer, New-York (1989)
9. Kim, G., Fishwick, P.: A validation method using fuzzy simulation in an object oriented physical modeling framework. In: *Proc. SPIE Aerosense Conference*, Orlando, Florida (1998)
10. Shen, Q., Leitch, R.: Fuzzy qualitative simulation. *IEEE Trans. on Systems, Man and Cybernetics* **23** (1993) 1038–1061

11. Ljung, L.: *System Identification - Theory for the User*. Prentice-Hall, Englewood Cliffs (1987)
12. Mamdani, E.: Applications of fuzzy algorithms for simple dynamic plant. *Proc. IEE* **121** (1974) 1585–1588
13. Abe, S., Lan, M.: Fuzzy rules extraction directly from numerical data for function approximation. *IEEE Trans. on Systems, Man and Cybernetics* **25** (1995) 119–129
14. Horikawa, S., Furuhashi, T., Uchikawa, Y.: On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm. *IEEE Trans. on Neural Networks* **3** (1992) 801–814
15. Pomares, H., Rojas, I., Gonzales, J., Prieto, A.: Structure identification in complete rule-based fuzzy systems. *IEEE Trans. on Fuzzy Systems* **10** (2002) 349–359
16. Bezdek, J.: *Pattern recognition with fuzzy objective function algorithms*. Plenum, New York (1981)
17. Sugeno, M., Yasukawa, T.: A fuzzy-logic based approach to qualitative modeling. *IEEE Trans. on Fuzzy Systems* **1** (1993) 7–31
18. Tikhonov, A., Arsenin, V.: *Solutions of ill-posed problems*. Winston, Washington DC (1977)
19. Jin, Y.: Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement. *IEEE Trans. on Fuzzy Systems* **8** (2000) 212–221
20. Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer, New York (1999)
21. Zeng, X., Singh, M.: Approximation accuracy analysis of fuzzy systems as functions approximators. *IEEE Trans. on Fuzzy Systems* **4** (1996) 44–63

Analysis of the TaSe-II TSK-Type Fuzzy System for Function Approximation

Luis Javier Herrera, Héctor Pomares, Ignacio Rojas, Alberto Guillén,
Mohammed Awad, and Olga Valenzuela

University of Granada, Department of Computer Architecture and Technology,
E.T.S. Computer Engineering, 18071 Granada, Spain
jherrera@atc.ugr.es
http://atc.ugr.es

Abstract. This paper reviews and analyzes the performance of the TaSe-II model, carrying out a statistical comparison among different TSK fuzzy system configurations for function approximation. The TaSe-II model, using a special type of rule antecedents, utilizes the Taylor Series Expansion of a function around a point to provide interpretability to the local models in a TSK approximator using a low number of rules. Here we will review the TaSe model basics and endow it with a full learning algorithm for function approximation from a set of I/O data points. Finally we present an ANOVA analysis about the modification of the different blocks that intervene in a TSK fuzzy model whose results support the use of the TaSe-II model.

1 Introduction

A Takagi-Sugeno-Kang (TSK) fuzzy model [1] consists of a set of K IF-THEN rules that typically have the form

$$\text{Rule}_k : \text{IF } x_1 \text{ is } MF_1^k \text{ AND } \dots \text{AND } x_n \text{ is } MF_n^k \text{ THEN } y = Y_k(\vec{x}) \quad (1)$$

where the MF_i^k are fuzzy sets characterized by membership functions $MF_i^k(x_i)$, x_i are the input variables and the consequents of the rules are not characterized by fuzzy sets but, as expounded by Buckley [2], by a general polynomial of the input variables $Y_k(\vec{x})$.

TSK fuzzy systems have demonstrated to be reasonably effective for function approximation problems [3], [4]. The TSK fuzzy model and its associated inference system have the advantage that the underlying model of the final designed system is transparent to the scientist/engineer designer. In particular, TSK Grid-Based Fuzzy Systems (GBFSs) [6] make use of a fixed group of fuzzy sets per input variable, thus improving the transparency of the resulting model. GBFSs provide a thorough coverage of the input space, thus being especially suited for control applications.

Nevertheless, GBFSs suffer from the curse of dimensionality problem, since the number of rules in this type of models is an exponential function of the number of input dimensions and the number of membership functions mf_i per input variable

$$K = \prod_{i=1}^n mf_i \quad (2)$$

Apart from this drawback, in general TSK fuzzy systems have associated a loss in interpretability of the local sub-models (rules) in relation to the global system output [5]. Furthermore, higher order polynomial consequents $Y_k(\vec{x})$, are typically associated with higher losses in interpretability.

The TaSe-II model, that was first presented in [6] and [20], overcomes both drawbacks by making use of second order polynomial consequent rules, thus keeping a low number of rules (comparing with zero or first order polynomials consequents), and making use of a specific rule antecedent and rule consequent pattern, that will allow us to interpret the output of each sub-model as the Taylor Series Expansion of the model output around each specific rule centre.

This paper analyzes the performance of the TaSe-II model, performing a statistical analysis ANOVA of the main parameters that intervene in the TSK fuzzy inference process. The multi-factorial ANalysis Of the VAriance (ANOVA) [7], consists of a set of statistical techniques that enable the analysis and comparison of experiments, by describing the interactions and interrelations between either the quantitative or the qualitative variables (called factors in this context) of the fuzzy system. The factors considered to be the most relevant variables in the design of a fuzzy system using the TSK models are: the aggregation operator, the type of membership function, the T-norm and the order of the polynomial rule consequents (order of the TSK system), considered for a number of different function examples and different system complexities. As we will see, the conclusions that can be drawn from this ANOVA analysis support the suitability of the TaSe-II model for function approximation.

The rest of the paper is organized as follows. Section 2 reviews the basics of the TaSe-II model. Section 3 presents a complete learning methodology for the TaSe-II model that includes parameter adjustment and structure identification. Section 4 briefly introduces the ANOVA statistical tool and Section 5 present and discuss the results drawn from the ANOVA analysis. Finally Section 6 summarizes the main conclusions of this paper.

2 The TaSe-II Model

The TaSe-II model makes use of second order TSK rules with a specific form in the consequents and in the antecedent part of the rules. Here we will review this methodology that obtains an interpretable and accurate model for function approximation.

2.1 TaSe-II Accuracy

The TaSe-II model makes use of TSK rules in the form shown in Eq. 1 with second order polynomial consequents

$$y = Y_k(\vec{x}) = w_0^k + \vec{w}_1^k \cdot \vec{x} + \vec{x}^T W_2^k \vec{x} \quad (3)$$

where w_0^k is the zero order coefficient, $\vec{w}_1^k$ is a vector of the first-order coefficients and W_2^k is a triangular matrix with the second-order coefficients.

We will make use of Least Squares in order to obtain the optimal rule consequents coefficients (given a fixed configuration of MFs per input variable) that minimize the error function

$$J = \sum_{m \in D} \left(y^m - F(\vec{x}^m) \right)^2 \quad (4)$$

where y^m is the desired output for point $\vec{x}^m$ in the data set D , and $F(\vec{x}^m)$ is the output of the approximator system using the weighted average aggregation strategy. The minimization process using the partial derivatives with respect to each of the coefficients leads to a linear equation system that can be solved using any of the many well-known mathematical methods for this purpose [8].

When facing a specific problem, it's arguable if it is more convenient to use a big amount of zero or first-order TSK rules than a few number of high-order rules. From the point of view of the number of rules it can be noticed the dramatic decrease that could be achieved using high-order TSK rules, thus increasing the transparency of the model. Nevertheless, from the computational-complexity point of view, the difference in cost for a given error tolerance when performing the approximation is not obvious; this is one of the aims of the study presented in this paper. Finally, from the interpretability point of view, traditionally zero-order fuzzy rules have been chosen due to its easier understandability. However, as the number of fuzzy rules increases, the understandability by a human operator of the whole rule-base is diminished. High-order TSK rules are commonly regarded as non-interpretable [5] but, as we will see in the next subsection, under certain conditions it is possible to endow this class of fuzzy rules with interpretability, thus gathering both the low number-of-rules property and the interpretability property in the same fuzzy system.

2.2 TaSe-II Interpretability

The Taylor Theorem states that given a function $f(\vec{x})$ defined in an interval that has derivatives of all orders, it can be approximated near a point $\vec{x} = \vec{a}$ as its Taylor Series Expansion around that point

$$f(\vec{x}) = f(\vec{a}) + (\vec{x} - \vec{a})^T \left[\frac{\partial f}{\partial \vec{x}_i}(\vec{a}) \right]_{i=1..n} + \frac{1}{2} (\vec{x} - \vec{a})^T W (\vec{x} - \vec{a}) + \dots + \xi \quad (5)$$

where ξ is an error expression that depends on a point $\vec{c}$ between $\vec{x}$ and $\vec{a}$. Note that the rule consequent form in Eq. 4 corresponds to the truncated Taylor Series Expansion of order 2; but some requirements have to be carried out in order to be able to interpret the rule consequents as the truncated Taylor Series Expansion of the global output of the system around the rule centers.

These requirements are on one hand the derivability of the model output, which is the requirement to apply the Taylor Theorem to a function around a point. And on the

other hand, the function output at each rule centre must be uniquely affected by its respective rule; this way, the polynomial expression of the rule consequent can be seen as the Taylor Series Expansion of the model output around the rule centre. Specifically, the OLMF basis of order p [9] have a set of general characteristics that comprise these two requirements and that will allow this interpretability:

- all MFs should be local, defined in a delimited domain and of the same shape
- every MF extreme point should coincide with the centre of the adjacent MF (they form a partition, thus avoiding a low level of overlapping of the MFs that would cause lack of transparency of the input space [10])
- all MFs are p times differentiable and the p -th derivative of the MF is continuous in all its domain
- the p -th derivative of the MF vanishes at its centre and at its boundaries
- the basis must accomplish the addition-to-unity property [11]

Marwan Bikdash in [9] presented the OLMF basis and a model that used the prior knowledge in the form of the derivatives values of the function to be approximated in some specific points, placing the rules at these specific points and using the derivatives to build the rule consequents, obtaining thus a strongly interpretable model. The interpretability of the model is verified using the following theorem by the same author:

“Given a complete TSK rule-base fuzzy system, where

1. the antecedent part of the rules are a grid-based fuzzy system with OLMF basis of order p for every input dimension and

2. the consequent-side of the rules is written in the rule-centered form shown in Eqs. (1) and (5), being $Y_k(\vec{x})$ polynomials of degree p').

then for $p' \leq p$, every $Y_k(\vec{x})$ can be interpreted as a truncated Taylor Series Expansion of order n of the output of the fuzzy system $F(\vec{x}^m)$ about the point $\vec{x} = \vec{a}$, the centre of the k -th rule”.

Note that the same theorem can be applied when the OLMF bases are used but the coefficients are not provided, but directly calculated. The TaSe-II model directly obtains the optimal rule consequents coefficients using Least Squares, obtaining in fact a model with the same interpretability properties and with better approximation than Bikdash’s model, since the directly provided (when possible) derivatives values are not optimal in the Least Squares sense, and hence, the system output barely represents a good approximation of the data we are modeling. Also, the TaSe model admits as we will see, a more complete learning algorithm to obtain the optimal rule centers positions (parameter adjustment) and the optimal number of membership functions per variable in the GBFS (structure identification) given an I/O dataset. It must be noted that there are other recent approaches that also try to overcome the lack of interpretability of the local models in the subdivision of the input space, by using a specific type of MF with similar properties to OLMF Bases as in [12].

3 Learning Methodology

We already mentioned how to obtain the optimal rule consequent coefficients for the TaSe-II model using Least Squares. The learning methodology for the TaSe is now completed with an automatic algorithm for obtaining the optimal subdivision of the input space (structure identification) and the optimal position of the MF centers (that are the single parameters of the MFs to be optimized due to the characteristics of the OLMF bases).

3.1 Parameter Adjustment

A gradient descent approach has been implemented (the Levenberg-Marquardt Algorithm, LMA [13]) to identify the optimal MF centers. Nevertheless, gradient descent approaches have the main drawback that there are high chances of falling in a bad local minimum. In order to avoid this problem, we have implemented the Error Equidistribution Method [14] in order to provide the LMA a good starting centers configuration to obtain a pseudo-optimal centers configuration.

3.2 Structure Identification

Structure identification, as explained before, deals with finding the optimal subdivision of the input space in order to obtain the best performance according to the I/O training data provided. In the case that we have noisy data, the structure identification approach should care about the well-known overfitting problem. A way to do this is to use a validation dataset that can be obtained by splitting the training dataset into both training and validation datasets, thus controlling the complexity-performance rate of the model. In this paper we are comparing different models having a similar complexity in number of parameters and thus, the structure identification process here will have to deal with finding the best system structure that has a number of parameters similar to the established limit.

Therefore a possible structure identification approach is to, starting with the simplest possible structure having one MF per input variable, iteratively add new MFs to the input variable chosen in each step, while the limit in the number of parameters is not exceeded. In every step of the iterative process, an identification method has to be performed in order to select the optimal input variable in which the new MF is added. Then the structure identification sub-algorithm will perform parameter adjustment using the available dataset in order to obtain a pseudo-optimal system. The identification process will simply test every possible new configuration that results from adding one MF on each input variable, and it will retain that configuration where the error is lower. This approach is possible since we have provided an efficient and automatic parameter adjustment algorithm that obtains the optimal rule consequent coefficients and a set of pseudo-optimal MF centers. This method is inefficient, specially for TSK systems with high number of rules, but very appropriate for higher order rules due to the low number of rules that will arise for a medium complexity problem and that a careful process has to be done since there is a high increase in the number of parameters when a single MF is added to an input variable. Despite this lack of efficiency

this is the method that we will apply in this paper for the sake of fairness in the comparisons, for all the possibilities of model architectures that will be analyzed using the ANOVA tool, briefly reviewed in the following section.

As an example, Fig. 1 shows the approximation obtained for the example function in f_1 taken from [17] (see Section V) as well as the interpretability of the local sub-models obtained using the TaSe model with 3 MFs per variable. See the high level of similarity of the local models in the vicinity of the centers of the respective rules in relation to the system output; not only in their functional values, but also in their trends (first and second derivatives).

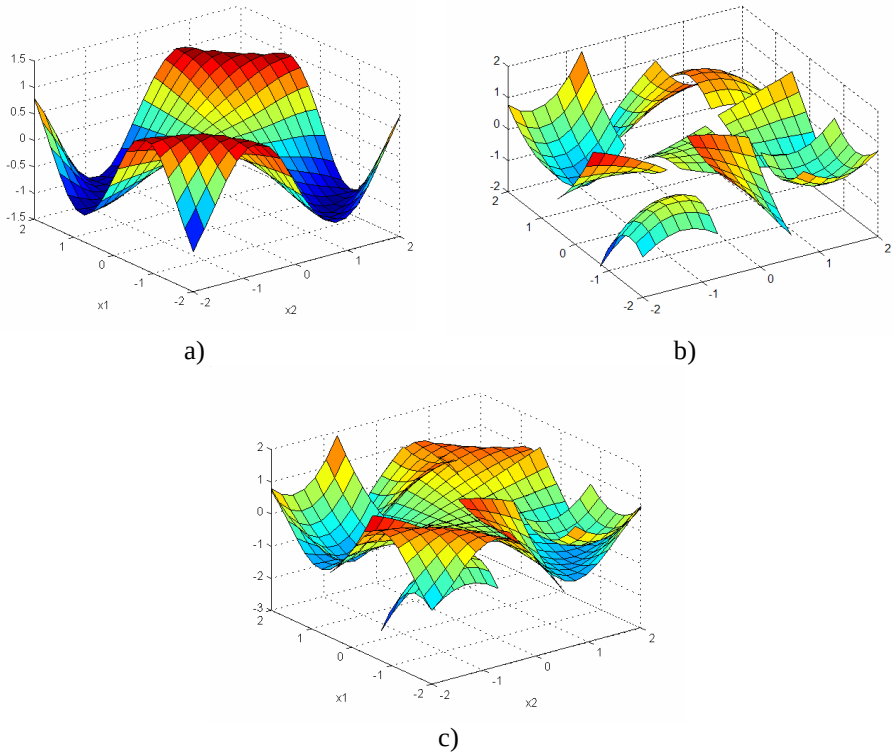


Fig. 1. a) Original function f_1 . b) Rule consequent polynomials of the 9 rules in the model with a 3×3 MF distribution. c) Output of the model with 3MFs per variable, i.e. with 9 rules combined with the rule consequent polynomials

4 The ANalysis Of VArance (ANOVA)

The ANalysis of VArance (ANOVA) theory and methodology were mainly first developed by R.A. Fisher [7]. It consist of a set of statistical techniques that allow to analyze and compare experiments in which there is a quantitative one-dimensional

output described as a function of a set of qualitative or quantitative variables called factors. ANOVA is used to estimate the direct and interaction effects that the factors have on the output. A direct effect takes place when there is a direct relation between the factor and the output; an interaction effect takes place when there is a combined relation of a number of factors with the output.

In order to apply the ANOVA tool, the values of the output variable of the model are expressed as an additive function of a set of components [15]. In general, if we have multiple factors, the observation $Y_{a,b,...,l;s}$ where

- $a, b, ..., l$ are the different variables or factors of the problem,
- s is the number of observations when there is repetition of real experiments over the same conditions or values of the factors,

may be admitted, in a first approximation, to be an additive consequent of the effects of the factors $a, b, ..., l$. Therefore the observation can be expressed as a lineal sum

$$Y_{a,b,...,l;s} = a + b + ... + l + \varepsilon \quad (6)$$

Being ε the effect of causes not assignable to the factors. If the factors $a, b, ..., l$ are kept constant, the values of $Y_{a,b,...,l;s}$ will present variations that should be considered to be ascribed to a big number of little causes indistinguishable among themselves, that is what is called random variation [16]. If we have an experiment in which we measure $Y_{a,b,...,l;s}$ at different levels of one or more factors, the set of measures obtained could be not-homogeneous, being compounded by two or more groups. The purpose of the technique introduced by Fisher is to contrast that heterogeneity, to check if those factors are really assignable causes of the variation that is under study or, if it should be ascribed to a random effect. That is, ANOVA is about separating the components of the variation that appears in a set of statistical data, determining if the discrepancy among the factors means are higher than what could be reasonable expectable for the variations that take place inside the factors.

Due to the abundant bibliography about TSK fuzzy systems sometimes with contradictory results about the order of the polynomial consequent, together with other parameters as the aggregation operator, the type of membership function, the T-norm, etc., in this article we will use the ANOVA tool to perform an analysis of the performance of a TSK fuzzy model when the different possibilities of the operators involved in the design of a fuzzy model are modified. As we will see, the ANOVA results will show the convenience of the use of the TaSe-II model due to its reduction in the number of rules and the maintenance of the interpretability characteristics.

5 ANOVA Results

5.1 Problem Specification

In this section we introduce the different example problems that have been used in the ANOVA analysis. We have selected a wide range of function approximation problems in order to provide enough experimental support to the results drawn from the

ANOVA analysis. The set of selected function approximation problems has been chosen from two previous works: in [17], Cherkassky et al. presented a comparison of different methodologies for function approximation using a set of 13 significant functions; another work by Rovatti et al. [18], presented a structure identification approach for fuzzy systems that is evaluated using five bi-dimensional example functions. In this work we will perform our analysis using the first five functions that Cherkassky used for his comparison ($f_1, \dots, f_5$) and the five functions used by Rovatti ($y_1, \dots, y_5$).

5.2 Results of the Analysis

To carry out the statistical study, a selection is made from a set of alternatives that are representative of each of the factors to be considered in the design of a TSK fuzzy system. In this study, the factors considered are: the aggregation operator, the type of membership function, the T-norm and the order of the TSK model, for different function examples and a number of levels of complexity of the system. By analyzing the different levels of each of these factors it is possible to determine their influence on the performance of a TSK fuzzy system. The response variable used to perform the

Table 1. Variables used in the statistical study

	Factors considered	Example function to approximate	Aggregation operator	Type of MF	T-Norm	Conseq. of the TSK rules	Complexity
Levels of the factors	Level 1	f_1 [17]	Weighted Average	OLMF	Product	Order 0	Low (54 params)
	Level 2	f_2 [17]	Weighted Sum	Triangular	Min	Order 1	Medium (120 params)
	Level 3	f_3 [17]		Gaussian	Hamacher $\lambda = 0.25$ [19]	Order 2	High (180 params)
	Level 4	f_4 [17]			Hamacher $\lambda = 0.5$ [19]		
	Level 5	f_5 [17]			Hamacher $\lambda = 0.75$ [19]		
	Level 6	f_6 [17]					
	Level 7	f_7 [17]					
	Level 8	f_8 [17]					
	Level 9	y_1 [18]					
	Level 10	y_2 [18]					
	Level 11	y_3 [18]					
	Level 12	y_4 [18]					
	Level 13	y_5 [18]					

Table 2. Analysis of Variance for Error using the “Mean Square Error” measure

Factors	Sum of Squares	DF	Mean Square	F-ratio	Sig.level
Functions	5,83	12	0,49	119,55	0,0000
Aggregation	2,05	1	2,05	503,44	0,0000
Type of MF	0,60	2	0,30	73,55	0,0000
T-Norm	2,60	4	0,65	159,93	0,0000
TSK Order	2,21	2	1,11	272,16	0,0000
Complexity	3,18	2	1,59	391,79	0,0000

statistical analysis is the mean square error in the output of the fuzzy system, when the factors considered change. *Table 1* gives the different levels considered in each factor when carrying out the multi-factorial ANOVA analysis. Therefore, all the possible configurations of factors used are evaluated for each of the different experiments considered. In order to make a comparison of all of these and to reach conclusions regarding the whole set, the error index is normalized in the range [0,1].

Table 2 gives the six-way variance analysis for the whole set of examples of the fuzzy system studied, using for all of the possibilities the algorithm presented in Section 3. The ANOVA table containing the sum of squares, degrees of freedom, mean squares, test statistics, etc., represents the initial analysis in a compact form. This kind of tabular representation is customarily used to set out the results of the ANOVA calculations.

Of all the information presented in the ANOVA table, the major interest of the researcher will most likely be focused on the value located in the “F-Ratio” and “Sig. Level” columns. If the numbers found in this column are less than the critical value (α) set by the experimenter, then the effect is said to be significant. Since this value is usually set at 0.05, any value less than this will result in significant effects, while any value greater than α this value will result in non-significant effects. If the effects are found to be significant using the above procedure, it implies that the means differ more than would be expected by chance alone. In terms of the above experiment, it would mean that the treatments were not equally effective. This table does not tell the researcher anything about what the effects were, just that there most likely were real effects. As can be seen from *Table 2* all the main factors are statistically significant, but it is important to note that the order of the TSK rules is a very important factor (that will be analyzed afterwards with the Multiple Range Test) together with the aggregation operator.

Let’s now make a detailed analysis of each of the factors examined. For the aggregation operator, *Fig. 2a* shows the means error differences for the two levels considered. It can be clearly noted that when using weighted average, the mean error is much lower than when using weighted sum. There is a case for which both weighted sum and weighted average have the same result, which is when the addition-to-unity property takes place, i.e. the cases in which the T-norm operator corresponds to the Product operator, both for OLMF basis and for triangular partition.

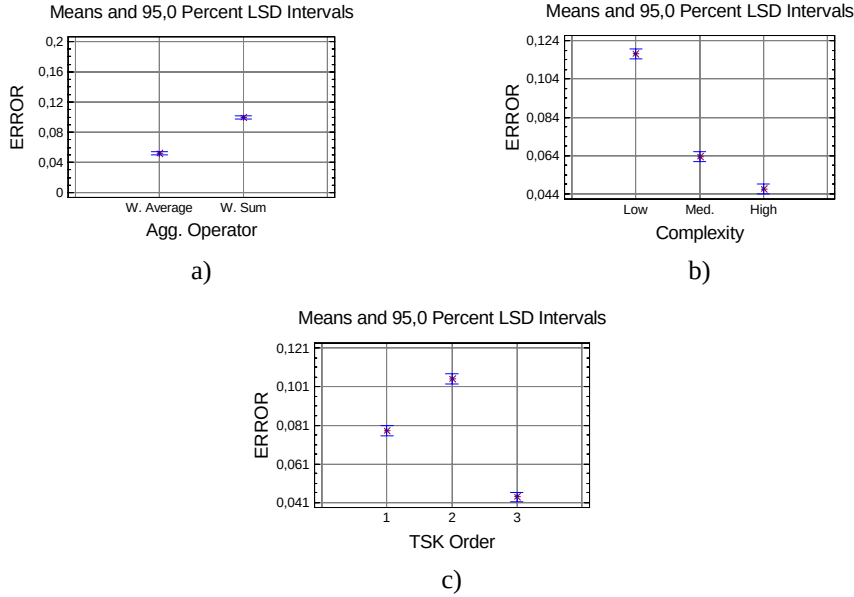


Fig. 2. a) Levels of the “Aggregation” factor and 95% Least Significant Difference (LSD) interval. b) Levels of the “Complexity” factor. c) Levels of the “TSK Order” factor

The factor “Complexity” is the next in importance; it is obvious that the higher the number of parameters used to perform the approximation the more precise the approximation is. This result can be confirmed in Fig. 2b.

In relation to the factor “TSK Order”, which refers to the degree of the polynomial consequent of the TSK rules, it is shown that the performance of the systems that use second order polynomial consequents is better in average than those that make use of first or zero-order TSK rules (see Fig. 2c). The first order TSK rules model present in average the highest error, with little difference with respect to zero order TSK rules model. We must highlight again that equality in number of parameters to optimize leads constant consequent rule systems to have a higher number of fuzzy rules than those that make use of first or second order TSK rules. This result points out that the exploitation of the parameters when using second order rules, is higher than when using lower order rules. Historically, higher order TSK rules haven’t been used due to the decrease in interpretability that it entails comparing with lower order TSK rules; but this barriers can be eliminated by using the TaSe methodology as expounded in the previous sections.

This result supports the utility and importance of using the proposed TaSe-II fuzzy system with Taylor Series Based Rules, since for a similar complexity, the number of rules needed is much lower, with better approximation accuracy, and additionally keeping the interpretability of the rules.

In relation to the rest of the factors: “MFS”, that refers to the type of Membership Functions used in the fuzzy system, the “T-Norm” operator and the factor “Functions”, the importance of these is much lower than those already commented.

6 Conclusions

This paper we have reviewed and analyzed the performance of the TaSe-II model, carrying out a statistical comparison among different TSK fuzzy system configurations for function approximation. For that purpose, we have performed an statistical analysis ANOVA (ANalysis Of VAriance [7]), in which we have studied the behavior of a fuzzy inference system when the different component blocks of the fuzzy inference model are modified (the aggregation operator, the type of membership function, the T-norm and the order of the TSK model), for different function examples and a number of levels of complexity of the system. The ANOVA results support the suitability of the TaSe-II model for function approximation using the learning algorithm proposed for this model.

Acknowledgements

This work has been partially supported by the Spanish CICYT Project TIN2004-01419.

References

1. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Syst. Man and Cyber.*, vol.15, (1985) 116-132
2. Buckley, J.J.: Sugeno-type Controllers are Universal Controllers, *Fuzzy Sets. Syst.*, vol.25, (1993) 299-303
3. Pomares, H., Rojas, I., González, J., Prieto, A.: Structure Identification in Complete Rule-Based Fuzzy Systems, *IEEE Trans. Fuzz. Vol.10 N. 3* (2002) 349-359
4. Wang, L.X.: Adaptive fuzzy systems and control. Design and stability analysis, Englewood Cliffs, N.J. Prentice Hall, (1994)
5. Yen, J., Wang, L., Gillespie, C.W., Improving the Interpretability of TSK Fuzzy Models by Combining Global Learning and Local Learning, *IEEE Trans. Fuz. Syst. Vol.6 No.4*, (1998) 530-537
6. Herrera, L.J., Pomares, H., Rojas, I., González, J., Valenzuela, O.: Function Approximation through Fuzzy Systems Using Taylor Series Expansion-Based Rules: Interpretability and Parameter Tuning, In: Mexican Int. Conf. Art. Intell. MICAI'2004, Lecture Notes in Computer Science, Vol. 2972. Springer-Verlag, Berlin Heidelberg New York (2004), 508-516
7. Fisher, R.A.: Contribution to Mathematical Statistics, New York: Wiley, (1950)
8. Golub, G., Loan, C.V.: Matrix Computations, The Johns Hopkins University Press, Baltimore, (1989)
9. Bikdash, M.: A Highly Interpretable Form of Sugeno Inference Systems, *IEEE Trans. Fuz. Syst. Vol.7 No.6*, (1999) 686-696.
10. Guillaume, S.: Designing fuzzy inference systems from data: an interpretability-oriented review, *IEEE Trans. Fuzz. Syst.*, vol. 9, (2001) 426-443
11. Ruspini, E.H.: A new approach to Clustering Info Control no.15, (1969) 22-32
12. Zhou, S.M., Gan, J.Q.: Improving the interpretability of Takagi-Sugeno fuzzy model by using linguistic modifiers and a multiple objective learning scheme. In : Int. Joint Conf. on Neural Networks IJCNN2004, (2004) 2385-2390

13. More, J. J. : The Levenberg-Marquardt algorithm implementation and theory, Lecture Notes in Mathematics, Vol. 630, (1978) 105-116
14. Pomares, H., Rojas, I., Ortega, J., Gonzalez, J., Prieto, A.: A Systematic Approach to a Self-generating Fuzzy Rule-table for Function Approximation, IEEE Trans. Syst. Man and Cyber., Vol.30, no.3, (2000) 431-447
15. Montgomery, D.C., Runger, G.C.: Applied Statistics and Probability for Engineers. John Wiley and Sons, 2nd edition, (1999)
16. DeGroot, M. H.: Probability and statistics. Menlo Park: Addison-Wesley Publishing Company, 2nd edition (1988)
17. Cherkassky, V., Gehring, D., Mulier, F.: Comparison of adaptive methods for function estimation from samples, IEEE Trans. Neural Net., Vol.7, no.4, (1996) 969-984
18. Rovatti, R. Guerrieri, R.: Fuzzy Sets of Rules for System Identification, IEEE Trans. Fuzz. Syst., Vol. 4, no. 2, (1996) 89-102
19. Fodor, J.C., Keresztfalvi, T.: A characterization of the Hamacher family of t-norms, Fuzzy Sets and Systems 65 (1994) 51-58
20. Herrera, L.J., Pomares, H., Rojas, I., Valenzuela, O., Prieto, A.: "TaSe, a Taylor Series Based Fuzzy System Model that Combines Interpretability and Accuracy", Fuzzy Sets Syst., *Accepted*

Non-deterministic Semantics for Paraconsistent C-Systems*

Arnon Avron

School of Computer Science, Tel-Aviv University,
Ramat Aviv 69978, Israel
aa@math.tau.ac.il

Abstract. We provide non-deterministic semantics for the 3 basic paraconsistent C-systems **C** (also known as **bC**), **Ci**, and **Cia**, as well as to all 9 extensions of them by one or two of the schemata **(I)** $\neg(\varphi \wedge \neg\varphi) \supset \circ\varphi$ and **(e)** $\varphi \supset \neg\neg\varphi$. This includes da Costa's original C_1 (which is equivalent to **Cila**). Our semantics is 3-valued for the systems without **(I)**, and infinite-valued for the systems with it. We prove that these results cannot be improved: neither of the systems without **(I)** has either a finite characteristic ordinary matrix or a two-valued characteristic non-deterministic matrix, and neither of the systems with **(I)** has a finite characteristic non-deterministic matrix. Still, our semantics suffices for providing decision procedures for all the systems investigated.

1 Introduction

One of the most difficult problems concerning reasoning with uncertainty is that of contradictory data. The following quotation from [14] describes the problem as follows:

“It is a fact of life that large knowledge bases are inherently inconsistent, in the same way large programs are inherently buggy. Moreover, within a conventional logic, the inconsistency of a knowledge base has the catastrophic consequence that $\perp$ is derivable from the knowledge base.”

It follows that in order to handle inconsistent knowledge bases in a reasonable way, one needs an $\perp$ -logic: a logic which allows nontrivial inconsistent theories. Logics of this sort are called $\perp$ -logics.

There are several approaches to the problem of designing a useful paraconsistent logic (see e.g. [6, 5, 11]). One of the oldest and best known is da Costa's approach ([12]), which seeks to allow the use of classical logic whenever it is safe to do so, but behaves completely differently when contradictions are involved. Da Costa's approach has led to the family of logics of formal (in)consistency

* This research was supported by THE ISRAEL SCIENCE FOUNDATION founded by The Israel Academy of Sciences and Humanities.

([10]). This family is based on two main ideas. The first is that propositions should be divided into two sorts: the “normal” (or consistent) propositions, and the “abnormal” (or inconsistent) ones. Classical logic can (and should) be applied freely to normal propositions, but not to abnormal ones. The second idea is to formally introduce this classification into the language, by employing a special (primitive or defined) unary connective $\circ$, where the intuitive meaning of $\circ\varphi$ is : “ φ is consistent”. Systems which provide such a connective, and in addition satisfy some certain basic conditions (to be described below) are known as *C*-systems ([9]).

The class of *C*-systems is the most important and useful subclass of the class of logics of formal (in)consistency. So far, the main shortcoming of this class has been the lack of an intuitive semantics for logics in it, which would be easy to use and would also provide real insight into these logics.¹ In this paper we remedy this by providing simple and modular bivaluations semantics for the 12 most basic *C*-systems (including da Costa’s original C_1). Our semantics is based on the use of non-deterministic matrices (Nmatrices). These are multi-valued structures where the value assigned by a valuation to a complex formula can be chosen non-deterministically out of a certain nonempty set of options. Nmatrices were introduced in [1, 2, 3], and it was shown there that while the use of finite Nmatrices has the benefit of preserving all the advantages of logics with ordinary finite-valued semantics (in particular: decidability and compactness), they are applicable to a much larger family of logics. In general, Nmatrices seem to be particularly useful for reasoning on uncertainty because uncertainty concerning the truth-value assigned to a formula is their most crucial feature. This potential in this area is demonstrated in this paper by applying them for the special case of paraconsistent reasoning.

The semantics we provide in this paper is 3-valued for the systems without da Costa’s Axiom (1) (see below), and infinite-valued for the systems with that axiom (but this infinite-valued semantics still provides decision procedures for these systems). We prove also that these results cannot be improved: neither of the systems without (1) has either a finite characteristic ordinary matrix or a two-valued characteristic Nmatrix, and neither of the systems with (1) has a finite characteristic Nmatrix.

2 Preliminaries

2.1 A Small Taxonomy of C-Systems

Let $\mathcal{L}_{cl}^+ = \{\wedge, \vee, \supset\}$, $\mathcal{L}_{cl} = \{\wedge, \vee, \supset, \neg\}$, and $\mathcal{L}_C = \{\wedge, \vee, \supset, \neg, \circ\}$.

Definition 1. Let $\mathbf{HCL}^+$ be some standard Hilbert-type system which has *MP* as the only inference rule, and is sound and strongly complete for the $\mathcal{L}_{cl}^+$ -

¹ The bivaluations semantics and the possible translations semantics described in [9, 10] are not really satisfactory from these points of view.

fragment of *CPL* (classical propositional logic). Define the following systems in $\mathcal{L}_C$:

C is the system obtained from **HCL**⁺ by adding the schemata:²

- (t) $\neg\varphi \vee \varphi$
- (c) $\neg\neg\varphi \supset \varphi$
- (b) $\circ\varphi \supset ((\varphi \wedge \neg\varphi) \supset \psi)$

Ci is the system obtained from **C** by the addition of the schema:

- (i) $\neg\circ\varphi \supset (\varphi \wedge \neg\varphi)$

Cia is the system obtained from **Ci** by the addition of the following schemata:

- (a) $\circ\varphi \wedge \circ\psi \supset \circ(\varphi \sharp \psi) \quad (\sharp \in \{\wedge, \vee, \supset\})$

Definition 2. For $L \in \{\mathbf{C}, \mathbf{Ci}, \mathbf{Cia}\}$:

Le is the system obtained from L by adding:

- (e) $\varphi \supset \neg\neg\varphi$

Ll is the system obtained from L by adding:³

- (l) $\neg(\varphi \wedge \neg\varphi) \supset \circ\varphi$

Lle is the system obtained from L by adding both (l) and (e).

2.2 Non-deterministic Matrices

Our main semantical tool in what follows will be the following generalization of the concept of a matrix given in [1, 2]:⁴

Definition 3.

1. A *non-deterministic matrix* (or *matrix* for short) for a propositional language $\mathcal{L}$ is a tuple $\mathcal{M} = \langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$, where:
 - (a) $\mathcal{V}$ is a non-empty set of *truth values*.
 - (b) $\mathcal{D}$ is a non-empty proper subset of $\mathcal{V}$.
 - (c) For every n -ary connective $\diamond$ of $\mathcal{L}$, $\mathcal{O}$ includes a corresponding n -ary function $\tilde{\diamond}$ from $\mathcal{V}^n$ to $2^{\mathcal{V}} - \{\emptyset\}$.

We say that $\mathcal{M}$ is *closed* if so is $\mathcal{V}$.

² In [9] this system is called **bC** (for “basic C”).

³ What we call here **Cial** was called **Cila** in [9]. It was shown there that this system is equivalent to **C**₁, da Costa most important system from [12].

⁴ A special two-valued case of this definition was essentially introduced in [4]. Another particular case of the same idea, using a similar name, was used in [8]. It should also be noted that Carnielli’s “possible-translations semantics” (see [7]) was originally called “non-deterministic semantics”, but later the name was changed to the present one. It is known that the semantics of non-deterministic matrices used here and possible-translations semantics are not identical, but it seems obvious that there are strong connections between them. The exact relations between the two types of semantics have not been clarified yet.

$$a \tilde{\wedge} b = \begin{cases} \mathcal{F} & \text{if either } a \in \mathcal{F} \text{ or } b \in \mathcal{F} \\ \mathcal{D} & \text{otherwise} \end{cases}$$

$$\begin{aligned}\simeq_a &= \begin{cases} \mathcal{F} & \text{if } a \in \mathcal{T} \\ \mathcal{T} & \text{if } a \in \mathcal{F} \\ \mathcal{D} & \text{if } a \in \mathcal{I} \end{cases} \\ \widetilde{\simeq}a &= \begin{cases} \mathcal{V} & \text{if } a \in \mathcal{F} \cup \mathcal{T} \\ \mathcal{F} & \text{if } a \in \mathcal{I} \end{cases}\end{aligned}$$

- A **C**-Nmatrix is an Nmatrix for $\mathcal{L}_C$ which is a refinement of some basic **C**-Nmatrix.
- $\mathcal{M}_C$ is the basic **C**-Nmatrix in which $\mathcal{T} = \{t\}$, $\mathcal{F} = \{f\}$, and $\mathcal{I} = \{I\}$.

It is easy to see that the nondeterministic truth tables corresponding to the operations in $\mathcal{M}_C$ are:

$\widetilde{\vee}$	f	I	t
f	$\{f\}$	$\{I, t\}$	$\{I, t\}$
I	$\{I, t\}$	$\{I, t\}$	$\{I, t\}$
t	$\{I, t\}$	$\{I, t\}$	$\{I, t\}$

$\widetilde{\wedge}$	f	I	t
f	$\{f\}$	$\{f\}$	$\{f\}$
I	$\{f\}$	$\{I, t\}$	$\{I, t\}$
t	$\{f\}$	$\{I, t\}$	$\{I, t\}$

$\widetilde{\supset}$	f	I	t
f	$\{I, t\}$	$\{I, t\}$	$\{I, t\}$
I	$\{f\}$	$\{I, t\}$	$\{I, t\}$
t	$\{f\}$	$\{I, t\}$	$\{I, t\}$

$\widetilde{\neg}$	f	I	t
	$\{t\}$	$\{I, t\}$	$\{f\}$

$\widetilde{\circ}$	f	I	t
	$\{t, I, f\}$	$\{f\}$	$\{t, I, f\}$

Theorem 1. $\mathbf{C}_{\langle \mathcal{M}_C, \widetilde{\vee}, \widetilde{\wedge}, \widetilde{\supset}, \widetilde{\neg}, \widetilde{\circ} \rangle}$ is sound.

Proof. It is straightforward to check that **C** is sound for any basic **C**-Nmatrix. Hence the theorem follows from Proposition 1. $\square$

Theorem 2. $\mathcal{M}_C \models_{\langle \mathcal{M}_C, \widetilde{\vee}, \widetilde{\wedge}, \widetilde{\supset}, \widetilde{\neg}, \widetilde{\circ} \rangle} \mathbf{C}$

Proof. The soundness of **C** with respect to $\mathcal{M}_C$ is an instance of Theorem 1. We prove completeness.⁵ Let **T** be a theory and φ_0 a sentence such that $\mathbf{T} \not\vdash_C \varphi_0$. We construct a model of **T** in $\mathcal{M}_C$ which is not a model of φ_0 . For this extend **T** to a maximal theory **T*** such that $\mathbf{T}^* \not\vdash_C \varphi_0$. **T*** has the following properties:

1. $\psi \notin \mathbf{T}^*$ iff $\psi \supset \varphi_0 \in \mathbf{T}^*$.
2. If $\psi \notin \mathbf{T}^*$ then $\psi \supset \varphi \in \mathbf{T}^*$ for every sentence φ of $\mathcal{L}_C$.
3. $\varphi \vee \psi \in \mathbf{T}^*$ iff either $\varphi \in \mathbf{T}^*$ or $\psi \in \mathbf{T}^*$.
4. $\varphi \wedge \psi \in \mathbf{T}^*$ iff both $\varphi \in \mathbf{T}^*$ and $\psi \in \mathbf{T}^*$.
5. $\varphi \supset \psi \in \mathbf{T}^*$ iff either $\varphi \notin \mathbf{T}^*$ or $\psi \in \mathbf{T}^*$.
6. For every sentence φ of $\mathcal{L}_C$, either $\varphi \in \mathbf{T}^*$ or $\neg\varphi \in \mathbf{T}^*$.
7. If $\neg\neg\varphi \in \mathbf{T}^*$ then $\varphi \in \mathbf{T}^*$.
8. If both $\varphi \in \mathbf{T}^*$ and $\neg\varphi \in \mathbf{T}^*$ then $\circ\varphi \notin \mathbf{T}^*$.

⁵ The completeness proof is very similar to the one that was given in [2] for the system C_{min} (see Corollary 1 below).

The proofs of Properties 1–5 are exactly as in the case of $\mathbf{HCL}^+$: Property 1 follows from the deduction theorem (which is obviously valid for C) and the maximality of $\mathbf{T}^*$. Property 2 is proved first for φ_0 using 1 and the tautology $((\varphi_0 \supset \varphi) \supset \varphi_0) \supset \varphi_0$. It then follows for all $\psi \notin \mathbf{T}^*$ by 1. Properties 3–5 are easy corollaries of 1, 2, and the maximality of $\mathbf{T}^*$. Finally, Property 6 is immediate from Property 3 and Axiom **(t)**, Property 7 follows from Axiom **(c)**, and Property 8 follows from Axiom **(b)**.

Define now a valuation v in $\mathcal{M}_C$ as follows:

$$v(\psi) = \begin{cases} f & \psi \notin \mathbf{T}^* \\ t & \psi \in \mathbf{T}^*, \neg\psi \notin \mathbf{T}^* \\ I & \psi \in \mathbf{T}^*, \neg\psi \in \mathbf{T}^* \end{cases}$$

Note that $v(\psi) \in \mathcal{D}$ iff $\psi \in \mathbf{T}^*$. We use this to prove that v is a legal valuation, i.e.: it respects the interpretations of the connectives in $\mathcal{M}_C$. That this is the case for the positive connectives easily follows from Properties 3–5 of $\mathbf{T}^*$. We prove next the cases of $\neg$ and $\circ$:

- Assume $v(\psi) = f$. Then $\psi \notin \mathbf{T}^*$. Hence $\neg\psi \in \mathbf{T}^*$ and $\neg\neg\psi \notin \mathbf{T}^*$ by Properties 6 and 7 of $\mathbf{T}^*$. Thus $v(\neg\psi) = t$.
- Assume $v(\psi) = t$. By definition, this implies $\neg\psi \notin \mathbf{T}^*$, whence $v(\neg\psi) = f$.
- Assume $v(\psi) = I$. By definition, this implies $\psi \in \mathbf{T}^*$ and $\neg\psi \in \mathbf{T}^*$. The latter implies $v(\neg\psi) \in \mathcal{D}$. Together with the former it also implies $\circ\psi \notin \mathbf{T}^*$, by Property 8 of $\mathbf{T}^*$. Hence $v(\circ\psi) = f$.

Since $v(\psi) \in \mathcal{D}$ iff $\psi \in \mathbf{T}^*$, $v(\psi) \in \mathcal{D}$ for every $\psi \in \mathbf{T}$, while $v(\varphi_0) \notin \mathcal{D}$. Hence v is a model of $\mathbf{T}$ which is not a model of $v(\varphi_0)$. $\square$

We turn next to **Ci** and **Cia**.

Definition 6.

- A **Ci**-Nmatrix is a **C**-Nmatrix which satisfies the following condition:

$$\text{Cond}(\mathbf{i}): \quad a \in \mathcal{T} \cup \mathcal{F} \Rightarrow \widetilde{\circ}a \subseteq \mathcal{T}$$

- $\mathcal{M}_{C_i}$ is the unique **Ci**-Nmatrix in which $\mathcal{T} = \{t\}$, $\mathcal{F} = \{f\}$, and $\mathcal{I} = \{I\}$.

Note that the only difference between $\mathcal{M}_{C_i}$ and $\mathcal{M}_C$ is that in $\mathcal{M}_{C_i}$ the truth table corresponding to $\circ$ is:

$\tilde{o}$	f	I	t
	$\{t\}$	$\{f\}$	$\{t\}$

Theorem 3.

- $\mathbf{Ci}_{i+1} = \mathbf{Ci}_i + \mathbf{Ci}_i \mathbf{C}_i$
- $\mathbf{M}_{\mathbf{Ci}_{i+1}} = \mathbf{M}_{\mathbf{Ci}_i} + \mathbf{C}_i \mathbf{C}_i^T \mathbf{M}_{\mathbf{Ci}_i} \mathbf{C}_i$

To show the first part it suffices by Theorem 1 to check that the validity of schema **(i)** follows from **Cond(i)**. This is easy. The proof of the second part is very similar to the proof of Theorem 2, using **Ci** instead of **C**. The only needed addition is a check that now $v(\circ\varphi) = t$ if $v(\varphi) \in \{t, f\}$ (where the valuation v is defined like in the proof of Theorem 2). Well, if $v(\varphi) \in \{t, f\}$ then $\varphi \wedge \neg\varphi \notin \mathbf{T}^*$. Hence (by schema **i**) $\neg\circ\varphi \notin \mathbf{T}^*$, and so $\circ\varphi \in \mathbf{T}^*$ (by schema **t**). By definition of v , these facts imply that $v(\circ\varphi) = t$. $\square$

Corollary 1. ... $C_{min}(\dots)$, $\mathbf{C}_{\psi}$, ..., (b) ... $\mathbf{C}_{\psi}$, ..., $C_{min}(\dots \vdash_C \psi, \dots \vdash_{C_{min}} \psi)$

• In [2] it was shown that the $\mathcal{L}_{cl}$ -reduct of $\mathcal{M}_C$ is a characteristic Nmatrix for C_{min} . The claim is immediate from this and Theorem 2. $\square$

Definition 7.

- A **Cia**-Nmatrix is a **Ci**-Nmatrix which satisfies the following condition:

$$\text{Cond}(\mathbf{a}): \quad a \in \mathcal{T} \cup \mathcal{F}, b \in \mathcal{T} \cup \mathcal{F} \Rightarrow a \widetilde{\#} b \subseteq \mathcal{T} \cup \mathcal{F} \quad (\# \in \{\vee, \wedge, \supset\})$$

- $\mathcal{M}_{Cia}$ is the unique **Cia**-Nmatrix in which $\mathcal{T} = \{t\}$, $\mathcal{F} = \{f\}$, and $\mathcal{I} = \{I\}$.

Note that in $\mathcal{M}_{Cia}$ the truth tables corresponding to $\{\vee, \wedge, \supset\}$ are:

$\widetilde{\vee}$	f	I	t	$\widetilde{\wedge}$	f	I	t	$\widetilde{\supset}$	f	I	t
f	{f}	{I, t}	{t}	f	{f}	{f}	{f}	f	{t}	{I, t}	{t}
I	{I, t}	{I, t}	{I, t}	I	{f}	{I, t}	{I, t}	I	{f}	{I, t}	{I, t}
t	{t}	{I, t}	{t}	t	{f}	{I, t}	{t}	t	{f}	{I, t}	{t}

Theorem 4.

- Cia₁, ..., Cia_n
- $\mathcal{M}_{Cia_1, \dots, Cia_n}$, Cia

To show the first part it suffices by Theorem 3 to check that the validity of schema **(a)** follows from $\text{Cond}(\mathbf{a})$. Again, this is easy. The proof of the second part is very similar to the proof of Theorem 3, with **Cia** used instead of **Ci**. The only needed addition is a check that now $v(\varphi \sharp \psi) \in \{t, f\}$ if $v(\varphi) \in \{t, f\}$ and $v(\psi) \in \{t, f\}$ (where the valuation v is defined like in the proof of Theorem 2). Well, if $v(\varphi) \in \{t, f\}$ and $v(\psi) \in \{t, f\}$ then by schemata **(i)** and **t**, $\circ(\varphi)$ and $\circ(\psi)$ are in $\mathbf{T}^*$, and so $\circ(\varphi \sharp \psi) \in \mathbf{T}^*$ by Schema **(a)**. Hence (by Schema **(b)**) either $\varphi \sharp \psi \notin \mathbf{T}^*$, or $\neg(\varphi \sharp \psi) \notin \mathbf{T}^*$. By definition of v , this yields $v(\varphi \sharp \psi) \in \{t, f\}$. $\square$

Examples:

1. $\vdash_{Cia} \neg(\varphi \wedge \psi) \supset (\neg\varphi \vee \neg\psi)$
2. $\not\vdash_{Ci} \neg(\varphi \wedge \psi) \supset (\neg\varphi \vee \neg\psi)$

To show the first part, let v be a legal valuation in **Cia**, and assume that $v(\neg\varphi \vee \neg\psi) = f$. Then $v(\neg\varphi) = f$ and $v(\neg\psi) = f$. Hence $v(\varphi) = v(\psi) = t$, and so $v(\varphi \wedge \psi) = t$. Thus $v(\neg(\varphi \wedge \psi)) = f$, implying that $v(\neg(\varphi \wedge \psi) \supset (\neg\varphi \vee \neg\psi)) \in \{t, f\}$.

For the second part, consider the following refuting valuation in $\mathcal{M}_{Ci}$: $v(\varphi) = t$, $v(\psi) = t$, $v(\neg\varphi) = f$, $v(\neg\psi) = f$, $v(\varphi \wedge \psi) = I$, $v(\neg(\varphi \wedge \psi)) = I$, $v(\neg\varphi \vee \neg\psi) = f$, $v(\neg(\varphi \wedge \psi) \supset (\neg\varphi \vee \neg\psi)) = f$.

Corollary 2. $\forall \epsilon, C_i, C_{ia}, C_{ib} \in \mathbb{R}_{\geq 0}$, $C_{min} = \max(C_i, C_{ia}, C_{ib})$.

We turn now to systems with (e).

Definition 8. For $L \in \{\mathbf{C}, \mathbf{Ci}, \mathbf{Cia}\}$:

- An *Le*-Nmatrix is an *L*-Nmatrix which satisfies the following condition:

$$\text{Cond}(\mathbf{e}): \quad a \in \mathcal{I} \Rightarrow \approx a \subseteq \mathcal{I}$$

- $\mathcal{M}_{Le}$ is the unique *Le*-Nmatrix in which $\mathcal{T} = \{t\}$, $\mathcal{F} = \{f\}$, and $\mathcal{I} = \{I\}$.

Note that in $\mathcal{M}_{L_e}$ ($L \in \{\mathbf{C}, \mathbf{Ci}, \mathbf{Cia}\}$) the truth table corresponding to $\neg$ is:

$\approx$	f	I	t
	$\{t\}$	$\{I\}$	$\{f\}$

Theorem 5. $\quad, L \in \{C, Ci, Cia\}$

- $Le_{\text{eff}} = Le_{\text{eff}}^{\text{eff}} + Le_{\text{eff}}^{\text{eff}}$
- $M_{Le} = M_{Le}^{\text{eff}} + M_{Le}^{\text{eff}}$

It is easy to see that (e) is valid in any $\mathbf{C}$ -Nmatrix which satisfies Cond(e). This entails the first part. The proof of the second part is similar to the proof that $\mathcal{M}_L$ is characteristic for L ($L \in \{\mathbf{C}, \mathbf{Ci}, \mathbf{Cia}\}$). The details are left for the reader. $\square$

Corollary 3. C, Ci, Cia, Ce, Cie, ..., Ciae, ...

Corollary 4. C, Ci, Cia, Ce, Cie, ..., Ciae, ..., Ciee, ..., Ciae, ...

Now we show that the semantics given in this section to the various systems cannot be simplified.

Theorem 6.

- C, Ci, Cia, Ce, Cie, Ciae
 C, Ci, Cia, Ce, Cie, Ciae

The proofs are identical to those given for C_{min} in [2].

4 The Systems with Schema 1

Definition 9. Let $\mathcal{T} = \{t_i^j \mid i \geq 0, j \geq 0\}$, $\mathcal{I} = \{I_i^j \mid i \geq 0, j \geq 0\}$, $\mathcal{F} = \{f\}$. Define the Nmatrix $\mathcal{M}_{Cl} = \langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$ for the language $\mathcal{L}_C$ by:

1. $\mathcal{V} = \mathcal{T} \cup \mathcal{I} \cup \mathcal{F}$
2. $\mathcal{D} = \mathcal{T} \cup \mathcal{I}$
3. $\mathcal{O}$ is defined by:

$$\begin{aligned} a \widetilde{\vee} b &= \begin{cases} \mathcal{D} & \text{if either } a \in \mathcal{D} \text{ or } b \in \mathcal{D}, \\ \mathcal{F} & \text{if } a, b \in \mathcal{F} \end{cases} \\ a \widetilde{\supset} b &= \begin{cases} \mathcal{D} & \text{if either } a \in \mathcal{F} \text{ or } b \in \mathcal{D} \\ \mathcal{F} & \text{if } a \in \mathcal{D} \text{ and } b \in \mathcal{F} \end{cases} \\ a \widetilde{\wedge} b &= \begin{cases} \mathcal{F} & \text{if either } a \in \mathcal{F} \text{ or } b \in \mathcal{F} \\ \mathcal{T} & \text{if } a = I_i^j \text{ and } b \in \{I_i^{j+1}, t_i^{j+1}\} \\ \mathcal{D} & \text{otherwise} \end{cases} \\ \widetilde{\sim} a &= \begin{cases} \mathcal{F} & \text{if } a \in \mathcal{T} \\ \mathcal{T} & \text{if } a \in \mathcal{F} \\ \{I_i^{j+1}, t_i^{j+1}\} & \text{if } a = I_i^j \end{cases} \\ \widetilde{\circ} a &= \begin{cases} \mathcal{D} & \text{if } a \in \mathcal{F} \cup \mathcal{T} \\ \mathcal{F} & \text{if } a \in \mathcal{I} \end{cases} \end{aligned}$$

Theorem 7. $\mathcal{M}_{Cl} \models \text{Schema 1}$

It is easy to check that $\mathcal{M}_{Cl}$ is a C-Nmatrix. To prove soundness, we need therefore to show only that (I) is valid in it. So let v be an $\mathcal{M}_{Cl}$ -legal valuation, and let φ be a sentence such that $v(\circ\varphi) \notin \mathcal{D}$. Then $v(\varphi) = I_i^j$ for some i, j . Hence $v(\neg\varphi) \in \{I_i^{j+1}, t_i^{j+1}\}$, and so $v(\varphi \wedge \neg\varphi) \in \mathcal{T}$, and $v(\neg(\varphi \wedge \neg\varphi)) = f$, as required.

To prove completeness, assume that $\mathbf{T} \not\models_{Cl} \varphi_0$. Extend $\mathbf{T}$ to a maximal theory $\mathbf{T}^*$ such that $\mathbf{T}^* \not\models_{Cl} \varphi_0$. Then $\mathbf{T}^*$ has the Properties 1-8 from the proof of Theorem 2. Let $\lambda_i.\alpha_i$ be an enumeration of all the formulas of $\mathcal{L}_C$ that do not begin with $\neg$. Then for any formula ψ of $\mathcal{L}_C$ there are unique $n(\psi), k(\psi)$ such that $\psi = \neg_{k(\psi)}\alpha_{n(\psi)}$, where $\neg_k\theta$ is θ preceded by k negation symbols. Define a valuation v in $\mathcal{M}_{Cl}$ as follows:

$$v(\psi) = \begin{cases} f & \psi \notin \mathbf{T}^* \\ t_{n(\psi)}^{k(\psi)} & \psi \in \mathbf{T}^*, \neg\psi \notin \mathbf{T}^* \\ I_{n(\psi)}^{k(\psi)} & \psi \in \mathbf{T}^*, \neg\psi \in \mathbf{T}^* \end{cases}$$

We show now that v is $\mathcal{M}_{Cl}$ -legal. The proofs that it respects the operations corresponding to $\vee$ and $\supset$ are like in the proof of Theorem 2. We consider next the cases of $\circ$, $\neg$ and $\wedge$.

- : That $v(\circ\psi) = f$ in case $v(\psi) \in \mathcal{I}$ is shown as in the proof of Theorem 2. Assume next that $v(\psi) \in \mathcal{T} \cup \mathcal{F}$. Then either $\psi \notin \mathbf{T}^*$, or $\neg\psi \notin \mathbf{T}^*$. It follows that $\psi \wedge \neg\psi \notin \mathbf{T}^*$, and so $\neg(\psi \wedge \neg\psi) \in \mathbf{T}^*$ (by Property 6). Hence $\circ\psi \in \mathbf{T}^*$ by (I), and so $v(\circ\psi) \in \mathcal{D}$.
 - ¬: The proofs that $v(\psi) = f$ implies $v(\neg\psi) \in \mathcal{T}$ and that $v(\psi) \in \mathcal{T}$ implies $v(\neg\psi) = f$ are like in the proof of Theorem 2. Assume next that $v(\psi) = I_n^k$. Then both ψ and $\neg\psi$ are in $\mathbf{T}^*$, and $\psi = \neg_k \alpha_n$. Thus $\neg\psi \in \mathbf{T}^*$, and $\neg\psi = \neg_{k+1} \alpha_n$. It follows by definition of v that $v(\neg\psi)$ is either I_n^{k+1} or t_n^{k+1} (depending whether $\neg\neg\psi$ is in $\mathbf{T}^*$ or not).
 - ∧: The proofs that if $v(\psi_1) = f$ or $v(\psi_2) = f$ then $v(\psi_1 \wedge \psi_2) = f$, and that $v(\psi_1 \wedge \psi_2) \in \mathcal{D}$ otherwise, are like in the proof of Theorem 2. Assume next that $v(\psi_1) = I_n^k$ and $v(\psi_2) \in \{I_n^{k+1}, t_n^{k+1}\}$. Then both ψ_1 and ψ_2 are in $\mathbf{T}^*$, and $\psi_1 = \neg_k \alpha_n$, $\psi_2 = \neg_{k+1} \alpha_n$. It follows that $\psi_2 = \neg\psi_1$, and that $\psi_1 \wedge \neg\psi_1 \in \mathbf{T}^*$. This entails (by Property 8) that $\circ\psi_1 \notin \mathbf{T}^*$. Hence schema (I) implies that $\neg(\psi_1 \wedge \neg\psi_1) \notin \mathbf{T}^*$, and so $\neg(\psi_1 \wedge \psi_2) \notin \mathbf{T}^*$. Thus $v(\psi_1 \wedge \psi_2) \in \mathcal{T}$.
- Obviously, $v(\psi) \in \mathcal{D}$ for every $\psi \in \mathbf{T}$, while $v(\varphi_0) = f$. Hence $\mathbf{T} \not\models_{\mathcal{M}_{Cl}} \varphi_0$. $\square$

Definition 10.

$\mathcal{M}_{Cil}$ is obtained from $\mathcal{M}_{Cl}$ through the modification:

$$a \in \mathcal{F} \cup \mathcal{T} \Rightarrow \widetilde{\circ}(a) = \mathcal{T}$$

$\mathcal{M}_{Cial}$ is obtained from $\mathcal{M}_{Cil}$ through the modifications:

$$a \in \mathcal{T} \text{ and } b \in \mathcal{T} \Rightarrow a \widetilde{\wedge} b = \mathcal{T}$$

$$a \in \mathcal{T} \text{ or } b \in \mathcal{T} \Rightarrow a \widetilde{\vee} b = \mathcal{T}$$

$$a \in \mathcal{F} \text{ or } b \in \mathcal{T} \Rightarrow a \widetilde{\supset} b = \mathcal{T}$$

$\mathcal{M}_{Cle}$, $\mathcal{M}_{Cile}$, $\mathcal{M}_{Ciale}$ are obtained from $\mathcal{M}_{Cl}$, $\mathcal{M}_{Cil}$, $\mathcal{M}_{Cial}$ (respectively) through the modification:

$$\widetilde{\circ}(I_i^j) = \{I_i^{j+1}\}$$

Theorem 8. $\mathbf{L} \in \{\mathbf{Cil}, \mathbf{Cial}, \mathbf{Cle}, \mathbf{Cile}, \mathbf{Ciale}\}$, $\mathcal{M}_L \models \mathbf{L}$.

Take **Cil** for example (the proofs for the other systems are similar). It is easy to check that $\mathcal{M}_{Cil}$ is both a **Ci**-matrix, and a refinement of $\mathcal{M}_{Cl}$. Hence the soundness of **Cil** for $\mathcal{M}_{Cl}$ follows from the first part of Theorem 3, Theorem 7, and Proposition 1. The proof of its completeness is a straightforward adaptation of the completeness proof of $\mathcal{M}_{Cl}$, similar to what was done in the proof of the second part of Theorem 3. $\square$

Corollary 5. $\mathbf{Cl}, \mathbf{Cil}, \mathbf{Cial}, \mathbf{Cle}, \mathbf{Cile}, \mathbf{Ciale}$ are decidable.⁶

⁶ The decidability results of these papers are not new. Thus the decidability of *Cial* ($=C_1$) is theorem 2.7.7 of [13].

Let $\mathbf{L}$ be one of these logics. From Theorem 8 and its proof it easily follows that to check whether a given formula φ is provable in $\mathbf{L}$, it suffices to check all legal partial valuations v in $\mathcal{M}_L$ which assign values in $\{f\} \cup \{t_i^j \mid 0 \leq i \leq n^*(\varphi), 0 \leq j \leq k^*(\varphi)\} \cup \{I_i^j \mid 0 \leq i \leq n^*(\varphi), 0 \leq j \leq k^*(\varphi)\}$ to subformulas of φ , where $n^*(\varphi)$ is the number of subformulas of φ which do not begin with $\neg$, and $k^*(\varphi)$ is the maximal number of consecutive negation symbols occurring within φ . This is a finite process. $\square$

The proof of Corollary 5 indicates that a simpler infinite Nmatrix would be sufficient for characterizing the set of provable formulas considered there.

Definition 11. For $\mathbf{L} \in \{\mathbf{Cl}, \mathbf{Cil}, \mathbf{Cial}, \mathbf{Cle}, \mathbf{Cile}, \mathbf{Ciale}\}$, Let $\mathcal{M}_L^{\text{weak}}$ be the simplest refinement of $\mathcal{M}_L$ in which the set of truth values is $\{f\} \cup \{t_0^j \mid 0 \leq j\} \cup \{I_0^j \mid 0 \leq j\}$.

Theorem 9. $\mathbf{L} \in \{\mathbf{Cl}, \mathbf{Cil}, \mathbf{Cial}, \mathbf{Cle}, \mathbf{Cile}, \mathbf{Ciale}\}$

$$\frac{\mathcal{M}_L^{\text{weak}} \models \varphi}{\mathcal{M}_L^{\text{weak}} \models \varphi} \quad \mathbf{L}$$

1. Since $\mathcal{M}_L^{\text{weak}}$ is a refinement of $\mathcal{M}_L$, $\mathbf{L}$ is sound for $\mathcal{M}_L^{\text{weak}}$. For completeness, assume $\not\models_{\mathbf{L}} \varphi$. Then φ has a refuting valuation in $\mathcal{M}_L$ of the type described in the proof of Corollary 5. Obtain from it a refuting valuation in $\mathcal{M}_L^{\text{weak}}$ by using $t_0^{(k^*(\varphi)+1)i+j}$ and $I_0^{(k^*(\varphi)+1)i+j}$ instead of t_i^j and I_i^j , respectively.
2. Let p, q and r be three distinct propositional variables. Let $N(p) = \{\neg_k p \mid k \geq 0\}$, $N(q) = \{\neg_k q \mid k \geq 0\}$, and let $\mathbf{T} = N(p) \cup N(q) \cup \{\neg(\varphi \wedge \psi) \mid \varphi \in N(p), \psi \in N(q)\} \cup \{\neg(\psi \wedge \varphi) \mid \varphi \in N(p), \psi \in N(q)\}$. Assume that v is a model in $\mathcal{M}_L^{\text{weak}}$ of $N(p) \cup N(q)$. Then in v all formulas in $N(p) \cup N(q)$ should get a value in $\{I_0^j \mid 0 \leq j\}$. It follows that for some m , either $v(p) = v(\neg_m q)$ or $v(q) = v(\neg_m p)$. In the first case $v(\neg(p \wedge \neg_{m+1} q)) = f$, while in the second $v(\neg(q \wedge \neg_{m+1} p)) = f$. Hence $\mathbf{T}$ has no model in $\mathcal{M}_L^{\text{weak}}$, and so $\mathbf{T} \vdash_{\mathcal{M}_L^{\text{weak}}} r$. On the other hand, it is easy to construct a model of $\mathbf{T}$ in $\mathcal{M}_L$ which is not a model of r . Hence $\mathbf{T} \not\models_{\mathbf{L}} r$. $\square$

We end the paper with a proof that the infinite-valued semantics given in this section cannot be replaced by a finite-valued one.

Theorem 10. $\mathbf{Cl} \not\vdash \mathbf{Ciale}$

Let $\mathbf{L}$ be such a logic. Assume for a contradiction that $\mathcal{M} = \langle \mathcal{T}_M, \mathcal{D}_M, \mathcal{O}_M \rangle$ is a finite characteristic Nmatrix for $\mathbf{L}$, and let n be the cardinality of $\mathcal{T}_M$. Define:

$$\begin{aligned} \mathbf{T} = & \{p_i \mid 1 \leq i \leq n+1\} \cup \{\neg p_i \mid 1 \leq i \leq n+1\} \\ & \cup \{\neg(p_i \wedge \neg p_j) \mid i \neq j, 1 \leq i \leq n+1, 1 \leq j \leq n+1\} \end{aligned}$$

Define a valuation v in $\mathcal{M}_{Ciale}$ by $v(p_{n+2}) = f$ and

$$v(p_i) = I_i^0 \quad v(\neg p_i) = I_i^1 \quad v(p_i \wedge \neg p_j) = I_0^0 \quad v(\neg(p_i \wedge \neg p_j)) = I_0^1$$

for $1 \leq i \leq n+1, 1 \leq j \leq n+1, i \neq j$. It is easy to check that v is a legal valuation in $\mathcal{M}_{Ciale}$ (actually, it is only a partial valuation, but since it is defined for a set of formulas closed under subformulas, it can be extended to a full legal valuation). Obviously, v is a model of **T** which is not a model of p_{n+2} . Hence $\mathbf{T} \not\models_{\mathbf{Ciale}} p_{n+2}$, and so $\mathbf{T} \not\models_{\mathbf{L}} p_{n+2}$. Since $\mathcal{M}$ is characteristic for **L**, there is a valuation v_0 in $\mathcal{M}$ which is a model of **T** but not a model of p_{n+2} . By the pigeonhole principle, there are $1 \leq i_0 < j_0 \leq n+1$ such that $v_0(p_{i_0}) = v_0(p_{j_0})$. Define now a new (partial) valuation v_1 by $v_1(p_{n+2}) = v_0(p_{n+2})$, and:

$$\begin{aligned} v_1(p_{i_0}) &= v_0(p_{i_0}) & v_1(\neg p_{i_0}) &= v_0(\neg p_{j_0}) & v_1(p_{i_0} \wedge \neg p_{i_0}) &= v_0(p_{i_0} \wedge \neg p_{j_0}) \\ v_1(\neg(p_{i_0} \wedge \neg p_{i_0})) &= v_0(\neg(p_{i_0} \wedge \neg p_{j_0})) \end{aligned}$$

Since v_0 is legal in $\mathcal{M}$, and $v_0(p_{i_0}) = v_0(p_{j_0})$, v_1 is also legal in $\mathcal{M}$. Now v_0 is a model of **T**, and so it is a model of $\{p_{i_0}, \neg p_{j_0}, \neg(p_{i_0} \wedge \neg p_{j_0})\}$. Hence v_1 is a model of $\{p_{i_0}, \neg p_{i_0}, \neg(p_{i_0} \wedge \neg p_{i_0})\}$. On the other hand, v_1 is not a model of p_{n+2} (because v_0 is not a model of p_{n+2}). Since **L** is sound for $\mathcal{M}$, it follows that $\{p_{i_0}, \neg p_{i_0}, \neg(p_{i_0} \wedge \neg p_{i_0})\} \not\models_{\mathbf{L}} p_{n+2}$. This contradicts the fact that $\{p_{i_0}, \neg p_{i_0}, \neg(p_{i_0} \wedge \neg p_{i_0})\} \vdash_{\mathbf{Cl}} p_{n+2}$, for **L** is an extension of **Cl**. $\square$

Note. The lack of finite characteristic $\vdash_{\mathbf{L}} \dots \vdash_{\mathbf{L}} \dots$ matrices for some C -systems has been known before (see [10]). However, what is proved in Theorem 10 is a much stronger result!

References

1. A. Avron and I. Lev, *Canonical Propositional Gentzen-Type Systems*, in **Proceedings of the 1st International Joint Conference on Automated Reasoning (IJCAR 2001)** (R. Goré, A. Leitsch, T. Nipkow, Eds), LNAI 2083, 529-544, Springer Verlag, 2001.
2. A. Avron, and I. Lev, *Non-deterministic Matrices*, in **Proceedings of the 34th IEEE International Symposium on Multiple-Valued Logic (ISMVL 2004)**, 282-287, IEEE Computer Society Press, 2004.
3. A. Avron, and I. Lev, *Non-deterministic Multiple-valued Structures*, to appear in the Journal of Logic and Computation.
4. D. Batens, K. De Clercq, and N. Kurtonina, *Embedding and Interpolation for Some Paralogics. The Propositional Case*, Reports on Mathematical Logic 33 (1999), 29-44.
5. D. Batens, C. Mortensen, G. Priest, J. P. Van Bendegem (eds.), **Frontiers of Paraconsistent Logic**, King's College Publications, Research Studies Press, Baldock, UK, 2000.
6. M. Bremer **An Introduction to Paraconsistent Logics**, Peter Lang GmbH, 2005.
7. W. A. Carnielli, *Possible-translations Semantics for Paraconsistent Logics*, in [5], 149-163.

8. J. M. Crawford and D. W. Etherington, *A Non-deterministic Semantics for Tractable Inference*, in **Proc. of the 15th International Conference on Artificial Intelligence and the 10th Conference on Innovative Applications of Artificial Intelligence**, 286-291, MIT Press, Cambridge, 1998.
9. W. A. Carnielli and J. Marcos, *A Taxonomy of C-systems*, in [11], 1-94.
10. W. A. Carnielli, M. E. Coniglio, J. Marcos, *Logics of Formal Inconsistency*, To appear in **Handbook of Philosophical Logic, 2nd edition** (D. Gabbay and F. Guenther, eds), Kluwer Academic Publishers, 2005.
11. W. A. Carnielli, M. E. Coniglio, I. L. M. D’ottaviano (eds.), **Paraconsistency — the logical way to the inconsistent** Lecture Notes in Pure and Applied Mathematics, Marcel Dekker, 2002.
12. N. C. A. da Costa, *On the theory of inconsistent formal systems*, Notre Dame Journal of Formal Logic 15 (1974), 497–510.
13. N. C. A. da Costa, D. Krause and O. Bueno, *Paraconsistent Logics and Paraconsistency: Technical and Philosophical Developments*, to appear in Dale Jacquette (ed.), **Philosophy of Logic** (D. Jacquette, ed.), Amsterdam North-Holland.
14. J. P. Delgrande and J. Mylopoulos, *Knowledge Representation: Features of Knowledge*, in **Fundamentals in Man-Machine Communication: Speech, Vision, and Natural Language** (J.P. Haton, ed.), Cambridge University Press, 1986. Reprinted in **Fundamentals of Artificial Intelligence** (W. Bibel et al., eds.), 1-36, Springer-Verlag, 1987.

Multi-valued Model Checking in Dense-Time^{*}

Ana Fernández Vilas, José J. Pazos Arias, A. Belén Barragáns Martínez,
Martín López Nores, Rebeca P. Díaz Redondo, Alberto Gil Solla,
Jorge García Duque, and Manuel Ramos Cabrer

Departamento de Ingeniería Telemática,
Universidad de Vigo. 36200, Vigo, Spain
avilas@det.uvigo.es

Abstract. In this paper we introduce $\mathcal{X}$ TCTL, a dense-time extension of the multi-valued Computation Tree Logic ($\mathcal{X}$ CTL) in [1]. Alternatively, $\mathcal{X}$ TCTL is a multi-valued extension of TCTL [2] over quasi-boolean algebras. A multi-valued quotient is defined which enables to reduce dense-time $\mathcal{X}$ TCTL model checking to the untimed case.

Keywords: formal methods, model checking, multi-valued logic, dense real-time.

1 Introduction

Temporal logic model checking, as a completely automatic verification technique, has been used successfully in practice to verify complex hardware and software designs. The model checking problem is easy to describe. Given some finite-state model $\mathcal{M}$ of the system and a temporal logic formula ϕ expressing some desired specification, model checking establishes whether the model satisfies the specification ($\mathcal{M} \models \phi$) or not ($\mathcal{M} \not\models \phi$). However, if the model of the system $\mathcal{M}$ is incomplete or contains inconsistencies, the response to the model checking problem cannot be established, in general, as $\models$ or $\not\models$ (two-valued approach). To effectively reason about this kind of models, multi-valued temporal logic model checking has been recently introduced in [1, 3]. Specifically, in [1] a multi-valued extension of CTL over quasi-boolean algebras ($\mathcal{X}$ CTL) is proposed and, for this family of logics, a symbolic multi-valued model checker algorithm is described ($\mathcal{X}$ Check). This extension allows temporal logic model checking to be used in presence of uncertainty and disagreement, however, real-time properties cannot be expressed. Real-time model checking has their early origin in [2], where $\dots$ introduced the first model checking algorithm for dense-time¹ models – Timed Automata (TA) with real-valued clocks – and the real-time logic TCTL – Timed Computation Tree Logic, the dense-time extension of CTL.

^{*} This work was partially supported by the Xunta de Galicia Basic Research Project PGIDIT04PXIB32201PR.

¹ It is well known that dense-time formal methods are more expressive and suitable than discrete ones for composition and refinement [4].

In order to reason in presence of uncertainty and disagreement when real-time is considered, this paper proposes a dense-time extension of $\mathcal{X}$ CTL, namely $\mathcal{X}$ TCTL. Alternatively, $\mathcal{X}$ TCTL is a multi-valued extension of TCTL over quasi-boolean algebras, in the same way that $\mathcal{X}$ CTL is a multi-valued extension of CTL. A dense-time structure, namely multi-valued Timed Automata ($\mathcal{X}$ TA), is also proposed as the multi-valued version of TA. In order to model check $\mathcal{X}$ TCTL formulas, a multi-valued abstraction is defined which enables to compute a finite quotient of a $\mathcal{X}$ TA and to reduce $\mathcal{X}$ TCTL to $\mathcal{X}$ CTL model checking. The rest of the paper is organized as follows. Section 2 gives some background on $\mathcal{X}$ CTL. Section 3 describes the real-time structures $\mathcal{X}$ TA and section 4 extends $\mathcal{X}$ CTL to dense-time by giving its semantics over $\mathcal{X}$ TA. Section 5 discusses the technique for reducing $\mathcal{X}$ TCTL to $\mathcal{X}$ CTL model checking. Finally, section 6 contains an application example.

2 Background on $\mathcal{X}$ CTL

This section contains some background about quasi-boolean algebras and $\mathcal{X}$ CTL taken from [1]. A partial order $(\mathcal{L}, \sqsubseteq)$ is a lattice if every two elements a, b in the set $\mathcal{L}$ have a unique least upper bound (meet: $a \sqcap b$) and a unique greatest lower bound (join: $a \sqcup b$). If the lattice is finite, $\top$ and $\perp$ represent the greatest and the least element of the lattice, respectively. A lattice $(\mathcal{L}, \sqsubseteq)$ is distributive iff $a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c)$ and $a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c)$. A quasi-boolean algebra is a tuple $(\mathcal{L}, \sqcap, \sqcup, \neg)$ where $(\mathcal{L}, \sqsubseteq)$ is a finite distributive lattice; and negation $\neg$ satisfies the following conditions:

$$\neg(a \sqcap b) = \neg a \sqcup \neg b \text{ (De Morgan)} \quad \neg\neg a = a \text{ (}\neg \text{ involution)} \quad (1)$$

$$\neg(a \sqcup b) = \neg a \sqcap \neg b \text{ (De Morgan)} \quad a \sqsubseteq b \text{ iff } \neg a \sqsupseteq \neg b \text{ (}\neg \text{ antimonotonic)} \quad (2)$$

A multi-valued Kripke ($\mathcal{X}$ Kripke) structure, as defined in [1], is a Kripke structure over a quasi-boolean algebra L . That is, each transition has a logical value in $\mathcal{L}$ and every atomic proposition is a variable which ranges over $\mathcal{L}$. Multi-valued CTL ($\mathcal{X}$ CTL) extends CTL to multi-valued reasoning by interpreting formulas in $\mathcal{X}$ Kripke structures. As L is described by a finite, quasi-boolean lattice $(\mathcal{L}, \sqsubseteq)$, the conjunction $\wedge$ ($\sqcap$ of the lattice), disjunction $\vee$ ($\sqcup$ of the lattice) and negation $\neg$ are available. The semantics of CTL operators is given by defining EX and AX with the standard treatment of multi-valued quantification (see section 4). AU and EU operators are defined using the AU and EU fixpoint properties and the definition of AX and EX . The remaining operators AF , EF , AG , EG are defined by the usual abbreviations in CTL (for instance $AF\phi$ is $A[\top U \phi]$). For a $\mathcal{X}$ Kripke structure $\mathcal{M}$ and a $\mathcal{X}$ CTL formula ϕ , the interpretation $\langle \phi \rangle_s^{\mathcal{X}\text{CTL}}$ denotes the truth value that ϕ takes in state s , that is, the degree at which the state s satisfies the property ϕ in the model $\mathcal{M}$.

3 Multi-valued Timed Automata: $\mathcal{XTA}$

In this section we define the multi-valued and dense-time structures $\mathcal{XTA}$ (multi-valued Timed Automata). $\mathcal{XTA}$ provides the dense-time counterpart of the $\mathcal{XKripke}$ structures which give the semantics to $\mathcal{XCTL}$. As in defining $\mathcal{XKripke}$ from Kripke structures, $\mathcal{XTA}$ incorporate logical values in the edges and propositions of TA, the well-known Timed Automata in [2].

Before defining $\mathcal{XTA}$, some definitions concerning clocks and valuations are in order. A clock is a simple real-valued variable drawn from the finite set $\mathcal{C}$. A clock constraint $\psi \in \Psi(\mathcal{C})$ is a boolean combination of atomic constraints on $\mathcal{C}$ of the form $x \prec c$ or $x - y \prec c$, where x, y are clocks, c is an integer constant, and $\prec$ is taken from $\{\leq, \geq, <, >\}$. A clock valuation γ is a point in $\mathbb{R}_+^{\#\mathcal{C}}$. If γ is a clock valuation, $\gamma + \tau$ for some $\tau \in \mathbb{R}_+$ stands for the clock valuation obtained by adding τ to the value of every clock. Given a clock constraint ψ and a valuation γ , $\psi(\gamma) = \top$ iff γ satisfies the clock constraint ψ , otherwise $\psi(\gamma) = \perp$. Given a clock constraint ψ , $\psi = \perp$ iff there is not valuation γ such that $\psi(\gamma) = \top$; and $\psi = \top$ iff $\psi(\gamma) = \top$ for every valuation γ . Finally, let $\lambda \subseteq \mathcal{C}$ be a set of clocks, then γ_λ is the clock valuation obtained from γ by setting every clock in λ to 0.

Definition 1. ($\mathcal{XTA}$ Syntax) A multi-valued Timed Automaton $\mathcal{M} = (Q, Q_0, E, T, P, I, A, \mathcal{C}, L)$ is a tuple

$$\begin{aligned} L &= (\mathcal{L}, \sqcap, \sqcup, \neg), \text{ where } \mathcal{L} \text{ is a lattice with } \perp, \top \text{ and } \mathcal{L} \text{ is a set of logical values} \\ A &= \{a_1, \dots, a_n\} \text{ is a finite set of atomic propositions} \\ \mathcal{C} &= \{c_1, \dots, c_m\} \text{ is a finite set of clocks} \\ Q_0 &\subseteq Q \text{ is the set of initial states} \\ E &\subseteq Q \times \Psi(\mathcal{C}) \times 2^{\mathcal{C}} \times Q \text{ is the set of edges} \\ T &: E \rightarrow \mathcal{L} \text{ is the transition function} \\ e &= (q, \psi, \lambda, q') \in E \implies T(e) \in \mathcal{L} \\ P &: Q \times A \rightarrow \mathcal{L} \text{ is the proposition function} \\ I &: Q \times \mathcal{L} \rightarrow \Psi(\mathcal{C}) \text{ is the invariant function} \\ q &\in Q \implies I(q, l) \in \mathcal{L} \end{aligned}$$

Informally, a $\mathcal{XTA}$ is well-formed when, for every single behaviour in the semantics of the model, the logical value is unique. A $\mathcal{XTA}$ is well-formed iff the following conditions hold:

- For every $q \in Q$ and for every pair $l \neq l' \in \mathcal{L}$, $I(q, l) \wedge I(q, l') = \perp$.
- For every pair of edges $e_1, e_2 \in E$ of the form $e_1 = (q, \psi_1, \lambda_1, q')$ and $e_2 = (q, \psi_2, \lambda_2, q')$ such that $T(e_1) \neq T(e_2)$, $\psi_1 \wedge \psi_2 = \perp$.

that is, the set of invariants (the set of guards) is disjoint for every location q (pair of locations q, q').

Semantics. The semantics of $\mathcal{M}$ is given in terms of a multi-valued structure $\mathcal{M}^S = (\mathcal{S}_{\mathcal{M}}, \mathcal{S}_{\mathcal{M}}^0, \mathcal{T}_{\mathcal{M}}, P_{\mathcal{M}})$ where (1) $\mathcal{S}_{\mathcal{M}} = \{Q \times \mathbb{R}_+^{\#\mathcal{C}}\}$ is the set of states; (2)

$\mathcal{S}_{\mathcal{M}}^0 = \{(q_0, \gamma^0) \mid q_0 \in Q_0\}$ is the subset of initial states where γ^0 assigns 0 to every clock; (3) $\mathcal{T}_{\mathcal{M}}$ is the transition relation; and (4) $P_{\mathcal{M}} : \mathcal{S}_{\mathcal{M}} \times A \rightarrow \mathcal{L}$ assigns to every state (q, γ) and to every proposition a the logical value $P_{\mathcal{M}}((q, \gamma), a) = P(q, a)$. In order $\mathcal{M}^S$ to be a completely connected graph (like $\mathcal{XKripke}$), a default logical value $l_d \in \mathcal{L}$ is designated. l_d is the logical value of the default edges and invariants in $\mathcal{M}$, that is:

- For every location q , $I(q, l_d) = \neg(\bigvee I(q, l_i))$ for all $I(q, l_i)$ with $l_i \neq l_d$.
- For every pair q and q' , $e_d = (q, \psi_d, -, q')$ is an edge in $\mathcal{M}$ such that $T(e_d) = l_d$, and $\psi_d = \neg(\bigvee \psi_i)$ for all ψ_i in an edge of the form $(q, \psi_i, -, q')$.

The multi-valued transition relation $\mathcal{T}_{\mathcal{M}} \subseteq \mathcal{S}_{\mathcal{M}} \times \mathbb{R}_+ \cup \{d\} \times \mathcal{L} \times \mathcal{S}_{\mathcal{M}}$ (denoted $\rightarrow$) contains discrete transitions (labeled by d) and time transitions (labeled by $\tau \in \mathbb{R}_+$). Consider a state $s = (q, \gamma)$, $\mathcal{T}_{\mathcal{M}}$ is defined by the following rules:

- Given an edge $e = (q, \psi, \lambda, q') \in E$ such that $T(e) = l$, a discrete transition $(q, \gamma) \xrightarrow{d, l} (q', \gamma_\lambda)$ exists if $\psi(\gamma) = \top$.
- Given $\tau \in \mathbb{R}_+$, a time transition $(q, \gamma) \xrightarrow{\tau, l} (q, \gamma + \tau)$ exists if $\forall 0 \leq \tau' \leq \tau$, the l -invariant holds $I(q, l)(\gamma + \tau') = \top^2$.

So, $\mathcal{M}^S$ is an infinite $\mathcal{XKripke}$ structure whose multi-valued transitions are augmented with labels $\in \mathbb{R}_+ \cup \{d\}$. For a state s , $out(s)$ denotes the set of transitions from s . For a transition t , $T(t)$ denotes the logical value of t ; $label(t)$ the label $\in \mathbb{R}_+ \cup \{d\}$; and $target(t)$ the target state of the transition t .

4 Multi-valued TCTL: $\mathcal{XTCTL}$

Following the main ideas in the definition of TCTL [2] (the dense-time extension of CTL), in this section, we extend $\mathcal{XCTL}$ with dense-time, namely multi-valued TCTL ($\mathcal{XTCTL}$).

Definition 2. ($\mathcal{XTCTL}$: Syntax) The multi-valued $\mathcal{XTCTL}$ formulas are defined by the following grammar:

$$\phi ::= \top \mid a \mid y. \phi \mid \psi \mid \neg\phi \mid \phi \vee \phi \mid E[\phi U \phi] \mid A[\phi U \phi]$$

where $a \in A$, $y \in \mathcal{C}$, $y. \phi$ is the clock constraint $y \in \mathcal{C}$ and ψ is a clock constraint $\psi \in \mathcal{C}$. The semantics of $\mathcal{XTCTL}$ is defined by the following function $\Psi(\mathcal{C} \cup \mathcal{E})$:

Semantics. Let $\mathcal{M}$ be a $\mathcal{XTA}$ over the quasi-boolean algebra L with a set of atomic propositions A and a set of clocks $\mathcal{C}$. $\mathcal{XTCTL}$ formulas are interpreted with respect to the infinite $\mathcal{XKripke}$ structure $\mathcal{M}_{\mathcal{E}}^S$ induced by $\mathcal{M}$ where $\mathcal{C}$ is

² The concatenation of $s \xrightarrow{\tau, l} s'$ and $s' \xrightarrow{\tau', l'} s''$ is the transition $s \xrightarrow{\tau + \tau', l \wedge l'} s''$.

extended by all clocks $\mathcal{E}$ mentioned in the formula ϕ . Given a state $s = (q, \gamma)$ of $\mathcal{M}_{\mathcal{E}}^S$, the interpretation $\langle \phi \rangle_s$ denotes the truth value that ϕ takes in state s . The complete semantics is defined in the following paragraphs.

As L is a quasi-boolean algebra, $\vee$, $\neg$ and $\top$ are well-defined. We start defining $\mathcal{X}$ TCTL by giving the semantics of propositional operators:

$$\langle a \rangle_s = P_{\mathcal{M}}(s, a) \quad \langle \neg \phi \rangle_s = \neg \langle \phi \rangle_s \quad \langle \phi \vee \phi' \rangle_s = \langle \phi \rangle_s \vee \langle \phi' \rangle_s \quad (3)$$

$$\langle \psi \rangle_s = \psi(\gamma) \quad \langle y.\phi \rangle_s = \langle \phi \rangle_{(q, \gamma_{\{y\}})} \quad (4)$$

where (3) correspond to the $\mathcal{X}$ CTL semantics, while (4) are specific of the timed extension $\mathcal{X}$ TCTL and correspond to the TCTL semantics.

Next, we define the semantics of until (EU and AU). In $\mathcal{X}$ CTL [1], these operators are defined by using the next state operators, EX and AX , as follows:

$$\langle EX\phi \rangle_s^{\mathcal{X}CTL} = \bigvee_{t \in out(s)} (T(t) \wedge \langle \phi \rangle_{target(t)}) \quad (5)$$

$$\langle AX\phi \rangle_s^{\mathcal{X}CTL} = \bigwedge_{t \in out(s)} (\neg T(t) \vee \langle \phi \rangle_{target(t)}) \quad (6)$$

However, when time is modeled by a dense domain, there is not notion of “next state”. We use a next operator $\triangleright$ similar to the “single-step until” in T_{μ} [6].

Informally, let t be a transition ($s \xrightarrow{label(t), T(t)} target(t)$), the formula $\phi \triangleright_t \phi'$ asserts that ϕ is true now and stays true until the transition t is taken and establishes ϕ' (footnote ³). The contribution of the transition t will be different according to the kind of quantification being considered (existential $\triangleright_t^{\exists}$, universal $\triangleright_t^{\forall}$):

$$\langle \phi \triangleright_t^{\exists} \phi' \rangle_s = \langle \phi \rangle_s \wedge (T(t) \wedge \langle \phi' \rangle_{target(t)}) \wedge \langle \phi \triangleright_{label(t)} \phi' \rangle_s \quad (7)$$

$$\langle \phi \triangleright_t^{\forall} \phi' \rangle_s = \langle \phi \rangle_s \wedge (\neg T(t) \vee \langle \phi' \rangle_{target(t)}) \wedge \langle \phi \triangleright_{label(t)} \phi' \rangle_s \quad (8)$$

where $\triangleright_d = \triangleright_0$, and $\phi \triangleright_{\tau} \phi'$ for some $\tau \geq 0$ is the dense next predicate which advances time an amount τ while $\phi \vee \phi'$ (due to the density of the time) continuously stays true:

$$\langle \phi \triangleright_{\tau} \phi' \rangle_s = \bigwedge_{s'} (\neg l \vee \langle \phi \vee \phi' \rangle_{s'}) \mid s \xrightarrow{\tau', l} s', 0 < \tau' \leq \tau \quad (9)$$

that is, when t is a discrete transition, the third element of the conjunct is $\top$ and does not contribute since the transition t is instantaneous. Finally, we define the binary next state operators $\phi AX \phi'$ and $\phi EX \phi'$ as the universal and existential

³ While in T_{μ} the operator is a combination of a time transition and a discrete transition, in our interpretation this transition is a single time or a single discrete transition.

“single-step until”, respectively; $\exists\mathcal{U}$ and $\forall\mathcal{U}$ operators, as in $\mathcal{X}\text{CTL}$, are obtained by replacing the next state operators by the binary next state operators⁴:

$$\langle\phi AX\phi'\rangle_s = \bigwedge_{t \in \text{out}(s)} \langle\phi \triangleright_t^\forall \phi'\rangle_s \quad (10)$$

$$\langle\phi EX\phi'\rangle_s = \bigvee_{t \in \text{out}(s)} \langle\phi \triangleright_t^\exists \phi'\rangle_s \quad (11)$$

$$A[\phi U\phi'] = \phi' \vee (\phi AX A[\phi U\phi'] \wedge \phi EX A[\phi U\phi']) \quad (12)$$

$$E[\phi U\phi'] = \phi' \vee (\phi EX E[\phi U\phi']) \quad (13)$$

The standard $EX\phi$ and $AX\phi$ operators could be defined as $\top EX\phi$, and $\top AX\phi$. With this definition the negation of “next” property is preserved, i.e., $\neg\langle AX\phi\rangle_s = \langle EX\neg\phi\rangle_s$. The remaining operators AF , EF , AG , EG are defined by the usual abbreviations. Note $\mathcal{X}\text{TCTL}$ reduces to TCTL when $\mathcal{L} = \{\top, \perp\}$, and $\{\wedge, \vee, \neg\}$ are the usual connectives in classical logic.

5 $\mathcal{X}\text{TCTL}$ Model Checking

Consider a $\mathcal{X}\text{TA}$ $\mathcal{M}$, a $\mathcal{X}\text{TCTL}$ formula is interpreted over the infinite structure $\mathcal{M}^S$ which give semantics to $\mathcal{M}$. To model check a formula ϕ , a finite and exact (with respect to the formula ϕ) abstraction of the infinite structure has to be provided. As in classical dense-time TCTL model checking, we use Strong Time-abstracting Bisimulations to reduce the verification problem to the untimed case.

Definition 3. (Multi-valued Strong Time-Abstracting Bisimulation, $\mathcal{X}\text{STaB}$.) $\mathcal{X}$

Let $\tau \in \mathbb{R}_+$ and $\epsilon \in \mathbb{R}_+$. $\mathcal{M} \in \mathcal{X}$ and $\mathcal{M}_\epsilon \in \mathcal{X}$ are $\mathcal{X}\text{STaB}$ iff $\mathcal{M} \cong \subseteq \mathcal{S}_\mathcal{M} \times \mathcal{S}_{\mathcal{M}_\epsilon}$ and $\mathcal{X}$ $s_1 \cong s_2$ iff

$$\text{I) } \forall s_1 \xrightarrow{d,l} s_1' \text{, } \exists s_2 \xrightarrow{d,l} s_2' \text{, } s_1' \cong s_2'$$

$$\text{II) } \forall s_1 \xrightarrow{\epsilon,l} s_1' \text{, } \exists s_2 \xrightarrow{\epsilon,l} s_2' \text{, } s_1' \cong s_2'$$

$$\text{III) } \forall s_1 \text{, } \exists s_2 \text{, } s_1 \cong s_2$$

Lemma 1. ($\mathcal{X}\text{TCTL}$ preservation), $\mathcal{X}$ $\mathcal{M} \in \mathcal{X}$ and $\mathcal{M}_\epsilon \in \mathcal{X}$ are $\mathcal{X}\text{STaB}$ iff $\phi \models \mathcal{M} \iff \phi \models \mathcal{M}_\epsilon$ $\mathcal{E}_\phi \subseteq \mathcal{M}_\epsilon$ $\Psi_\phi \subseteq \mathcal{M}_\epsilon$ $A_\phi \subseteq \mathcal{M}_\epsilon$ $\mathcal{M}_\epsilon \cong \subseteq \mathcal{S}_{\mathcal{M}_\epsilon} \times \mathcal{S}_{\mathcal{M}^S}$ $\mathcal{E}_\phi \cong \subseteq \mathcal{X}$ $\mathcal{M}_\epsilon^S \cong \subseteq \mathcal{S}_{\mathcal{M}_\epsilon^S} \times \mathcal{S}_{\mathcal{M}^S}$ $y_i = 0$ $y_i \in \mathcal{E}_\phi$ $y_i \in \Psi_\phi$ $y_i \in A_\phi$ $\cong$ $\phi \models \mathcal{M} \iff \phi \models \mathcal{M}_\epsilon$ $s_1 \cong s_2$ $\langle\phi\rangle_{s_1} = \langle\phi\rangle_{s_2}$

⁴ As in $\mathcal{X}\text{CTL}$, AU includes an additional conjunct to preserve an “strong until” semantics for states that have no outgoing $\top$ transitions.

(Hint on proof) The proof is similar to the one for TCTL and Timed Automata (lemma 13 in [7]), but considering a logical value in transitions of the semantic structure. Remark that, the region equivalence in [2] induces a $\mathcal{XSTaB}$ with the preservation characteristics above. Since this equivalence induces a finite partition of the state-space, a $\mathcal{XSTaB}$ with these preservation characteristics exists and has a finite number of classes.

Given a $\mathcal{XSTaB} \cong$ on the semantic structure $\mathcal{M}^S$ of a $\mathcal{XTA} \mathcal{M}$, it is easy to prove that $\mathcal{M}^S$ and its quotient with respect to $\cong$ are themselves bisimilar. Since $\cong$ exists and is finite, $\mathcal{XTCTL}$ model checking can be computed over the quotient with respect to $\cong$. This multi-valued quotient can be obtained by adapting the timed minimization algorithm in [7] to multi-valued transitions. In general, minimization algorithms are based on the facts that a bisimulation induces a pre-stable partition and vice-versa. So minimization is done by partition refinement, that is, the coarser pre-stable partition ρ of the state-space $\mathcal{S}_{\mathcal{M}}$ is computed starting from an initial partition ρ_0 and successively refining it until it becomes pre-stable.

Partitions. Let $\mathcal{M}^S = (\mathcal{S}_{\mathcal{M}}, \mathcal{S}_{\mathcal{M}}^0, \mathcal{T}_{\mathcal{M}}, P_{\mathcal{M}})$ be the semantic structure of a $\mathcal{XTA} \mathcal{M}$. A partition ρ of $\mathcal{S}_{\mathcal{M}}$ is a set of disjoint classes $r \subseteq \mathcal{S}_{\mathcal{M}}$ the union of which yields $\mathcal{S}_{\mathcal{M}}$. A class r is a region in $\mathcal{S}_{\mathcal{M}}$ of the form $[q, \zeta]$ where q is a location in $\mathcal{M}$ and ζ is a $\mathcal{C}$ -polyhedron⁵. Let $\mathcal{C}$ be the clocks in $\mathcal{M}$, a $\mathcal{C}$ -hyperplane is a set of valuations satisfying an atomic clock constraint $\Psi(\mathcal{C})$ (see section 3). A $\mathcal{C}$ -polyhedron is a subset in $\mathbb{R}_+^{\# \mathcal{C}}$ defined by the union, intersection and complementation of $\mathcal{C}$ -hyperplanes.

Operations on Polyhedra. By definition, intersection ($\zeta_1 \cap \zeta_2$), union ($\zeta_1 \cup \zeta_2$) and complementation ($\bar{\zeta}$) are well-defined operations on polyhedra. Polyhedra difference is defined via complementation as $\zeta_1 - \zeta_2 = \zeta_1 \cap \bar{\zeta}_2$. Besides, the following are common operations which will be used for obtaining the multi-valued quotient:

$$\zeta \downarrow = \{\gamma \mid \exists \tau \in \mathbb{R}_+ \cdot \gamma + \tau \in \zeta\} \quad (14)$$

$$\zeta_{\lambda}^{-1} = \{\gamma \mid \gamma_{\lambda} \in \zeta\} \quad (15)$$

$$until(\zeta, \zeta') = \{\gamma \mid \exists \gamma' \in \zeta, \tau \in \mathbb{R}_+ \cdot \gamma + \tau = \gamma', \forall \tau' \leq \tau \cdot \gamma + \tau' \in \zeta'\} \quad (16)$$

where, given a $\mathcal{C}$ -polyhedron ζ , the operation $\zeta \downarrow$ decrements the valuations in ζ an arbitrary amount of time; the pre-reset operation ζ_{λ}^{-1} for some subset of clocks $\lambda \subseteq \mathcal{C}$ computes all valuations which, after resetting clocks in λ , yield a valuation in ζ . Finally, the operation $until(\zeta, \zeta')$ contains all the valuations in a $\mathcal{C}$ -polyhedron ζ' which can let time pass and reach ζ continuously staying in ζ' , $until(\zeta, \zeta') = (\zeta \cap \zeta') \downarrow \cap \zeta'$. All the above polyhedra operations can be computed over a DBM (Difference Bound Matrix) representation (see [7]).

⁵ We adopt this definition for simplicity. The definitions in the following and the model checking remain equal if a more general definition is adopted, that is, a class being the union of elements $[q, \zeta]$.

Pre-stability. The notion of pre-stability in $\mathcal{X}$ Kripke models is a multi-valued extension of the classical pre-stability [7], and similar abstract predecessors are defined in order to compute a pre-stable partition being a $\mathcal{X}$ STaB. Let ρ be a partition of the state-space $\mathcal{S}_{\mathcal{M}}$ and r a class in this partition, the abstract predecessor $pred_d$ ($pred_\epsilon$) of r compute the set of states which are predecessors of a state in r by a d -labeled discrete transition (by a ϵ -labeled time transitions) with a logical value $l \in \mathcal{L}$:

$$pred_k(r, l) = \{s \in \mathcal{S}_{\mathcal{M}} \mid \exists s' \in r \cdot s \xrightarrow{k, l} s'\} \text{ with } k \in \{d, \epsilon\} \quad (17)$$

Given a class $r = [q, \zeta]$, $pred_k(r, l)$ obtains a region, holding the formal definition of the operator above, which can be computed symbolically as follows:

$$pred_\epsilon([q, \zeta], l) = [q, \text{until}(\zeta, I(s, l))] \quad (18)$$

$$pred_d([q, \zeta], l) = \bigcup_{t \in \mathcal{T}_{\mathcal{M}}} [q', (\zeta \cap \psi)_{\lambda}^{-1}] \mid T(t) = l, t = (q', < a, \psi, \lambda >, q) \quad (19)$$

Definition 4. (Pre-stable partition) Let ρ be a partition of $\mathcal{S}_{\mathcal{M}}$ and $(k, l) \in \mathcal{L} \times \mathcal{L}$. ρ is a pre-stable partition with respect to (k, l) if for every $r_1, r_2 \in \rho$, $r_1 \subseteq pred_k(r_2, l)$ implies $r_1 \cap pred_\epsilon(r_2, l) = \emptyset$. If ρ is a pre-stable partition with respect to (k, l) , then $r_1 \cap pred_\epsilon(r_2, l) = \emptyset$ for every $r_1, r_2 \in \rho$.

Given two classes r_1 and r_2 such that r_1 is unstable in (k, l) with respect to r_2 , refinement consists in replacing r_1 by $r_1^1 = r_1 \cap pred_k(r_2, l)$ and $r_1^2 = r_1 - pred_k(r_2, l)$. In this way, both r_1^1, r_1^2 are stables in (k, l) with respect to r_2 .

Multi-valued Quotient. The multi-valued quotient of a $\mathcal{X}$ TA $\mathcal{M}$ with respect to a partition ρ is the structure $(\rho, \rho_0, \rightarrow', P_\rho)$ where (1) ρ_0 is set of initial class in ρ , the ones containing a state (q_0, γ^0) ; (2) the transition relation $\rightarrow' = \rho \times \mathcal{L} \times \rho$ is as follows:

$$r_1 \xrightarrow{l} r_2 \text{ iff } \exists s_1 \in r_1, s_2 \in r_2 \cdot s_1 \xrightarrow{d, l} s_2 \text{ or } s_1 \xrightarrow{\epsilon, l} s_2 \quad (20)$$

and (3) P_ρ is defined as $P_\rho(r, a) = P(q, a)$ for every class $r = [q, \zeta]$. The multi-valued quotient is a finite $\mathcal{X}$ Kripke structure which has as nodes the classes in ρ . Such quotient can be obtained during the refinement process, when the abstract predecessors are computed.

As the algorithms proposed in [8, 7], our algorithm only computes the classes which are reachable. A class is reachable if it contain at least one state (q, γ) which can be reached, by transitions with specification condition $> \perp$, from some initial state (q_0, γ^0)

$\mathcal{X}$ TCTL Model Checking. $\mathcal{X}$ TCTL model checking can be computed as follows. Firstly, an initial partition ρ is selected which preserves A_ϕ, Ψ_ϕ and $y_i = 0$.

Secondly, the minimization algorithm computes a pre-stable refinement ρ of the initial partition ρ_0 (every refinement of ρ_0 also preserves A_ϕ , Ψ_ϕ and $y_i = 0$). The algorithm which obtains the pre-stable refinement ρ is analogous to the one in [7] for Timed Automata, but distinguishing the abstract predecessors by the logical value of the transitions. After obtaining the pre-stable refinement, the finite and multi-valued quotient is constructed. Finally, the model checking can be computed over such quotient by using the untimed model checker $\mathcal{X}\text{check}$ [9]. To do this, $\mathcal{X}\text{check}$ has to be augmented with time and reset propositions.

6 Application

Many-valued logics has been applied in software engineering to a variety of situations which need to model additional truth values between true and false (see [10] for a survey). For instance, the intermediate truth values of the logic can represent incomplete information (or uncertainty). Such applications typically use a 3-valued logic, with the values $\top$, $\perp$ and M for “Maybe” knowledge (figure 1(a)). In this case, model checking can determine (1) if a property holds (does not hold), even when the model is incomplete; or (2) if the property is unknown to be hold, when the missing information affects the property. Also, the intermediate values can represent disagreement ($\top\perp$ or $\perp\top$) between two different system’s viewpoints (figure 1(b)); or both uncertainty M and disagreement D when a number of incomplete viewpoints are combined (figure 1(c)).

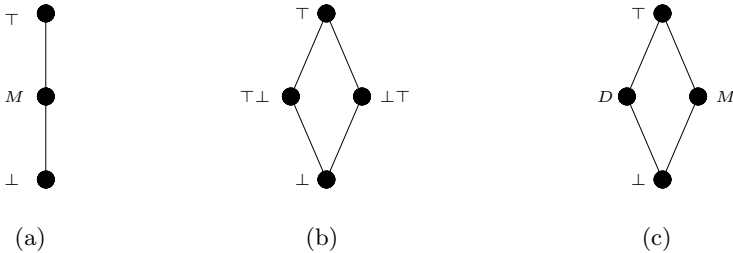
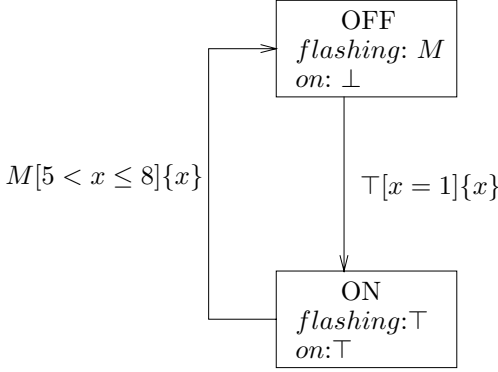


Fig. 1. Some example lattices for expressing uncertainty, disagreements and both

However, all the works in the literature assume that the truth only depends on a qualitative order on the time, and does not on timed behavior. With the results in this paper, many-valued reasoning in presence of real-time constraints is enabled. We illustrate this new kind of reasoning on a simple example of a flashing light. Let $(\mathcal{L}, \sqsubseteq)$ be the example lattice for expressing uncertainty in figure 1(a) with negation defined by horizontal symmetry. Also, let $\mathcal{M}$ be the proposed $\mathcal{X}\text{TA}$ in figure 2 which represents an incomplete specification of the flashing light with $l_d = \perp$.

For this example, we identify the following property: Is the light ever flashing within less than 3 time units? (eq. 21). The formalization of this property in $\mathcal{X}\text{TCTL}$ is given by the following equation:



$$\begin{aligned}
 I(\text{OFF}, \top) : x \leq 1, I(\text{OFF}, M) : x > 1 \\
 I(\text{ON}, \top) : x \leq 8 (asl_d = \perp, I(\text{ON}, \perp) : x > 8)
 \end{aligned}$$

Fig. 2. An incomplete $\mathcal{X}$ TA over the example lattice in figure 1(a)

$$y. AF(\text{flashing} \wedge y \leq 3) \quad (21)$$

where AF is defined in terms of AU as usual, that is, $AF\phi \triangleq A[\top U \phi]$.

Let $\mathcal{M}_{\mathcal{E}}$ be $\mathcal{M}$ extended with the specification clock y in the formula. In order to model check the property in eq. 21, a finite quotient being a $\mathcal{X}$ STaB has to be computed which preserves the time proposition $y \leq 3$ and the truth value of flashing . For this, an initial partition ρ_0 preserving $y \leq 3$ is selected:

$$\rho_0 := \{[\text{OFF}, y \leq 3], [\text{OFF}, y > 3], [\text{ON}, y \leq 3], [\text{ON}, y > 3]\}$$

The minimization algorithm obtains a pre-stable refinement of ρ_0 guaranteeing the classes in this partition are classes in a $\mathcal{X}$ STaB. Additionally, as the initial partition ρ_0 preserves $y \leq 3$ and flashing , every refinement of ρ_0 will also preserve both. The many-valued quotient is shown in figure 3 where dashed arrows represent M -transitions. For clarity, transitions are labeled with ϵ (time transitions) or d (discrete transitions), and besides, $\perp$ transitions and unreachable classes (9 and 10) have been removed.

This quotient model is an $\mathcal{X}$ Kripke structure where the truth value of the formula can be computed by using $\mathcal{X}$ Check. For this, it is necessary to add a new proposition $y \leq 3$ to the $\mathcal{X}$ Kripke structure. Being $\zeta_{y \leq 3}$ the polyhedron defined by the atomic clock constraint $y \leq 3$, the new proposition $y \leq 3$ is $\top$ in every class $[q, \zeta]$ such that $\zeta \subseteq \zeta_{y \leq 3}$, and $\perp$ in another case. The result of the evaluation of the eq. 21 is M (“Maybe”) since there is a deadlock class (number 3 in figure 3) where flashing is M . So, maybe the property will hold in the final specification of the problem, or maybe not. If the missing information about the progress of the time in OFF is provided as $I(OFF, \top) : x > 1$, the property will not hold. However, if this information is specified as $I(OFF, \perp) : x > 1$, the property will hold.

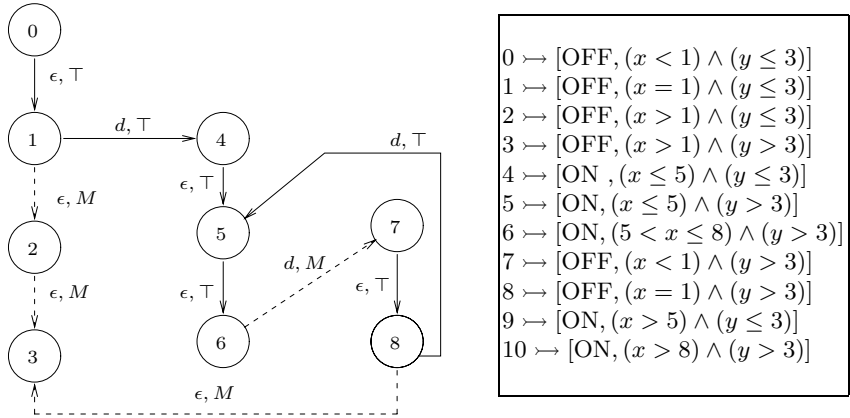


Fig. 3. A finite quotient for evaluating eq. 21. $\perp$ -transitions are not showed

7 Discussion

In this paper, we have defined $\mathcal{X}$ TCTL, a dense-time extension of the multi-valued logic $\mathcal{X}$ CTL which enables to express timing properties when uncertainty and disagreement arise. We have defined multi-valued Strong Time-abstracting Bisimulations and applied them to reduce model checking for dense-time and multi-valued models to the untimed multi-valued case. In that way, the existing model checking tool for $\mathcal{X}$ CTL ($\mathcal{X}$ Check) [9] can be exploited. Also, as it has been proposed in [11, 12], the multi-valued quotient could be used to reduce model checking from multi-valued $\mathcal{X}$ TCTL to classical two-valued CTL.

As a concrete application of multi-valued reasoning, we have been working for several years in the formalization of an iterative and incremental methodology, namely SCTL/MUS [13], which has been extended for dense-time in [14, 15, 16]. This methodology defines a 6-valued quasi-boolean logic which measures the capability of an intermediate model in an incremental process to satisfy a given property. So, we think providing a multi-valued counterpart of TA and TCTL can open some other applications of multi-valued reasoning in environments where uncertainty and disagreement appear in presence of time dependence.

References

1. Marsha Chechik, Benet Devereux, Arie Gurfinkel, and Steve Easterbrook. Multi-Valued Symbolic Model-Checking. *ACM Transactions on Software Engineering and Methodology*, 12(4):1–38, October 2003.
2. Rajeev Alur, Costas Courcoubetis, and David Dill. Model Checking in Dense Real-time. *Information and Computation*, 104(1):2–34, 1993.
3. Beata Konikowska and Wojciech Penczek. *Beyond Two: Theory and Applications of Multiple-valued Logic*, chapter Model Checking for Multi-Valued CTL*, pages 193–210. Springer, 2003.

4. Rajeev Alur. *Techniques for Automatic Verification of Real-Time Systems*. PhD thesis, Department of Computer Science, Stanford University, 1991.
5. Rajeev Alur and Thomas Henzinger. A Really Temporal Logic. *Journal of the ACM*, 41(1):181–204, 1994.
6. Thomas A. Henzinger, Xavier Nicollin, Joseph Sifakis, and Sergio Yovine. Symbolic Model-checking for Real Time Systems. *Information and Computation*, 111(2):193–244, 1994.
7. Stavros Tripakis and Sergio Yovine. Analysis of Timed Systems using Time-abstraction Bisimulations. *Formal Methods in System Design*, 18(1):25–68, 2001.
8. Mihalis Yannakakis and David Lee. An Efficient Algorithm for Minimizing Real-Time Transition Systems. *Formal Methods in System Design*, 11(2):113–136, 1997.
9. Marsha Chechik, Benet Devereux, and Arie Gurfinkel. XChек: A Multi-Valued Model-Checker. In *Computer-Aided Verification, 14th International Conference (CAV'02)*, volume 2404 of *Lecture Notes in Computer Science*, pages 505–509. Springer, 2002.
10. Leonard Bolc and Piotr Borowik. *Many-Valued logics 2. Automated Reasoning and Practical Applications*, chapter 7, pages 169–198. Springer, 2003.
11. Beata Konikowska and Wojciech Penczek. Reducing Model Checking from Multi-Valued CTL * to CTL*. In *Concurrency Theory, 13th International Conference (CONCUR'02)*, volume 2421 of *Lecture Notes in Computer Science*, pages 226–239. Springer, 2002.
12. Arie Gurfinkel and Marsha Chechik. Multi-valued Model Checking via Classical Model Checking. In *14th International Conference on Concurrency Theory (CONCUR'03)*, volume 2761 of *Lecture Notes in Computer Science*. Springer, 2003.
13. José J. Pazos Arias and Jorge García Duque. SCTL-MUS: A Formal Methodology for Software Development of Distributed Systems. A Case Study. *Formal Aspects of Computing*, 13:50–91, 2001.
14. Ana Fernández Vilas, José J. Pazos Arias, and Rebeca P. Díaz Redondo. Extending Timed Automaton and Real-time Logic to Many-valued Reasoning. In *Formal Techniques in Real-Time and Fault-tolerant Systems, 7th International Symposium (FTRTFT'02)*, volume 2469 of *Lecture Notes in Computer Science*, pages 185–204. Springer, 2002.
15. Ana Fernández Vilas, José J. Pazos Arias, Alberto Gil Solla, Rebeca P. Díaz Redondo, Jorge García Duque, and A. Belén Barragáns Martínez. Incremental Specification with SCTL/MUS-T: a Case Study. *The Journal of Systems & Software*, 70(1-2):189–208, 2004.
16. Ana Fernández Vilas, José J. Pazos Arias, Rebeca P. Díaz Redondo, Alberto Gil Solla, and Jorge García Duque. A Many-valued Logic with Imperative Semantics for Incremental Specification of Timed Models. In *International Conference on Integrated Formal Methods (IFM'04)*, volume 2999 of *Lecture Notes in Computer Science*, pages 382–401. Springer, 2004.

Brun Normal Forms for Co-atomic Łukasiewicz Logics

Stefano Aguzzoli¹, Ottavio M. D'Antona², and Vincenzo Marra²

¹ Dipartimento di Scienze dell'Informazione,
Università degli Studi di Milano,
Via Comelico 39-41, 20135 Milano, Italy
`aguzzoli@dsi.unimi.it`

² Dipartimento di Informatica e Comunicazione,
Università degli Studi di Milano,
Via Comelico 39-41, 20135 Milano, Italy
`{dantona, marra}@dico.unimi.it`

Abstract. We construct a class of Łukasiewicz formulae whose associated McNaughton functions constitute a family of Schauder hats having special properties. Our technique is inspired by the well-known algorithm of Brun [3, 4, 2] for simultaneous diophantine approximations. As a first application of Brun hats we construct normal forms for co-atomic Łukasiewicz logics. We also show how to combine Brun hats to obtain normal forms for all finite-valued Łukasiewicz logics.

1 Introduction and Preliminaries

In order to make this paper easily accessible to as wide an audience as possible, we include in this section a larger amount of background than customary.

We denote by ω the set $\{0, 1, 2, \dots, n \dots\}$ of nonnegative integers, and by $\omega + 1$ the set $\omega \cup \{\omega\}$. Łukasiewicz infinite-valued logic L_ω is axiomatized by

$$\begin{array}{ll} \text{L1: } \varphi \rightarrow (\psi \rightarrow \varphi) & \text{L2: } (\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \vartheta) \rightarrow (\varphi \rightarrow \vartheta)) \\ \text{L3: } ((\varphi \rightarrow \psi) \rightarrow \psi) \rightarrow ((\psi \rightarrow \varphi) \rightarrow \varphi) & \text{L4: } (\neg\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \varphi) \end{array}$$

with *modus ponens*

$$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi}$$

as the only inference rule.

By Chang's Completeness Theorem [5], L_ω is the logic of the real unit interval $[0, 1]$ equipped with the operations

$$x \rightarrow y := \min(1, 1 - x + y) \qquad \neg x := 1 - x,$$

in the sense that a formula is provable if and only if it evaluates constantly to 1 over $[0, 1]$ (i.e. is an L_ω -*tautology*) when connectives are interpreted as above. We

write $\mathbf{L}_\omega \models \varphi$ iff φ is an $\mathbf{L}_\omega$ -tautology iff $\mathbf{L}_\omega$ proves φ . The set of $\mathbf{L}_\omega$ -tautologies is denoted by $Taut(\mathbf{L}_\omega)$. Two formulae φ and ψ are *logically equivalent*, in symbols $\varphi \equiv \psi$, iff $\mathbf{L}_\omega \models (\varphi \rightarrow \psi) \odot (\psi \rightarrow \varphi)$.

Common derived connectives are

$$\begin{aligned} \varphi \oplus \psi &:= \neg\varphi \rightarrow \psi & \varphi \odot \psi &:= \neg(\neg\varphi \oplus \neg\psi) \\ \varphi \ominus \psi &:= \varphi \odot \neg\psi & \varphi \leftrightarrow \psi &:= (\varphi \rightarrow \psi) \odot (\psi \rightarrow \varphi) \\ \varphi \vee \psi &:= (\varphi \rightarrow \psi) \rightarrow \psi & \varphi \wedge \psi &:= \neg(\neg\varphi \vee \neg\psi) \end{aligned}$$

Their interpretation in $[0, 1]$ is given, for all $x, y \in [0, 1]$, by

$$\begin{aligned} x \oplus y &= \min(1, x + y) & x \odot y &= \max(0, x + y - 1) \\ x \ominus y &= \max(0, x - y) & x \leftrightarrow y &= 1 - |x - y| \\ x \vee y &= \max(x, y) & x \wedge y &= \min(x, y) \end{aligned}$$

Upon choosing an arbitrary formula ξ , one introduces the 0-ary connectives $\top := \xi \rightarrow \xi$ and $\perp := \neg\top$, and the choice of ξ is immaterial because $\top \equiv \psi$ for any $\psi \in Taut(\mathbf{L}_\omega)$. Evidently, the interpretation of $\top$ in $[0, 1]$ is 1, and that of $\perp$ is 0. For any integer $n > 0$, we let $n\varphi$ denote the formula $\varphi \oplus \varphi \oplus \cdots \oplus \varphi$, with n many occurrences of φ . Analogously, we denote by φ^n the formula $\varphi \odot \varphi \odot \cdots \odot \varphi$. We also stipulate that 0φ is $\perp$ and φ^0 is $\top$ for any φ .

Equivalent formulations of Lukasiewicz logic can be obtained by choosing a set of functionally complete connectives other than $\{\rightarrow, \neg\}$. When the main focus is on the algebraic semantics of Lukasiewicz logic the most popular approach is through Chang's MV algebras, and then the usual choice of connectives is $\{\oplus, \neg, \perp\}$ ($\{\oplus, \neg\}$ suffices).

For any integer $n > 0$, the set $S_n = \{0, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n-1}{n}, 1\}$ is closed with respect to all aforementioned operations. *Lukasiewicz* $(n+1)$ -valued logic, denoted $\mathbf{L}_n$, is the logic of $S_n \subseteq [0, 1]$ equipped with the restriction $\rightarrow$ and $\neg$. The notions of tautology and logical equivalence in $\mathbf{L}_n$ are, *mutatis mutandis*, the same as in $\mathbf{L}_\omega$. The set $\{\mathbf{L}_i \mid 0 < i \in \omega + 1\}$ of many-valued Lukasiewicz logics ordered by

$$\mathbf{L}_i \leq \mathbf{L}_j \quad \text{iff} \quad Taut(\mathbf{L}_i) \subseteq Taut(\mathbf{L}_j)$$

forms a bounded lattice $\mathcal{L}$, with top element classical Boolean logic $B = \mathbf{L}_1$ and bottom element $\mathbf{L}_\omega$, that turns out to be isomorphic (via $\mathbf{L}_i \mapsto i$, $\mathbf{L}_\omega \mapsto 0$), to the lattice of non-negative integers $(\mathbb{N}, \vee, \wedge)$ equipped with reversed divisibility order, where $m \vee n = \gcd(m, n)$ and $m \wedge n = \text{lcm}(m, n)$. Then co-atoms of $\mathcal{L}$ are exactly the logics $\mathbf{L}_p$ for p a prime number.

Via the Lindenbaum-Tarski functor, the algebraic counterpart of $\mathbf{L}_\omega$ is the variety (a.k.a. equational class) $\mathbf{MV}$ of MV algebras. An *MV algebra* is a structure $(A, \oplus, \neg, \perp)$ satisfying the equations:

$$\begin{aligned} \text{MV1} \quad & (x \oplus y) \oplus z = x \oplus (y \oplus z) & \text{MV4} \quad & \neg\neg x = x \\ \text{MV2} \quad & x \oplus y = y \oplus x & \text{MV5} \quad & x \oplus \neg\perp = \neg\perp \\ \text{MV3} \quad & x \oplus \perp = x & \text{MV6} \quad & \neg(\neg x \oplus y) \oplus y = \neg(\neg y \oplus x) \oplus x \end{aligned}$$

For each positive integer k , the logic $\mathbf{L}_k$ is finitely axiomatizable [8]. Hence, each $\mathbf{L}_k$ naturally corresponds to a subvariety of $\mathbf{MV}$, denoted $\mathbf{MV}_k$.

By Chang's Completeness Theorem, the MV algebra $([0, 1], \oplus, \neg, \perp)$ generates $\mathbf{MV}$. Hence, by standard universal algebra, the free MV algebra over n generators $\mathcal{M}_n$ is the subset of $[0, 1]^{[0, 1]^n}$ generated by the projections (coordinate functions) $\pi_i(x_1, x_2, \dots, x_n) = x_i$, for $1 \leq i \leq n$. There is a neat characterisation of such functions: by McNaughton's Representation Theorem [9], $\mathcal{M}_n$ coincides with the set of all continuous (finitely) piecewise linear functions from $[0, 1]^n$ to $[0, 1]$, with each piece a linear polynomial with *integer* coefficients. Operations are defined pointwise. Hence, any formula φ in the variables $x_1, \dots, x_n$ corresponds to (an element of the free algebra $\mathcal{M}_n$, i.e. to) a function $\varphi^{\mathcal{M}_n}$ of the above type, which shall henceforth be called a *McNaughton function*. We then have $\mathbb{L}_\omega \models \varphi$ iff $\varphi^{\mathcal{M}_n}(t_1, \dots, t_n) = 1$ for all $(t_1, \dots, t_n) \in [0, 1]^n$.

For each integer $k > 0$, the set of restrictions of McNaughton functions in n variables to $S_k^n \subseteq [0, 1]^n$ is the free n -generated algebra in the variety $\mathbf{MV}_k$.

We now need to recall some notions from integral polyhedral geometry. For further background on MV algebras and polyhedral geometry see e.g. [6] and [7], respectively.

For each integer $n > 0$, we let $\mathbf{0}$ be the n -tuple $(0, 0, \dots, 0)$ and $\mathbf{1}$ be the n -tuple $(1, 1, \dots, 1)$. We let $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ denote the canonical basis of $\mathbb{R}^n$.

A set C of points (vectors) in $\mathbb{R}^n$ is *affinely independent* iff $\sum_{\mathbf{x} \in C} a_{\mathbf{x}} \mathbf{x} = \mathbf{0}$ and $\sum_{\mathbf{x} \in C} a_{\mathbf{x}} = 0$ for $a_{\mathbf{x}} \in \mathbb{R}$, implies $a_{\mathbf{x}} = 0$ for all $\mathbf{x} \in C$. The *convex hull* of a set $C \subseteq \mathbb{R}^n$ is the set of all convex combinations of points in C : $\text{conv}(C) = \{\sum_{\mathbf{x} \in C} a_{\mathbf{x}} \mathbf{x} \mid 0 \leq a_{\mathbf{x}} \in \mathbb{R}, \sum_{\mathbf{x} \in C} a_{\mathbf{x}} = 1\}$. A *polytope* is the convex hull of a finite set of points. If $C = \{\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{x}_{n+1}\} \subseteq [0, 1]^n$ is affinely independent, the polytope $S = \text{conv}(C)$ is a *simplex* of dimension n (or an n -simplex), and $\mathbf{x}_1, \dots, \mathbf{x}_{n+1}$ are its *vertices*, which are said to *span* the simplex S . The convex hull of a subset of the set of vertices of a n -simplex S is again a simplex, called a *face* of S .

Let $\mathbf{x} \in ([0, 1] \cap \mathbb{Q})^n$ be a rational point. Display $\mathbf{x}$ as $(\frac{a_1}{d_1}, \frac{a_2}{d_2}, \dots, \frac{a_n}{d_n})$ with $\gcd(a_i, d_i) = 1$ for $i \in \{1, \dots, n\}$. Let $d = \text{lcm}(d_1, d_2, \dots, d_n)$. Then $\frac{a_i}{d_i} = \frac{b_i}{d}$, for $b_i = \frac{a_i d}{d_i}$. The point $H(\mathbf{x}) := (b_1, b_2, \dots, b_n, d) \in \mathbb{Z}^{n+1}$ is the expression of $\mathbf{x}$ in *homogeneous coordinates*, and $\text{den } \mathbf{x} := d$ is the *denominator* of $\mathbf{x}$.

With each n -simplex $S = \text{conv}(\mathbf{x}_1, \dots, \mathbf{x}_{n+1})$ such that each $\mathbf{x}_i \in ([0, 1] \cap \mathbb{Q})^n$, we associate the $(n+1) \times (n+1)$ integral matrix M_S whose rows are the homogeneous expressions $H(\mathbf{x}_i)$ of the vertices of S . We say that S is *unimodular* iff $\det(M_S) = \pm 1$.

Let $S = \text{conv}(\mathbf{x}_1, \dots, \mathbf{x}_{n+1})$ be a unimodular n -simplex. Observe that $\mathbf{v} \in S \cap \mathbb{Q}^n$ iff there exists exactly one $(n+1)$ -tuple $(c_1, \dots, c_{n+1})$ of nonnegative integers such that $H(\mathbf{v}) = \sum_{i=1}^{n+1} c_i H(\mathbf{x}_i)$. Hence, $\text{den } \mathbf{v} = \sum_{i=1}^{n+1} c_i \text{den } \mathbf{x}_i$. The *Farey mediant* of a subset $\{\mathbf{x}_{\iota(1)}, \dots, \mathbf{x}_{\iota(k)}\}$ of $\{\mathbf{x}_1, \dots, \mathbf{x}_{n+1}\}$, for $k \in \{1, \dots, n+1\}$, $\iota: \{1, \dots, k\} \rightarrow \{1, \dots, n+1\}$ injective, is the only rational point $\mathbf{x}$ such that $H(\mathbf{x}) = \sum_{i=1}^k H(\mathbf{x}_{\iota(i)})$. Note that $\text{den } \mathbf{x} = \sum_{i=1}^k \text{den } \mathbf{x}_{\iota(i)}$. The *Farey subdivision* of S through $\mathbf{x}$ is the set $\{\text{conv}(V_1), \text{conv}(V_2), \dots, \text{conv}(V_k)\}$ of n -dimensional simplices, where each V_i is obtained by replacing in $\{\mathbf{x}_1, \dots, \mathbf{x}_{n+1}\}$ the vertex $\mathbf{x}_{\iota(i)}$ with $\mathbf{x}$. Note that each $\text{conv}(V_i)$ is unimodular, $\bigcup_{i=1}^k \text{conv}(V_i) = S$,

and $\text{conv}(V_i) \cap \text{conv}(V_j) = \text{conv}(V_i \cap V_j)$. When $k = 2$, we call the Farey subdivision on an *edge subdivision*. For any $k \leq n$, a k -simplex in $[0, 1]^n$ is *unimodular* iff it can be completed to a unimodular n -simplex.

A *unimodular partition* $\mathcal{U}$ of $[0, 1]^n$ is a finite set of simplices such that:

1. Each 0-simplex $\{\mathbf{v}\} \in \mathcal{U}$ (by abuse of notation we shall write $\mathbf{v} \in \mathcal{U}$) is a rational point, that is $\mathbf{v} \in ([0, 1] \cap \mathbb{Q})^n$ (0-simplices are called *vertices* of $\mathcal{U}$);
2. If an n -simplex belongs to $\mathcal{U}$ then all of its k -dimensional faces, for each $-1 \leq k \leq n$, belong to $\mathcal{U}$, too (note that, by definition, $\emptyset$ is the only (-1) -dimensional face of any polytope);
3. The intersection of any two simplices of $\mathcal{U}$ is a face of both of them.
4. $\bigcup \mathcal{U} = [0, 1]^n$.
5. Each simplex $S \in \mathcal{U}$ is unimodular.

We call a unimodular partition $\mathcal{U}$ of $[0, 1]^n$ *principal* iff its set of vertices coincides with the set $\{0, 1\}^n$ of vertices of the n -cube. Principal partitions will play a key rôle in the sequel — let us briefly recall how they arise.

If $\text{Sym}(n)$ is the group of permutations of n elements, let

$$\mathcal{T}_n = \{T_n^\sigma \mid \sigma \in \text{Sym}(n)\}, \quad T_n^\sigma = \{(t_1, t_2, \dots, t_n) \mid t_{\sigma(1)} \leq t_{\sigma(2)} \leq \dots \leq t_{\sigma(n)}\}.$$

It is easy to see that $\mathcal{T}_n$ is the set of all n -simplices of a uniquely determined unimodular partition $\mathcal{U}_n$ of $[0, 1]^n$ such that:

- $\mathcal{U}_n$ is principal.
- $\mathcal{U}_n$ has $n!$ simplices of dimension n , each of them congruent — through a suitable permutation $\sigma \in \text{Sym}(n)$ — to the *principal* n -simplex T_n :

$$T_n := T_n^{\text{id}} = \{(t_1, t_2, \dots, t_n) \mid t_1 \leq t_2 \leq \dots \leq t_n\}.$$

Note that $T_n = \text{conv}(\mathbf{0}, \mathbf{e}_n, \mathbf{e}_{n-1} + \mathbf{e}_n, \dots, \mathbf{e}_1 + \mathbf{e}_2 + \dots + \mathbf{e}_n)$, and that the vertices of T_n are linearly ordered by $\mathbf{x} \leq \mathbf{y}$ iff $\mathbf{x} \cdot \mathbf{1} \leq \mathbf{y} \cdot \mathbf{1}$. Moreover, $\mathbf{x} \leq \mathbf{y}$ iff $\sigma(\mathbf{x}) \leq \sigma(\mathbf{y})$ for each $\sigma \in \text{Sym}(n)$. Throughout, we shall use the same symbol σ to denote both a permutation in $\text{Sym}(n)$ and the substitution of variables specified by $x_i \mapsto x_{\sigma(i)}$, that is $\sigma(x_i) = x_{\sigma(i)}$. We shall also denote by $\sigma(\mathbf{t})$ the point $(t_{\sigma(1)}, \dots, t_{\sigma(n)})$, for each point $\mathbf{t} \in T_n$.

For every McNaughton function $f \in \mathcal{M}_n$ it is possible to find a unimodular complex $\mathcal{U}$ such that f is linear over each simplex in $\mathcal{U}$. We say such a $\mathcal{U}$ is *linearly adequate* to f . Linear adequacy allows one to express f as a suitable $\oplus$ -sum of a special class of McNaughton functions defined over $\mathcal{U}$, namely *Schauder hats*. This technique was pioneered in [10]. Let us summarise the construction.

The *support* $\text{supp}(f)$ of a function $f : [0, 1]^n \rightarrow [0, 1]$ is the set $\{\mathbf{x} \mid f(\mathbf{x}) > 0\}$. Given a vertex $\mathbf{v}$ in $\mathcal{U}$, the *star* $\text{star } \mathbf{v}$ of $\mathbf{v}$ in $\mathcal{U}$ is the set of all simplices of $\mathcal{U}$ having $\mathbf{v}$ among their vertices.

Definition 1. *The Schauder hat at $\mathbf{v}$ over $\mathcal{U}$ is the continuous function $h_{\mathbf{v}}$ uniquely determined by the following prescriptions:*

- $h_{\mathbf{v}}(\mathbf{v}) = 1/\text{den } \mathbf{v}$.
- $h_{\mathbf{v}}(\mathbf{x}) = 0$ for all $\mathbf{x} \in [0, 1]^n \setminus \bigcup \text{star } \mathbf{v}$.
- $h_{\mathbf{v}}$ is linear over each simplex of $\mathcal{U}$.

Note that $\bigcup \text{star } \mathbf{v}$ is the closure of $\text{supp}(f)$. An easy computation shows that unimodularity of the simplices in $\text{star}(\mathbf{v})$ guarantees that the linear pieces of $h_{\mathbf{v}}$ have integer coefficients, whence $h_{\mathbf{v}} \in \mathcal{M}_n$.

Observe that given an arbitrary $\mathbf{v} \in ([0, 1] \cap \mathbb{Q})^n$, with $\text{den } \mathbf{v} = d$, and an arbitrary function $f \in \mathcal{M}_n$, we have $f(\mathbf{v}) \in \{0, \frac{1}{d}, \frac{2}{d}, \dots, \frac{d-1}{d}, 1\}$. Hence, by linearity, one can show that given a unimodular complex $\mathcal{U}$ linearly adequate to f , and given the set $\{(\mathbf{v}, k_{\mathbf{v}}) \mid \mathbf{v} \text{ is a vertex of } \mathcal{U}, f(\mathbf{v}) = \frac{k_{\mathbf{v}}}{\text{den } \mathbf{v}}\}$, then

$$f = \bigoplus_{\mathbf{v}} k_{\mathbf{v}} h_{\mathbf{v}}.$$

See [11] for a proof of the completeness of L_{ω} exploiting Schauder hats.

We now have marshalled enough mathematical machinery to embark on the construction of Brun hats. By way of motivation, note that the task of effectively writing down a formula whose interpretation is a given but arbitrary Schauder hat is computationally nontrivial:

1. Known algorithms to find a unimodular partition linearly adequate to a McNaughton function $\varphi^{\mathcal{M}_n}$ are more easily specified in semantic rather than syntactic terms.
2. The “linear” formula ψ_S associated with the restriction $f|_S$, for $S \in \text{star}(\mathbf{v})$ an n -simplex, is costly to compute from the values f takes on the vertices of S , at least on the basis of known algorithms (see [12, 6]).
3. Patching the formulae $\{\psi_S\}_{S \in \text{star } \mathbf{v}}$ to obtain a single formula associated with $h_{\mathbf{v}}$ entails a costly $\bigwedge \bigvee$ -combination of the ψ_S ’s. This is because, in general, $\bigcup \text{star } \mathbf{v}$ is not a convex subset of $[0, 1]^n$ (see [6]); when $\bigcup \text{star } \mathbf{v}$ is convex, the desired formula boils down to $\bigwedge_{S \in \text{star } \mathbf{v}} \psi_S$.

However, in producing a Schauder hats-based normal form φ for L_k , $k > 0$ an integer, one may take advantage of the liberty to choose values of $\varphi^{\mathcal{M}_n}$ at points in $[0, 1]^n \setminus S_k^n$. Our main result is that exploiting this basic observation one can obtain a defensible generalisation of the familiar minterms in classical propositional logic. Recall that a (*Boolean*) *minterm* is a conjunction of variables and negated variables which yields 1 under exactly one Boolean evaluation (that is, in geometric language, over exactly one vertex of the hypercube $[0, 1]^n$). Then each Boolean formula is logically equivalent to a disjunction of minterms.

Below we shall introduce, for each prime number p , a class of formulae we call L_p -*minterms*. An L_p -minterm $\mu_{\mathbf{v}}$ evaluates to $\frac{1}{\text{den } \mathbf{v}}$ at exactly one point $\mathbf{v} \in S_p^n$, while $\mu_{\mathbf{v}}(\mathbf{w}) = 0$ for all other points $\mathbf{v} \neq \mathbf{w} \in S_p^n$. We then proceed to show that every formula of L_p is logically equivalent to an $\oplus$ -sum of L_p -minterms.

The McNaughton function associated with an L_p -minterm is a specialised Schauder hat that we call *Brun hat*, because the notion is inspired by a well-known simultaneous diophantine approximation algorithm due to Brun [3, 4]. See [2] for an analysis of Brun’s algorithm and its variants.

2 An Algorithm Producing Brun Hats

We define Brun hats in intrinsic terms, and we then produce an algorithm that constructs precisely such objects.

Let $\text{cl } A$ denote the closure of the set $A \subseteq \mathbb{R}^n$. If A is a simplex, we write $\text{relint } A$ for its relative interior.

Definition 2 (Brun Expansion). Let $\mathbf{v} \in ([0, 1] \cap \mathbb{Q})^n$ and R be a unimodular k -simplex such that $\mathbf{v} \in \text{relint } R$. A finite sequence of edge subdivisions $R = \mathcal{R}_0, \mathcal{R}_1, \dots, \mathcal{R}_u$ is a Brun expansion at $\mathbf{v}$ from R iff for each $j \in \{0, \dots, u\}$ there exists exactly one k -simplex $S_j \in \mathcal{R}_j$ such that $\mathbf{v} \in \text{relint } S_j$. We say that S_u is obtained via $\mathcal{R}_0, \dots, \mathcal{R}_u$. A Brun expansion $\mathcal{R}_0, \dots, \mathcal{R}_u$ at $\mathbf{v}$ from R is maximal iff $\mathbf{v}$ is the Farey mediant of all the vertices of S_u and R is a simplex of a principal unimodular partition of $[0, 1]^n$.

Definition 3 (Brun Hat). Fix a principal partition $\mathcal{U}$ of $[0, 1]^n$, and a rational point $\mathbf{v} \in ([0, 1] \cap \mathbb{Q})^n$. Then there exists a unique dimension-minimal k -simplex $R_{\mathbf{v}}$ of $\mathcal{U}$ such that $\mathbf{v} \in \text{relint } R_{\mathbf{v}}$; let $\{R_i\} \subseteq \mathcal{U}$ be the (finite) collection of maximal simplices containing $R_{\mathbf{v}}$. A Brun hat $h_{\mathbf{v}}: [0, 1]^n \rightarrow [0, 1]$ at $\mathbf{v}$ is a McNaughton function having the following properties:

1. (Relative simpliciality.) $\text{cl}(\text{supp } h_{\mathbf{v}} \cap R_{\mathbf{v}})$ is a unimodular k -simplex S .
2. (Brunness.) S is obtainable via a maximal Brun expansion $\mathcal{R}_0, \dots, \mathcal{R}_u$ at $\mathbf{v}$ from $R_{\mathbf{v}}$.
3. (Completeness.) $h_{\mathbf{v}}$ coincides with the hat at $\mathbf{v}$ over the unimodular partition $\mathcal{S}^*$ defined as follows. Let $\{\mathbf{v}_k\}$ be the list of all vertices of all R_i 's, except those of $R_{\mathbf{v}}$; let $\{\mathbf{w}_j\}$ be the list of all vertices of all simplices in $\mathcal{R}_u$. Define the partition $\mathcal{S}$ as the one whose simplices are precisely those of the form $\text{conv}(\mathbf{v}_{k_1}, \dots, \mathbf{v}_{k_s}, \mathbf{w}_{j_1}, \dots, \mathbf{w}_{j_t})$, where $\{\mathbf{v}_{k_1}, \dots, \mathbf{v}_{k_s}\}$ span a simplex of $\mathcal{U}$ and $\{\mathbf{w}_{j_1}, \dots, \mathbf{w}_{j_t}\}$ span a simplex of $\mathcal{R}_u$, together with all simplices of $\mathcal{U}$ not contained in any R_i . Then let $\mathcal{S}^*$ be the partition obtained from $\mathcal{S}$ via subdivision through the Farey mediant of all vertices of $S \in \mathcal{S}$.

Remark 1.

1. It is a simple exercise to show that $\mathcal{S}$ (and hence $\mathcal{S}^*$) indeed is a unimodular partition of $[0, 1]^n$. In fact, $\mathcal{S}$ is unimodular when S is *any* unimodular simplex, not necessarily Brun. A detailed proof derived for different purposes may be found in [1].
2. Brun hats are in bijective correspondence with maximal Brun expansions.
3. For any $\mathbf{v} \neq \mathbf{w} \in ([0, 1] \cap \mathbb{Q})^n$ such that $\text{den } \mathbf{w} \leq \text{den } \mathbf{v}$ we have that $h_{\mathbf{v}}(\mathbf{w}) = 0$.

Throughout the paper we shall identify the formula φ of $\mathbf{L}_{\omega}$ in the variables $x_1, \dots, x_n$ with the McNaughton function $\varphi^{\mathcal{M}_n}: [0, 1]^n \rightarrow [0, 1]$ given by the interpretation of the term φ in the free MV algebra $\mathcal{M}_n$.

Algorithm 1. Let $\mathbf{v} \in ([0, 1] \cap \mathbb{Q})^n$ and let $S \subseteq [0, 1]^n$ be a unimodular n -simplex such that $\mathbf{v} \in S$. Let $\mathbf{b}_1, \dots, \mathbf{b}_{n+1}$ be the vertices of S , and assume $\vartheta_1, \dots, \vartheta_{n+1}$ are formulae such that, for each $i \neq j \in \{1, \dots, n+1\}$, we have ϑ_i is linear over S , and $\vartheta_i(\mathbf{b}_i) = \frac{1}{\det \mathbf{b}_i}$ while $\vartheta_i(\mathbf{b}_j) = 0$. We call $\Theta = (\vartheta_1, \dots, \vartheta_{n+1})$ the initial base for S . For uniquely determined non-negative integers $a_1, \dots, a_{n+1}$,

$$H(\mathbf{v}) = \sum_{i=1}^{n+1} a_i H(\mathbf{b}_i).$$

Let $\mathbf{w} = (a_1, \dots, a_{n+1})$. Let $N = \{i \mid a_i \neq 0\}$ and $m = |N|$.

We construct the sequences $\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_k, \dots$ and $\Psi_0, \Psi_1, \dots, \Psi_k, \dots$ inductively as follows:

$$\mathbf{w}_0 := (a_1, \dots, a_{n+1}), \quad \Psi_0 := (\vartheta_1, \dots, \vartheta_{n+1}).$$

For any integer $k \geq 0$, either $\mathbf{w}_k = \sum_{j \in N} \mathbf{e}_j$ and then we terminate the algorithm, or we construct $\mathbf{w}_{k+1}$ as follows:

Let us display $\mathbf{w}_k$ as $(w_{k,1}, \dots, w_{k,n+1})$ and Ψ_k as $(\psi_{k,1}, \dots, \psi_{k,n+1})$. Let p_k and q_k be two non-deterministically chosen elements in N such that

$$w_{k,p_k} - w_{k,q_k} > 0.$$

(Call p_k and q_k the first and second pivot at step k , respectively.) Then

$$\mathbf{w}_{k+1} := \mathbf{w}_k - w_{k,q_k} \mathbf{e}_{p_k}$$

and

$$\Psi_{k+1} := (\psi_{k+1,1}, \dots, \psi_{k+1,n+1}),$$

for

$$\psi_{k+1,i} = \begin{cases} \psi_{k,p_k} \ominus \psi_{k,q_k} & \text{if } i = p_k, \\ \psi_{k,i} & \text{otherwise.} \end{cases}$$

Remark 2. A heuristic to choose pivots which makes Algorithm 1 deterministic is the following: at each step k , let $p_k = \text{lex min}\{i \mid w_{k,i} = \max_{j \in N} \{w_{k,j}\}\}$ and $q_k = \text{lex min}\{i \mid w_{k,i} = \max_{j \in N} \{w_{k,j} \mid w_{k,j} \neq w_{k,p_k}\}\}$. It can be shown that the length of the sequence of steps computed by the algorithm under this choice of pivots is not always minimal.

Lemma 1. Algorithm 1 always terminates for all admissible choices of the pivots at each step. More specifically, for some $h < -m + \sum_{j=1}^{n+1} a_j$, we have $\mathbf{w}_h = \sum_{j \in N} \mathbf{e}_j$.

Proof. By the constraint on the admissible choices of pivots in Algorithm 1, we have at each step $i > 0$ that $w_{i,j} = 0$ iff $j \notin N$. Moreover, exactly one component of $\mathbf{w}_i$ is strictly smaller than the component with the same index of $\mathbf{w}_{i-1}$, while all the others remain unchanged. $\mathbf{w}_h = \sum_{j \in N} \mathbf{e}_j$ is reached after at most $-m + \sum_{i=1}^{n+1} a_i$ steps. $\square$

We call the $(2h)$ -tuple $P = (p_0, q_0, \dots, p_{h-1}, q_{h-1})$ the *pivot vector*. For each input $(\mathbf{v}, S, \Theta)$ to Algorithm 1 there are finitely many admissible pivot vectors.

Lemma 2. *Let $\mathbf{v} \in S \cap \mathbb{Q}^n$, and let h be the last step of Algorithm 1 under some pivot choice P . For each integer $i \in \{0, \dots, h\}$, let $\mathbf{w}_i$ and Ψ_i be as in Algorithm 1. Then there exists an $(n+1)$ -tuple of affinely independent points $(\mathbf{b}_{i,1}, \dots, \mathbf{b}_{i,n+1})$ of $([0, 1] \cap \mathbb{Q})^n$ such that*

1. *Let S_i denote $\text{conv}(\mathbf{b}_{i,1}, \dots, \mathbf{b}_{i,n+1})$. If $i > 0$, then S_i is obtained from S_{i-1} by one edge subdivision.*
2. *$\mathbf{w} = \sum_{j=1}^{n+1} w_{i,j} H(\mathbf{b}_{i,j})$, hence $\mathbf{v}$ lies in $\text{conv}(\mathbf{b}_{i,1}, \dots, \mathbf{b}_{i,n+1})$.*
3. *For each $j \in \{1, \dots, n+1\}$, if $i > 0$ then $\text{den } \mathbf{b}_{i-1,j} \leq \text{den } \mathbf{b}_{i,j}$.*
4. *For each $j \in \{1, \dots, n+1\}$, $\psi_{i,j}$ is linear over $\text{conv}(\mathbf{b}_{i,1}, \dots, \mathbf{b}_{i,n+1})$.*
5. *For each $j \in \{1, \dots, n+1\}$, $\psi_{i,j}(\mathbf{b}_{i,j}) = 1/\text{den } \mathbf{b}_{i,j}$, while, for $l \neq j$, $\psi_{i,j}(\mathbf{b}_{i,l}) = 0$.*

Proof. For $i = 0$, there is nothing to prove. Let $p = p_{i-1}$ and $q = q_{i-1}$ be respectively the first and second pivot at step $i-1$. Let $\mathbf{b}_{i,j}$ be the Farey mediant of $\{\mathbf{b}_{i-1,p}, \mathbf{b}_{i-1,q}\}$. Let $H(\mathbf{b}_{i,j}) = H(\mathbf{b}_{i-1,j})$ for each $q \neq j \in \{1, \dots, n+1\}$, and $H(\mathbf{b}_{i,q}) = H(\mathbf{b}_{i-1,p}) + H(\mathbf{b}_{i-1,q})$. Hence S_i is an n -dimensional simplex obtained from S_{i-1} by one edge subdivision. Note that $w_{i,p}H(\mathbf{b}_{i,p}) + w_{i,q}H(\mathbf{b}_{i,q}) = (w_{i-1,p} - w_{i-1,q})H(\mathbf{b}_{i-1,p}) + w_{i-1,q}(H(\mathbf{b}_{i-1,q}) + H(\mathbf{b}_{i-1,p})) = w_{i-1,p}H(\mathbf{b}_{i-1,p}) + w_{i-1,q}H(\mathbf{b}_{i-1,q})$. Then 1, 2, 3 are settled.

An easy computation involving linear functions shows that $\psi_{i-1,p} \geq \psi_{i-1,q}$ over S_i and $\psi_{i-1,p}(\mathbf{b}_{i,q}) = \psi_{i-1,q}(\mathbf{b}_{i,q}) = 1/\text{den } \mathbf{b}_{i,q}$, hence $\psi_{i,p}(\mathbf{b}_{i,q}) = 0$ and $\psi_{i,p}$ is linear over all S_i . This settles the remaining points. $\square$

Remark 3. Lemma 2.1 and a moment's reflection show that if $\mathbf{v} \in \text{relint } S$, then each Brun expansion $\mathcal{R}_0, \dots, \mathcal{R}_u$ at $\mathbf{v}$ from S is obtainable from Algorithm 1 under a suitable choice of the pivots P .

Definition 4. *Let $(\mathbf{v}, S, \Theta)$ be input to Algorithm 1, and let h be the last step of the algorithm under a pivot choice P . The Brun hat at $\mathbf{v}$ relative to S (or to Θ) under P is the formula:*

$$\beta_{\mathbf{v}}^{S, \Theta, P} = \bigwedge_{j \in N} \psi_{h,j}.$$

We take the liberty of dropping some of the superscripts S, Θ, P when the corresponding parameters of $\beta_{\mathbf{v}}$ are clear from the context or are immaterial.

Lemma 3. *For any relative Brun hat $\beta_{\mathbf{v}}^S$, let $T = \text{cl}(\text{supp}(\beta_{\mathbf{v}}^S) \cap S)$, $F = \text{conv}(\{\mathbf{b}_j \mid j \in N\})$ (for $\mathbf{b}_1, \dots, \mathbf{b}_{n+1}$ the vertices of S) and $F' = T \cap F$. Then:*

- T is an n -simplex;
- F' is obtained via a Brun expansion at $\mathbf{v}$ from the unique $(m-1)$ -dimensional face of S containing $\mathbf{v}$ in its relative interior, which is F .

- $\beta_{\mathbf{v}}^S$ is linear over each simplex in the Farey subdivision of T through $\mathbf{v}$.
- $\beta_{\mathbf{v}}^S(\mathbf{v}) = 1/\text{den } \mathbf{v}$, while, for every point $\mathbf{v} \neq \mathbf{v}' \in S \cap \mathbb{Q}^n$, if $\text{den } \mathbf{v}' \leq \text{den } \mathbf{v}$, then $\beta_{\mathbf{v}}^S(\mathbf{v}') = 0$.

Proof. Assume Algorithm 1 terminates at the h th step, hence $\mathbf{w}_{h,j} = 1$ for all $j \in N$. Then we have $\mathbf{w} = \sum_{j \in N} H(\mathbf{b}_{h,j})$, and by Lemma 2 and Definition 4, all requirements are easily seen to be met. $\square$

Example 1. Let $\mathbf{v} = (2/11, 3/11)$. Note $\mathbf{v} \in \text{conv}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{0})$. Then, we set $\mathbf{w}_0 = (2, 3, 6)$ and $\Psi_0 = (x_1, x_2, \neg(x_1 \oplus x_2))$. In all examples in the paper we choose the pivots in accordance with Remark 2. In the following table we write x_3 for $\neg(x_1 \oplus x_2)$.

$\mathbf{w}_i$	Ψ_i	$\mathbf{b}_{i,1}$	$\mathbf{b}_{i,2}$	$\mathbf{b}_{i,3}$
$(2, 3, 6)$	x_1, x_2, x_3	$(1, 0, 1)$	$(0, 1, 1)$	$(0, 0, 1)$
$(2, 3, 3)$	$x_1, x_2, x_3 \ominus x_2$	$(1, 0, 1)$	$(0, 1, 2)$	$(0, 0, 1)$
$(2, 1, 3)$	$x_1, x_2 \ominus x_1, x_3 \ominus x_2$	$(1, 1, 3)$	$(0, 1, 2)$	$(0, 0, 1)$
$(2, 1, 1)$	$x_1, x_2 \ominus x_1, (x_3 \ominus x_2) \ominus x_1$	$(1, 1, 4)$	$(0, 1, 2)$	$(0, 0, 1)$
$(1, 1, 1)$	$x_1 \ominus (x_2 \ominus x_1), (x_2 \ominus x_1), (x_3 \ominus x_2) \ominus x_1$	$(1, 1, 4)$	$(1, 2, 6)$	$(0, 0, 1)$

Hence, $\beta_{\mathbf{v}}^S = (x_1 \ominus (x_2 \ominus x_1)) \wedge (x_2 \ominus x_1) \wedge ((\neg(x_1 \oplus x_2) \ominus x_2) \ominus x_1)$, and $\text{star}_S(\mathbf{v}) = \text{conv}((1/4, 1/4), (1/6, 1/3), (0, 0))$.

Remark 4. It is clear from Algorithm 1 (cf. also Example 1) that the rôle played by the sets of formulae Ψ_i is purely formal, and we can assume the initial $(n+1)$ -tuple Ψ_0 is just the *formal base* $(x_1, \dots, x_{n+1})$, to be replaced by the actual *initial base* $\Theta = (\vartheta_1, \dots, \vartheta_{n+1})$ just after the construction of Ψ_h , for h the last step of the algorithm on input $(\mathbf{v}, S, \Psi_0)$. Let τ_{Θ} be the substitution of variables defined by $\tau_{\Theta}(x_i) = \vartheta_i$ for all $i \in \{1, \dots, n+1\}$. We summarise these considerations in the following lemma:

Lemma 4. *Let $\Psi_0 = (x_1, x_2, \dots, x_{n+1})$ and $(\mathbf{v}, S, \Psi_0)$ be the input to Algorithm 1. Then, for each admissible pivot vector P*

$$\beta_{\mathbf{v}}^{\Theta, P} = \tau_{\Theta}(\beta_{\mathbf{v}}^{\Psi_0, P}).$$

Lemma 5. *Let $\mathbf{v} \in T_n^{\sigma} \cap \mathbb{Q}^n$ for some permutation σ . There is an algorithm to produce a formula coinciding with $h_{\mathbf{v}}$ over T_n^{σ} , for each Brun hat $h_{\mathbf{v}}$.*

Proof. Let $\mathbf{v}' = \sigma^{-1}(\mathbf{v}) \in T_n$. Display $H(\mathbf{v}')$ as $(c_1, \dots, c_n, c_{n+1})$. Then $H(\mathbf{v}') = (c_{n+1} - c_n)H(\mathbf{0}) + (c_n - c_{n-1})H(\mathbf{e}_n) + \dots + (c_2 - c_1)H(\sum_{i=2}^n \mathbf{e}_i) + c_1 H(\sum_{i=1}^n \mathbf{e}_i)$. We set

$$\Theta_{\sigma} = (\neg\sigma(x_n), \sigma(x_n) \ominus \sigma(x_{n-1}), \dots, \sigma(x_2) \ominus \sigma(x_1), \sigma(x_1)),$$

and $\mathbf{w}_0 = (c_{n+1} - c_n, c_n - c_{n-1}, \dots, c_2 - c_1, c_1)$. By Definition 4 and Lemmata 3, 4, for each admissible pivot vector P the formula $\beta_{\mathbf{v}}^{\Theta_{\sigma}, P}$ coincides over T_n^{σ} with a Brun hat at $\mathbf{v}$. By Remarks 1.2 and 3, we only need to run Algorithm 1 under each one of the finitely many admissible choices for P . $\square$

Example 2. Consider the point $\mathbf{v} = (6/11, 9/11) \in \text{relint } T_2$. Then we set $\Theta = (\neg x_2, x_2 \ominus x_1, x_1)$, and $\mathbf{w}_0 = (2, 3, 6)$. The relative Brun hat $\beta_{\mathbf{v}}^{\Theta}$ is the infimum of the three formulae in the last line of the table of Example 1, after the substitution τ_{Θ} specified by $\tau_{\Theta}(x_1) = \neg x_2$, $\tau_{\Theta}(x_2) = x_2 \ominus x_1$, $\tau_{\Theta}(x_3) = x_1$. Moreover, $\mathbf{v}$ is the Farey median of the vertices of the 2-simplex $\text{conv}((1/2, 3/4), (1/2, 5/6), (1, 1))$. The hat for $(9/11, 6/11)$ is obtained from the previous one by substituting all occurrences of x_1 with x_2 , and viceversa.

If $\mathbf{v} \in \text{relint } T_n^{\sigma}$, or, more generally, if there exists exactly one T_n^{σ} to which $\mathbf{v}$ belongs, then each $\beta_{\mathbf{v}}^{\Theta_{\sigma}, P}$ in Lemma 5 is a Brun hat at $\mathbf{v}$. There remains to consider the case when $\mathbf{v}$ lies on a face common to several n -simplices T_n^{σ} .

Theorem 1. *Let $\mathbf{v} \in ([0, 1] \cap \mathbb{Q})^n$. Then there is an algorithm to produce every Brun hat at $\mathbf{v}$.*

Proof. Let $\mathbf{v} \in ([0, 1] \cap \mathbb{Q})^n$, and let $\Delta = \{\sigma \mid \mathbf{v} \in T_n^{\sigma}\}$. Let $F = \bigcap_{\Delta} T_n^{\sigma}$. For some $0 \leq k \leq n$, F is a k -dimensional face of T_n^{σ} , for every $\sigma \in \Delta$. Let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_{k+1}\}$ be the set of vertices of F . Recall that each $\mathbf{x}_i \in \{0, 1\}^n$. Let Θ_{σ} be the initial base for T_n^{σ} as constructed in Lemma 5, and let $\vartheta_{\sigma,1}, \dots, \vartheta_{\sigma,k+1}$ be the formulae in Θ_{σ} corresponding to vertices in X , that is, $\vartheta_{\sigma,j}$ is the only formula in Θ_{σ} such that $\vartheta_{\sigma,j}(\mathbf{x}_j) = 1$. Let $\lambda_j = \bigwedge_{\Delta} \vartheta_{\sigma,j}$, for every $j \in \{1, \dots, k+1\}$. Observe that $\lambda_j(\mathbf{x}_j) = 1$, and $\lambda_j(\mathbf{y}) = 0$ for all vertices $\mathbf{x}_j \neq \mathbf{y} \in \{0, 1\}^n \cap \bigcup_{\Delta} T_n^{\sigma}$. Moreover $\lambda_j(\mathbf{x}_j)$ is linear over each n -simplex of $\mathcal{U}_n$. Fix arbitrarily $\varsigma \in \Delta$. Let the initial base be $\Theta = (\vartheta_1, \dots, \vartheta_{n+1})$, where $\vartheta_i = \lambda_j$ if $\mathbf{v}_{\varsigma(i)} = \mathbf{x}_j \in X$, $\vartheta_i = \perp$ otherwise. Observe that $\mathbf{v}_{\varsigma(i)} = \mathbf{x}_j$ implies $\mathbf{v}_{\sigma(i)} = \mathbf{x}_j$ for all $\sigma \in \Delta$. Then, by Lemma 5, $\{\beta_{\mathbf{v}}^{\Theta_{\sigma}, P} \mid P \text{ admissible}\}$ is the set of all Brun hats at $\mathbf{v}$ (note that components of Θ equal to $\perp$ are never used to construct any $\beta_{\mathbf{v}}^{\Theta_{\sigma}, P}$). $\square$

Example 3. Let $\mathbf{v} = (2/5, 2/5, 3/5)$. Then $\mathbf{v}$ lies on the frontier of T_3 and of T_3^{σ} , for $\sigma(x_1) = x_2$, $\sigma(x_2) = x_1$, $\sigma(x_3) = x_3$. Let ς be the identity. Then $\Theta = (\neg x_3 \wedge \neg x_3, (x_3 \ominus x_2) \wedge (x_3 \ominus x_1), \perp, x_1 \wedge x_2)$ and $\mathbf{w}_0 = (2, 1, 0, 2)$. The Brun hat at $\mathbf{v}$ is $(\neg x_3 \ominus ((x_3 \ominus x_2) \wedge (x_3 \ominus x_1))) \wedge ((x_3 \ominus x_2) \wedge (x_3 \ominus x_1)) \wedge ((x_1 \wedge x_2) \ominus (\neg x_3 \ominus ((x_3 \ominus x_2) \wedge (x_3 \ominus x_1))))$. Note that $\vartheta_3 = \perp$ is never used.

3 Minterms and Normal Forms for Co-atomic Lukasiewicz Logics

Recall from Section 1 the shorthands $n\varphi$ and φ^n for iterated disjunctions and conjunctions of φ with itself. To improve readability we shall write $m\varphi^n$ for $m(\varphi^n)$. Throughout this section let p be a prime number. Then L_p is a co-atom in the lattice $\mathcal{L}$. Let $R_p^n = S_p^n \setminus \{0, 1\}^n$. Let $\beta_{\mathbf{v}}$ the Brun hat output by the algorithm of Theorem 1 under the choice of pivots of Remark 2. The properties of Brun hats ensure that each $\beta_{\mathbf{v}}$, for $\mathbf{v} \in R_p^n$ is an L_p -minterm for $\mathbf{v}$. To deal with vertices $\mathbf{v} \in \{0, 1\}^n$, we just take $\beta_{\mathbf{v}}^p$ — note that $\beta_{\mathbf{v}}$ coincides with the Boolean minterm at $\mathbf{v}$.

Theorem 2. For any formula φ in the variables $x_1, \dots, x_n$:

$$\mathbb{L}_p \models \varphi \leftrightarrow \bigoplus_{\mathbf{v} \in R_p^n} ((\varphi(\mathbf{v})p)\beta_{\mathbf{v}}) \oplus \bigoplus_{\mathbf{v} \in \{0,1\}^n} (\varphi(\mathbf{v})\beta_{\mathbf{v}}^p).$$

Proof. Immediate, after noticing that for each $\mathbf{v} \in \{0,1\}^n$ and each $\mathbf{w} \in R_p^n$, we have $\beta_{\mathbf{v}}^p(\mathbf{w}) = 0$. $\square$

We define inductively on the structure of φ the *Brun normal form* $BNF(\varphi)$. For each $i \in \{1, \dots, n\}$ and each $j \in \{1, \dots, p\}$ let $\Phi(i, j) = \{\mathbf{v} \mid \mathbf{v} \cdot \mathbf{e}_i = j/p\}$. Then

$$BNF_p(x_i) := \bigoplus_{j=1}^{p-1} \bigoplus_{\mathbf{v} \in \Phi(i, j)} j\beta_{\mathbf{v}} \oplus \bigoplus_{\mathbf{v} \in \Phi(i, p)} \beta_{\mathbf{v}}^p.$$

Assume that for every subformula ζ of φ there are uniquely determined non-negative integers $\{k_{\mathbf{v}}^{\zeta}\}_{\mathbf{v} \in S_n^p}$ such that $BNF_p(\zeta) = \bigoplus_{\mathbf{v} \in R_p^n} k_{\mathbf{v}}^{\zeta}\beta_{\mathbf{v}} \oplus \bigoplus_{\mathbf{v} \in \{0,1\}^n} k_{\mathbf{v}}^{\zeta}\beta_{\mathbf{v}}^p$.

Then, if $\varphi = \neg\psi$:

$$BNF_p(\neg\psi) := \bigoplus_{\mathbf{v} \in R_p^n} (p - k_{\mathbf{v}}^{\psi})\beta_{\mathbf{v}} \oplus \bigoplus_{\mathbf{v} \in \{0,1\}^n} (1 - k_{\mathbf{v}}^{\psi})\beta_{\mathbf{v}}^p,$$

if $\varphi = \psi \oplus \vartheta$:

$$BNF_p(\psi \oplus \vartheta) := \bigoplus_{\mathbf{v} \in R_p^n} \min(p, k_{\mathbf{v}}^{\psi} + k_{\mathbf{v}}^{\vartheta})\beta_{\mathbf{v}} \oplus \bigoplus_{\mathbf{v} \in \{0,1\}^n} \min(1, k_{\mathbf{v}}^{\psi} + k_{\mathbf{v}}^{\vartheta})\beta_{\mathbf{v}}^p.$$

Corollary 1. For each formula φ in the variables $x_1, \dots, x_n$:

$$BNF_p(\varphi) = \bigoplus_{\mathbf{v} \in R_p^n} ((\varphi(\mathbf{v})p)\beta_{\mathbf{v}}) \oplus \bigoplus_{\mathbf{v} \in \{0,1\}^n} (\varphi(\mathbf{v})\beta_{\mathbf{v}}^p).$$

Further,

$$\mathbb{L}_p \models \varphi \quad \text{iff} \quad BNF_p(\varphi) = \bigoplus_{\mathbf{v} \in R_p^n} p\beta_{\mathbf{v}} \oplus \bigoplus_{\mathbf{v} \in \{0,1\}^n} \beta_{\mathbf{v}}^p.$$

Brun normal forms have the following strong properties:

Corollary 2. For each integer $q < p$ and for any formula φ in the variables $x_1, \dots, x_n$:

1. $\mathbb{L}_q \models BNF_p(\varphi) \leftrightarrow \bigoplus_{\mathbf{v} \in \{0,1\}^n} \varphi(\mathbf{v})\beta_{\mathbf{v}}^p$.
2. $\mathbb{L}_q \models \neg \bigoplus_{\mathbf{v} \in R_p^n} t_{\mathbf{v}}\beta_{\mathbf{v}}$ for any set of non-negative integers $\{t_{\mathbf{v}} \mid \mathbf{v} \in R_p^n\}$.
3. $B \models \neg\varphi$ iff $\mathbb{L}_q \models \neg BNF_p(\varphi)$.

Proof. $\beta_{\mathbf{v}}(\mathbf{w}) = 0$ for all $\mathbf{w} \in ([0, 1] \cap \mathbb{Q})^n$ such that $\text{den } \mathbf{w} < \text{den } \mathbf{v}$. $\square$

4 Normal Forms for All Finite-Valued Łukasiewicz Logics

Algorithm 1 does not directly yield all L_k -minterms for each positive integer k . To see the problem, consider the Brun hat $\beta_{(\frac{1}{2}, \frac{1}{2})} = (\neg x_2 \wedge \neg x_1) \wedge (x_1 \wedge x_2)$. Then we have $\beta_{(\frac{1}{2}, \frac{1}{2})}(\frac{1}{4}, \frac{1}{4}) = \frac{1}{4}$, while an L_4 -minterm at $(\frac{1}{2}, \frac{1}{2})$ must evaluate to 0 over $(\frac{1}{4}, \frac{1}{4})$. However, we can easily extend our basic algorithm to produce any needed L_k -minterm as a suitable combination of Brun hats.

Theorem 3. *Let $\mathbf{v} \in ([0, 1] \cap \mathbb{Q})^n$ be a point of denominator $d = \text{den } \mathbf{v}$ and let k be a multiple of d . There is an algorithm to produce an L_k -minterm $\mu_{\mathbf{v}}^{(k)}$ at $\mathbf{v}$.*

Proof. We proceed by induction on k . If k is a prime number then either $d = k$ or $d = 1$. If $d = k$ then we just set $\mu_{\mathbf{v}}^{(k)} = \beta_{\mathbf{v}}$, while if $d = 1$ we set $\mu_{\mathbf{v}}^{(k)} = \beta_{\mathbf{v}}^k$, as in Theorem 2. If $k = cd$ for some $c > 1$ then we set $\mu_{\mathbf{v}}^{(k)} = (d\mu_{\mathbf{v}}^{(d)})^c \wedge \mu_{\mathbf{v}}^{(d)}$. It is easy to check that $\mu_{\mathbf{v}}^{(k)}$ is an L_k -minterm at $\mathbf{v}$, though in general not a Brun hat.

Corollary 3. *For any positive integer k and for any formula φ we have*

$$L_k \models \varphi \leftrightarrow \bigoplus_{\mathbf{v} \in S_k^n} (\varphi(\mathbf{v}) \text{den } \mathbf{v}) \mu_{\mathbf{v}}^{(k)}.$$

References

1. Aguzzoli, S., Marra, V.: Finitely presented MV algebras with finite automorphism group. *Submitted*.
2. Brentjes, A. J.: Multi-dimensional continued fraction algorithms. *Mathematical Centre Tracts*. Amsterdam **145**. (1981)
3. Brun, V.: En generalisation av kjederbrøken I. *Skr. Vidensk. Selsk. Kristiania* **6** (1919)
4. Brun, V.: En generalisation av kjederbrøken II. *Skr. Vidensk. Selsk. Kristiania* **6** (1920)
5. Chang, C. C.: Algebraic analysis of many valued logics. *Trans. Amer. Math. Soc.* **88** (1958) 467–490
6. Cignoli, R., D'Ottaviano I. M. L., Mundici, D.: *Algebraic Foundations of Many-Valued Reasoning*. Trends in Logic, Kluwer, Dordrecht **7** (2000)
7. Ewald, G.: *Combinatorial Convexity and Algebraic Geometry*. Graduate Texts in Mathematics, Springer-Verlag, Berlin **168** (1996)
8. Grigolia, R.: An algebraic analysis of Łukasiewicz–Tarski's n -valued logical systems. *Selected Papers on Łukasiewicz Sentential Calculi*, R. Wójcicki and G. Malinowski eds., Ossolineum, Wrocław (1977) 81–92
9. McNaughton, R.: A theorem about infinite-valued sentential logic. *Journal of Symbolic Logic* **16** (1951) 1–13
10. Mundici, D.: A constructive proof of McNaughton's Theorem in infinite-valued logics. *Journal of Symbolic Logic* **59** (1994) 596–602
11. Panti, G.: A geometric proof of the completeness of the Łukasiewicz calculus. *Journal of Symbolic Logic* **60** (1995) 563–578
12. Rose, A., Rosser, J.B.: Fragments of many-valued statement calculi. *Trans. Amer. Math. Soc.* **87** (1958) 1–53

Poset Representation for Gödel and Nilpotent Minimum Logics

Stefano Aguzzoli¹, Brunella Gerla², and Corrado Manara²

¹ Dipartimento di Scienze dell'Informazione,
Università degli Studi di Milano,
Via Comelico 39-41, 20135 Milano, Italy
`aguzzoli@dsi.unimi.it`

² Dipartimento di Matematica e Informatica,
Università degli Studi di Salerno,
Via Ponte don Melillo, 84084 Fisciano (SA), Italy
`{bgerla, cmanara}@unisa.it`

Abstract. MTL is the logic of all left-continuous t -norms and their residua. Its algebraic semantics is constituted by the variety $\mathbb{V}(\text{MTL})$ of MTL-algebras. Among schematic extensions of MTL there are infinite-valued logics $\mathcal{L}$ such that the finitely generated free algebras in the corresponding subvariety $\mathbb{V}(\mathcal{L})$ of $\mathbb{V}(\text{MTL})$ are finite. In this paper we focus on Gödel and Nilpotent Minimum logics. We give concrete representations of their associated free algebras in terms of finite algebras of sections over finite posets.

1 Introduction

Concrete representations of free algebras in varieties constituting the algebraic semantics of a logic are powerful analysis tools for that logic. A notable example is McNaughton's representation theorem [12] of free n -generated MV algebra (the variety of MV algebras is the algebraic counterpart of Łukasiewicz logic) as the set of continuous piecewise linear functions from $[0, 1]^n$ to $[0, 1]$, with each of the finitely many pieces having integer coefficients, equipped with pointwise defined operations. McNaughton's theorem allowed Mundici to prove NP-completeness of the satisfiability problem for Łukasiewicz infinite-valued logic [13].

In some cases free algebras turn out to be even simpler objects. In this paper we shall deal with some infinite-valued logics corresponding with subvarieties of $\mathbb{V}(\text{MTL})$ whose finitely generated free algebras are finite, and we represent them combinatorially as algebras of sections over finite posets. As a byproduct, we shall count the number of elements in the free algebras.

In particular we shall deal with Gödel and Nilpotent Minimum logics. Functional representations analogous to McNaughton's for free algebras associated with those logics have been given in [6] and [16], while [10] deals with Gödel algebras as the subvariety L of Heyting algebras where *prelinearity* $(x \rightarrow y) \vee (y \rightarrow x) = 1$ holds, and [11] counts the elements of free L -algebras.

It turns out that operations of free algebras in those varieties are understood in order-theoretical terms (they are comparisons between variables or negated variables), while arithmetic plays no significant rôle: this is in sharp contrast for instance with Łukasiewicz logic, where operations are truncated sums and subtractions. The same approach can be also used for other logics with similar characteristics. In this paper we shall also provide the poset representation for the logic of Gödel hoops, while, for lack of space, the description of the logic NMG [15], corresponding to the t -norm obtained as ordinal sum of nilpotent minimum and Gödel t -norms, will appear elsewhere.

2 Preliminary Notions and Background

A t -norm is a binary operation from $[0, 1]^2$ into $[0, 1]$ that is associative, commutative, non-decreasing in both arguments, and has 0 as absorbing element and 1 as unit. Given a left-continuous t -norm $\odot$, its associated *residuum* is the binary operation $x \rightarrow y = \max\{z \mid z \odot x \leq y\}$. The algebra $[0, 1]_{\odot} = ([0, 1], \odot, \rightarrow, \wedge, 0)$, where $x \wedge y = \min(x, y)$, is called a *standard algebra*.

Monoidal t -norm based logic (MTL, for short) is introduced in [3]. The language of MTL contains the constant 0 and binary connectives $\odot, \rightarrow, \wedge$. Its only rule is Modus Ponens, and its axioms are the following:

(A1) $(\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \vartheta) \rightarrow (\varphi \rightarrow \vartheta))$	(A4c) $(\varphi \wedge \psi) \rightarrow (\psi \wedge \varphi)$
(A2) $(\varphi \odot \psi) \rightarrow \varphi$	(A5a) $(\varphi \rightarrow (\psi \rightarrow \vartheta)) \rightarrow ((\varphi \odot \psi) \rightarrow \vartheta)$
(A3) $(\varphi \odot \psi) \rightarrow (\psi \odot \varphi)$	(A5b) $((\varphi \odot \psi) \rightarrow \vartheta) \rightarrow (\varphi \rightarrow (\psi \rightarrow \vartheta))$
(A4a) $(\varphi \odot (\varphi \rightarrow \psi)) \rightarrow (\varphi \wedge \psi)$	(A6) $((\varphi \rightarrow \psi) \rightarrow \vartheta) \rightarrow (((\psi \rightarrow \varphi) \rightarrow \vartheta) \rightarrow \vartheta)$
(A4b) $(\varphi \wedge \psi) \rightarrow \varphi$	(A7) $0 \rightarrow \varphi$

The algebraic counterpart of MTL, via the usual Lindenbaum construction, is the variety $\mathbb{V}(\text{MTL})$ of MTL-algebras. MTL turns out to be the logic of all left-continuous t -norms and their residua, in the sense that $\mathbb{V}(\text{MTL})$ is generated by the set of standard algebras [3].

Each MTL-algebra is lattice-ordered by $x \leq y$ iff $x \rightarrow y = 0 \rightarrow 0$ (for standard algebras this order is total). Note that the top element 1 is $0 \rightarrow 0$ and $x \wedge y = \inf(x, y)$. The connective $\vee$ whose interpretation is $\sup(x, y)$ in every MTL-algebra, can be defined as $\varphi \vee \psi := ((\varphi \rightarrow \psi) \rightarrow \psi) \wedge ((\psi \rightarrow \varphi) \rightarrow \varphi)$. In each MTL-algebra the unary connective of negation is defined by $\neg\varphi := \varphi \rightarrow 0$.

Hájek's *Basic logic* (BL for short, [9]) is obtained from the axioms of MTL by replacing (A4a), (A4b), (A4c) with (A4) $(\varphi \odot (\varphi \rightarrow \psi)) \rightarrow (\psi \odot (\psi \rightarrow \varphi))$.

The algebraic counterpart of BL is the variety $\mathbb{V}(\text{BL})$ of BL-algebras. BL is the logic of all continuous t -norms and their residua, in the sense that $\mathbb{V}(\text{BL})$ is generated by all standard algebras $[0, 1]_{\odot}$ for $\odot$ a continuous t -norm [1]. In BL the connective $\wedge$ is derived from $\odot$ and $\rightarrow$ as follows: $\varphi \wedge \psi := \varphi \odot (\varphi \rightarrow \psi)$.

Each axiomatic extension $\mathcal{L}$ of MTL determines a subvariety $\mathbb{V}(\mathcal{L})$ of the variety $\mathbb{V}(\text{MTL})$ of MTL-algebras. Clearly, $\mathbb{V}(\text{BL})$ is a subvariety of $\mathbb{V}(\text{MTL})$, and each schematic extension of BL is a schematic extension of MTL, too. We shall denote the free n -generated algebra in a variety $\mathbb{V}(\mathcal{L})$ by $\text{Free}_n(\mathcal{L})$.

Gödel logic (G for short, [7]) is the axiomatic extension of BL given by adding the *idempotency* axiom $\varphi \rightarrow (\varphi \odot \varphi)$ to (A1), ..., (A7). The variety $\mathbb{V}(G)$ of Gödel algebras is formed by the BL-algebras satisfying the equation $x \odot x = x$. Gödel logic is standard complete in the strong sense, i.e., $\mathbb{V}(G)$ is generated by the standard Gödel algebra $([0, 1], \odot, \rightarrow, \wedge, 0)$ given by

$$x \odot y = \min(x, y), \quad x \rightarrow y = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise.} \end{cases}$$

Note that $(\varphi \wedge \psi) \rightarrow (\varphi \odot \psi)$ is a theorem of G. Since the converse implication holds in MTL, in Gödel algebras the interpretations of $\odot$ and $\wedge$ coincide. Hence, Gödel algebras may be specified as structures of type $(A, \wedge, \rightarrow, 0)$.

Nilpotent Minimum logic (NM for short, [3]) is the axiomatic extension of MTL obtained by adding the *involutiveness* axiom $\neg\neg\varphi \rightarrow \varphi$ and the so-called *weak nilpotent minimum* axiom $\neg(\varphi \odot \psi) \vee ((\varphi \wedge \psi) \rightarrow (\varphi \odot \psi))$. Note that $x \odot x \odot x = x$ holds in any algebra in the variety $\mathbb{V}(\text{NM})$ of NM-algebras. The following are logical equivalences of NM: $\varphi \rightarrow \psi \equiv \neg(\varphi \odot \neg\psi)$, $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$, $\varphi \vee \psi \equiv ((\varphi \rightarrow \psi) \rightarrow \psi) \odot (((\varphi \rightarrow \psi) \rightarrow \psi) \rightarrow ((\psi \rightarrow \varphi) \rightarrow \varphi))$. Hence, $0, \odot, \rightarrow$ suffice to express all formulas of NM.

In [3] it is proved that NM is standard complete in the strong sense, that is $\mathbb{V}(\text{NM})$ is generated by the standard algebra $([0, 1], \odot, \rightarrow, \wedge, 0)$ given by nilpotent minimum t-norm [5] and its residuum, i.e.,

$$x \odot y = \begin{cases} \min(x, y) & \text{if } x > 1 - y \\ 0 & \text{otherwise,} \end{cases} \quad x \rightarrow y = \begin{cases} 1 & \text{if } x \leq y \\ \max(1 - x, y) & \text{otherwise.} \end{cases}$$

Since the standard Gödel algebra $[0, 1]$ generates $\mathbb{V}(G)$, a well-known fact of universal algebra states that $\text{Free}_n(G)$ is the subalgebra of $[0, 1]^{[0, 1]^n}$ generated by the projections $x_i(t_1, \dots, t_n) = t_i$. The same fact can be applied to get $\text{Free}_n(\text{NM})$. Direct inspection of standard algebras shows that both Gödel and Nilpotent Minimum free n -generated algebras have a finite number of elements. We refer to element of $\text{Free}_n(G)$ and $\text{Free}_n(\text{NM})$ as Gödel functions and NM-functions, respectively.

We shall now give some definitions holding for any variety $\mathbb{V}(\mathcal{L})$ in which the free algebra over n generators is a finite set of piecewise linear functions from $[0, 1]^n$ into $[0, 1]$.

Definition 1. *The domains of linearity for a function $f \in \text{Free}_n(\mathcal{L})$ are the subsets of $[0, 1]^n$ over which f is linear. The linear functions arising as restriction of any function $f \in \text{Free}_n(\mathcal{L})$ to any domain of linearity are called basic functions of $\mathcal{L}$.*

In this paper we are interested in varieties in which the domains of linearity only depends on the order between basic functions, hence having the form

$$D = \{\mathbf{t} \in [0, 1]^n \mid f_1(\mathbf{t}) \prec_1 \dots \prec_{k-1} f_k(\mathbf{t})\},$$

where $f_1, \dots, f_k$ are basic functions and each $\prec_i$ is either $<$ or $=$. To each such domain we associate a *domain chain* $C_D = \langle A_D, \leq \rangle$ where $A_D \subseteq 2^{\{f_1, \dots, f_k\}}$

and elements $X \in A_D$ are such that for every $\mathbf{t} \in D$, $f_i(\mathbf{t}) = f_j(\mathbf{t})$ if and only if $f_i, f_j \in X$, and where $X \leq Y$ if and only if there exist $f_i \in X$ and $f_j \in Y$ with $i \leq j$. Viceversa, with any chain $C = \langle \{W_1, \dots, W_k\}, \leq \rangle$ such that $W_i \subseteq \{f_1, \dots, f_k\}$, $W_h \cap W_k = \emptyset$ for $h \neq k$ and $W_i \leq W_j$ iff $i \leq j$, we associate the domain

$$D_C = \left\{ \mathbf{t} \in [0, 1]^n \left| \begin{array}{l} f_i(\mathbf{t}) = f_j(\mathbf{t}) \text{ iff } f_i, f_j \in W_h, \\ f_i(\mathbf{t}) < f_j(\mathbf{t}) \text{ iff } f_i \in W_l \text{ and } f_j \in W_h \text{ with } l < h \end{array} \right. \right\}.$$

Notation. In the following we take the liberty of using the same symbol to denote connectives and their corresponding algebraic operations, as well as basic functions and their values, that is, by x_i we denote the i th propositional variable, the projection function $x_i(t_1, \dots, t_n) = t_i$ and the i th component of the n -tuple $\mathbf{x} = (x_1, \dots, x_n)$. If $f : A \rightarrow B$ and $D \subset A$, we denote by $f \upharpoonright D$ the restriction of f to D . For any (finite) set A , we denote by $|A|$ the cardinality of A .

3 Poset Basics

In this section we give some background on partially ordered sets (*posets*, for short), needed for our results. For more details see [2].

The (disjoint) union, or *horizontal sum* $A \cup B$ of two disjoint posets A and B is the poset formed by defining $x \leq y$ if and only if either $x, y \in A$ and $x \leq y$ in A , or $x, y \in B$ and $x \leq y$ in B . The operation $\cup$ is associative and commutative. The Hasse diagram for $A \cup B$ is formed by placing side by side diagrams of A and B .

The linear sum, or *vertical sum* $A \oplus B$ of two disjoint posets A and B is defined by taking the following ordered relation: $x \leq y$ if and only if either $x, y \in A$ and $x \leq y$ in A , or $x, y \in B$ and $x \leq y$ in B , or $x \in A$ and $y \in B$. The operation $\oplus$ is associative. The Hasse diagram for $A \oplus B$ is obtained by placing the diagram for A directly below the diagram for B and then adding a line segment from each maximal element of A to each minimal element of B . A special case of vertical sum is the *lifting* construction $A_\perp := \{\perp\} \oplus A$, with $\perp \notin A$. In the following we shall always assume that posets involved in horizontal and vertical sums are disjoint, by taking isomorphic copies of operands when necessary.

By nA we denote $A \cup A \cup \dots \cup A$ n -times. We write $\mathbf{1}$ to denote the poset containing only one element. A *chain* is a totally order poset. A chain with n elements is isomorphic to $\mathbf{1} \oplus \dots \oplus \mathbf{1}$, n times, which we denote by $\mathbf{n}$. Up to isomorphism, $\mathbf{n}_\perp = \mathbf{n} + \mathbf{1}$. For every chain $C = \{a_1, \dots, a_u\}$ with $a_1 < a_2 < \dots < a_u$ we write $C = \{a_1 < a_2 < \dots < a_u\}$.

The *dual* of a poset $\langle P, \leq \rangle$ is a poset $\langle P^\partial, \leq^\partial \rangle$ where $P^\partial = P$ and $x \leq^\partial y$ if and only if $y \leq x$.

Each poset $\langle A, \leq \rangle$ is order isomorphic to a poset $\langle o(A), \leq \rangle$ obtained by replacing each element of A with a copy of $\mathbf{1}$. We call $o(A)$ the *type* of A , since $o(A)$ retains only the order theoretic information about A .

Definition 2. A finite poset A is nice if its type $o(A)$ is described using only operations $\mathbf{1}$, $\cup$ and $\oplus$.

For example the set $\{a, b, c, d\}$ with the order defined by $a < c$, $a < d$, $b < c$ is not a nice poset. Every chain $\mathbf{n}$ is a nice poset. Note that for any family $\{C_i\}$ of chains, $o((\bigcup C_i)^\theta) = o(\bigcup C_i)$.

Let $M(A)$ be the set of maximal elements of a poset A and let $m(A)$ be the set of minimal elements of A .

Definition 3. Let A be a finite poset. A branch B of A is a chain of maximal length in A , i.e., is a set of elements $b_1 < \dots < b_u$ with $b_1 \in m(A)$, $b_u \in M(A)$ and such that if $B \subseteq B' \subseteq A$ and B' is a chain then $B = B'$ or $B' = A$.

We denote by $\mathcal{B}(A)$ the set of branches of A .

Definition 4. A section over A is a sequence of elements $(p_i)_{i \in I}$ of A such that for each branch $B \in \mathcal{B}(A)$ there exists exactly one $p_i \in B$. We shall denote a section over A by $[p_B]_{B \in \mathcal{B}(A)}$, for $p_B \in B$. The set of sections over A is denoted by $S(A)$.

Note that in general a section $[p_B]_{B \in \mathcal{B}(A)}$ is a sequence with a number of elements less or equal than $|\mathcal{B}(A)|$, since distinct branches may share a common element (see the end of Example 2). The set of sections $S(A)$ can be equipped with an order structure in the following way: for any two sections $[p_B]_{B \in \mathcal{B}(A)}$ and $[q_B]_{B \in \mathcal{B}(A)}$ we have $[p_B]_{B \in \mathcal{B}(A)} \leq [q_B]_{B \in \mathcal{B}(A)}$ if and only if for every $B \in \mathcal{B}(A)$, $p_B \leq q_B$, i.e., the order is defined *branch-wise*.

Note that if $A = A_1 \cup A_2$, then $S(A) = S(A_1) \times S(A_2)$ and if $A = A_1 \oplus A_2$ then $S(A) = S(A_1) \oplus S(A_2)$. Moreover, the number of elements of a nice poset and of the set of its sections is easily computed from its type:

$$\begin{array}{lll} |1| = 1 & |A \cup B| = |A| + |B| & |A \oplus B| = |A| + |B| \\ |S(1)| = 1 & |S(A \cup B)| = |S(A)| \cdot |S(B)| & |S(A \oplus B)| = |S(A)| + |S(B)| \end{array}$$

4 Poset Representation for Gödel Logic

The set of Gödel functions in n variables is properly included in the set of piecewise linear functions coinciding on each linear domain with one of the following *basic functions*: $0, x_1, \dots, x_n, 1$.

Let us consider domains of linearity of the form

$$D = \{\mathbf{x} \in [0, 1]^n \mid 0 \prec x_{\sigma(1)} \prec \dots \prec x_{\sigma(n)}\}, \quad (1)$$

where $\sigma \in \text{Sym}_n$ is a permutation of $\{1, \dots, n\}$ and each occurrence of $\prec$ is either $<$ or $=$. Hence domain chains have the form $C_D = \{W_1 < \dots < W_k\}$ where each W_i is a set of basic functions. Let $\mathcal{D}_n^G$ be the set of all domains D satisfying (1) and $\mathcal{C}_n^G = \{C_D \mid D \in \mathcal{D}_n^G\}$. Throughout this section we shall write $\mathcal{C}_n$ and $\mathcal{D}_n$ instead of $\mathcal{C}_n^G$ and $\mathcal{D}_n^G$.

Definition 5. Let P be a finite poset with a unique minimal element m . Then a prefix of P is a chain $V \subseteq P$ such that either $V = \emptyset$ or $m \in V$ and $P \setminus V = \{p \in P \mid p > \max V\}$. The length of a prefix is its cardinality.

Let $G_D := C_D \oplus \{1\} = \{W_1 < W_2 < \dots < W_k < \{1\}\}$.

Example 1. The domain $D = \{(x, y, z) \mid 0 < x = y < z\}$ is associated with the chain $C_D = \{\{0\} < \{x, y\} < \{z\}\}$. Further $G_D = \{\{0\} < \{x, y\} < \{z\} < \{1\}\}$ and $\{\{0\} < \{x, y\}\}$ is a prefix of C_D of length 2.

Hence representation theorem of Gödel functions in [6], [11] can be expressed as follows:

Theorem 1. *The free Gödel algebra over n generators is the algebra of functions $f: [0, 1]^n \rightarrow [0, 1]$ such that:*

1. f is piecewise linear, each piece being a basic function.
2. f is linear over each domain $D \in \mathcal{D}_n$.
3. Let $U = \{U_1 < \dots < U_u\}$ and $V = \{V_1 < \dots < V_v\}$ be two domain chains and let $W = \{W_1 < \dots < W_k\}$ be their longest common prefix. Then $f|_{D_U} \in W_i$ if and only if $f|_{D_V} \in W_i$, for each $i \in \{1, \dots, k\}$.

Proposition 1. *For each domain $D \in \mathcal{D}_n$, the algebra of restrictions of Gödel functions to D is isomorphic to G_D as a Gödel chain. Further, for each $i \in \{1, \dots, n\}$, let $(W_D^i)_{D \in \mathcal{D}_n}$ be the i -th projection, i.e., the unique element of the direct product $\prod_{D \in \mathcal{D}_n} G_D$ such that $x_i \in W_D^i$. Then the free Gödel algebra over n generators is the subalgebra of $\prod_{D \in \mathcal{D}_n} G_D$ generated by the projections, i.e., $\text{Free}_n(G) = \langle (W_D^i)_{D \in \mathcal{D}_n} \mid i = 1, \dots, n \rangle \subseteq \prod_{D \in \mathcal{D}_n} G_D$.*

We want to characterize this subalgebra. We shall do this by constructing an algebra of sections over a finite poset G_n obtained by making explicit the constraints given by common prefixes.

Definition 6. *Let $\mathcal{C}$ be a set of chains such that the longest chain in $\mathcal{C}$ has length $l + 1$. The poset $\Gamma(\mathcal{C})$ is defined in the following way: let $\Gamma_l = \{C \oplus \{1\} \mid C \in \mathcal{C}\}$. For every $1 \leq i \leq l$, let $\mathcal{V}_i = \{V \mid V \text{ is a prefix of length } i \text{ of } W \in \Gamma_i\}$. For every $V \in \mathcal{V}_i$ let $\bar{V} = \{W \in \Gamma_i \mid V \text{ is prefix of } W\}$. Further, let $\mathcal{V}_i^c = \Gamma_i \setminus \bigcup_{V \in \mathcal{V}_i} \bar{V}$. Then let $\Gamma_{i-1} = \{V^+ \mid V \in \mathcal{V}_i\} \cup \{V \mid V \in \mathcal{V}_i^c\}$ where $V^+ = V \oplus \bigcup_{W \in \bar{V}} (W \setminus V)$. We set $\Gamma(\mathcal{C}) = \bigcup \Gamma_0$.*

It is easy to check that each Γ_i is a finite set of finite posets, constructed from Γ_{i+1} by merging all the common prefixes of length $i + 1$. Observe that if V is a common prefix of the chains $C_1, \dots, C_u$, then $V^+ = V \oplus \Gamma(\{C_i \setminus V\}_{i=1}^u)$.

When Γ is applied to $\mathcal{C}_n$ we have $l = n$ and $\Gamma_n = \{G_D \mid D \in \mathcal{D}_n\}$. We set

$$G_n = \Gamma(\mathcal{C}_n).$$

Let $S(G_n)$ be the poset of sections over G_n and let $\mathcal{B} = \mathcal{B}(G_n)$ be the set of branches of the poset G_n . Consider the algebra $\mathcal{G}_n = (S(G_n), \wedge, \rightarrow, 0)$, where $0 = [\min B]_{B \in \mathcal{B}} = [p_B \mid 0 \in p_B]_{B \in \mathcal{B}}$ and the operations are defined branch-wise, that is, $[x_B]_{B \in \mathcal{B}} \wedge [y_B]_{B \in \mathcal{B}} = [x_B \wedge y_B]_{B \in \mathcal{B}}$, $[x_B]_{B \in \mathcal{B}} \rightarrow [y_B]_{B \in \mathcal{B}} = [x_B \rightarrow y_B]_{B \in \mathcal{B}}$, where $\wedge_B$ and $\rightarrow_B$ are the operations of the Gödel algebra $(B, \wedge_B, \rightarrow_B, \min B)$. Note that such operations are well defined. Indeed for every branch B , $x_B \rightarrow y_B$ is equal either to 1_B or to y_B , hence it is the unique element of $[x_B \rightarrow y_B]_{B \in \mathcal{B}}$ belonging to B .

Theorem 2. $\mathcal{G}_n$ is isomorphic to the free Gödel algebra over n variables.

Proof. Let $[p_B]_{B \in \mathcal{B}}$ be a section of G_n . Each branch B of G_n is a chain G_{D_B} for some domain D_B . We set for every $\mathbf{t} \in D_B$, $\psi([p_B]_{B \in \mathcal{B}})(\mathbf{t}) = f(\mathbf{t})$, for an arbitrarily chosen $f \in p_B$. Note that if $f, g \in p_B$ then for every $\mathbf{t} \in D_B$ we have $f(\mathbf{t}) = g(\mathbf{t})$. Then $\psi([p_B]_{B \in \mathcal{B}})$ is a function from $[0, 1]^n$ to $[0, 1]$ such that its restriction to each domain of linearity is a basic function. The construction of G_n assures that condition 3 in Theorem 1 is satisfied, hence ψ is a map from $\mathcal{G}_n$ to $Free_n(\mathbf{G})$.

Let $f \in Free_n(\mathbf{G})$. Then for every $B \in \mathcal{B}$ we consider $\phi(f) = [p_B]_{B \in \mathcal{B}}$ as the unique section such that $f|_{D_B} \in p_B$. Then it is easy to check that ϕ is the inverse of ψ . The proof that ψ is an isomorphism of Gödel algebras then follows easily (for instance, note that $\psi([\min B]_{B \in \mathcal{B}})(\mathbf{t}) = 0$ for every $\mathbf{t} \in [0, 1]^n$). $\square$

The free generators of $\mathcal{G}_n$ are the sections $[p_B \mid x_i \in p_B]_{B \in \mathcal{B}}$, for $i \in \{1, \dots, n\}$.

We are going to prove that G_n is nice by giving a recursive law to effectively compute its type. We recall that the Stirling number of the second kind $\left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\}$ gives the number of ways a set of n elements can be partitioned into k many nonempty subsets [8].

Theorem 3. G_n is nice, its type being $o(G_n) = H_n \cup (H_n)_\perp$ with $H_0 = \mathbf{1}$ and

$$H_n = \bigcup_{i=0}^{n-1} \binom{n}{i} (H_i)_\perp.$$

Moreover, $|m(G_n)| = 2^n$ and $|\mathcal{C}_n| = |M(G_n)| = 2 \sum_{k=0}^n k! \left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\}$.

Proof. First we partition $\mathcal{C}_n$ into two sets $\mathcal{C}_n^0$ and $\mathcal{C}_n^1$ as follows: $C = \{W_1 < W_2 < \dots < W_w\} \in \mathcal{C}_n^0$ iff $W_1 = \{0\}$. Then, $\mathcal{C}_n^1 = \mathcal{C}_n \setminus \mathcal{C}_n^0$. Let $V = \{V_1 < V_2 < \dots < V_v\} \in \mathcal{C}_n^1$, with $V_1 = \{0, x_{\sigma(1)}, \dots, x_{\sigma(k)}\}$ for some $1 \leq k \leq n$ and some $\sigma \in Sym_n$. We define the map $V \mapsto V^0$ by setting $V^0 = \{\{0\} < V_1 \setminus \{0\} < V_2 < \dots < V_v\}$. It is obvious that $(\cdot)^0$ maps bijectively $\mathcal{C}_n^1$ to $\mathcal{C}_n^0$.

A straightforward computation shows that

$$|\mathcal{C}_n^0| = \sum_{k=0}^n k! \left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\},$$

since each $C \in \mathcal{C}_n^0$ is obtained by first specifying a partition $\{W_1, \dots, W_k\}$ of the set of basic functions $\{x_1, \dots, x_n\}$ and then by choosing a permutation $\sigma \in Sym_k$ to get $C = \{\{0\} < W_{\sigma(1)} < \dots < W_{\sigma(k)}\}$. Hence, $|\mathcal{C}_n| = |M(G_n)| = 2 \sum_{k=0}^n k! \left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\}$. Each minimal element of G_n has the form $\{0\} \cup \{x_i \mid i \in S\}$ for some $S \subseteq 2^n$ and, viceversa, each such S corresponds to an element of $m(G_n)$. Then $|m(G_n)| = 2^n$.

Note that $\{\{0\}\}$ is the common prefix of all chains in $\mathcal{C}_n^0$, while the common prefix of all chains in $\mathcal{C}_n^1$ is the empty set. We can display G_n as $G_n = \Gamma(\mathcal{C}_n) = \Gamma(\mathcal{C}_n^1) \cup \Gamma(\mathcal{C}_n^0)$. Note that $\Gamma(\mathcal{C}_n^0) = \{\{0\}\}^+$ and by our definition of the map $(\cdot)^0$ it follows that the type of $\Gamma(\mathcal{C}_n^0)$ is $o(\Gamma(\mathcal{C}_n^1))_\perp$. Let $H_n = o(\Gamma(\mathcal{C}_n^1))$. A direct

inspection shows that $H_1 = \mathbf{2}$. We shall give a recursive law to compute the structure of each H_n . For later convenience we write $H_1 = (H_0)_\perp$, with $H_0 = \mathbf{1}$.

For each $k \in \{1, \dots, n\}$ we consider the set $\mathcal{Z}_k$ of all prefixes V of length 1 in $\mathcal{C}_n^1$ that have the form $V = \{\{0, x_{\sigma(1)}, \dots, x_{\sigma(k)}\}\}$ for some $\sigma \in \text{Sym}_n$. Let $\bar{V} = \{W \in \mathcal{C}_n^1 \mid V \text{ is a common prefix of } W\}$ and $V^+ = V \oplus \Gamma(\{W \setminus V \mid W \in \bar{V}\})$ as in Definition 6. Then $o(V^+) = (o(\Gamma(\{W \setminus V \mid W \in \bar{V}\})))_\perp$, since V has only one element. The type of $\Gamma(\{W \setminus V \mid W \in \bar{V}\})$ is H_{n-k} , since it is isomorphic to $\Gamma(\mathcal{C}_{n-k}^1)$ (to see this, just delete 0 from each minimal element of $\Gamma(\mathcal{C}_{n-k}^1)$ and suitably rename variables). Observe that $|\mathcal{Z}_k| = \binom{n}{k} = \binom{n}{n-k}$, since the subset $\{x_{\sigma(1)}, \dots, x_{\sigma(k)}\}$ of the variables occurring in V can be chosen in $\binom{n}{n-k}$ ways. Then, summing over all $\mathcal{Z}_k$ for $k \in \{1, \dots, n\}$ we have (for $i = n - k$):

$$H_n = \bigcup_{i=0}^{n-1} \binom{n}{i} (H_i)_\perp.$$

Whence, G_n is nice. □

The following first appeared in [11]:

Corollary 1. $|\mathcal{G}_n| = |S(H_n)|(|S(H_n)|+1)$, for $|S(H_n)| = \prod_{i=0}^{n-1} (|S(H_i)|+1) \binom{n}{i}$.

Remark 1. The number $b_n := \sum_{k=0}^n k! \binom{n}{k}$ is called the *n*th ordered Bell number [14], and gives the number of ways n competitors can rank in a competition, allowing for the possibility of ties.

Example 2. The type of the poset underlying the representation of $\text{Free}_2(\mathbf{G})$ as an algebra of sections is

$$o(G_2) = \mathbf{2} \cup \mathbf{3} \cup \mathbf{3} \cup (\mathbf{2} \cup \mathbf{3} \cup \mathbf{3})_\perp.$$

The diagram of G_2 is depicted in Fig. 1. The elements of $\mathcal{G}_2$ are the sections over such poset, i.e. sequences either of the form $(p_1, p_2, p_3, \{0\})$ or $(p_1, p_2, p_3, p_4, p_5, p_6)$, with $p_1 \in \{\{0, x, y\}, \{1\}\}$, $p_2 \in \{\{0, x\}, \{y\}, \{1\}\}$, $p_3 \in \{\{0, y\}, \{x\}, \{1\}\}$, $p_4 \in \{\{x, y\}, \{1\}\}$, $p_5, p_6 \in \{\{x\}, \{y\}, \{1\}\}$.

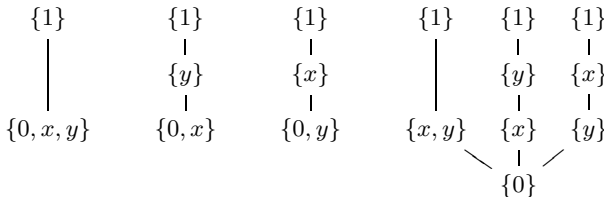


Fig. 1. Poset structure for the free Gödel algebra over two variables

4.1 Gödel Hoops

BH is the logic obtained by dropping the *ex falso quodlibet* axiom (A7) from BL, and the constant 0 from the language. BH then turns out to be the falsity-free fragment of BL, and its algebraic semantics is given by the variety of *basic hoops* [4]. The logic GH is obtained by adding the idempotency axiom $\varphi \rightarrow (\varphi \odot \varphi)$ to BH and the corresponding subvariety of $\mathbb{V}(\text{BH})$ is the variety $\mathbb{V}(\text{GH})$ of Gödel hoops.

$\mathbb{V}(\text{GH})$ is generated by the standard Gödel hoop $((0, 1], \wedge, \rightarrow)$ where operations are obtained by restricting the Gödel standard operations to $(0, 1]$.

It is straightforward to see that domains of linearity of $\text{Free}_n(\text{GH})$ have the form $\{\mathbf{x} \in (0, 1]^n \mid x_{\sigma(1)} \prec_1 \cdots \prec_{n-1} x_{\sigma(n)}\}$ where each $\prec_i \in \{<, =\}$ and $\sigma \in \text{Sym}_n$. Hence:

Theorem 4. *$\text{Free}_n(\text{GH})$ is the algebra of sections over (a poset whose type is) H_n , with operations defined branch-wise as those of $\mathcal{G}_n$.*

5 Poset Representation for Nilpotent Minimum Logic

The set of basic functions for NM is $\{0, x_i, \neg x_j, 1 \mid i, j = 1, \dots, n\}$. If x is a variable we denote by x^{-1} its negation $\neg x$ and by x^1 the variable x itself. Let us consider domains of linearity of the form

$$D = \left\{ \mathbf{x} \in [0, 1]^n \mid x_{\sigma(1)}^{\delta_{\sigma(1)}} \prec \dots \prec x_{\sigma(n)}^{\delta_{\sigma(n)}} \prec x_{\sigma(n)}^{-\delta_{\sigma(n)}} \prec \dots \prec x_{\sigma(1)}^{-\delta_{\sigma(1)}} \right\}, \quad (2)$$

where $\sigma \in \text{Sym}_n$, $\delta_{\sigma(i)} \in \{-1, 1\}$ and each occurrence of $\prec$ is either $<$ or $=$. Hence domain chains have the form

$$C_D = \{W_1 < \dots < W_k \triangleleft \neg W_k < \dots < \neg W_1\},$$

where $\triangleleft \in \{\cup, <\}$, W_i is a subset of $\{x_1^{\delta_1}, \dots, x_n^{\delta_n}\}$ and $\neg W_i = \{x_h^{-\delta_h} \mid x_h^{\delta_h} \in W_i\}$.

Let $\mathcal{D}_n^{NM}$ be the set of all domains satisfying (2) and let $\mathcal{C}_n^{NM} = \{C_D \mid D \in \mathcal{D}_n^{NM}\}$. Throughout this section we shall write $\mathcal{D}_n$ and $\mathcal{C}_n$ instead of $\mathcal{D}_n^{NM}$ and $\mathcal{C}_n^{NM}$.

Definition 7. Let $C = \{W_1 < W_2 < \dots < W_k \triangleleft \neg W_k < \dots < \neg W_1\} \in \mathcal{C}_n$. If $\triangleleft$ is $\cup$, then we denote by C^* the set $C^* = W_k \cup \neg W_k$. If $\triangleleft$ is $<$, then by C^* we mean the empty set. Subchains W^1 and W^2 of C are defined univocally by $C = W^1 \cup C^* \cup W^2$ and $X < Y < Z$ for every $X \in W^1$, $Y \in C^*$ and $Z \in W^2$.

In the subset C^* of C we hence collect the basic functions coinciding over D_C with their negation.

Definition 8. The sign of a chain $C = W^1 \cup C^* \cup W^2$ is a sequence $\Lambda(C) = (\lambda_1, \dots, \lambda_n) \in \{0, -1, 1\}^n$ such that

$$\lambda_i = \begin{cases} \delta_i & \text{if } x_i^{\delta_i} \notin C^* \\ 0 & \text{otherwise.} \end{cases}$$

The length of $\Lambda(C)$, denoted by $\|\Lambda(C)\|$, is the number of components of $\Lambda(C)$ different from 0. The support of $\Lambda(C)$ is the sequence $(|\lambda_1|, \dots, |\lambda_n|) \in \{0, 1\}^n$.

Note that if $U, V \in \mathcal{C}_n$ are such that $\Lambda(U) = \Lambda(V)$ then $U^* = V^*$.

Example 3. Let $D = \{(x, y, z) \mid y < \neg x < z = \neg z < x < \neg y\}$. Then $C_D = \{\{y\} < \{\neg x\} < \{z, \neg z\} < \{x\} < \{\neg y\}\}$, $C_D^* = \{z, \neg z\}$, $\Lambda(C_D) = (-1, 1, 0)$, $\|\Lambda(C_D)\| = 2$ and its support is $(1, 1, 0)$. Note that $D = \{(x, y, \frac{1}{2}) \mid \frac{1}{2} < x < \neg y\}$.

We state the functional representation of free NM-algebras given in [16] using our notation.

Theorem 5. *The free NM-algebra over n variables is the algebra of functions $f: [0, 1]^n \rightarrow [0, 1]$ satisfying the following conditions:*

1. f is piecewise linear, each piece being an NM basic function.
2. f is linear over each domain $D \in \mathcal{D}_n$.
3. Let $\preceq \in \{<, >, =\}$. If $f(\mathbf{t}) \preceq 1/2$ for every $\mathbf{t} \in D$, then for every domain D' with the same sign as D we have $f(\mathbf{u}) \preceq 1/2$ for every $\mathbf{u} \in D'$.

If $C_D = \{W_1 < W_2 < \dots < W_k \triangleleft \neg W_k < \dots < \neg W_1\}$ we set $N_D := \{0\} \oplus C_D \oplus \{1\} = \{\{0\} < W_1 < \dots < W_k \triangleleft \neg W_k < \dots < \neg W_1 < \{1\}\}$. Note that if C_D^* is the empty set, then N_D has exactly $2k+2$ elements. In this case we set $N_D^1 = \{\{0\} < W_1 < W_2 < \dots < W_k\}$ and $N_D^2 = \{\neg W_k < \dots < \neg W_1 < \{1\}\}$. Otherwise N_D has $2k+1$ elements and we set $N_D^1 = \{\{0\} < W_1 < W_2 < \dots < W_{k-1}\}$ and $N_D^2 = \{\neg W_{k-1} < \dots < \neg W_1 < \{1\}\}$.

Any chain N_D can be equipped with the structure of NM-algebra. The operation $\neg$ mapping C_D^* in itself, W_i in $\neg W_i$, $\{0\}$ in $\{1\}$ and viceversa is the negation of N_D . Further, $\neg$ is an order isomorphism between N_D^1 and $(N_D^2)^\partial$.

Proposition 2. *The free NM-algebra over n variables is the subalgebra of the direct product $\prod_{D \in \mathcal{D}_n} N_D$ generated by the projections (the definition of projection is mutatis mutandis the same as in Proposition 1).*

We want now to characterize this subalgebra of $\prod_{D \in \mathcal{D}_n} N_D$, in order to obtain a representation of the free NM-algebra as an algebra of sections.

Definition 9. *Consider the partition of $\mathcal{C}_n = \{C_D \mid D \in \mathcal{D}_n\}$ obtained by considering the equivalence relation given by $U \sim V$ if and only if $\Lambda(U) = \Lambda(V)$. Then, for every $[U] \in \mathcal{C}_n / \sim$, we set $N_{[U]}^1 = \bigcup_{V \in [U]} N_{D_V}^1$, $N_{[U]}^2 = \bigcup_{V \in [U]} N_{D_V}^2$ and*

$$N_{[U]} := \begin{cases} N_{[U]}^1 \oplus N_{[U]}^2 & \text{if } \|\Lambda(U)\| = n \\ N_{[U]}^1 \oplus \{U^*\} \oplus N_{[U]}^2 & \text{if } \|\Lambda(U)\| < n. \end{cases}$$

Let

$$NM_n = \bigcup_{[U] \in \mathcal{C}_n / \sim} N_{[U]}.$$

Let $S(NM_n)$ be the poset of sections over NM_n and let $\mathcal{B} = \mathcal{B}(NM_n)$ be the set of branches of NM_n . Further, for every $U \in \mathcal{C}_n$, let $\mathcal{B}(U) = \mathcal{B}(N_{[U]})$

be the set of branches contained in $N_{[U]}$. If $[p_B]_{B \in \mathcal{B}}$ is a section of NM_n , then $[p_B]_{B \in \mathcal{B}(U)}$ is a section over $N_{[U]}$ and then either is a section over $\bigcup_{V \in [U]} N_{D_V}^1$, or is a section over $\bigcup_{V \in [U]} N_{D_V}^2$, or $p_B = \{U^*\}$ for all $B \in \mathcal{B}(U)$.

We set $\neg[p_B]_{B \in \mathcal{B}} = [\neg p_B]_{B \in \mathcal{B}}$. If $[p_B]_{B \in \mathcal{B}(U)}$ is a section over $\bigcup_{V \in [U]} N_{D_V}^1$ then $\neg[p_B]_{B \in \mathcal{B}(U)}$ is a section over $\bigcup_{V \in [U]} N_{D_V}^2$ and viceversa (if $[p_B] = \{U^*\}$ for $B \in \mathcal{B}(U)$, then $\neg[p_B]_{B \in \mathcal{B}(U)} = [p_B]_{B \in \mathcal{B}(U)}$). Let $\overline{\mathcal{B}} = \{B \in \mathcal{B} \mid p \in B \text{ implies } \neg p \in B\} = \{N_D \mid D \in \mathcal{D}_n\}$. Note that $[p_B]_{B \in \mathcal{B}} = [p_B]_{B \in \overline{\mathcal{B}}}$ for all $[p_B]_{B \in \mathcal{B}} \in S(NM_n)$. We define $[p_B]_{B \in \mathcal{B}} \odot [q_B]_{B \in \mathcal{B}}$ and $[p_B]_{B \in \mathcal{B}} \rightarrow [q_B]_{B \in \mathcal{B}}$ as the unique sections $[r_B]_{B \in \mathcal{B}}$ and $[s_B]_{B \in \mathcal{B}}$, respectively, such that for every $B \in \overline{\mathcal{B}}$,

$$r_B = \begin{cases} p_B \wedge q_B & \text{if } q_B > \neg p_B \\ 0_B & \text{if } q_B \leq \neg p_B, \end{cases} \quad s_B = \begin{cases} 1_B & \text{if } p_B \leq q_B \\ \neg p_B \vee q_B & \text{if } p_B > q_B. \end{cases}$$

Further, let $[p_B]_{B \in \mathcal{B}} \wedge [q_B]_{B \in \mathcal{B}} = [p_B \wedge q_B]_{B \in \mathcal{B}}$, $[p_B]_{B \in \mathcal{B}} \vee [q_B]_{B \in \mathcal{B}} = [p_B \vee q_B]_{B \in \mathcal{B}}$, $0 = [\min B]_{B \in \mathcal{B}} = [\min N_D]_{D \in \mathcal{D}_n} = [\{0\}]_{D \in \mathcal{D}_n}$ and $1 = [\max B]_{B \in \mathcal{B}} = [\max N_D]_{D \in \mathcal{D}_n} = [\{1\}]_{D \in \mathcal{D}_n}$. Note that $p \rightarrow 0 = \neg p$ for every $p \in S(NM_n)$.

Theorem 6. $\mathcal{NM}_n = (S(NM_n), \odot, \rightarrow, \wedge, 0)$ is isomorphic to the free NM-algebra over n variables.

Proof. The proof is analogous to the proof of Theorem 2 for Gödel logic. $\square$

The free generators of $\mathcal{NM}_n$ are the sections $[p_B \mid x_i \in p_B]_{B \in \mathcal{B}}$, for each $i \in \{1, \dots, n\}$.

Theorem 7. NM_n is nice. Further, $|\mathcal{C}_n / \sim| = 3^n$, $|m(NM_n)| = |M(NM_n)| = \sum_{i=0}^n \binom{n}{i} 2^i b_i$, where b_i is the i th ordered Bell number (see Remark 1).

Proof. It is trivial to see that $|\mathcal{C}_n / \sim| = |\{-1, 0, 1\}^n| = 3^n$. Note that $3^n = \sum_{i=0}^n \binom{n}{i} 2^i$. As a matter of fact there are $\binom{n}{i}$ supports of signs of length i , and each one of them is the support of 2^i many signs.

Let $U \in \mathcal{C}_n$, and let h be the length of $A(U)$. Then $|[U]| = b_h$. Let $\mathcal{U}_k$ be the subset of $[U]$ formed by all chains of the form $C = \{W_1 < W_2 < \dots < W_k \triangleleft \neg W_k < \dots < \neg W_1\}$. Then $N_{D_C}^1 = \{\{0\} < W_1 < \dots < W_l\}$ where $l = k$ if $\triangleleft$ is $<$, $l = k - 1$ otherwise. Note that $N_{[U]}^1 = \bigcup_{k=0}^n \bigcup_{V \in \mathcal{U}_k} N_{D_V}^1$ and $o(N_{[U]}^1) = o(N_{[U]}^2)$. For each $i \in \{0, \dots, n\}$ we let

$$K_i = \bigcup_{k=0}^i k! \left\{ \begin{matrix} i \\ k \end{matrix} \right\} (\mathbf{k}_\perp),$$

and then $o(N_{[U]}) = K_h \oplus \mathbf{1} \oplus K_h$ if $h < n$; in case $h = n$, $o(N_{[U]}) = K_n \oplus K_n$. Summing over all equivalence classes of $\mathcal{C}_n / \sim$ we obtain

$$o(NM_n) = \left(\bigcup_{i=0}^{n-1} \binom{n}{i} 2^i (K_i \oplus \mathbf{1} \oplus K_i) \right) \cup 2^n (K_n \oplus K_n),$$

and hence NM_n is nice. The number of elements in $m(NM_n)$ equals the number of elements of $M(NM_n)$ and it is given by summing the number of occurrences

of chains in each K_i , hence $|m(NM_n)| = |M(NM_n)| = \sum_{i=0}^n \binom{n}{i} 2^i |m(K_i)| = \sum_{i=0}^n \binom{n}{i} 2^i b_i$. $\square$

Corollary 2. $|\mathcal{NM}_n| = |S(K_n)|^{2^{n+1}} \prod_{i=0}^{n-1} (2|S(K_i)| + 1)^{2^i \binom{n}{i}}$, where $|S(K_i)| = \prod_{k=0}^i (k+1)^{k! \binom{i}{k}}$.

Example 4. For $n = 2$ we have 17 different domains of linearity, hence 17 chains: $\{\{x, y, \neg y, \neg x\}\}$ with sign $(0, 0)$, $\{\{x\} < \{y, \neg y\} < \{\neg x\}\}$ with sign $(1, 0)$, $\{\{y\} < \{x, \neg x\} < \{\neg y\}\}$ with sign $(0, 1)$, $\{\{\neg x\} < \{y, \neg y\} < \{x\}\}$ with sign $(-1, 0)$, $\{\{\neg y\} < \{x, \neg x\} < \{y\}\}$ with sign $(0, -1)$, $\{\{x, y\} < \{\neg y, \neg x\}\}$, $\{\{x\} < \{y\} < \{\neg y\} < \{\neg x\}\}$, $\{\{y\} < \{x\} < \{\neg x\} < \{\neg y\}\}$, with sign $(1, 1)$, $\{\{\neg x, y\}, \{\neg y, x\}\}$, $\{\{\neg x\}, \{y\}, \{\neg y\}, \{x\}\}$, $\{\{y\} < \{\neg x\} < \{x\} < \{\neg y\}\}$ with sign $(-1, 1)$, $\{\{\neg x, \neg y\} < \{y, x\}\}$, $\{\{\neg x\} < \{\neg y\} < \{y\} < \{x\}\}$, $\{\{\neg y\} < \{\neg x\} < \{x\} < \{y\}\}$ with sign $(-1, -1)$, $\{\{x, \neg y\} < \{y, \neg x\}\}$, $\{\{x\} < \{\neg y\} < \{y\} < \{\neg x\}\}$, $\{\{\neg y\} < \{x\} < \{\neg x\} < \{y\}\}$ with sign $(1, -1)$. The type of NM_2 is given by:

$$o(NM_2) = \mathbf{3} \cup 4(\mathbf{5}) \cup 4((\mathbf{3} \cup \mathbf{2} \cup \mathbf{3}) \oplus (\mathbf{3} \cup \mathbf{2} \cup \mathbf{3})).$$

References

1. Cignoli, R., Esteva, F., Godo, L., Torrens A.: Basic fuzzy logic is the logic of continuous t-norms and their residua. *Soft Computing* **4** (2000) 106–112
2. Davey, B.A., Priestley, H.A.: *Introduction to Lattices and Order*. Cambridge Mathematical Textbooks, Cambridge University Press (1991)
3. Esteva, F., Godo, L.: Monoidal t-norm based logic: towards a logic for left-continuous t-norms. *Fuzzy Sets and Systems* **124** (2001) 271–288
4. Esteva, F., Godo, L., Hájek, P., Montagna F.: Hoops and fuzzy logic. *Journal of Logic and Computation* **13** (2003) 531–555
5. Fodor, J.: Nilpotent minimum and related connectives for fuzzy logic. *Proc. FUZZ-IEEE'95* (1995) 2077–2082
6. Gerla, B.: A characterization of Gödel functions. *Soft Computing* **4** (2000) 206–209
7. Gödel, K.: Zum intuitionistischen Aussagenkalkül. *Anzeiger Akademie der Wissenschaften Wien, Math.-naturwissensch. Klasse* **69** (1932) 65–66
8. Graham, R.L., Knuth, D.E., Patashnik, O.: *Concrete mathematics: a foundation for computer science*. 2nd edition. Addison-Wesley, Boston San Francisco (1994)
9. Hájek, P.: *Metamathematics of Fuzzy Logic*. Kluwer, Dordrecht (1998)
10. Horn, A.: Logic with Truth Values in a Linearly Ordered Heyting Algebra. *Journal of Symbolic Logic* **34** (1969) 395–408
11. Horn, A.: Free L -Algebras. *Journal of Symbolic Logic* **34** (1969) 475–480
12. McNaughton, R.: A theorem about infinite-valued sentential logic. *Journal of Symbolic Logic* **16** (1951) 1–13
13. Mundici, D.: Satisfiability in many-valued sentential logic is NP-complete. *Theoretical Computer Science* **52** (1987) 145–153

14. Sloane, N.J.A.: The On-Line Encyclopedia of Integer Sequences. Published electronically at <http://www.research.att.com/~njas/sequences> (2004)
15. Wang, S.M., Wang, B.S., Pei, D.W.: A fuzzy logic for an ordinal sum t-norm. *Fuzzy Sets and Systems*, **149** (2005) 297–307
16. Wang, S.M., Wang, B.S., Wang, X.Y.: A characterization of truth-functions in the nilpotent minimum logic. *Fuzzy Sets and Systems* **145** (2004) 253–266

Possibilistic Inductive Logic Programming

M. Serrurier and H. Prade

IRIT - Universit Paul Sabatier,
118 route de Narbonne 31062 Toulouse France
{serrurier, henri.prade}@irit.fr

Abstract. Learning rules with exceptions may be of interest, especially if the exceptions are not important in some sense. Standard Inductive Logic Programming (ILP) algorithms and classical first order logic are not well-suited for managing rules with exceptions. Indeed, a hypothesis that is induced accumulates all the exceptions of the rules contained in it. Moreover, with multiple-class problems, classifying an example in two different classes (even if one is the right one) is not correct, so a rule that contains some exceptions may prevent another rule which has no exception from being useful. This paper proposes a new possibilistic logic framework for weighted ILP. It induces rules which are progressively more and more accurate, and allows us to manage exceptions by controlling their accumulation. In this setting, we first propose an algorithm for learning rules when the background knowledge and the examples are stratified into layers having different levels of priority or certainty. This allows the induction of general but uncertain rules together with more specific and less uncertain rules. A second algorithm is presented, which does not require an initial weighted database, but still learn a default set of rules in the possibilistic setting.

1 Introduction

Inductive Logic Programming (ILP) [11, 10] provides a general framework for learning hypotheses under the form of sets of classical first order logic rules. Reasonably efficient algorithms have been developed (Progol [9], FOIL [12]...) and several applications have been described in data mining, especially for biochemistry databases, for natural language processing and more generally, in relational databases, since first order logic is well-adapted for describing them.

Inducing rules with exceptions is of interest for several reasons. First of all, for some databases, there may not exist rules without exception, or too few for covering a large part of the examples. Rules with exceptions may also have a larger support and may be less complex than ones without exception. In this scope, these rules can encode a default behavior w.r.t. some examples. The problem is that first order logic is not well-suited for managing exceptions. Indeed, exceptions w.r.t. the rules constituting a hypothesis accumulate, in the sense that there is no way to compensate the existence of an exception to a rule by means of the other rules. Moreover, when dealing with more than two classes,

one rule with exceptions may spoil the correctness of another rule. This would be the case when one example is well classified by the latter and misclassified by the first rule. Then, since there is no way for selecting the rule that best applies, the example is considered as misclassified. This problem still remains for binary classification, this is why the most common method in ILP is to describe examples in distinguishing between positive and negative ones. Thus, only one class is learnt, the second one is used for default classification. So, in this way, an example cannot be classified into two classes. However, assuming that the second class is just the complement of the first one, may lead to lose some information that appears in the description of negative examples. Moreover, this method becomes very costly in terms of space and time when dealing with more than two classes. A proper handling of exception-tolerant reasoning is provided by monotonic reasoning and can be embedding in a possibilistic logic framework. We propose two different approaches for using possibilistic logic in ILP. First, one may think of having examples and background knowledge associated with a level of exceptionality/priority supposedly provided by an expert. Possibilistic logic is used for handling these levels. What is obtained by an extended ILP procedure is a layered set of rules, with the property that the rules in the highest layers do not encounter exceptions among the most normal examples. This method enables us to find both simple rules that handle normal cases, but which may have some exceptions, as well as more sophisticated rules which cope with marginal examples. This method is an extension from the one presented in [6]. A second approach, which avoids the difficulty of structuring the knowledge base into layers, is also proposed. It consists in learning a stratified set of rules together with a stratification of the examples from a classical setting. What is learnt is then a set of default rules encoded with possibilistic logic. This method is defined with a classical ILP algorithm and allows us to improve the effectivity of this one.

The paper is organized as follows. First, we describe the ILP setting in a multiple-class context with an inconsistency-based treatment of exceptions. Next, we establish how to transpose possibilistic logic in a first order setting and we propose a definition of the possibilistic ILP problem. Then, the two above methods are described. In the last section, an illustrative example and a benchmark on the mutagenesis¹ database are provided.

2 Inductive Logic Programming and Classification

Stated in the general context of first-order logic, the task of ILP is to find a non-trivial set of formulas H such that :

$$B \wedge H \models E \quad (1)$$

given a background theory B and a set of observations E (training set). E , B and H here denote sets of Horn clauses. When dealing with two class problems, it is

¹ <http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/mutagenesis.html>

common to introduce negative examples in order to describe some exceptions of a rule. A rule in H has some exceptions if it covers a negative example. Moreover, it allows to learn only one class, the second class is deduced by default. As already said, in a problem that has a finite number of classes, (but more than two), describing negative examples may become very costly and all classes must be learnt. In this context an example that is derived in two different classes from H , even if one is the right one, is considered as misclassified because there no way for choosing one class rather than other. According to these remarks, we reformulate the ILP problems, by adding a classification constraint D , as follows in the spirit of [2]:

- E is a set of ground facts of the form $C(x, y)$ where x denotes an identification key of an example and y a class.
- B is a set of rules that describes background knowledge
- D is the constraint :

$$D \equiv \forall X, Y, Z \ C(X, Y) \wedge C(X, Z) \rightarrow Y = Z, \quad (2)$$

which expresses that an example can be only classified into one class.

Then, given B , D and E , the goal is to find H such as

$$B \wedge D \wedge H \models E \quad (3)$$

Natural notions of ILP are $\text{ILP}_{\text{correct}}$ and $\text{ILP}_{\text{complete}}$, defined as follows:

Definition 1. H is $\text{ILP}_{\text{correct}}$ w.r.t. B, D and E if and only if $B \wedge D \wedge H \models E$.

A complete hypothesis cover all examples in E but can make some errors.

Definition 2. H is $\text{ILP}_{\text{complete}}$ w.r.t. B, D and E if and only if $B \wedge D \wedge H \wedge E \not\models \perp$.

There are two ways for not being correct with respect to B , D and E . First H may classify an example in two different classes and then we have

$$B \wedge D \wedge H \models \perp.$$

Second, H may misclassify an example $e \in E$ i.e.

$$B \wedge D \wedge H \wedge e \models \perp$$

An ILP algorithm is said to be correct and complete if it always produces correct and complete hypotheses if any exists. Thanks to the use of D , there is no need to introduce negative examples for having a notion of exception and then the ILP problem can be completely stated in logical terms. There is a large number of ILP algorithms [9][12]. A majority of them can be correct with an appropriate use, but only one is theoretically correct and complete [7].

3 First Order Possibilistic Logic

In this section, we first provide a brief refresher on propositional possibilistic logic before proposing an extension to the first order setting in order to define possibilistic ILP in the next section.

3.1 Propositional Possibilistic Logic

We briefly restate the possibilistic logic framework for propositional logic [5, 4]. Ω denotes a set of possible interpretations of a set of propositional variables.

Definition 3. A possibility distribution π is a function from Ω to $[0, 1]$ such that $\exists \omega \in \Omega, \pi(\omega) = 1$.

A possibility distribution can encode certainty degrees or can be used only for encoding a complete pre-order on possible worlds. We consider only normalized distributions. From the definition of π , we can define the possibilistic measure of a propositional formula ϕ .

Definition 4. $\Pi(\phi) = \max\{\pi(\omega), \omega \in \Omega, \omega \models \phi\}$. $\Pi(\phi)$ is the maximum possibility degree of ϕ in Ω .

Necessity is the dual notion of possibility. It refers to the possibility degrees of the counter-models of a formula.

Definition 5. $N(\phi) = 1 - \Pi(\neg\phi)$. $N(\phi)$ is the minimum possibility degree of ϕ in Ω .

We can now define a notion of possibilistic model $\models_\pi$.

Definition 6. A formula (ϕ, α) is satisfied by ω if $\pi(\omega) \geq \alpha$ and $\omega \models \phi$.

Given K a set of possibilistic formulas, we define the α -cut of K as :

Definition 7. $K_\alpha = \{(\phi, \beta) \in K, \beta \geq \alpha\}$

Then we have :

Definition 8. $K \models_\pi (\phi, \alpha)$ if $\exists \beta \geq \alpha, K_\beta \not\models \perp$ and $K_\beta \models \phi$.

Possibilistic logic was proved to be sound and complete with respect to a semantics in terms of a complete plausibility preorder on the interpretations for refutation based on an extended resolution rule [5].

3.2 First-Order Case

In order to extend possibilistic logic to first-order settings, we replace Ω by S , which is a set of Herbrand interpretations. Then a possibility distribution is defined on Herbrand interpretations. Then definitions 3-7 remain valid. Nevertheless, definition 8 must be extended for avoiding some drowning of desirable

consequences in the inference process. Let us take the example of the following set of formulas in order to illustrate the problem:

$$K = \begin{cases} (C(X, Y), C(X, Z) \rightarrow Y = Z, 1) \\ (P(X) \rightarrow C(X, 2), 0.9) \\ (Q(X) \rightarrow C(X, 3), 0.7) \\ (S(X), R(X, Y), Q(Y) \rightarrow C(X, 5), 0.4) \\ (V(X) \rightarrow C(X, 4), 0.3) \\ (P(x_1), 1) \\ (Q(x_1), 1) \\ (Q(x_2), 1) \\ (S(x_3), 1) \\ (R(x_3, x_2), 1) \\ (V(x_3), 1) \end{cases}$$

Applying definition 8, only $(C(x_1, 2), 0.9)$ is deducible from K since $K_{0.7}$ is inconsistent. Formulas $(C(x_1, 3), 0.7)$, $(C(x_2, 3), 0.7)$, $(C(x_3, 5), 0.4)$ and $(C(x_3, 4), 0.3)$ are not obtained. However, formulas $(C(x_2, 3), 0.7)$ and $(C(x_3, 5), 0.4)$ should not be rejected, in particular the last one since it does not have any connection with the inconsistency of $K_{0.7}$. In the spirit of [1], we extend the notion of possibilistic models for first-order inference in the following way.

Definition 9. $T \models \phi$ iff $T \models \phi$ and $\forall \psi \in T, T \models \psi \rightarrow \phi$

Definition 10. $K \models_\pi (\phi, \alpha)$ iff $\exists K' \subset K_\alpha, K' \models \phi$ and $\nexists K'' \subset K_\alpha, K'' \models \phi$ and $K' \subset K'' \subseteq K_\alpha$

If we use definition 10, formulas $(C(x_2, 3), 0.7)$ and $(C(x_3, 5), 0.4)$ are now accepted and only $(C(x_1, 3), 0.7)$ and $(C(x_3, 4), 0.3)$ are rejected as expected.

4 ILP with Possibilistic Background and Examples

Possibilistic ILP is an adaptation of ILP to the possibilistic first-order logic setting. In this context, let B_p and E_p be sets of possibilistic formulas. D_p denote formula 2 associated with the maximal necessity level, namely 1, since the unicity of class assignments has priority over all other formulas. According to these definitions the goal of possibilistic ILP is to find a possibilistic set of rules H_p , given B_p, D_p and E_p such as :

$$B_p \wedge D_p \wedge H_p \models_\pi E_p$$

Obviously, only all the formulas in B_p and E_p have a necessity level strictly greater than 0.

Completeness and correctness are defined as follows :

Definition 11. $H_p \models_{\alpha} C(x, y), \alpha \in E_p, B_p \wedge D_p \models_{\pi} (C(x, y), \alpha).$

Definition 12. $H_p \models_{\alpha} C(x, y), \alpha \in E_p, \nexists (C(x, z), \beta), y \neq z, B_p \wedge D_p \wedge H_p \models_{\pi} (C(x, z), \beta).$

Moreover, we need the notion of a weakly-complete hypothesis. A weakly complete hypothesis covers all examples, but may fail on predicting the right necessity degree.

Definition 13. $H_p \models_{\alpha} C(x, y), \alpha \in E_p, \exists \beta > 0, B_p \wedge H_p \models_{\pi} (C(x, y), \beta).$

Note that this latter definition is not in conflict with correctness since deducing the right class of an example without having the right necessity level does not make the hypothesis incorrect.

Necessity degrees can encode different types of information. First, necessity degrees can represent levels of certainty. The higher the level, the more certain the formula. In this case, having correct and complete hypotheses is interesting since examples are well-classified with the right certainty level. This kind of description is useful for introducing uncertain rules in the background knowledge and for describing examples for which the class is not certain. Besides, necessity levels can also encode priority or normality levels. It may indicate that some information is very typical and then must be used with high priority and some other is more marginal. For instance, when describing E_p , the examples that have the highest necessity degree must be learnt in priority and should not be misclassified. On the contrary, making classification errors on examples that have a lower necessity is less important. In this context, having a correct and weakly complete hypothesis is sufficient since necessity degrees only provide information for biasing the learning task. Lastly, necessity levels may encode partial order on formulas in order to learn sets of default rules.

5 Learning Possibilistic Hypothesis

In this section we propose two different algorithms. The first algorithm describes induction from background and examples that are weighted with necessity levels. These levels are supposed to be supplied on the basis of a partition of the set of examples into classes of different reliability levels. In case such levels are not available, we propose a second algorithm that induces a possibilistic hypothesis together with a stratification of the set of examples starting from a standard background and a non-stratified set of examples. We denote M_{ilp} a standard ILP algorithm.

5.1 Possibilistic Background and Examples

The main idea of our algorithm is to repeatedly use the classical ILP machinery on sets of formulas that correspond to decreasing necessity level cuts of B_p and E_p . Then, all the hypotheses that are found are merged into a unique one. Lastly, we compute the necessity level of each rule.

Let $\alpha_0, \dots, \alpha_n$, $\alpha_0 = 1 > \dots > \alpha_n$ be the sequence of all necessity levels that appear in B_p and E_p . Then, necessity level cuts are defined as follows :

- $B_\alpha = \{b | (b, \beta) \in B_p, \beta \geq \alpha\}$
- $E_\alpha = \{e | (e, \beta) \in E_p, \beta \geq \alpha\}$
- $D_\alpha = D$

Alg. 1 PossibilisticILP(M_{ILP}, B_p, D_p, E_p)

```

1:  $H = \emptyset$ 
2: for  $i = 0$  to  $i = n$  do
3:   compute  $H_{\alpha_i}$  with  $M_{ILP}$  such that :  $B_{\alpha_i} \wedge D \wedge H_{\alpha_i} \models E_{\alpha_i}$ 
4:    $H = H \cup H_{\alpha_i}$ 
5: end for
6:  $H_p = \text{computeLevel}(H)$ 
7: return  $H_p$ 
    
```

Our algorithm is detailed in Alg. 1. A classical induction process is repeatedly applied to the sets E_{α_1} , $E_{\alpha_1} \cup E_{\alpha_2}$, ..., $E_{\alpha_1} \cup \dots \cup E_{\alpha_n}$. The hypothesis found at each step is merged with the rules previously induced. However, each rule remains associated with a level as for the sets of examples. The necessity degree of a rule depends on the levels of its exceptions. The higher the levels of exceptions, the lower the level of the rule. The level associated with a rule $h \in H_p$ is computed as follows :

$$\text{level}(h) = 1 - \max(\{\alpha | (C(x, y), \alpha) \in E_p, \exists z \neq y, \exists \beta > 0, D_p \wedge B_p \wedge h \models_\pi (C(x, z), \beta)\}) \quad (4)$$

with the convention $\max(\{\emptyset\}) = 0$. Thus, a rule that does not encounter any exception on E_p is associated with level 1. Moreover, the rules which encounter no exception, or whose exceptions correspond to examples that are marginal (i.e. with a low necessity level) have the highest levels.

Given properties on M_{ILP} we can deduce properties of our possibilistic machinery.

Proposition 1. *If M_{ILP} is correct and complete, then the hypothesis H_p computed by Alg. 1 has the following properties:*

- E_{α_n} contains all examples that appear in E_p . Since M_{ILP} is correct and complete, the hypothesis H_{α_n} covers all examples in E_p and has no exception. It means that all the rules in H_{α_n} have a necessity level equal to 1 in H_p . Then all

examples are covered by a rule with 1 for necessity level and since no formulas in B_p have 0 for necessity level, H_p is weakly complete. ■

We can also say that if M_{ILP} is correct then H_p is not necessarily correct. In fact, a hypothesis may be correct at one level and encounter exception in lower levels. In the same way, if M_{ILP} is complete then H_p is not necessarily complete nor weakly complete. A complete hypothesis, in the classical sense, may have some exceptions with 1 for necessity levels and then, the levels of the rules concerned is 0 and the hypothesis may not be weakly complete.

Proposition 2. *If M_{ILP} is correct and complete, then B_p is correct and complete, H_p is correct and complete.*

Proof. E_{α_n} contains all examples that appear in E_p . Since M_{ILP} is correct and complete, the hypothesis H_{α_n} covers all examples in E_p and has no exception. It means that all the rules in H_{α_n} have a necessity level equal to 1 in H_p . Then, all examples are covered by a rule with 1 for necessity level and, since all the formulas in B_p have 1 for necessity level, all examples in E_p are deducible with 1 as necessity level. Then, H_p is complete. For making H_p uncorrect, we need to deduce an exception with a necessity level equal to 1 according to equation 4. Thus, it must exist a rule h in H_p that has 1 as necessity level and have some exception. Since no example has a necessity degree equal to 0, the value $\max(\{\alpha | (e, \alpha) \in E_p, B \wedge h \wedge e \models \perp\})$ is greater than 0 because h has some exception and then $level(h) < 1$. So H_p is correct. ■

Thus, algorithm 1 preserves some key properties of the classical machinery used. The algorithm ensures that the hypothesis found will in priority cover examples with high necessity levels and will not make any error on these examples. Moreover, since it works with an incremental subset of the set of examples it allows us to find very general but uncertain rules in the first steps and very specific but more certain rules in the last steps.

5.2 Standard Background and Examples

Now, we propose an algorithm that learns a possibilistic hypothesis from an ordinary ILP setting. In contrast with the previous algorithm, the stratification of the set of examples is now automatically produced during the computation process. The goal of this algorithm is to increase the power of classical machineries by adding the possibility to learn sets of default rules. The algorithm is based on the fact that an example for one rule may be in our setting a counter-example for another rule. At each step of the algorithm (Alg. 2) a hypothesis is found. Then, we remove from the example set, all the examples that are covered by the hypothesis found, even if they are misclassified. We repeat this until the set of examples is empty or no more rule is found. We construct E_p by considering that the necessity level of an example depends on the step where it is covered. Examples that are covered at the first step have the lowest level because since they are removed for all the following steps, and may be counter-examples for the rules that are learnt after. In this scope, examples that are covered in the

last step get the highest level. H_p is made of all the rules learnt at each step. The level of a rule in H_p corresponds to the step where it was found because the rules found in the first step are induced from the whole sets of examples and the rules found at the last steps are induced from a small part of the examples. The background is considered as totally sure. So given D, B and E we search H_p and E_p such that $B_p \wedge D_p \wedge H_p \models_{\pi} E_p$, where B_p contain all formulas that appear in B with 1 as necessity level.

Alg. 2 DefaultILP(M_{ILP}, B, D, E)

```

1:  $H = \{\emptyset\}$ 
2: flag=true
3: i=0
4: while flag do
5:   compute  $H_i$  with  $M_{ILP}$  such that :  $B \wedge D \wedge H_i \models E$ 
6:    $H = H \cup H_i$ 
7:    $E_i = \{C(x, y) \in E \mid \exists z, B \wedge H_i \models C(x, z)\}$ 
8:    $E = E - E_i$ 
9:   i=i+1
10:  if  $E = \emptyset$  or  $H_i = \emptyset$  then
11:    flag=false
12:  end if
13: end while
14: Necessity level of examples in  $E_j = 1 - \frac{j}{i}$ 
15: Necessity level of a rule in  $H_j = \frac{j}{i}$ 
16: return  $H_p$  and  $E_p$ 
    
```

Proposition 3. *If M_{ILP} is correct, then H_p is correct and weakly complete.*

Proof. Since M_{ILP} is correct, so rules induced at step 1 have no exception. A rule h_i at a step $i > 1$ can only have an exception on an example that is covered at step $j < i$. Then, since M_{ILP} is correct, this example is correctly covered by a rule with a greater necessity level than h_i and then according to possibilistic definition of models, the example is correctly classified by the hypothesis. ■

Obviously, if M_{ILP} is correct and complete, then H_p is correct and weakly complete.

Proposition 4. *If M_{ILP} is correct and complete, then H_p is correct and weakly complete.*

Proof. According to Proposition 3 H_p is correct. If $E = \{\emptyset\}$ at the end of the algorithm, all example have been covered. Then, H_p is correct and weakly complete. ■

This algorithm is clearly well-suited for increasing the effectiveness of correct ILP algorithms by allowing to induce a set of default rules. This approach is reminiscent of decision list methods [13, 8, 3]. However, decisions list can be viewed as a particular case of possibilistic logic sets of rules.

6 Illustration and Experimentation

We first provide a complete example for the purpose of illustrating algorithm 1. Because of lack of stratified benchmarks, an experimentation is only proposed for algorithm 2.

6.1 Illustrative Example

Let us apply algorithm 1 on a well-known example.

$$B_p = \begin{cases} (penguin(X, true) \rightarrow bird(X, true), 1) \\ (bird(x_1, true), 1) \\ (penguin(x_2, true), 0.8) \\ (fish(x_3, true), 1) \end{cases}$$

$$E_p = \begin{cases} (C(x_1, fly), 1) \\ (C(x_2, notfly), 0.8) \\ (C(x_3, notfly), 1) \end{cases}$$

We use the notation $penguin(X, true)$, $bird(x_1, true)$ in order to state that if an individual is, for instance, a penguin or not, without using negation. Here levels are priority levels. In this scope, classical *bird* or *fish* are more important to explain, contrarily to the example that describes a penguin which is more marginal.

According to Alg. 1 we compute H_1 such as $D \wedge B_1 \wedge H_1 \models E_1$. One correct and complete hypothesis may be :

$$H_1 = \begin{cases} bird(X, true) \rightarrow C(X, fly) \\ fish(X, true) \rightarrow C(X, notfly) \end{cases}$$

Note that, if x_3 was not in B_1 and E_1 the hypothesis $\rightarrow bird(X, true)$ will be correct and complete.

Next, we compute $H_{0.8}$ such as $D \wedge B_{0.8} \wedge H_{0.8} \models E_{0.8}$

$$H_{0.8} = \begin{cases} bird(X, true) \wedge penguin(X, false) \rightarrow C(X, fly) \\ penguin(X, true) \rightarrow C(X, notfly) \\ fish(X, true) \rightarrow C(X, notfly) \end{cases}$$

We compute $H = H_1 \cup H_{0.8}$ and the level of each rule. The only rule that has an exception is $bird(X, true) \rightarrow C(X, fly)$ which misclassifies x_2 . Since $N(C(x_2, notfly) \geq 0.8)$ we have, according to equation 4, $N(bird(X, true) \rightarrow C(X, fly)) \geq 0.2$. Thus, we have:

$$H_p = \begin{cases} (bird(X, true) \rightarrow C(X, fly), 0.2) \\ (bird(X, true) \wedge penguin(X, false) \rightarrow C(X, fly), 1) \\ (penguin(X, true) \rightarrow C(X, not fly), 1) \\ (fish(X, true) \rightarrow C(X, not fly), 1) \end{cases}$$

H_p is correct and complete. As expected, we found both certain rules and more general rules. We can simplify H_p into a weakly-complete hypothesis :

$$H_p = \begin{cases} (bird(X, true) \rightarrow C(X, fly), 0.2) \\ (penguin(X, true) \rightarrow C(X, not fly), 1) \\ (fish(X, true) \rightarrow C(X, not fly), 1) \end{cases}$$

This version remains correct and contains very simple rules.

6.2 Mutagenesis

We illustrate the algorithm 2 with the database “mutagenesis” which is a classical benchmark for ILP. This database describes molecules. The concept we want to learn is the activity level of a molecule. The database contains 188 instances described with 7 predicates. The two classes to be learnt are “active” and “inactive” molecules. We use an algorithm inspired from the FOIL algorithm [12]. With the classical ILP setting we found a hypothesis with 62% of accuracy (proportion of well and uniquely classified examples) when confidence degree threshold is equal to 1 (in this case, FOIL is correct) and 72% of accuracy with a confidence degree threshold equal to 0.90. These values may seem to be low regarding to literature on the mutagenesis database, but results reported in these cases corresponds to the learning of the “active” class only (inactive class is deduced by default and corresponds to negative examples) and learning one class is an easier problem. By using the algorithm 2 with FOIL algorithm, we obtain an accuracy value equal to 75% and 85% w.r.t. the two above thresholds respectively. It clearly shows that using our algorithm allows us to increase the accuracy, up to 13% w.r.t. a classical algorithm by allowing to learn default rules. For instance we found, using the correct version of FOIL, very precise rules such as:

$(ind1(A, 1.0), atm(A, B, C, 27, D) \rightarrow active(A, active), 1)$

and more general rules such as:

$(ind1(A, 1.0) \rightarrow active(A, inactive), 0.33)$

in a hypothesis which is correct in the sense of possibilistic induction.

7 Conclusion

In this paper we have described a possibilistic version of ILP that allows us to handle exceptions and default rules. This have been made possible by extending the possibilistic logic setting to first-order logic. The two algorithms described serve different purposes. The first method induces rules when background and examples are stratified. This can be very useful for databases that contain uncertain pieces of information, or if an expert is able to express preferences about

using or learning some information. With the second algorithm we learn default rules from a classical database. This allows us to increase the expressivity and then the learning power of ILP.

In the future, it may be interesting to make an automatic stratification of a classical database in order to test the algorithm 1 on real database. The linkage with decision list method needs also to be investigated.

References

1. S. Benferhat, D. Dubois, and H. Prade. Representing default rules in possibilistic logic. In *Proceedings of the 11 th International Conference of on Principles of Knowledge Representation and Reasoning KR92*, pages 673–684, 1992.
2. H. Blockeel, L. De Raedt, N. Jacobs, and B. Demoen. Scaling up inductive logic programming by learning from interpretations. *Data Mining and Knowledge Discovery*, 3:59–93, 1999.
3. M. E. Califf. Efficient and effective induction of first order decision lists. In S. Matwin and C. Sammut, editors, *Proceedings of the 12 th International Conference of Inductive Logic Programming, ILP 2002*, pages 17–31. Springer, 2002.
4. D. Dubois, J. Lang, and H. Prade. Automated reasoning using possibilistic logic: semantics, belief revision and variable certainty weights. *IEEE Trans. on Data and Knowledge Engineering*, 6:64–71, 1994.
5. D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D. Gabbay et al., editor, *Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 3*, pages 439–513. Oxford University Press, 1994.
6. Didier Dubois, Henri Prade, Gilles Richard, and Mathieu Serrurier. Stratified induction in possibilistic logic . In *Proc. of the Ninth International Conference, Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU 2002), Annecy- France*, pages 617–623. ESIA-Universit de Savoie, 2002.
7. K. Inoue. Induction, abduction and consequence-finding. In C. Rouveirol and M. Sebag, editors, *Proceedings of the 11 th International Conference of Inductive Logic Programming, ILP 2001*, pages 65–79. Springer, 2001.
8. R. Mooney and M. E. Califf. Induction of first-order decision lists: results on learning the past tense of english verbs. *Journal of Artificial Intelligence Research*, 3:1–24, 1995.
9. S.H. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.
10. S.H. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *New Generation Computing*, 19 20:629–680, 1994.
11. S.-H. Nienhuys-Cheng and R. de Wolf. *Foundations of Inductive Logic Programming*. Number 1228 in LNAI. Springer, 1997.
12. J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.
13. R. L. Rivest. Learning decision lists. *Machine Learning*, 2:229–246, 1987.

Query Answering in Normal Logic Programs Under Uncertainty

Umberto Straccia

ISTI - CNR, Via G. Moruzzi, 1 I-56124 Pisa (PI), Italy

Abstract. We present a simple, yet general top-down query answering procedure for normal logic programs over lattices and bilattices, where functions may appear in the rule bodies. Its interest relies on the fact that many approaches to para-consistency and uncertainty in logic programs with or without non-monotonic negation are based on bilattices or lattices, respectively.

1 Introduction

The management of uncertainty in logic programming has attracted the attention of many researchers and numerous frameworks have been proposed. Essentially, they differ in the underlying notion of uncertainty ¹ (e.g. probability theory [15, 18], fuzzy set theory [22], multi-valued logic [4, 12, 13, 14], possibilistic logic [8]) and how uncertainty values, associated to rules and facts, are managed. Roughly, these frameworks can be classified into *annotation based* (AB) and *implication based* (IB). In the AB approach (e.g. [12, 18]), a rule is of the form $A: f(\beta_1, \dots, \beta_n) \leftarrow B_1: \beta_1, \dots, B_n: \beta_n$, which asserts “the certainty of atom A is at least (or is in) $f(\beta_1, \dots, \beta_n)$, whenever the certainty of atom B_i is at least (or is in) β_i , $1 \leq i \leq n$ ”. Here f is an n -ary computable function and β_i is either a constant or a variable ranging over an appropriate certainty domain. In the IB approach (see [4, 13] for a more detailed comparison between the two approaches), a rule is of the form $A \stackrel{\alpha}{\leftarrow} B_1, \dots, B_n$, which says that the certainty associated with the implication $B_1 \wedge \dots \wedge B_n \rightarrow A$ is α . Computationally, given an assignment v of certainties to the B_i , the certainty of A is computed by taking the “conjunction” of the certainties $v(B_i)$ and then somehow “propagating” it to the rule head. The truth-values are taken from a certainty lattice. More recently, [4, 13, 22] show that most of the frameworks can be embedded into the IB framework (some exceptions deal with probability theory). However, most of the approaches stress an important limitation, as they do not address any mode of *non-monotonic negation*. Exception to this limitation are e.g. [16, 17] in which the stable semantics has been considered, but limited to the case where the underlying uncertainty formalism is probability theory; in [7] the underlying truth-space are lattices, but its formulations goes over *bilattices* [11] (a slightly more general structure than lattices); while [14] uses normal logic programs over bilattices under the IB framework.

¹ See e.g. [19] for an extensive list of references.

In many frameworks, in order to answer to a query, we have to compute the whole intended model by a bottom-up fixed-point computation and then answer with the evaluation of the query in this model. This always requires to compute a whole model, even if not all the atom's truth is required to determine the answer. To the best of our knowledge the only work presenting top-down procedures are [5, 12, 13, 22], but in none of them non-monotonic negation is considered.

In this paper we present a general top-down query answering procedure for normal logic programs over lattices and bilattices [9, 11] in the IB framework. Its interest relies on the fact that many approaches to paraconsistency and uncertainty in logic programs with or without non-monotonic negation are based on bilattices or lattices, respectively.

We proceed as follows. In the next section, we will briefly recall definitions and properties of lattices, bilattices and normal logic programs over bilattices. Section 3 is the main part of this work, where we present our top-down query procedure and the computational complexity analysis, while Section 4 concludes.

2 Preliminaries

Lattice. In a complete lattice $\mathcal{L} = \langle L, \preceq \rangle$, with L a countable set, bottom $\perp$ and top element $\top$, a function $f: L \rightarrow L$ is *monotone*, if $\forall x, y \in L, x \preceq y$ implies $f(x) \preceq f(y)$, while f is *antitone* if $x \preceq y$ implies $f(y) \preceq f(x)$. A *fixed-point* of f is an element $x \in L$ such that $f(x) = x$. The basic tool for studying fixed-points of functions on lattices is the well-known Knaster-Tarski theorem [20]. Let f be a monotone function on a complete lattice $\langle L, \preceq \rangle$. Then f has a fixed-point, the set of fixed-points of f is a complete lattice and, thus, f has a $\preceq$ -least fixed-point. The $\preceq$ -least fixed-point can be obtained by iterating f over $\perp$, i.e. is the limit of the non-decreasing sequence $y_0, \dots, y_i, y_{i+1}, \dots, y_\lambda, \dots$, where for a successor ordinal $i \geq 0, y_0 = \perp, y_{i+1} = f(y_i)$, while for a limit ordinal $\lambda, y_\lambda = \text{lub}_{\preceq} \{y_i: i < \lambda\}$. We denote the $\preceq$ -least fixed-point by $\text{lfp}_{\preceq}(f)$. For ease, we will specify the initial condition y_0 and the next iteration step y_{i+1} only, while the condition for a limit ordinal is considered as implicit.

Bilattice. A *bilattice* [11] is a structure $\langle \mathcal{B}, \preceq_t, \preceq_k \rangle$ where $\mathcal{B}$ is a non-empty, countable set and $\preceq_t$ (the *truth order*) and $\preceq_k$ (the *knowledge order*) are both partial orderings giving $\mathcal{B}$ the structure of a *complete lattice* with a top and bottom element. *Meet* (or *greatest lower bound*) and *join* (or *least upper bound*) under $\preceq_t$, denoted $\wedge$ and $\vee$, correspond to extensions of classical conjunction and disjunction. On the other hand, *meet* and *join* under $\preceq_k$ are denoted $\otimes$ and $\oplus$. $x \otimes y$ corresponds to the maximal information x and y can agree on, while $x \oplus y$ simply combines the information represented by x with that represented by y . *Top* and *bottom* under $\preceq_t$ are denoted $\mathbf{t}$ and $\mathbf{f}$, and *top* and *bottom* under $\preceq_k$ are denoted $\top$ and $\perp$, respectively. We will assume that bilattices are *infinitary distributive bilattices* in which all distributive laws connecting $\wedge, \vee, \otimes$ and $\oplus$ hold. We also assume that every bilattice satisfies the *infinitary interlacing conditions*, i.e. each of the lattice operations $\wedge, \vee, \otimes$ and $\oplus$ is monotone w.r.t. both orderings. An example of interlacing condition is: $x \preceq_t y$ and $x' \preceq_t y'$ implies $x \otimes x' \preceq_t y \otimes y'$. Finally, we assume that each bilattice has a *negation*, i.e. an operator $\neg$ that reverses the $\preceq_t$ ordering, leaves unchanged the $\preceq_k$ ordering, and verifies $\neg\neg x = x$. For instance, the simplest non-

trivial bilattice, called *FOUR*, is due to Belnap [1], who introduced a logic intended to deal with incomplete and/or inconsistent information – see also [6]. *FOUR* already illustrates many of the basic properties concerning bilattices. Essentially, *FOUR* extends the classical truth set $\{\mathbf{f}, \mathbf{t}\}$ to $\{\mathbf{f}, \mathbf{t}, \perp, \top\}$, where $\perp$ stands *unknown*, and $\top$ stands for *inconsistent*. In *FOUR*, $\perp \preceq_k \mathbf{f} \preceq_k \top$, $\perp \preceq_k \mathbf{t} \preceq_k \top$, $\mathbf{f} \preceq_t \perp \preceq_t \mathbf{t}$ and $\mathbf{f} \preceq_t \top \preceq_t \mathbf{t}$. Furthermore, we have that $\neg \mathbf{f} = \mathbf{t}$, $\neg \perp = \perp$, $\neg \top = \top$. In addition to the usual bilattice approach, we provide a family $\mathcal{F}$ of $\preceq_k$ and $\preceq_t$ -continuous binary and unary functions $f: \mathcal{B} \times \mathcal{B} \rightarrow \mathcal{B}$ and $f: \mathcal{B} \rightarrow \mathcal{B}$ to manipulate truth values. That is, for any $\preceq_k$ -monotone chain $x_0, x_1, \dots$ of values in $\mathcal{B}$, $f(\oplus_i x_i) = \oplus_i f(x_i)$ and for any $\preceq_t$ -monotone chain $x_0, x_1, \dots$ of values in $\mathcal{B}$, $f(\vee_i x_i) = \vee_i f(x_i)$. The binary case is similar. Notably, $\wedge, \vee, \otimes$ and $\vee$ are both $\preceq_k$ -continuous and $\preceq_t$ -continuous, while $\neg$ is $\preceq_k$ -continuous but not $\preceq_t$ -continuous (it is $\preceq_t$ -antitone). Interestingly, bilattices come up in two natural ways and are widely used. We just sketch them here in order to give a feeling of their application. The first bilattice construction method comes from [11]. Suppose we have two complete distributive lattices $\langle L_1, \preceq_1 \rangle$ and $\langle L_2, \preceq_2 \rangle$. Think of L_1 as a lattice of values we use when we measure the degree of belief of a statement, while think of L_2 as the lattice we use when we measure the degree of doubt of it. Now, we define the structure $L_1 \odot L_2$ as follows. The structure is $\langle L_1 \times L_2, \preceq_t, \preceq_k \rangle$, where (i) $\langle x_1, x_2 \rangle \preceq_t \langle y_1, y_2 \rangle$ if $x_1 \preceq_1 y_1$ and $y_2 \preceq_2 x_2$, while (ii) $\langle x_1, x_2 \rangle \preceq_k \langle y_1, y_2 \rangle$ if $x_1 \preceq_1 y_1$ and $x_2 \preceq_2 y_2$. In $L_1 \odot L_2$ the idea is: knowledge goes up if both degree of belief and degree of doubt go up; truth goes up if the degree of belief goes up, while the degree of doubt goes down. It is easily verified that $L_1 \odot L_2$ is a bilattice. Furthermore, if $L_1 = L_2 = L$, i.e. we are measuring belief and doubt in the same way, then negation can be defined as $\neg \langle x, y \rangle = \langle y, x \rangle$. That is, negation switches the roles of belief and doubt. Notably, under this approach fall work on paraconsistent logic programming (see, e.g. [6]) and anti-tonic logic programming (see, e.g. [7]). In the second construction method, suppose we have a complete distributive lattice of truth values $\langle L, \preceq \rangle$ (like e.g. in Many-valued Logics). Think of these values as the ‘actual’ values we are interested in, but due to lack of knowledge we are able just to ‘approximate’ the exact values. That is, rather than considering a pair $\langle x, y \rangle \in L \times L$ as indicator for degree of belief and doubt, $\langle x, y \rangle$ is interpreted as the set of elements $z \in L$ such that $x \preceq z \preceq y$. Therefore, a pair $\langle x, y \rangle$ is interpreted as an *interval*. An interval $\langle x, y \rangle$ may be seen as an approximation of an exact value. Formally, given a distributive lattice $\langle L, \preceq \rangle$, the *bilattice of intervals*, denoted $\mathcal{K}(L)$, is $\langle L \times L, \preceq_t, \preceq_k \rangle$, where: (i) $\langle x_1, x_2 \rangle \preceq_t \langle y_1, y_2 \rangle$ if $x_1 \preceq y_1$ and $x_2 \preceq y_2$, while (ii) $\langle x_1, x_2 \rangle \preceq_k \langle y_1, y_2 \rangle$ if $x_1 \preceq y_1$ and $y_2 \preceq x_2$. The intuition of those orders is that truth increases if the interval contains greater values, whereas the knowledge increases when the interval becomes more *precise*. Negation can be defined as $\neg \langle x, y \rangle = \langle \neg y, \neg x \rangle$, where $\neg$ is a negation operator on L . This approach has been used, e.g. in [14], where L is the unit interval $[0, 1]_{\mathbb{Q}} = [0, 1] \cap \mathbb{Q}$ with standard ordering.

Logic Programs. Fix a bilattice. We start with the definitions given in [9] and extend it to the case *arbitrary computable functions* $f \in \mathcal{F}$ are allowed in rule bodies to manipulate the truth values. For the ease of presentation, we limit our attention to propositional logic programs. The first order case can be handled by grounding. So, consider an alphabet of propositional letters. An *atom*, denoted A is a propositional letter. A literal, l ,

is of the form A or $\neg A$, where A is an atom. A *formula*, φ , is an expression built up from the literals, the members of a bilattice $\mathcal{B}$ using $\wedge, \vee, \otimes$ and $\oplus$ and the functions $f \in \mathcal{F}$. Note that members of the bilattice may appear in a formula, as well as functions: e.g. in *FOUR*, $f(p \wedge q, r \otimes \mathbf{f}) \oplus v$ is a formula. The intuition here is that the truth value of the formula $f(p \wedge q, r \otimes \mathbf{f}) \oplus v$ is obtained by determining the truth value of $p \wedge q$ and $r \otimes \mathbf{f}$, then apply the function f to them and join the result to the truth value of v . A *rule* is of the form $A \leftarrow \varphi$, where A is an atom and φ is a formula. The atom A is called the *head*, and the formula φ is called the *body*. A *normal logic program* (in the following, simply *logic program*), denoted with $\mathcal{P}$, is a finite set of rules. The *Herbrand base* of $\mathcal{P}$ (denoted $B_{\mathcal{P}}$) is the set of atoms occurring in $\mathcal{P}$. Given $\mathcal{P}$, the set $\mathcal{P}^*$ is constructed as follows; (i) if an atom A is not head of any rule in $\mathcal{P}^*$, then add the rule $A \leftarrow \mathbf{f}$ to $\mathcal{P}^*$; ² and (ii) replace several rules in $\mathcal{P}^*$ having same head, $A \leftarrow \varphi_1, A \leftarrow \varphi_2, \dots$ with $A \leftarrow \varphi_1 \vee \varphi_2 \vee \dots$. Note that in $\mathcal{P}^*$, each atom appears in the head of *exactly one* rule.

Example 1 ([14]). Consider $\mathcal{K}([0, 1]_{\mathbb{Q}})$, with $\wedge = \min$ and $\vee = \max$. Consider an insurance company, which has information about its customers used to determine the risk coefficient of each customer. Suppose a value of the risk coefficient is already known, but has to be re-evaluated (the client is a new client and his risk coefficient is given by his precedent insurance company). The company may have: (i) data grouped into a set of facts $\{(\text{Experience}(\text{john}) \leftarrow [0.7, 0.7]), (\text{Risk}(\text{john}) \leftarrow [0.5, 0.5]), (\text{Sport\_car}(\text{john}) \leftarrow [0.8, 0.8])\}$; and (ii) a set of rules, which after grounding are:

$$\begin{aligned} \text{Good\_driver}(\text{john}) &\leftarrow \text{Experience}(\text{john}) \wedge \neg \text{Risk}(\text{john}) \\ \text{Risk}(\text{john}) &\leftarrow 0.8 \cdot \text{Young}(\text{john}) \\ \text{Risk}(\text{john}) &\leftarrow 0.8 \cdot \text{Sport\_car}(\text{john}) \\ \text{Risk}(\text{john}) &\leftarrow \text{Experience}(\text{john}) \wedge \neg \text{Good\_driver}(\text{john}) \end{aligned}$$

Interpretations. An *interpretation* of a logic program on the bilattice $\langle \mathcal{B}, \preceq_t, \preceq_k \rangle$ is a mapping from atoms to members of $\mathcal{B}$. An interpretation I is extended from atoms to formulae as follows: (i) for $b \in \mathcal{B}$, $I(b) = b$; (ii) for formulae φ and φ' , $I(\varphi \wedge \varphi') = I(\varphi) \wedge I(\varphi')$, and similarly for $\vee, \otimes, \oplus$ and $\neg$; and (iii) for formulae $f(\varphi)$, $I(f(\varphi)) = f(I(\varphi))$, and similarly for binary functions. The truth and knowledge orderings are extended from $\mathcal{B}$ to the set $\mathcal{I}(\mathcal{B})$ of all interpretations point-wise: (i) $I_1 \preceq_t I_2$ iff $I_1(A) \preceq_t I_2(A)$, for every ground atom A ; and (ii) $I_1 \preceq_k I_2$ iff $I_1(A) \preceq_k I_2(A)$, for every ground atom A . We define $(I_1 \wedge I_2)(A) = I_1(A) \wedge I_2(A)$, and similarly for the other operations. With $\mathbf{I}_{\mathbf{f}}$ and $\mathbf{I}_{\perp}$ we denote the bottom interpretations under $\preceq_t$ and $\preceq_k$ respectively (they map any atom into $\mathbf{f}$ and $\perp$, respectively). It is easy to see that $\langle \mathcal{I}(\mathcal{B}), \preceq_t, \preceq_k \rangle$ is an infinitary interlaced and distributive bilattice as well.

Models. An interpretation I is a *model* of a logic program $\mathcal{P}$, denoted by $I \models \mathcal{P}$, iff for the *unique* rule involving A , $A \leftarrow \varphi \in \mathcal{P}^*$, $I(A) = I(\varphi)$ holds. Note that usually a model has to satisfy $I(\varphi) \preceq_t I(A)$ only, i.e. $A \leftarrow \varphi \in \mathcal{P}^*$ specifies the necessary condition on A , “ A is at least as true as φ ”. But, as $A \leftarrow \varphi \in \mathcal{P}^*$ is the unique rule with head A , the constraint becomes also sufficient (see e.g. [9]).

² It is a standard practice in logic programming to consider such atoms as *false*.

Query. A *query*, denoted q , is an expression of the form $?A$ (*query atom*), intended as a question about the truth of the atom A in the selected intended model of $\mathcal{P}$. We also allow a query to be a *set* $\{?A_1, \dots, ?A_n\}$ of query atoms. In that latter case we ask about the truth of all the atoms A_i in the intended model of a logic program $\mathcal{P}$.

Semantics of Logic Programs. The semantics of a logic program $\mathcal{P}$ is determined by selecting a particular model, or a set of models, of $\mathcal{P}$. In our context we will consider two possible intended semantics over bilattices, namely the *Kripke-Kleene* (KK) and the *Well-Founded semantics* (WF) [9, 21], which are well-established semantics for non-monotonic negation over bilattices. It is well-known that the WF semantics is more informative (provides more knowledge) than the KK semantics.

Example 2. Consider $\mathcal{FOUR}$ and $\mathcal{P} = \{(p \leftarrow p), (q \leftarrow \neg r), (r \leftarrow \neg q \wedge \neg p)\}$. Let us identify an interpretation I as a triple for the truth-values of $\langle p, q, r \rangle$. Then the models of $\mathcal{P}$ are $I_1 = \langle \perp, \perp, \perp \rangle$, $I_2 = \langle \perp, \mathbf{t}, \mathbf{f} \rangle$, $I_3 = \langle \mathbf{f}, \perp, \perp \rangle$, $I_4 = \langle \mathbf{f}, \mathbf{f}, \mathbf{t} \rangle$, $I_5 = \langle \mathbf{f}, \mathbf{t}, \mathbf{f} \rangle$, $I_6 = \langle \mathbf{f}, \top, \top \rangle$, $I_7 = \langle \mathbf{t}, \mathbf{t}, \mathbf{f} \rangle$, $I_8 = \langle \top, \mathbf{t}, \mathbf{f} \rangle$ and $I_9 = \langle \top, \top, \top \rangle$. The KK semantics will be I_1 , while the WF semantics will be I_3 . Note that $I_1 \preceq_k I_3$.

Formally, the *Kripke-Kleene semantics* has a simple and intuitive characterization, as it corresponds to the $\preceq_k$ -least model of a logic program, i.e. the *Kripke-Kleene model* of a logic program $\mathcal{P}$ is $KK(\mathcal{P}) = \min_{\preceq_k} \{I : I \models \mathcal{P}\}$. The *existence and uniqueness* of $KK(\mathcal{P})$ is guaranteed by the fixed-point characterization, by means of the *immediate consequence operator* $\Phi_{\mathcal{P}}$. For an interpretation I , for any ground atom A with (unique) $A \leftarrow \varphi \in \mathcal{P}^*$, $\Phi_{\mathcal{P}}(I)(A) = I(\varphi)$. We can show that (based on [9, 14]) the function $\Phi_{\mathcal{P}}$ is $\preceq_k$ -continuous over $\mathcal{I}(\mathcal{B})$, the set of fixed-points of $\Phi_{\mathcal{P}}$ is a complete lattice under $\preceq_k$ and, thus, $\Phi_{\mathcal{P}}$ has a $\preceq_k$ -least fixed-point and I is a model of a program $\mathcal{P}$ iff I is a fixed-point of $\Phi_{\mathcal{P}}$. Therefore, the Kripke-Kleene model of $\mathcal{P}$ coincides with the least fixed-point of $\Phi_{\mathcal{P}}$ under $\preceq_k$, which can be computed in the usual way by iterating $\Phi_{\mathcal{P}}$ over $I_{\perp}$ and is attained after at most ω iterations.

Example 3. Consider $\mathcal{K}([0, 1]_{\mathbb{Q}})$, the function $f(\langle x, 1 \rangle) = \langle \frac{x+a}{2}, 1 \rangle$ ($0 < a \leq 1, a \in \mathbb{Q}$), and $\mathcal{P} = \{A \leftarrow f(A)\}$. Then the KK model is attained after ω steps of $\Phi_{\mathcal{P}}$ iterations over $I_{\perp} = \langle 0, 1 \rangle$ and is $KK(\mathcal{P})(A) = \langle a, 1 \rangle$.

The *Well-Founded semantics* over bilattices is derived directly from Fitting's formulation [9]. Informally, an interpretation I is the *well-founded model* of a logic program $\mathcal{P}$ if I is the $\preceq_k$ -least interpretation satisfying $I = I'$, where I' is computed according to the so-called *Gelfond-Lifschitz transformation*: (i) substitute (fix) in $\mathcal{P}^*$ the negative literals by their evaluation with respect to I . Let $\mathcal{P}^I$ be the resulting *positive* program, called *reduct* of $\mathcal{P}$ w.r.t. I ; and (ii) compute the truth-minimal model I' of $\mathcal{P}^I$. For instance, given $\mathcal{P}$ and I_3 in Example 2, $\mathcal{P}^{I_3}$ is $\{(p \leftarrow p), (q \leftarrow \perp), (r \leftarrow \perp \wedge \mathbf{t})\}$, whose $\preceq_t$ -least model is I_3 . Also I_3 is the $\preceq_k$ -least model satisfying the above condition. Therefore, I_3 is well-founded model. Note that the $\preceq_t$ -least model of $\mathcal{P}^{I_1} (= \mathcal{P}^{I_3})$, is I_3 , so I_1 does not satisfy the Gelfond-Lifschitz transformation. Formally, Fitting [9] relies on a binary immediate consequence operator $\Psi_{\mathcal{P}}$, which accepts two input interpretations I and J over a bilattice, the first one is used to assign meanings to positive literals, while the second one is used to assign meanings to negative literals. Let I and J be two interpretations in the bilattice $\langle \mathcal{I}(\mathcal{B}), \preceq_t, \preceq_k \rangle$. The notion of *pseudo-interpretation*

$I \triangle J$ over the bilattice is defined as follows: for an atom A : $(I \triangle J)(A) = I(A)$ and $(I \triangle J)(\neg A) = \neg J(A)$. Pseudo-interpretations are extended to non-literals in the obvious way. For instance, $(I \triangle J)(f(\neg A \wedge B)) = f((I \triangle J)(\neg A \wedge B)) = f((I \triangle J)(\neg A) \wedge (I \triangle J)(B)) = f(\neg J(A) \wedge I(B))$. We can now define $\Psi_{\mathcal{P}}$ as follows. For $I, J \in \mathcal{I}(\mathcal{B})$, $\Psi_{\mathcal{P}}(I, J)$ is the interpretation, which for any atom A with $A \leftarrow \varphi \in \mathcal{P}^*$, satisfies $\Psi_{\mathcal{P}}(I, J)(A) = (I \triangle J)(\varphi)$. Note that $\Phi_{\mathcal{P}}$ is a special case of $\Psi_{\mathcal{P}}$, as by construction $\Phi_{\mathcal{P}}(I) = \Psi_{\mathcal{P}}(I, I)$. Similarly to [9], we can show that the operator $\Psi_{\mathcal{P}}$ is $\preceq_k$ -continuous in both arguments, $\preceq_t$ -continuous in its first argument and $\preceq_t$ -antitone in its second argument. To define the well-founded semantics, Fitting [9] further introduces the $\Psi'_{\mathcal{P}}$ operator, whose $\preceq_k$ -least fixed-point will be the WF model of a program. For any interpretation I , $\Psi'_{\mathcal{P}}(I)$ is the $\preceq_t$ -least fixed-point of the operator $\lambda x. \Psi_{\mathcal{P}}(x, I)$, i.e.

$$\Psi'_{\mathcal{P}}(I) = \text{lfp}_{\preceq_t}(\lambda x. \Psi_{\mathcal{P}}(x, I)) . \quad (1)$$

Due to the $\preceq_t$ -continuity of $\Psi_{\mathcal{P}}$ on its first argument, $\Psi'_{\mathcal{P}}$ is well defined. $\Psi'_{\mathcal{P}}(I)$ can be computed by iterating $\Psi_{\mathcal{P}}(x, I)$ over $\mathcal{I}_{\mathcal{f}}$ and the limit is attained in at most ω iterations. In particular, we can show that the operator $\Psi'_{\mathcal{P}}$ is $\preceq_k$ -continuous, $\preceq_t$ -antitone and every fixed-point of $\Psi'_{\mathcal{P}}$ is also a fixed-point of $\Phi_{\mathcal{P}}$, i.e. a model of $\mathcal{P}$. Therefore, the set of fixed-points of $\Psi'_{\mathcal{P}}$ is a complete lattice under $\preceq_k$ and, thus, $\Psi'_{\mathcal{P}}$ has a $\preceq_k$ -least fixed-point, which is denoted $WF(\mathcal{P})$. $WF(\mathcal{P})$ is the *Well-Founded model* of $\mathcal{P}$. Of course, the well-founded model can be computed by iterating $\Psi'_{\mathcal{P}}$ starting from $\mathcal{I}_{\perp}$ and the limit is attained in at most ω iterations.

Example 4. Consider $\mathcal{K}([0, 1]_{\mathbb{Q}})$ and $\mathcal{P} = \{ (A \leftarrow A \vee B), (B \leftarrow (\neg C \wedge A) \vee \langle 0.3, 0.5 \rangle), (C \leftarrow \neg B \vee \langle 0.2, 0.4 \rangle) \}$. Then the computation of $KK(\mathcal{P})$, as $\preceq_k$ -least fixed-point of $\Phi_{\mathcal{P}}$, converges to $KK(\mathcal{P})(A, B, C) = \langle \langle 0.3, 1 \rangle, \langle 0.3, 0.8 \rangle, \langle 0.2, 0.7 \rangle \rangle$. The computation of $WF(\mathcal{P})$, as $\preceq_k$ -least fixed-point of $\Psi'_{\mathcal{P}}$, converges to $WF(\mathcal{P})(A, B, C) = \langle \langle 0.3, 0.5 \rangle, \langle 0.3, 0.5 \rangle, \langle 0.5, 0.7 \rangle \rangle$. Notice that $KK(\mathcal{P}) \preceq_k WF(\mathcal{P})$, as expected.

Example 5. Consider Example 1. It turns out that the KK semantics is I_1 , while the WF semantics is I_2 , where (for ease, we use first letter only) $I_1(R(j)) = [0.64, 0.8]$, $I_1(S(j)) = [0.8, 0.8]$, $I_1(Y(j)) = [0, 1]$, $I_1(G(j)) = [0.2, 0.36]$, $I_1(E(j)) = [0.7, 0.7]$, while $I_2(R(j)) = [0.64, 0.7]$, $I_2(S(j)) = [0.8, 0.8]$, $I_2(Y(j)) = [0, 0]$, $I_2(G(j)) = [0.3, 0.36]$, $I_2(E(j)) = [0.7, 0.7]$. Note that $I_1 \preceq_k I_2$. In fact, I_2 establish that j ohn's degree of Risk is in between $[0.64, 0.7]$, while I_1 is less precise. Also note that $I_2(Y(j)) = [0, 0]$ (= false), while $I_1(Y(j)) = [0, 1]$ (= unknown).

3 Top-Down Query Answering

Given a logic program $\mathcal{P}$ and either the KK or the WF model, one way to answer to a query $?A$ is to compute the intended model I of $\mathcal{P}$ by a bottom-up fixed-point computation and then answer with $I(A)$. This always requires to compute a whole model, even if in order to determine $I(A)$, not all the atom's truth is required. Our goal is to present a simple, yet general top-down method, which relies on the computation of just a part of an intended model. Essentially, we will try to determine the value

of a single atom by investigating only a part of the program $\mathcal{P}$. Our method is based on a transformation of a program into a system of equations of monotonic functions over lattices and bilattices for which we compute the least fixed-point in a top-down style. The idea is the following. Let $\langle \mathcal{B}, \preceq_t, \preceq_k \rangle$ be a bilattice and let $\mathcal{P}$ be a logic program. Consider the Herbrand base $B_{\mathcal{P}} = \{A_1, \dots, A_n\}$ of $\mathcal{P}$ and consider $\mathcal{P}^*$. Let us associate to each atom $A_i \in B_{\mathcal{P}}$ a variable x_i , which will take a value in the domain $\mathcal{B}$ (sometimes, we will refer to that variable with x_A as well). An interpretation I may be seen as an assignment of bilattice values to the variables $x_1, \dots, x_n$. For an immediate consequence operator O , e.g. $\Phi_{\mathcal{P}}$, a fixed-point is such that $I = O(I)$, i.e. for all atoms $A_i \in B_{\mathcal{P}}$, $I(A_i) = O(I)(A_i)$. Therefore, we may identify the fixed-points of O as the solutions over $\mathcal{B}$ of the system of equations of the following form:

$$\begin{aligned} x_1 &= f_1(x_{1_1}, \dots, x_{1_{a_1}}), \\ &\vdots \\ x_n &= f_n(x_{n_1}, \dots, x_{n_{a_n}}), \end{aligned} \tag{2}$$

where for $1 \leq i \leq n$, $1 \leq k \leq a_i$, we have $1 \leq i_k \leq n$. Each variable x_{i_k} will take a value in the domain $\mathcal{B}$, each (monotone) function f_i determines the value of x_i (i.e. A_i) given an assignment $I(A_{i_k})$ to each of the a_i variables x_{i_k} . The function f_i implements $O(I)(A_i)$. Of course, we are especially interested in the computation of the least fixed-point of the above system. For instance, by considering the logic program in Example 2, the fixed-points of the $\Phi_{\mathcal{P}}$ operator are the solutions over a bilattice of the system of equations ($p \mapsto x_1, q \mapsto x_2, r \mapsto x_3$)

$$x_1 = x_1, \quad x_2 = \neg x_3, \quad x_3 = \neg x_2 \wedge \neg x_1. \tag{3}$$

It is easily verified that all nine interpretations I_i in Example 2 are bijectively related to the solutions of the system (3) over $\mathcal{FOUR}$ and $(x_1, x_2, x_3) = (\perp, \perp, \perp)$ is the $\preceq_k$ -least solution and corresponds to the Kripke-Kleene model of $\mathcal{P}$.

Now, at first present the general procedure for the top-down computation of the value of variable in the $\preceq$ -least solution of the system (2), given a lattice $\mathcal{L} = \langle L, \preceq \rangle$. Then, we will customize it to the particular case of the Kripke-Kleene semantics and the well-founded semantics. We use some auxiliary functions. $\mathbf{s}(x)$ denotes the set of *sons* of x , i.e. $\mathbf{s}(x_i) = \{x_{i_1}, \dots, x_{i_{a_i}}\}$ (the set of variables appearing in the right hand side of the definition of x_i). $\mathbf{p}(x)$ denotes the set of *parents* of x , i.e. the set $\mathbf{p}(x) = \{x_i : x \in \mathbf{s}(x_i)\}$ (the set of variables depending on the value of x). In the general case, we assume that each function $f_i: L^{a_i} \mapsto L$ in Equation (2) is $\preceq$ -monotone. We also use f_x in place of f_i , for $x = x_i$. We refer to the monotone system as in Equation (2) as the tuple $\mathcal{S} = \langle \mathcal{L}, V, \mathbf{f} \rangle$, where $\mathcal{L}$ is a lattice, $V = \{x_1, \dots, x_n\}$ are the variables and $\mathbf{f} = \langle f_1, \dots, f_n \rangle$ is the tuple of functions. As it is well known, a monotonic equation system as (2) has a $\preceq$ -least solution, $\text{lfp}_{\preceq}(\mathbf{f})$, the $\preceq$ -least fixed-point of $\mathbf{f}$ is given as the least upper bound of the $\preceq$ -monotone sequence, $\mathbf{y}_0, \dots, \mathbf{y}_i, \dots$, where $\mathbf{y}_0 = \perp$ and $\mathbf{y}_{i+1} = \mathbf{f}(\mathbf{y}_i)$.

Example 6. Consider Example 4. The equational system is $\{x_A = x_A \vee x_B, \quad x_B = (\neg x_C \wedge x_A) \vee \langle 0.3, 0.5 \rangle, \quad x_C = \neg x_B \vee \langle 0.2, 0.4 \rangle\}$. The $\preceq_k$ -least fixed-point computation is $\mathbf{y}_0 = \perp = \langle [0, 1]_{\mathbb{Q}}, [0, 1]_{\mathbb{Q}}, [0, 1]_{\mathbb{Q}} \rangle$ (the triples represent (x_A, x_B, x_C)),

$\mathbf{y}_1 = \langle [0, 1]_{\mathbb{Q}}, [0.3, 1]_{\mathbb{Q}}, [0.2, 1]_{\mathbb{Q}} \rangle$, $\mathbf{y}_2 = \langle [0.3, 1]_{\mathbb{Q}}, [0.3, 0.8]_{\mathbb{Q}}, [0.2, 0.7]_{\mathbb{Q}} \rangle$ and $\mathbf{y}_3 = \mathbf{y}_2$, which corresponds to the KK model of the program, as expected.

Informally our algorithm works as follows. Assume we are interested in the value of x_0 in the least fixed-point of the system. We associate to each variable x_i a marking $v(x_i)$ denoting the current value of x_i (the mapping v contains the current value associated to the variables). Initially, $v(x_i)$ is $\perp$. We start with putting x_0 in the *active* list of variables A , for which we evaluate whether the current value of the variable is identical to whatever its right-hand side evaluates to. When evaluating a right-hand side it might of course turn out that we do indeed need a better value of some sons, which will assumed to have the value $\perp$ and put them on the list of active nodes to be examined. In doing so we keep track of the dependencies between variables, and whenever it turns out that a variable changes its value (actually, it can only $\preceq$ -increase) all variables that might depend on this variable are put in the active set to be examined. At some point (even if cyclic definitions are present) the active list will become empty and we have actually found part of the fixed-point, sufficient to determine the value of the query x_0 . The algorithm is given below.

Procedure $Solve(\mathcal{S}, Q)$

Input: $\preceq$ -monotonic system $\mathcal{S} = \langle \mathcal{L}, V, f \rangle$, where $Q \subseteq V$ is the set of query variables;

Output: A set $B \subseteq V$, with $Q \subseteq B$ such that the mapping v equals $\text{lfp}_{\preceq}(f)$ on B .

1. $A := Q$, $\text{dg} := Q$, $\text{in} := \emptyset$, **for all** $x \in V$ **do** $v(x) = \perp$, $\text{exp}(x) = \text{false}$
2. **while** $A \neq \emptyset$ **do**
3. **select** $x_i \in A$, $A := A \setminus \{x_i\}$, $\text{dg} := \text{dg} \cup s(x_i)$
4. $r := f_i(v(x_{i_1}), \dots, v(x_{i_{a_i}}))$
5. **if** $r \succ v(x_i)$ **then** $v(x_i) := r$, $A := A \cup (p(x_i) \cap \text{dg})$ **fi**
6. **if not** $\text{exp}(x_i)$ **then** $\text{exp}(x_i) = \text{true}$, $A := A \cup (s(x_i) \setminus \text{in})$, $\text{in} := \text{in} \cup s(x_i)$ **fi**
- od**

The variable dg collects the variables that may influence the value of the query variables, the array variable exp traces the equations that has been “expanded” (the body variables are put into the active list), while the variable in keeps track of the variables that have been put into the active list so far due to an expansion (to avoid, to put the same variable multiple times in the active list due to function body expansion). The attentive reader will notice that the *Solve* procedure has much in common with the so-called *tabulation* procedures, like [3, 5]. Indeed, it is a generalization of it to arbitrary monotone equational systems over lattices.

Example 7. Consider Example 6 and query variable x_A . Below is a sequence of $Solve(\mathcal{S}, \{x_A\})$ computation w.r.t. $\preceq_k$. Each line is a sequence of steps in the ‘while loop’. What is left unchanged is not reported.

1. $A := \{x_A\}$, $x_i := x_A$, $A := \emptyset$, $\text{dg} := \{x_A, x_B\}$, $r := \perp$, $\text{exp}(x_A) := \text{true}$, $A := \{x_A, x_B\}$,
 $\text{in} := \{x_A, x_B\}$
2. $x_i := x_B$, $A := \{x_A\}$, $\text{dg} := \{x_A, x_B, x_C\}$, $r := \langle 0.3, 1 \rangle$, $v(x_B) := \langle 0.3, 1 \rangle$, $A := \{x_A, x_C\}$,
 $\text{exp}(x_B) := \text{true}$, $\text{in} := \{x_A, x_B, x_C\}$
3. $x_i := x_C$, $A := \{x_A\}$, $r := \langle 0.2, 0.7 \rangle$, $v(x_C) := \langle 0.2, 0.7 \rangle$, $A := \{x_A, x_B\}$, $\text{exp}(x_C) := \text{true}$
4. $x_i := x_B$, $A := \{x_A\}$, $r := \langle 0.3, 0.8 \rangle$, $v(x_B) := \langle 0.3, 0.8 \rangle$, $A := \{x_A, x_C\}$

5. $x_i := x_C, A := \{x_A\}, r := \langle 0.2, 0.7 \rangle$
6. $x_i := x_A, A := \emptyset, r := \langle 0.3, 1 \rangle, v(x_A) := \langle 0.3, 1 \rangle, A := \{x_A, x_B\}$
7. $x_i := x_B, A := \{x_A\}, r := \langle 0.3, 0.8 \rangle,$
8. $x_i := x_A, A := \emptyset, r := \langle 0.3, 1 \rangle$
9. **stop. return** $v(x_A, x_B, x_C) = \langle \langle 0.3, 1 \rangle, \langle 0.3, 0.8 \rangle, \langle 0.2, 0.7 \rangle \rangle$

The fact that only a part of the model is computed becomes evident, as the computation does not change if we add any program $\mathcal{P}'$ to $\mathcal{P}$ in which A, B and C do not occur.

Given a system $\mathcal{S} = \langle \mathcal{L}, V, \mathbf{f} \rangle$, where $\mathcal{L} = \langle L, \preceq \rangle$, let $h(\mathcal{L})$ be the *height* of the truth-value set L , i.e. the length of the longest strictly $\preceq$ -increasing chain in L minus 1, where the length of a chain $v_1, \dots, v_\alpha, \dots$ is the cardinal $|\{v_1, \dots, v_\alpha, \dots\}|$. The *cardinal* of a countable set X is the least ordinal α such that α and X are *equipollent*, i.e. there is a bijection from α to X . For instance, $h(\mathcal{FOUR}) = 2$ w.r.t. $\preceq_k$ as well as w.r.t. $\preceq_t$, while $h(\mathcal{K}([0, 1]_{\mathbb{Q}})) = \omega$. It can be shown that the above algorithm behaves correctly.

Proposition 8. *Given a monotone system of equations $\mathcal{S} = \langle \mathcal{L}, V, \mathbf{f} \rangle$, then there is a limit ordinal λ such that after $|\lambda|$ steps $\text{Solve}(\mathcal{S}, Q)$ determines a set $B \subseteq V$, with $Q \subseteq B$ such that the mapping v equals $\text{lfp}_{\preceq}(\mathbf{f})$ on B , i.e. $v|_B = \text{lfp}_{\preceq}(\mathbf{f})|_B$.*

From a computational point of view, by means of appropriate data structures, the operations on A , v , dg , in , exp , p and s can be performed in constant time. Therefore, Step 1. is $O(|V|)$, all other steps, except Step 2. and Step 4. are $O(1)$. Let $c(f_x)$ be the maximal cost of evaluating function f_x on its arguments, so Step 4. is $O(c(f_x))$. It remains to determine the number of loops of Step 2. In case the height $h(\mathcal{L})$ of the bilattice $\mathcal{L}$ is finite, observe that any variable is increasing in the $\preceq$ order as it enters in the A list (Step 5.), except it enters due to Step 6., which may happen one time only. Therefore, each variable x_i will appear in A at most $a_i \cdot h(\mathcal{L}) + 1$ times, where a_i is the arity of f_i , as a variable is only re-entered into A if one of its son gets an increased value (which for each son only can happen $h(\mathcal{L})$ times), plus the additional entry due to Step 6. As a consequence, the worst-case complexity is $O(\sum_{x_i \in V} (c(f_i) \cdot (a_i \cdot h(\mathcal{L}) + 1)))$. Therefore:

Proposition 9. *Given a monotone system of equations $\mathcal{S} = \langle \mathcal{L}, V, \mathbf{f} \rangle$. If the computing cost of each function in $\mathbf{f}$ is bounded by c , the arity bounded by a , and the height is bounded by h , then the worst-case complexity of the algorithm Solve is $O(|V|cah)$.*

In case the height of a bilattice is not finite, the computation may not terminate after a finite number of steps (see Example 3). Fortunately, under reasonable assumptions on the functions, we may guarantee the termination of Solve . We exploit two of such conditions. Consider a monotonic equational system $\mathcal{S} = \langle \mathcal{L}, V, \mathbf{f} \rangle$. Consider a function $f: L \rightarrow L$, where $\langle L, \preceq \rangle$ is a lattice. Let $[\perp]_f$ be the *f-closure* of $\{\perp\}$, i.e. the smallest set that contains $\{\perp\}$ and is closed under f . We say that f has a *finite generation* (see also [2] for more on this issue) iff $[\perp]_f$ is finite. For instance, it can be verified that the functions $\wedge, \vee, \otimes, \oplus, \neg$ have a finite generation on *any* finite set $X \subseteq \mathcal{B}$. More concretely, over the interval bilattice on $[0, 1]_{\mathbb{Q}}$, $\min, \max, 1 - x$ and Lukasiewicz t-norm and t-conorm, $\max(x + y - 1, 0), \min(x + y, 1)$ have a finite generation, while e.g. the product t-norm $x \cdot y$ and its t-conorm $x + y - x \cdot y$ have not. Note also that

if f, g have a finite generation over X then so has $f \circ g$. Therefore, given an equational system $\mathcal{S} = \langle \mathcal{L}, V, \mathbf{f} \rangle$. If $\mathbf{f}$ has a finite generation, then $[\perp]_{\mathbf{f}}$ is finite. That is, $\{\perp, \mathbf{f}(\perp), \mathbf{f}^2(\perp), \dots\}$ is finite. In particular, on induction on the computation of the $\preceq$ -least fixed-point of $\mathcal{S}$ it can be shown that at each step of the bottom-up computation of the $\preceq$ -least fixed-point, the values of the variables are in $[\perp]_{\mathbf{f}}$. Therefore, the *height* of $[\perp]_{\mathbf{f}}$, $h([\perp]_{\mathbf{f}})$, is finite. On the other hand, it can easily be seen that *Solve* terminates if the sequence, $\perp, \mathbf{f}(\perp), \mathbf{f}^2(\perp), \dots$ converges after a finite number of steps. Therefore:

Proposition 10. *Given a monotone system of equations $\mathcal{S} = \langle \mathcal{L}, V, \mathbf{f} \rangle$. Then *Solve* terminates iff $\mathbf{f}$ has a finite generation. If the cost of computing each of the functions in $\mathbf{f}$ is bounded by c and the arity bounded by a then the worst-case complexity of the algorithm *Solve* is $O(|V|cah)$, where h is the height of $[\perp]_{\mathbf{f}}$.*

The second condition, which guarantees the termination of *Solve*, is inspired directly by [4] and is a special case of above. On bilattices, we say that a function $f: \mathcal{B}^n \rightarrow \mathcal{B}$ is *bounded* iff $f(x_1, \dots, x_n) \preceq_k \otimes_i x_i$. Now, consider a monotone system of equations $\mathcal{S} = \langle \mathcal{L}, V, \mathbf{f} \rangle$. We say that $\mathbf{f}$ is *bounded* iff each f_i is a composition of functions, each of which is either bounded, or a constant in $\mathcal{B}$ or one of $\vee, \wedge, \oplus, \otimes$ and $\neg$. For instance, the function in Example 3 is not bounded, while $f_i(\langle x, y \rangle) = \langle \max(0, x + y - 1), 1 \rangle \wedge \langle 0.3, 0.4 \rangle$ over $\mathcal{K}([0, 1]_{\mathbb{Q}})$ is. The idea is to prevent the existence of an infinite ascending chain of the form $\perp \prec_k \mathbf{f}(\perp) \prec_k \dots \prec_k \mathbf{f}^m(\perp) \prec_k \dots$. In fact, roughly, consider a $\preceq_k$ -monotone function $\mathbf{f} = \mathbf{g} \circ \mathbf{h}$, where $\mathbf{g}$ is a bounded function, while $\mathbf{h}$ is the composition of constants in $\mathcal{B}$ or functions among $\vee, \wedge, \oplus, \otimes$ and $\neg$. Then $\perp \preceq_k \mathbf{f}(\perp) = \mathbf{g} \circ \mathbf{h}(\perp) = \mathbf{g}(\mathbf{h}(\perp)) \preceq_k \mathbf{h}(\perp)$. But $\mathbf{h}$ has a finite generation and, thus, so has $\mathbf{f}$. The argument for $\mathbf{f} = \mathbf{h} \circ \mathbf{g}$ is similar. Therefore:

Proposition 11. *Given a monotone system of equations $\mathcal{S} = \langle \mathcal{L}, V, \mathbf{f} \rangle$, where $\mathbf{f}$ is bounded. Then *Solve* terminates.*

Note that for bounded functions $\mathbf{f} = \mathbf{g} \circ \mathbf{h}$, the height of $[\perp]_{\mathbf{f}}$ is given by the height of $[\perp]_{\mathbf{h}}$. We believe that this latter height is bounded by the number $n = |V|$ as we conjecture that $\mathbf{h}^n(\perp) = \mathbf{h}^{n+1}(\perp)$ (this is compatible with [4]). This would imply that the worst-case complexity of the algorithm *Solve* is $O(|V|^2ca)$ in that case.

3.1 Top-Down Query Answering Under the Kripke-Kleene Semantics

We start with the Kripke-Kleene semantics, for which we have almost anticipated how we will proceed. Let $\mathcal{P}$ be a logic program and consider $\mathcal{P}^*$. As already pointed out, each atom appears exactly once in the head of a rule in $\mathcal{P}^*$. The system of equations that we build from $\mathcal{P}^*$ is straightforward. Assign to each atom A a variable x_A and substitute in $\mathcal{P}^*$ each occurrence of A with x_A . Finally, substitute each occurrence of $\leftarrow$ with $=$ and let $\mathcal{S}_{KK}(\mathcal{P}) = \langle \mathcal{L}, V, \mathbf{f}_{\mathcal{P}} \rangle$ be the resulting equational system (see Equation 3). Of course, $|V| = |\mathcal{B}_{\mathcal{P}}|$, $|\mathcal{S}_{KK}(\mathcal{P})|$ can be computed in time $O(|\mathcal{P}|)$ and all functions in $\mathcal{S}_{KK}(\mathcal{P})$ are $\preceq_k$ -continuous. As $\mathbf{f}_{\mathcal{P}}$ is one to one related to $\Phi_{\mathcal{P}}$, it follows that the $\preceq_k$ -least fixed-point of $\mathcal{S}_{KK}(\mathcal{P})$ corresponds to the Kripke-Kleene semantics of $\mathcal{P}$. The algorithm *Solve*_{KK}($\mathcal{P}, ?A$), first computes $\mathcal{S}_{KK}(\mathcal{P})$ and then calls *Solve*($\mathcal{S}_{KK}(\mathcal{P}), \{x_A\}$) and returns the output v on the query variable, where v is the output of the call to *Solve*. *Solve*_{KK} behaves correctly (see Example 7).

Proposition 12. *Let $\mathcal{P}$ and $?A$ be a logic program and a query, respectively. Then $KK(\mathcal{P})(A) = \text{Solve}_{KK}(\mathcal{P}, \{?A\})$ ³.*

From a computational point of view, we can avoid the cost of translating $\mathcal{P}$ into $\mathcal{S}_{KK}(\mathcal{P})$ as we can directly operate on $\mathcal{P}$. So the cost $O(|\mathcal{P}|)$ can be avoided. In case the height of the bilattice is finite, from Proposition 9 it follows immediately that the worst-case complexity for top-down query answering under the Kripke-Kleene semantics of a logic program $\mathcal{P}$ is $O(|B_{\mathcal{P}}|cah)$. Furthermore, often the cost of computing each of the functions of $\mathbf{f}_{\mathcal{P}}$ is in $O(1)$. By observing that $|B_{\mathcal{P}}|a$ is in $O(|\mathcal{P}|)$ we immediately have that in this case the complexity is $O(|\mathcal{P}|h)$. It follows that over the bilattice $\mathcal{FOUR}$ ($h = 2$) the top-down algorithm works in linear time. Moreover, if the height is a fixed parameter, i.e. a constant, we can conclude that the additional expressive power of Kripke-Kleene semantics of logic programs over bilattices (with functions with constant cost) does not increase the computational complexity of classical propositional logic programs, which is linear. The computational complexity of the case where the height of the bilattice is not finite is determined by Proposition 10 and Proposition 11. In general, the continuity of the functions in $\mathcal{S}_{KK}(\mathcal{P})$ guarantees the termination after at most ω steps.

3.2 Top-Down Query Answering Under the Well-Founded Semantics

We address now the issue of a top-down computation of the value of a query under the well-founded semantics. As we have seen, according to Fitting's formulation, the well-founded semantics of a logic program $\mathcal{P}$ is the $\preceq_k$ -least fixed-point of the operator $\Psi'_{\mathcal{P}}(I) = \text{lfp}_{\preceq_t}(\lambda x. \Psi_{\mathcal{P}}(x, I))$. Before we are going to present our top-down procedure for the well-founded semantics, we roughly explain the approach. To this purpose, let us consider Example 2. Assume that our query is $?r$ and consider the related equational system (3). So, our query variable is x_3 . Following the *Solve* algorithm, x_3 becomes the active variable. We have to introduce a major change in Step 4.: it is not hard to see that, due to Equation (1), in order to compute $r := \neg x_2 \wedge \neg x_1$, we have to compute the values of x_1 and x_2 w.r.t. the $\preceq_t$ -least fixed-point of another equational system, where the current partial evaluation $\mathbf{v}$ acts as the interpretation I . That is, we have to make a call to another instance of the *Solve* algorithm, which computes the values of x_1 and x_2 w.r.t. to the current evaluation $\mathbf{v}(x_1, x_2, x_3)$. In our case, we consider the equational system (3) in which negated variables have been replaced with their value w.r.t. to the current evaluation and, thus, we replace $\neg x_1, \neg x_2$ and $\neg x_3$ with $\mathbf{v}(x_1)$ and $\mathbf{v}(x_2)$, and $\mathbf{v}(x_3)$ respectively. Once the sub-routine call gives us back the values of the arguments x_1, x_2 we compute $r := \neg x_2 \wedge \neg x_1$ and continue with Step 5.

Let us formalize the above illustrated concept. Given a logic program $\mathcal{P}$, given a truth value assignment I , let us denote $\mathcal{S}(\mathcal{P}^I)$ the equational system obtained from $\mathcal{S}_{KK}(\mathcal{P})$ in which all occurrences of $\neg x$ have been replaced with $\neg I(x)$, but $\mathcal{S}(\mathcal{P}^I)$ is based on the $\preceq_t$ order rather than on $\preceq_k$. Then it can be verified that $\text{Solve}(\mathcal{S}(\mathcal{P}^I), Q)$ outputs a set $B \subseteq V$, with $Q \subseteq B$, s.t. the mapping $\mathbf{v}$ equals to the $\preceq_t$ -least fixed-point on B of the functions in $\mathcal{S}(\mathcal{P}^I)$ and $\mathbf{v}|_B = \Psi'_{\mathcal{P}}(I)|_B$. Moreover, from a computational

³ The extension to a set of query atoms is straightforward.

complexity point of view, the same properties of *Solve* hold for $Solve(\mathcal{S}(\mathcal{P}^I), Q)$ as well. Finally, $Solve_{WF}(\mathcal{P}, ?A)$ is as $Solve_{KK}(\mathcal{P}, ?A)$, except that Step 4. is replaced with the statements $Q := s(x_i)$; $I := v$; $v' := Solve(\mathcal{S}(\mathcal{P}^I), Q)$; $r := f_i(v'(x_{i_1}), \dots, v'(x_{i_{a_i}}))$. It can be shown that the following holds:

Proposition 13. *Let $\mathcal{P}$ and $?A$ be a logic program and a query, respectively. Then $WF(\mathcal{P})(A) = Solve_{WF}(\mathcal{P}, ?A)$.*

Example 14. *Consider Example 6 and query variable x_A . Below is a sequence of $Solve_{WW}(\mathcal{P}, ?A)$ computation. It resembles the one we have seen in Example 7. Each line is a sequence of steps in the ‘while loop’. What is left unchanged is not reported.*

1. $A := \{x_A\}, x_i := x_A, A := \emptyset, dg := \{x_A, x_B\}, Q := \{x_A, x_B\}, v' := \langle \langle 0.3, 0.5 \rangle, \langle 0.3, 0.5 \rangle, \langle 0, 1 \rangle \rangle, r := \langle 0.3, 0.5 \rangle, v(x_A) := \langle 0.3, 0.5 \rangle, A := \{x_A, x_B\}, \mathbf{exp}(x_A) := \mathbf{true},$
 $\mathbf{in} := \{x_A, x_B\}$
2. $x_i := x_B, A := \{x_A\}, dg := \{x_A, x_B, x_C\}, Q := \{x_A, x_C\}, v' := \langle \langle 0.3, 0.5 \rangle, \langle 0.3, 0.5 \rangle, \langle 0.5, 0.7 \rangle \rangle, r := \langle 0.3, 0.5 \rangle, v(x_B) := \langle 0.3, 0.5 \rangle, A := \{x_A, x_C\}, \mathbf{exp}(x_B) := \mathbf{true},$
 $A := \{x_A, x_C\}, \mathbf{in} := \{x_A, x_B, x_C\}$
3. $x_i := x_C, A := \{x_A\}, Q := \{x_B\}, v' := \langle \langle 0.3, 0.5 \rangle, \langle 0.3, 0.5 \rangle, \langle 0.5, 0.7 \rangle \rangle,$
 $r := \langle 0.5, 0.7 \rangle, v(x_C) := \langle 0.5, 0.7 \rangle, A := \{x_A, x_B\}, \mathbf{exp}(x_C) := \mathbf{true}$
4. $x_i := x_B, A := \{x_A\}, Q := \{x_A, x_C\}, v' := \langle \langle 0.3, 0.5 \rangle, \langle 0.3, 0.5 \rangle, \langle 0.5, 0.7 \rangle \rangle, r := \langle 0.3, 0.5 \rangle$
5. $x_i := x_A, A := \emptyset, Q := \{x_A, x_B\}, v' := \langle \langle 0.3, 0.5 \rangle, \langle 0.3, 0.5 \rangle, \langle 0.5, 0.7 \rangle \rangle, r := \langle 0.3, 0.5 \rangle$
6. **stop. return** $v(x_A, x_B, x_C)|_{x_A} = \langle \langle 0.3, 0.5 \rangle, \langle 0.3, 0.5 \rangle, \langle 0.5, 0.7 \rangle \rangle|_{x_A} = \langle 0.3, 0.5 \rangle$

The computational complexity analysis of $Solve_{WF}$ parallels the one we have made for $Solve_{KK}$. If the height of a bilattice is finite then, like $Solve_{KK}$, each variable x_j will appear in A at most $a_j \cdot (h(\mathcal{L}) + 1)$ times and, thus, the worst-case complexity is $O(\sum_{x_j \in V} (c(f_j) \cdot (a_j \cdot (h(\mathcal{L}) + 1)))$. But now, the cost of $c(f_j)$ is the cost of a recursive call to *Solve*, which is $O(|B_{\mathcal{P}}|^{cah})$. Therefore, $Solve_{WF}$ runs in time $O(|B_{\mathcal{P}}|^2 a^2 h^2 c)$. That is, $Solve_{WF}$ runs in time $O(|\mathcal{P}|^2 h^2 c)$. If the bilattice is fixed, then the height parameter is a constant. Furthermore, often we can assume that c is $O(1)$ and, thus, the worst-case complexity reduces to $O(|\mathcal{P}|^2)$. In the case the height of a bilattice is not finite, the continuity of the functions $f \in \mathcal{F}$ guarantees that each recursive call to *Solve* requires at most ω steps. Thus, we have at most ω^2 steps for $Solve_{WF}$. In case the functions have a finite generation or are bounded, Proposition 10 and Proposition 11 can be applied.

4 Conclusions

We have presented a general top-down algorithm to answer queries for normal logic programs over lattices as well as over bilattices (for which no top-down algorithm was known yet). We believe that its interest relies on the fact that many approaches to paraconsistency and uncertainty of logic programming with or without non-monotonic negation are based on bilattices or lattices, respectively. Therefore, the presented algorithms give us general query-solving procedures for many of them.

References

1. N. D. Belnap. A useful four-valued logic. In G. Epstein and J. M. Dunn, editors, *Modern uses of multiple-valued logic*, pages 5–37. Reidel, Dordrecht, NL, 1977.
2. E. Böhrer, C. Glaer, B. Schwarz, and K. Wagner. Generation problems. In *29th Int. Symp. on Mathematical Foundations of Computer Science (MFCS-04)*, LNCS 3153, pages 392–403. Springer Verlag, 2004.
3. W. Chen and D. S. Warren. Tabled evaluation with delaying for general logic programs. *Journal of the ACM*, 43(1):20–74, 1996.
4. C. V. Damásio, J. Medina, and M. O. Aciego. Sorted multi-adjoint logic programs: Termination results and applications. In *Proc. of the 9th Europ. Conf. on Logics in Art. Intelligence (JELIA-04)*, LNCS 3229, pages 252–265. Springer Verlag, 2004.
5. C. V. Damásio, J. Medina, and M. O. Aciego. A tabulation proof procedure for residuated logic programming. In *Proc. of the 6th Europ. Conf. on Art. Intelligence (ECAI-04)*, 2004.
6. C. V. Damásio and L. M. Pereira. A survey of paraconsistent semantics for logic programs. In D. Gabbay and P. Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pages 241–320. Kluwer, 1998.
7. C. Viegas Damásio and L. M. Pereira. Antitonic logic programs. In *Proc. of the 6th Int. Conf. on logic programming and Nonmonotonic Reasoning (LPNMR-01)*, LNCS 2173. Springer-Verlag, 2001.
8. D. Dubois, J. Lang, and H. Prade. Towards possibilistic logic programming. In *Proc. of the 8th Int. Conf. on Logic Programming (ICLP-91)*, pages 581–595. The MIT Press, 1991.
9. M. C. Fitting. Fixpoint semantics for logic programming - a survey. *Theoretical Computer Science*, 21(3):25–51, 2002.
10. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of the 5th Int. Conf. on Logic Programming*, pages 1070–1080, Cambridge, Massachusetts, 1988. The MIT Press.
11. M. L. Ginsberg. Multi-valued logics: a uniform approach to reasoning in artificial intelligence. *Computational Intelligence*, 4:265–316, 1988.
12. M. Kifer and V.S. Subrahmanian. Theory of generalized annotated logic programming and its applications. *Journal of Logic Programming*, 12:335–367, 1992.
13. Laks V.S. Lakshmanan and N. Shiri. A parametric approach to deductive databases with uncertainty. *IEEE Transactions on Knowledge and Data Engineering*, 13(4):554–570, 2001.
14. Y. Loyer and U. Straccia. The approximate well-founded semantics for logic programs with uncertainty. In *28th Int. Symp. on Mathematical Foundations of Computer Science (MFCS-2003)*, LNCS 2747, pages 541–550, 2003. Springer-Verlag.
15. T. Lukasiewicz. Probabilistic logic programming. In *Proc. of the 13th European Conf. on Artificial Intelligence (ECAI-98)*, pages 388–392, 1998.
16. T. Lukasiewicz. Fixpoint characterizations for many-valued disjunctive logic programs with probabilistic semantics. In *Proc. of the 6th Int. Conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR-01)*, LNAI 2173, pages 336–350. Springer-Verlag, 2001.
17. R. Ng and V.S. Subrahmanian. Stable model semantics for probabilistic deductive databases. In *Proc. of the 6th Int. Sym. on Methodologies for Intelligent Systems (ISMIS-91)*, LNAI 542, pages 163–171. Springer-Verlag, 1991.
18. R. Ng and V.S. Subrahmanian. Probabilistic logic programming. *Information and Computation*, 101(2):150–201, 1993.
19. U. Straccia. Top-down query answering for logic programs over bilattices. Technical Report, ISTI-CNR, Pisa, Italy, 2004.

20. A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, (5):285–309, 1955.
21. A. van Gelder, K. A. Ross, and J. S. Schlimpf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, January 1991.
22. P. Vojtáš. Fuzzy logic programming. *Fuzzy Sets and Systems*, 124:361–370, 2004.

A Logical Treatment of Possibilistic Conditioning

Enrico Marchioni

Departamento de Lógica, Universidad de Salamanca,
Edificio FES - Campus Unamuno, 37007 Salamanca, Spain
`marchioni@usal.es`

Abstract. The notion of conditional possibility derived from marginal possibility measures has received different treatments. However, as shown by Bouchon-Meunier et al., conditional possibility can be introduced as a primitive notion generalizing simple possibility measures. In this paper, following an approach already adopted by the author w.r.t. conditional probability, we build up the fuzzy modal logic $FCII$, relying on the fuzzy logic $LII \frac{1}{2}$, so as to reason about coherent conditional possibilities and necessities. First we apply a modal operator $\diamond$ over conditional events $\varphi|\chi$ to obtain modal formulas of the type $(\varphi|\chi)^\diamond$ whose reading is “ $\varphi|\chi$ is possible”. Then we define the truth-value of the modal formulas as corresponding to a conditional possibility measure. The logic $FCII$ is shown to be strongly complete for finite theories w.r.t. to the class of the introduced conditional possibility Kripke structures. Then, we show that any rational assessment of conditional possibilities is coherent iff a suitable defined theory over $FCII$ is consistent. We also prove compactness for rational coherent assessments of conditional possibilities. Finally, we derive the notion of generalized conditional necessity from that of generalized conditional possibility, and we show how to represent them introducing the logic $GFCII$.

1 Introduction

Possibility Theory was first introduced by Zadeh in [23] so as to deal with that kind of uncertainty induced by vague information based on fuzzy sets. Such a theory provides both a qualitative and quantitative treatment of partial belief [11, 13] relying on a normalized function called possibility distribution π , assigning to each possible situation a value from a totally ordered scale. A new information is then evaluated w.r.t. π by means of the dual pair of *possibility* and *necessity measures* which capture the degrees of compatibility and certainty of such information, respectively. Measures of possibility Π and necessity N can be also introduced independently from a possibility distribution. Indeed, they are fuzzy measures which are weakly compositional w.r.t. union and intersection respectively:

$$\bullet \quad \Pi(A \vee B) = \max\{\Pi(A), \Pi(B)\} \quad \bullet \quad N(A \wedge B) = \min\{N(A), N(B)\}.$$

The notion of conditioning for possibility measures has received different treatments. In general, the conditional possibility $\Pi(A \mid B)$ can be viewed as the solution to the equation

$$\Pi(A \wedge B) = T(x, \Pi(B)),$$

where T is a t-norm, and $\Pi(A \mid B)$ is defined as the greatest solution¹. This would then be equivalent to the following equation [8]:

$$\Pi(A \mid B) = I(\Pi(B), \Pi(A \wedge B)),$$

where I is the residuum of the t-norm.

A classical treatment [12] consists in taking the minimum t-norm, to obtain a qualitative definition. This, however, yields some technical problems in the infinite case, given that Gödel $\Rightarrow_G$ implication is not continuous. This can be avoided by defining a probability-like conditioning by means of the product t-norm. Indeed, as shown in [9], not any t-norm can be used if we want the mapping $\Pi(\cdot|B)$ to be a possibility measure. If we rely on an arbitrary universe, T must be a strict t-norm, i.e. continuous, Archimedean and with no zero-divisors. If the universe is finite, T needs not be Archimedean, and then we can recover the minimum t-norm.

In [1, 2, 3], a different treatment of possibilistic conditioning based on conditional events is presented. Conditional events $E|H$, in effect, can be investigated through their truth-value V , so that $V(E \mid H)$ is true, whenever $|E| = |H| = 1$, false if $|H| = 1$ and $|E| = 0$, and $V(E \mid H)$ takes a third value $t(E \mid H)$, when $|H| = 0$ ². This can be represented in a discrete random variable form by

$$V(E \mid H) = 1 \cdot |E \wedge H| + 0 \cdot |E^c \wedge H| + t(E \mid H) \cdot |H^c|.$$

Now, we can define among such random variables two commutative, associative and monotone operations $\sqcap$, $\sqcup$ from $[0, 1]^2$ to $[0, 1]$ having 1 and 0 as neutral element, respectively, so that, under some restrictions³:

$$t(E \wedge A \mid H) = t(E \mid H) \sqcap t(A \mid E \wedge H), \text{ and } t(E \vee A \mid H) = t(E \mid H) \sqcup t(A \mid H).$$

By interpreting those operations as the minimum and the maximum, the third value $t(\cdot|\cdot)$ is forced to behave like a function which can be defined as a *conditional possibility*⁴.

Definition 1 ([2]). A real function Π defined on $\mathcal{C} = \mathcal{E} \times \mathcal{H}$, where $\mathcal{E}$ is a Boolean algebra, $\mathcal{H}^0$ an additive set, with $\mathcal{H}^0 \subseteq \mathcal{E}$ and $\mathcal{H}^0 = \mathcal{H} \setminus \{\emptyset\}$, is a *conditional possibility* if the following conditions hold:

¹ That is driven by the principle of minimum specificity which states that from a set of possibility distributions the best representative is the least specific one, i.e. the one which assigns the greatest degree of possibility to each situation [13].

² Notice that the indicator function $|\cdot|$ maps unconditional events to their truth-value.

³ See [1, 7] for details on constructing measures from conditional events.

⁴ Notice, for instance, that by interpreting those operations as the product and the sum we obtain the axioms for conditional probability [7].

- (i) $\Pi(E \mid H) = \Pi(E \wedge H \mid H)$, for every $E \in \mathcal{E}$ and $H \in \mathcal{H}$
- (ii) $\Pi(\cdot \mid H)$ is a possibility measure, for any given $H \in \mathcal{H}$
- (iii) $\Pi(E \wedge A \mid H) = \min\{\Pi(E \mid H), \Pi(A \mid E \wedge H)\}$, for every $A \in \mathcal{E}$ and $E, H \in \mathcal{H}$, $E \wedge H \neq 0$.

The dual notion of *conditional necessity* $N(\cdot \mid \cdot)$ is introduced in [2] by means of the standard involutive negation $n_s(x) = 1 - x$. The first two axioms correspond to those for conditional possibility (properly translated for the conditional necessity N), while the third is defined as follows

$$N(E \vee A \mid H) = \max\{N(E \mid H), N(A \mid E^c \wedge H)\}, \text{ for every } A, E \in \mathcal{E} \text{ and } E^c, H \in \mathcal{H}, E^c \wedge H \neq 0.$$

Being possibilistic logic a relevant exception [10], there are not many logical treatments of possibility theory. In this paper, we exploit an idea already carried out in [16, 17, 18, 19, 20, 22]. Similarly to what done w.r.t. conditional probabilities in [22, 18], we take as the truth-value of the modal proposition “ $\varphi \mid \chi$ is possible” the corresponding measure of conditional possibility. Then we obtain modal propositions of the form $(\varphi \mid \chi)^\diamond$ which can be combined by means of the connectives of the fuzzy logic $L\Pi_{\frac{1}{2}}$ which allow to represent the characteristic operations of conditional possibility. We obtain the logic $FC\Pi$, which is shown to be strongly complete for finite theories and which does capture the notion of coherent assessment of conditional possibilities. We will also show how to extend such a logical treatment in order to deal with generalized conditional possibilities and necessity.

In the following section we provide some background notions concerning $L\Pi_{\frac{1}{2}}$, while in the third we introduce both the syntax and the semantics of the logic $FC\Pi$, along with the completeness result. In the next two sections we prove that the coherence (in the sense of [7]) of a rational assessment of conditional possibility is tantamount to the consistency of a suitable defined theory in $FC\Pi$, and moreover that such assessments are compact. Finally, we introduce the notion of generalized conditional possibility and necessity and modify the previous logic so as to (partially) represent such more general measures.

2 Logical Background: the $L\Pi_{\frac{1}{2}}$ Logic

The language of the $L\Pi$ logic is built in the usual way from a countable set of propositional variables, three binary connectives $\rightarrow_L$ (Łukasiewicz implication), $\odot$ (Product conjunction) and $\rightarrow_\Pi$ (Product implication), and the truth constant $\bar{0}$. A truth-evaluation is a mapping e that assigns to every propositional variable a real number from the unit interval $[0, 1]$ and extends to all formulas as follows:

$$\begin{aligned} e(\bar{0}) &= 0, \\ e(\varphi \rightarrow_L \psi) &= \min(1 - e(\varphi) + e(\psi), 1), \\ e(\varphi \odot \psi) &= e(\varphi) \cdot e(\psi), \\ e(\varphi \rightarrow_\Pi \psi) &= \begin{cases} 1, & \text{if } e(\varphi) \leq e(\psi) \\ e(\psi)/e(\varphi), & \text{otherwise} \end{cases}. \end{aligned}$$

The truth constant $\bar{1}$ is defined as $\varphi \rightarrow_L \varphi$. In this way we have $e(\bar{1}) = 1$ for any truth-evaluation e . Moreover, many other connectives can be defined from those introduced above:

$$\begin{array}{ll} \neg_L \varphi \text{ is } \varphi \rightarrow_L \bar{0}, & \neg_{II} \varphi \text{ is } \varphi \rightarrow_{II} \bar{0}, \\ \varphi \wedge \psi \text{ is } \varphi \& (\varphi \rightarrow_L \psi), & \varphi \vee \psi \text{ is } \neg_L (\neg_L \varphi \wedge \neg_L \psi), \\ \varphi \oplus \psi \text{ is } \neg_L \varphi \rightarrow_L \psi, & \varphi \& \psi \text{ is } \neg_L (\neg_L \varphi \oplus \neg_L \psi), \\ \varphi \ominus \psi \text{ is } \varphi \& \neg_L \psi, & \varphi \equiv \psi \text{ is } (\varphi \rightarrow_L \psi) \& (\psi \rightarrow_L \varphi), \\ \Delta \varphi \text{ is } \neg_{II} \neg_L \varphi, & \nabla \varphi \text{ is } \neg_{II} \neg_{II} \varphi, \end{array}$$

with the following interpretations:

$$\begin{array}{ll} e(\neg_L \varphi) = 1 - e(\varphi), & e(\neg_{II} \varphi) = \begin{cases} 1, & \text{if } e(\varphi) = 0 \\ 0, & \text{otherwise} \end{cases}, \\ e(\varphi \wedge \psi) = \min(e(\varphi), e(\psi)), & e(\varphi \vee \psi) = \max(e(\varphi), e(\psi)), \\ e(\varphi \oplus \psi) = \min(1, e(\varphi) + e(\psi)), & e(\varphi \& \psi) = \max(0, e(\varphi) + e(\psi) - 1), \\ e(\varphi \ominus \psi) = \max(0, e(\varphi) - e(\psi)), & e(\varphi \equiv \psi) = 1 - |e(\varphi) - e(\psi)|, \\ e(\Delta \varphi) = \begin{cases} 1, & \text{if } e(\varphi) = 1 \\ 0, & \text{otherwise} \end{cases}, & e(\nabla \varphi) = \begin{cases} 1, & \text{if } e(\varphi) > 0 \\ 0, & \text{otherwise} \end{cases}. \end{array}$$

The logic LII is defined Hilbert-style as the logical system whose axioms and rules are the following⁵:

- (i) Axioms of Łukasiewicz Logic;
- (ii) Axioms of Product Logic;
- (iii) The following additional axioms relating Łukasiewicz and Product logic connectives:
 - ($\neg$) $\neg_{II} \varphi \rightarrow_L \neg_L \varphi$
 - (Δ) $\Delta(\varphi \rightarrow_L \psi) \equiv \Delta(\varphi \rightarrow_{II} \psi)$
 - (LII) $\varphi \odot (\psi \ominus \chi) \equiv (\varphi \odot \psi) \ominus (\varphi \odot \chi)$
- (iv) Deduction rules of LII are modus ponens for $\rightarrow_L$ (modus ponens for $\rightarrow_{II}$ is derivable), and necessitation for Δ : from φ derive $\Delta \varphi$.

The logic $LII_{\frac{1}{2}}$ is the logic obtained from LII by expanding the language with a propositional variable $\frac{\bar{1}}{2}$ and adding the axiom:

$$(LII_{\frac{1}{2}}) \frac{\bar{1}}{2} \equiv \neg_L \frac{\bar{1}}{2}$$

Obviously, a truth-evaluation e for LII is easily extended to an evaluation for $LII_{\frac{1}{2}}$ by further requiring $e(\frac{\bar{1}}{2}) = \frac{1}{2}$.

From the above axiom systems, the notion of proof from a theory (a set of formulas) in both logics, denoted $\vdash_{LII}$ and $\vdash_{LII_{\frac{1}{2}}}$ respectively, is defined as usual. Strong completeness of both logics for finite theories with respect to the given semantics has been proved in [15]. In what follows we will restrict ourselves to the logic $LII_{\frac{1}{2}}$.

⁵ See [15, 4] for details.

As it is also shown in [15], for each rational $r \in [0, 1]$ a formula $\bar{r}$ is definable in $LII_{\frac{1}{2}}$ from the truth constant $\frac{1}{2}$ and the connectives, so that $e(\bar{r}) = r$ for each evaluation e . Therefore, in the $LII_{\frac{1}{2}}$ -language we have a truth constant for each rational in $[0, 1]$, and due to $LII_{\frac{1}{2}}$ -completeness, the following book-keeping axioms for rational truth constants are provable:

$$\begin{aligned} (RLII1) \quad & \neg_L \bar{r} \equiv \overline{1 - r} \\ (RLII2) \quad & \bar{r} \rightarrow_L \bar{s} \equiv \overline{\min(1, 1 - r + s)} \\ (RLII3) \quad & \bar{r} \odot \bar{s} \equiv \overline{r \cdot s} \\ (RLII4) \quad & \bar{r} \rightarrow_{II} \bar{s} \equiv \overline{r \Rightarrow_P s} \end{aligned}$$

where $r \Rightarrow_P s = 1$ if $r \leq s$, $r \Rightarrow_P s = \frac{s}{r}$ otherwise.

3 A Logic of Conditional Possibility

In this section we introduce the modal fuzzy logic $FCII$ in order to reason about conditional possibility in the sense of the above definition. $FCII$ is built up over $LII_{\frac{1}{2}}$ extending its language by including modal formulas which represent the possibility of conditional events. We define the language in two steps. First, we take into account classical Boolean formulas φ , ψ , etc., defined in the usual way from the classical connectives $(\wedge, \neg)$. Then, modal sentences are formulas of the form $(\varphi \mid \chi)^\diamond$, where $\diamond$ is a unary operator taking as arguments conditional events $\varphi \mid \chi$, such that φ and χ are Boolean sentences. Compound modal formulas are built by means of the $LII_{\frac{1}{2}}$ -connectives $(\rightarrow_L, \&, \wedge, \vee, \dots)$ and the truth constants $\bar{r}$, for each rational $r \in [0, 1]$. We also introduce a derived modal operator N such that $(\varphi \mid \chi)^N \equiv \neg_L(\neg\varphi \mid \chi)^\diamond$. Nested modalities are not allowed.

Definition 2. *The axioms of the logic $FCII$ are the following:*

- (i) *Axioms of Classical propositional Logic for Boolean formulas*
- (ii) *Axioms of $LII_{\frac{1}{2}}$ for modal formulas*
- (iii) *Possibilistic modal axioms:*
 - (CII1) $(\varphi \mid \chi)^\diamond \rightarrow (\varphi \wedge \chi \mid \chi)^\diamond$
 - (CII2) $(\varphi \vee \psi \mid \chi)^\diamond \equiv (\varphi \mid \chi)^\diamond \vee (\psi \mid \chi)^\diamond$
 - (CII3) $(\varphi \wedge \psi \mid \chi)^\diamond \equiv (\varphi \mid \chi)^\diamond \wedge (\psi \mid \varphi \wedge \chi)^\diamond$
 - (CII4) $\neg(\perp \mid \chi)^\diamond$

Deduction rules of $FCII$ are those of $LII_{\frac{1}{2}}$ (i.e. modus ponens and necessitation for Δ), plus:

- (iv) *modalization: from φ derive $(\varphi \mid \chi)^\diamond$*
- (v) *substitution of equivalents for the conditioning event: from $\chi \leftrightarrow \chi'$, derive $(\varphi \mid \chi)^\diamond \equiv (\varphi \mid \chi')^\diamond$*
- (vi) *monotonicity: from $\varphi \rightarrow \psi$ derive $(\varphi \mid \chi)^\diamond \rightarrow (\psi \mid \chi)^\diamond$.*

A formula Φ follows from a theory T , i.e. $T \vdash_{FCII} \Phi$, if it is an axiom or it is the result of applying the rules of inference to formulas in T . Notice that the rule of modalization can be applied only to Boolean theorems.

We now define the semantics for *FCII* by introducing *conditional possibility Kripke structures*.

Definition 3. A conditional possibility Kripke model is a structure $K = \langle W, \mathcal{U}, e, \Pi \rangle$, where:

- W is a non-empty set of possible worlds.
- $\mathcal{U}$ is a Boolean algebra of subsets of W .
- $e : V \times W \rightarrow \{0, 1\}$ is a Boolean evaluation of the propositional variables, that is, $e(p, w) \in \{0, 1\}$ for each propositional variable $p \in V$ and each world $w \in W$. Any given truth-evaluation $e(\cdot, w)$ is extended to Boolean propositions as usual. For a Boolean formula φ , we will denote by $[\varphi]_W$ the set of worlds in which φ is true, i.e. $[\varphi]_W = \{w \in W \mid e(\varphi, w) = 1\}$.
- $\Pi : \mathcal{U} \times \mathcal{U}^0 \rightarrow [0, 1]$ is a conditional possibility over $\mathcal{U} \times \mathcal{U}^0$, where $\mathcal{U}^0 = \mathcal{U} \setminus \{\emptyset\}$, and such that $([\varphi]_W, [\chi]_W)$ is Π -measurable for any non-modal φ and χ (with $[\chi]_W \neq \emptyset$).⁶
- $e(\cdot, w)$ is extended to elementary modal formulas by defining

$$e((\varphi \mid \chi)^\circ, w) = \Pi([\varphi]_W \mid [\chi]_W)^7,$$

and to arbitrary modal formulas according to $L\Pi_{\frac{1}{2}}$ -semantics, that is:

$$\begin{aligned} e((\varphi \mid \chi)^N, w) &= 1 - \Pi([\neg\varphi]_W \mid [\chi]_W), \\ e(\bar{r}, w) &= r, \\ e(\Phi \rightarrow_L \Psi, w) &= \min(1 - e(\Phi, w) + e(\Psi, w), 1), \\ e(\Phi \odot \Psi, w) &= e(\Phi, w) \cdot e(\Psi, w), \\ e(\Phi \rightarrow_\Pi \Psi, w) &= \begin{cases} 1, & \text{if } e(\Phi, w) \leq e(\Psi, w) \\ e(\Psi, w)/e(\Phi, w), & \text{otherwise} \end{cases}. \end{aligned}$$

Notice that if Φ is a modal formula the truth-evaluations $e(\Phi, w)$ depend only on the conditional possibility measure Π and not on the particular world w .

The truth-degree of a formula Φ in a conditional possibility Kripke structure $K = \langle W, \mathcal{U}, e, \Pi \rangle$, written $\|\Phi\|^K$, is defined as $\|\Phi\|^K = \inf_{w \in W} e(\Phi, w)$. When $\|\Phi\|^K = 1$ we will say that Φ is valid in K or that K is a model for Φ . We say that K is a model of a set of formulas T if $\|\Phi\|^K = 1$ for all $\Phi \in T$. Now let $\mathcal{M}$ be a class of conditional possibility Kripke structures. Then we define the truth-degree $\|\Phi\|_T^{\mathcal{M}}$ of a formula in a theory T relative to the class $\mathcal{M}$ as $\|\Phi\|_T^{\mathcal{M}} = \inf\{\|\Phi\|^K \mid K \in \mathcal{M}, K \text{ being a model of } T\}$. The notion of logical entailment relative to the class $\mathcal{M}$, written $\models_{\mathcal{M}}$, is then defined as follows:

$$T \models_{\mathcal{M}} \Phi \text{ iff } \|\Phi\|_T^{\mathcal{M}} = 1.$$

If $\mathcal{M}$ denotes the whole class of conditional probability Kripke structures we shall write $T \models_{FCII} \Phi$ and $\|\Phi\|_T^{FCII}$.

⁶ Notice that in our definition the factors of the Cartesian product are the same Boolean algebra. This is clearly a special case of what stated in Definition 1.

⁷ When $[\chi]_W = \emptyset$, we define $e((\varphi \mid \chi)^\circ, w) = 1$.

It is easy to check that axioms *CII1-CII4* are valid formulas in the class of all conditional possibility Kripke structures. Moreover, the inference rule of substitution of equivalents preserves truth in a model, while the modalization rule preserves validity in a model. Therefore we have the following soundness result.

Proposition 1 (Soundness). *The logic FCII is sound with respect to the class of conditional possibility Kripke structures.*

Let $\sim$ be an equivalence relation in the Boolean language $\mathcal{L}$, and $[\varphi]$ the equivalence class of propositions provably equivalent to φ . Obviously, the quotient set $\mathcal{L}/\sim$ forms a Boolean algebra which is isomorphic to a subalgebra $\mathbf{B}(\Omega)$ of the power set of the set Ω of Boolean interpretations $\mathcal{L}$. For each $\varphi \in \mathcal{L}$, we shall identify the equivalence class $[\varphi]$ with the set $\{\omega \in \Omega \mid \omega(\varphi) = 1\} \in \mathbf{B}(\Omega)$ of interpretations that make φ true. We shall denote by $\mathcal{CII}(\mathcal{L})$ the set of conditional possibilities over $\mathcal{L}/\sim \times (\mathcal{L}/\sim \setminus [\perp])$ or equivalently on $\mathbf{B}(\Omega) \times \mathbf{B}(\Omega)^0$.

Each conditional possibility $\Pi \in \mathcal{CII}(\mathcal{L})$ induces a conditional possibility Kripke structure $\langle \Omega, \mathbf{B}(\Omega), e_\Pi, \Pi \rangle$ where $e_\Pi(p, \omega) = \omega(p) \in \{0, 1\}$ for each $\omega \in \Omega$ and each propositional variable p . Denote by $\mathcal{CII\mathcal{S}}$ the class of Kripke structures induced by conditional possibilities $\Pi \in \mathcal{CII}(\mathcal{L})$, i.e.:

$$\mathcal{CII\mathcal{S}} = \{ \langle \Omega, \mathbf{B}(\Omega), e_\Pi, \Pi \rangle \mid \Pi \in \mathcal{CII}(\mathcal{L}) \}.$$

Then, we can safely say that a conditional possibility $\Pi \in \mathcal{CII}(\mathcal{L})$ is a *model* of a modal theory T whenever the induced Kripke structure $\Omega_\Pi = \langle \Omega, \mathbf{B}(\Omega), e_\Pi, \Pi \rangle$ is a model of T . For the sake of simplicity we shall also often write $\Pi(\varphi \mid \chi)$ actually meaning $\Pi([\varphi] \mid [\chi])$. Moreover, we can just take into account only the class of conditional possibility Kripke structures $\mathcal{CII\mathcal{S}}$. Indeed, it is easy to prove the following proposition.

Proposition 2. *For each conditional possibility Kripke structure $K = \langle W, \mathcal{U}, e, \Pi \rangle$ there is a conditional possibility $\Pi^* : \mathbf{B}(\Omega) \times \mathbf{B}(\Omega)^0 \rightarrow [0, 1]$ such that $\|(\varphi \mid \chi)^\diamond\|^K = \Pi^*(\varphi \mid \chi)$ for all $\varphi, \chi \in \mathcal{L}$ such that $[\chi] \neq \emptyset$. Therefore, it also holds that $\|\Phi\|_T = \|\Phi\|_T^{\mathcal{CII\mathcal{S}}}$ for any modal formula Φ and any modal theory T .*

Then, we have the following completeness result for finite modal theories with respect to the given possibilistic semantics.

Theorem 1 (Strong Finite Possibilistic Completeness). *Let T be a finite modal theory over FCII and Φ a modal formula. Then $T \vdash_{FCII} \Phi$ iff $e_\Pi(\Phi) = 1$ for each conditional possibility model Π of T .*

To conclude, notice that not only the axioms of conditional possibility are carved in *FCII* but also those of conditional necessity.

4 Consistency and Coherent Assessments

In many real-life situations assessments of uncertainty are not precisely made over a set of events with a specific algebraic structure. Effectively, the approach

to conditional measures developed in [7, 1, 2, 3] allows partial assessments over an arbitrary family $\mathcal{C}$ of conditional events. However, such assessments are required to be coherent, that is: they must satisfy the axioms of a conditional measure whenever they are extended over the set $\mathcal{E} \times \mathcal{H}^0$ of Boolean supports of the conditional events in $\mathcal{C}$.

Definition 4 ([3]). *Let $\mathcal{C}$ be an arbitrary finite family of conditional events, and Π a real-valued function defined on $\mathcal{C}$. We call Π a coherent conditional possibility if for every $\mathcal{C}' \supseteq \mathcal{C}$, where $\mathcal{C}' = \mathcal{E} \times \mathcal{H}$, with $\mathcal{E}$ a Boolean algebra, $\mathcal{H}$ an additive set, $\mathcal{H} \subseteq \mathcal{E}$ and $\emptyset \notin \mathcal{H}$, there exists a conditional possibility defined on $\mathcal{C}'$ extending Π .*

We can easily show, similarly to the case of conditional probability in [22, 18], that checking coherence of a possibilistic assessment over an arbitrary family of conditional events is tantamount to checking consistency of a suitable defined theory in $FC\Pi$. Indeed, we need theories of the form $\Gamma = \{(\varphi_i \mid \chi_i)^\diamond \equiv \bar{\alpha}_i\}$ in order to have models in which possibilistic assessments are not only 1-valued. But then, we cannot take into account real-valued assessments, since we only have rational truth-constants in our language. Then we obtain that for any rational assessment its coherence is equivalent to the consistency of the respective theory in $FC\Pi$, given that its extension induces a possibilistic Kripke structure which is a model of such a theory.

Theorem 2. *Let $\kappa = \{Poss(\varphi_i \mid \chi_i) = \alpha_i : i = 1, \dots, n\}$ be a rational possibilistic assessment. Then κ is coherent iff the theory $\Gamma_\kappa = \{(\varphi_i \mid \chi_i)^\diamond \equiv \bar{\alpha}_i : i = 1, \dots, n\}$ is consistent in $FC\Pi$, i.e. $\Gamma_\kappa \not\models_{FC\Pi} \bar{0}$.*

5 Compactness of Coherent Assessments

Several results concerning compactness of many fuzzy logics were presented in [6]. The notion of *satisfiability* proposed generalizes the classical one, since it admits various degrees of simultaneous satisfiability.

Definition 5 ([6]). *For a set Γ of formulas in a fuzzy logic and $K \subseteq [0, 1]$, we say that Γ is K -satisfiable if there exists an evaluation e such that $e(\varphi) \in K$ for all $\varphi \in \Gamma$. The set Γ is said to be finitely K -satisfiable if each finite subset of Γ is K -satisfiable. We say that a logic is K -compact if K -satisfiability is equivalent to finite K -satisfiability. A logic satisfies the compactness property if it is K -compact for each closed subset of $[0, 1]$.*

In particular we should mention that fuzzy logics only having continuous truth-functions, like Łukasiewicz Logic, do enjoy the compactness property.

Theorem 3. *Let L be a given fuzzy logic whose connectives only have continuous truth-functions. Then L has the compactness property.*

From this, it clearly follows that compactness also holds for any theory whose connectives only have continuous truth-functions, even if the theory is based on a logic which is not compact. Therefore, if we focus on modal $FC\Pi$ -theories which do not include the product implication $\rightarrow_{\Pi}$ we can easily obtain the following compactness result.

Theorem 4. *Let Γ be a modal theory over $FC\Pi$ whose formulas only involve (at most) truth-constants and the $\&$, $\rightarrow_L$, $\odot$ connectives. Then Γ is consistent iff every finite subtheory of Γ is consistent.*

Given the above theorem, we directly obtain that compactness also holds for coherent possibilistic assessments. This means that when we have a rational possibilistic assessment to an arbitrary family of conditional events, such an assessment is coherent if and only if its restriction to each finite subset of that family also is coherent. Indeed, since any of such coherent restrictions can be translated into a suitable theory which is consistent by Theorem 2, the whole corresponding theory is consistent, and consequently, again by Theorem 2, the corresponding assessment is coherent.

Theorem 5. *Let $\kappa = \{Poss(\varphi_i \mid \chi_i) = \alpha_i\}$ be a rational possibilistic assessment over a family of conditional events $\mathcal{C}$. Let $\kappa_{\downarrow \mathcal{I}}$ be the restriction of κ to each finite $\mathcal{I}$, such that $\mathcal{I} \subset \mathcal{C}$. Then:*

κ is coherent iff $\kappa_{\downarrow \mathcal{I}}$ is coherent for every $\mathcal{I}$.

6 Generalized Conditional Possibility and Necessity

As mentioned in the introduction, by specifying certain operations between conditional events we can obtain different measures. In [2], the authors introduced *generalized possibility measures* by taking as operations the maximum and any t -norm. This clearly generalizes the case of conditional possibilities, where we just have the maximum and the minimum.

Definition 6 ([2]). *A real function Π defined on $\mathcal{C} = \mathcal{E} \times \mathcal{H}$, where $\mathcal{E}$ is a Boolean algebra, $\mathcal{H}^0$ an additive set, with $\mathcal{H}^0 \subseteq \mathcal{E}$ and $\mathcal{H}^0 = \mathcal{H} \setminus \{\emptyset\}$, is a generalized conditional possibility if the following conditions hold:*

- (i) $\Pi(E \mid H) = \Pi(E \wedge H \mid H)$, for every $E \in \mathcal{E}$ and $H \in \mathcal{H}$
- (ii) $\Pi(\cdot \mid H)$ is a possibility measure, for any given $H \in \mathcal{H}$
- (iii) $\Pi(E \wedge A \mid H) = T(\Pi(E \mid H), \Pi(A \mid E \wedge H))$, for every $A \in \mathcal{E}$ and $E, H \in \mathcal{H}$, $E \wedge H \neq 0$ for a t -norm T .

No concept of generalized conditional necessity is derived from that of possibility. However, it can be easily introduced by taking into account De Morgan triples. Remind that a triple (T, S, n) is called a *De Morgan triple* if and only if

n is a strong negation, T a t-norm, and S is the t-conorm n -dual of T . So we define the concept of *generalized conditional necessity*.

Definition 7. A real function N defined on $\mathcal{C} = \mathcal{E} \times \mathcal{H}$, where $\mathcal{E}$ is a Boolean algebra, $\mathcal{H}^0$ an additive set, with $\mathcal{H}^0 \subseteq \mathcal{E}$ and $\mathcal{H}^0 = \mathcal{H} \setminus \{\emptyset\}$, is a generalized conditional necessity if the following conditions hold:

- (i) $N(E \mid H) = N(E \wedge H \mid H)$, for every $E \in \mathcal{E}$ and $H \in \mathcal{H}$
- (ii) $N(\cdot \mid H)$ is a necessity measure, for any given $H \in \mathcal{H}$
- (iii) $N(E \vee A \mid H) = S(N(E \mid H), N(A \mid E^c \wedge H))$, for every $A, E \in \mathcal{E}$ and $E^c, H \in \mathcal{H}$, $E^c \wedge H \neq \emptyset$ for a t-conorm S n -dual of T .

Now, we want to study a logic for reasoning about generalized conditional possibilities and necessities. We need a system which could carry out the presence of t-norms, yielding by means of an involutive negation (not dependent on the t-norm) the dual t-conorms. However, we will not be able to represent any t-norm, since left-continuity is a necessary requirement in order to obtain the residuum [21]. We could rely on the *IMTL* Logic, which extends *MTL* (the logic of left-continuous t-norms [14]) by means of an involutive negation. Still this would not be a good choice since the *IMTL*-negation depends on the t-norm, so we would end up dealing only with those left-continuous t-norms which yield an involutive negation. The best solution seems to be provided by $L\Pi_{\frac{1}{2}}$ itself. Indeed, as shown in [5], any continuous t-norm finitely constructed by means of the ordinal sum technique⁸ can be represented in $L\Pi_{\frac{1}{2}}$ by introducing new connectives which effectively reproduce such a finite construction. We can then define two operations⁹

$$\bullet \quad \varphi \&^R \psi \equiv \bigwedge_{i=1}^m (\varphi \&_{A_i}^{r_i, s_i} \psi) \quad \bullet \quad \varphi \rightarrow^R \psi \equiv \bigvee_{i=1}^m (\varphi \rightarrow_{A_i}^{r_i, s_i} \psi),$$

where A is either Π or L ; $\varphi, \psi, \chi, \gamma$ are formulas; m is a natural number and R a sequence of pairs $([r_i, s_i], A_i)$, where $[r_i, s_i]$ is a closed interval from $[0, 1]$, $r, s \in [0, 1] \cap \mathbb{Q}$, $s_i \leq r_{i+1}$, and $P^{r,s}, \&_A^{r,s}$ and $\rightarrow_A^{r,s}$ are defined as follows:

- $P^{r,s}(\varphi, \psi, \chi, \gamma) \equiv [\chi \wedge \Delta((\bar{r} \rightarrow_{\Pi} \varphi \wedge \psi) \wedge (\varphi \vee \psi \rightarrow_{\Pi} \bar{s}))] \vee [\gamma \wedge \neg_{\Pi} \Delta((\bar{r} \rightarrow_{\Pi} \varphi \wedge \psi) \wedge (\varphi \vee \psi \rightarrow_{\Pi} \bar{s}))];$
- $\varphi \&_A^{r,s} \psi \equiv P^{r,s}([((\bar{s} \ominus \bar{r}) \rightarrow_{\Pi} (\varphi \ominus \bar{r})) \&_A((\bar{s} \ominus \bar{r}) \rightarrow_{\Pi} (\psi \ominus \bar{r})) \&_{\Pi}(\bar{s} \ominus \bar{r}) \oplus \bar{r}, \varphi \wedge \psi, \varphi, \psi);$
- $\varphi \rightarrow_A^{r,s} \psi \equiv P^{r,s}([((\bar{s} \ominus \bar{r}) \rightarrow_{\Pi} (\varphi \ominus \bar{r})) \rightarrow_A((\bar{s} \ominus \bar{r}) \rightarrow_{\Pi} (\psi \ominus \bar{r})) \&_{\Pi}(\bar{s} \ominus \bar{r}) \oplus \bar{r}, \psi, \varphi, \psi) \vee \Delta(\varphi \rightarrow_{\Pi} \psi).$

The truth-functions obtained for $\&^R$ and $\rightarrow^R$ effectively are finitely constructed continuous t-norms and their residua, respectively.

Now, even if we must take into account only a restricted class of t-norms, we can safely rely on $L\Pi_{\frac{1}{2}}$, since we can also exploit Łukasiewicz standard

⁸ Consult [21] for the ordinal sum construction of t-norms.

⁹ See [5] for details.

negation so as to obtain the n_s -dual t-conorms. Then, we introduce the logic *GFCII* - Generalized Fuzzy Conditional Possibility - by employing the very *FCII*-language and *FCII*-axioms, with the exception of *CII3*, which is now substituted by the axiom *GCII1*:

$$(\varphi \wedge \psi \mid \chi)^\diamond \equiv (\varphi \mid \chi)^\diamond \&^R(\psi \mid \varphi \wedge \chi)^\diamond.$$

Clearly $(\varphi \mid \chi)^N \equiv \neg_L(\neg\varphi \mid \chi)^\diamond$ as before.

The semantics for *GFCII* is given by *generalized conditional possibility Kripke structures* $K = \langle W, \mathcal{U}, e, \Pi \rangle$, where W , $\mathcal{U}$ and e are defined as above, and Π is a generalized conditional possibility restricted to the class of finitely constructed continuous t-norms so that:

$$e(\Phi \&^R \Psi) = T^R(e(\Phi), e(\Psi)),$$

where, T^R is a finitely constructed continuous t-norm. Notice that by defining $\Phi \oplus^R \Psi \equiv \neg_L(\neg_L \Phi \&^R \neg_L \Psi)$, then $e(\Phi \oplus^R \Psi) = S^R(e(\Phi), e(\Psi))$, where S^R is the continuous t-conorm n_s -dual of T^R obtained by means of the standard involutive negation $n_s(x) = 1 - x$. Then, it is easy to prove the following results.

Proposition 3 (Soundness). *The logic GFCII is sound with respect to the class of generalized conditional possibility Kripke structures.*

Theorem 6 (Strong Finite Possibilistic Completeness). *Let T be a finite modal theory over GFCII and Φ a modal formula. Then $T \vdash_{GFCII} \Phi$ iff $e_\Pi(\Phi) = 1$ for each generalized conditional possibility model Π of T .*

7 Conclusions and Future Work

In this paper we have studied the logics *FCII* and *GFCII* to reason about conditional possibility and generalized conditional possibility, and we have shown that such logics are strongly complete for finite theories. In particular, we have introduced the notion of generalized conditional necessity as a dual notion w.r.t. generalized conditional possibility.

Both for *FCII* and *GFCII* we have relied on $L\Pi^{\frac{1}{2}}$ for the sake of simplicity, but we could have chosen a weaker base logic for *FCII*. Indeed, also Łukasiewicz Logic would work, since we need nothing more than the minimum and maximum operations along with an involutive negation and a t-norm. That would be interesting if we aim at studying complexity, since, considering that L complexity is lower than $L\Pi^{\frac{1}{2}}$ complexity, we argue that by choosing L as a basis we could obtain a lower complexity for *FCII* as well. This would reflect the computational simplicity of Possibility Theory. Clearly, no similar reasoning holds for *GFCII* since we cannot give up $L\Pi^{\frac{1}{2}}$ without losing the possibility of defining finitely constructed continuous t-norms. The study of complexity of such logics would then be an interesting path of research to follow.

To conclude, we mention as another worthwhile line of investigation the study of a logical treatment of both simple and conditional possibility measures whose arguments are not Boolean events but fuzzy events.

Acknowledgements. Marchioni recognizes support of the grant No. AP2002-1571 of the Ministerio de Educación y Ciencia of Spain.

References

1. BOUCHON-MEUNIER B., COLETTI G. AND MARSALA C. Possibilistic conditional events. In *Proc. of IPMU 2000*, Madrid, Spain, 1561–1566, 2000.
2. BOUCHON-MEUNIER B., COLETTI G. AND MARSALA C. Independence and possibilistic conditioning. *Annals of Mathematics and Artificial Intelligence*, 35, 107–123, 2002.
3. BOUCHON-MEUNIER B., COLETTI G. AND MARSALA C. Conditional possibility and necessity. In *Technologies for Constructing Intelligent Systems*, Physica Verlag, Heidelberg, 59–71, 2002.
4. CINTULA P. The LII and $LII\frac{1}{2}$ propositional and predicate logics. *Fuzzy Sets and Systems* 124, 21–34, 2001.
5. CINTULA P. Advances in the LII and $LII\frac{1}{2}$ logics. *Archive for Mathematical Logic* 42, 449–468, 2003.
6. CINTULA P. AND NAVARA M. Compactness of fuzzy logics. *Fuzzy Sets and Systems* 143, 59–73, 2004.
7. COLETTI, G. AND SCOZZAFAVA R. *Probabilistic Logic in a Coherent Setting*. Kluwer Academic Publisher, Dordrecht, The Netherlands, 2002.
8. DE BAETS B. AND KERRE E. A primer on solving fuzzy relational equations on the unit interval. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2, 205–225, 1994.
9. DE BAETS B., TSIPORKOVA E. AND MESIAR R. Conditioning in possibility theory with strict order norms. *Fuzzy Sets and Systems* 106, 121–129, 1999.
10. DUBOIS D., LANG J. AND PRADE H. Possibilistic Logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, D. M.Gabbay et al. (eds), Vol. 3, 439–513, Oxford University Press, 1994.
11. DUBOIS D. AND PRADE H. *Possibility Theory*. Plenum Press, New York, 1988.
12. DUBOIS D. AND PRADE H. The logical view of conditioning and its application to possibility and evidence theories. *Int. J. of Approximate Reasoning*, 4, 23–46, 1990.
13. DUBOIS D. AND PRADE H. Possibility theory: Qualitative and quantitative aspects. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, D. M.Gabbay and Ph. Smets (eds), Vol. 1, 169–226, Kluwer Academic Publisher, Dordrecht, The Netherlands, 1998.
14. ESTEVA F. AND GODO L. Monoidal t-norm based logic: towards a logic for left-continuous t-norms. *Fuzzy Sets and Systems*, 124, 271–288, 2001.
15. ESTEVA F., GODO L. AND MONTAGNA F. The LII and $LII\frac{1}{2}$ logics: two complete fuzzy logics joining Łukasiewicz and product logic. *Archive for Mathematical Logic*, 40, pp. 39–67, 2001.
16. FLAMINIO T. AND MONTAGNA F. A logical and algebraic treatment of conditional probability. *Archive for Mathematical logic*, 44, No. 2, 245–262, 2005.

17. GODO L., ESTEVA F. AND HÁJEK P. Reasoning about probability using fuzzy logic. *Neural Network World*, 10, No. 5, 811–824, 2000.
18. GODO L. AND MARCHIONI E. Reasoning about coherent conditional probability in the logic $FCP(LII)$. In *Proc. of the Workshop on Conditional, Information and Inference*, Ulm, Germany, 2004.
19. HÁJEK P. *Metamathematics of fuzzy logic*, Kluwer 1998.
20. HÁJEK P., GODO L. AND ESTEVA F. Fuzzy Logic and Probability. In *Proc. of UAI'95*, 237–244, 1995.
21. KLEMENT E. P., MESIAR R. AND PAP E. *Triangular Norms*. Kluwer Academic Publisher, Dordrecht, The Netherlands, 2000.
22. MARCHIONI E. AND GODO L. A logic for reasoning about coherent conditional probability: A modal fuzzy logic approach. In *Logics in Artificial Intelligence*, LNAI 3229, Springer-Verlag, Berlin Heidelberg, 213–225, 2004.
23. ZADEH L. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1, 3–28, 1978.

A Zero-Layer Based Fuzzy Probabilistic Logic for Conditional Probability

Tommaso Flaminio

Department of Mathematics and Computer Science,
University of Siena, Pian dei Mantellini 44, 53100 Siena, Italy
flaminio@unisi.it

Abstract. This paper is devoted to a logical and algebraic treatment of conditional probability. Unlike the other approaches to this problem (cf [7], [10]) we base our work on the notion of *zero-layer* (cf [4]). Thus we define the fuzzy modal logic $FP_k(RL_\Delta)$ with k modalities for *non-conditional probably*, built up over the many-valued logic RL_Δ (obtained by adding to the Rational Łukasiewicz logic the Baaz connective Δ). The main result of this paper tells us that it is possible to characterize the coherence of an assessment of conditional probability by the consistence of a suitable theory over $FP_k(RL_\Delta)$.

1 Introduction

The two most relevant mathematical theories allowing a treatment of uncertainty and vagueness are probability theory and fuzzy set theory respectively. Their logical counterparts are probabilistic logic and fuzzy logic.

Such logical formalisms are related each other from the need of intermediate values in their semantics. Hájek's book [11] contains a treatment of probabilistic logic inside a modal fuzzy logic. The idea is that the probability of an event φ may be regarded as the truth degree of the modal formula φ , formula denoted by $P\varphi$. The necessity of intermediate values suggests to use a many-valued logic in order to describe such modal logic of probability. The most appropriate many-valued logic seems to be Łukasiewicz Logic (L in symbols) (see [11]). In fact the Łukasiewicz Logic has a connective ($\oplus$) whose natural interpretation is the Łukasiewicz t-conorm which behaves like a truncated sum:

$$x \oplus y = \min\{1, x + y\},$$

so, if φ_1 and φ_2 are two disjoint events, then we can express the additivity of the modal operator P by the formula:

$$P(\varphi_1 \vee \varphi_2) \leftrightarrow P(\varphi_1) \oplus P(\varphi_2).$$

Moreover it is useful to have some constants corresponding to rational truth-values. Thus Hájek introduced in [11] a probabilistic many-valued logic with

rational truth values denoted by $FP(RPL)$, where FP stands for Fuzzy Probabilistic and RPL (Rational Pavelka Logic) is obtained by adding to Łukasiewicz Logic constants $\bar{r}$ for every $r \in [0, 1]_{\mathbb{Q}}$ and axioms for the arithmetic in the rational field.

A crucial concept in probability theory is that of conditional probability. In [5], the authors introduce a probabilistic logic based on $LII_{\frac{1}{2}}$, logic obtained joining both Łukasiewicz and Product Logic (II) and by the adding of a propositional constant $\frac{1}{2}$ together with the axiom $\frac{1}{2} \leftrightarrow \neg \frac{1}{2}$ (see [6], [1] and [2] for more details). The main idea is the following: product conjunction ($\&_{II}$) has a residuation ($\Rightarrow_{II}$) whose truth function behaves as a truncated division:

$$x \Rightarrow_{II} y = \begin{cases} \frac{y}{x} & \text{if } y < x, \\ 1 & \text{otherwise.} \end{cases}$$

Thus, if the truth value of $P(\psi)$ is non-zero we can express the conditional probability of φ given ψ by $P(\varphi | \psi) = P(\psi) \Rightarrow_{II} P(\varphi \wedge \psi)$. Thus the definition works only in the case where the truth value of $P(\psi)$ is not zero.

In [7] and [10] two approaches are presented intended to overcome the difficulty of defining $P(\varphi | \psi)$ only in the case where $P(\psi) \neq 0$. In particular in [7] we presented a non-standard approach to probability introducing the logic $FP(SLII)$ where $SLII$ is obtained by adding to $LII_{\frac{1}{2}}$ a unary connective S and axioms reflecting the standard part operator in order to relate non-standard probability to standard probability. In [10] a conditional probability $\mu(\cdot | \cdot)$ is introduced as a primitive notion. Namely for each couple of classical propositions φ and ψ the probability of a conditional event $\varphi | \psi$ is taken as the truth value of the modal formula $P(\varphi | \psi)$. Based on this idea Godo and Marchioni define the fuzzy modal logic $FCP(LII)$ (where FCP stands for fuzzy conditional probability) built up over the logic $LII_{\frac{1}{2}}$.

An important problem in probability theory is the coherence of a probabilistic assessment. We say that an assessment $\kappa : P(\varphi_i) = \alpha_i, i = 1, \dots, n$ is coherent iff there exists a probability measure Θ on the space of events φ_i extending the assessment, that is, for every $i = 1, \dots, n$ one has $P(\varphi_i) = \Theta(\varphi_i)$. Similarly an assessment of conditional probability $\chi : P(\varphi_i | \psi_i) = \beta_i, i = 1, \dots, n$ is coherent iff there exists a conditional probability ρ (see [4]) such that for any $i = 1, \dots, n$ one has $P(\varphi_i | \psi_i) = \rho(\varphi_i | \psi_i)$.

Both [7] and [10] are devoted to the treatment of the coherence of assessments of conditional probability and the main results of these papers tells us that there exists a standard way of associating to every assessment χ of conditional probability a many-valued theory T_{χ} over the related logic ($FP(SLII)$ and $FCP(LII)$ respectively) which is logically consistent iff χ is coherent. Unfortunately in these papers a good algorithm for checking the coherence of an assessment of conditional probability is not presented. In [12] Hájek and Tulipani studied the complexity of the fuzzy probabilistic logics $FP(L)$ and $FP(LII_{\frac{1}{2}})$, that is, fuzzy probabilistic logics built up over the Łukasiewicz logic and $LII_{\frac{1}{2}}$ logic respectively. In particular they proved the following:

Theorem 1. *Let $\mathcal{A} = \langle \mathcal{A}, \mathcal{B} \rangle$ be a $\mathcal{C}$ -bimodule and $\mathcal{C}$ be a Frobenius algebra. Then $\mathcal{A}$ is a Frobenius algebra if and only if $\mathcal{A}$ is a Frobenius algebra.*

Even if Theorem 1 provide an upper (and not lower) bound to the complexity for the satisfaction problem in $FP(L\Pi\frac{1}{2})$, we believe that a good strategy for determining a nice algorithm testing the coherence of an assessment of conditional probability, could be to find a fuzzy probabilistic logic allowing a treatment of an assessment of conditional probability and such that the upper bound to the complexity for the satisfaction problem for its modal formulas be lower than PSPACE.

In the same paper (cf [12]) Hájek and Tulipani have shown that, if we build a fuzzy probabilistic logic on the Łukasiewicz logic, then its complexity diminishes. In particular they proved the following:

[illegible]

So, in this paper, we present an approach where the ground many-valued logic which the fuzzy probabilistic logic is based on is obtained by adding to the Łukasiewicz Logic the following connectives:

δ_n (each for every $n \in \mathbb{N}$) whose truth function behaves as a division by n : for each $x \in [0, 1]$, $\delta_n(x) = x/n$,

Δ whose truth function behaves as the following $\{0, 1\}$ -projection: for each $x \in [0, 1]$ $\Delta(x) = 1$ if $x = 1$ and $\Delta(x) = 0$ if $x < 1$.

In other words the logic so obtained is the Rational Lukasiewicz Logic (RL for short) (see [9]) extended with the Baaz operator Δ . In [9] it is shown that RL is a proper extension of RPL . Thus, also in RL_{Δ} , for every $r \in [0, 1] \cap \mathbb{Q}$, rational constants $\bar{r}$ are definable. More formally we have the following:

Definition 1. Let (R, Δ) be a $\mathbb{K}$ -algebra with a comultiplication Δ . For $n \in \mathbb{N}$, Δ^n is the n -th iteration of Δ .

- $$\begin{array}{ll} (D_1) & n.\delta_n\varphi \leftrightarrow \varphi, \\ (D_2) & \neg\delta_n\varphi \oplus (n-1).(\neg(\delta_n\varphi)), \\ (\Delta_1) & \Delta(\varphi \rightarrow \psi) \rightarrow (\Delta\varphi \rightarrow \Delta\psi), \\ (\Delta_2) & \Delta\varphi \vee \neg\Delta\varphi, \\ (\Delta_3) & \Delta\varphi \rightarrow \varphi, \\ (\Delta_4) & \Delta\varphi \rightarrow \Delta(\Delta\varphi), \\ (\Delta_5) & \Delta(\varphi \vee \psi) \leftrightarrow (\Delta\varphi \vee \Delta\psi) \end{array}$$

$$(\varphi \rightarrow \psi) \& (\psi \rightarrow \varphi), \quad \varphi \oplus \psi, \quad \neg \varphi \rightarrow \psi, \quad \varphi \leftrightarrow \psi, \\ (\varphi \rightarrow \psi) \& (\psi \rightarrow \varphi), \quad n \in \mathbb{N}, \quad n, \varphi, \quad \varphi \oplus \dots \oplus \varphi,$$
$$\begin{array}{l} R_{\Delta} \text{ modus ponens } ((\Delta) \vdash \varphi, \varphi \rightarrow \psi \vdash \psi) \\ \Delta\text{-necessitation } ((\Delta) \vdash \varphi \vdash \Delta(\varphi)) \\ \Gamma \vdash_{R_{\Delta}} \varphi, \Gamma \vdash_{R_{\Delta}} \varphi \rightarrow \psi \vdash_{R_{\Delta}} \psi \end{array}$$

In the following of this paper we use the following further abbreviations:

$\nabla\varphi$ stands for $\neg\Delta\neg\varphi$ and its natural interpretation is given by: for each $x \in [0, 1]$, $\nabla(x) = 0$ if $x = 0$ and $\nabla(x) = 1$ if $x > 0$.

For every $n \leq m$, $\overline{n/m}$ stands for $n.\delta_m(\overline{1})$. The natural interpretation of n/m is given by the rational value $\frac{n}{m}$.

The algebraic counterpart of RL is a variety of algebras called in [8] DMV -algebras (divisible MV -algebras). Now we introduce in each DMV -algebra the operator Δ . In this way we define the class of DMV_Δ -algebras. Moreover we require that in each DMV_Δ -algebra, properties capturing the behavior of the Baaz operator hold.

Definition 2. A DMV_Δ -algebra $\mathcal{A} = \langle A, \oplus, \neg, \{\delta_n\}_{n \in \mathbb{N}}, \Delta, 0, 1 \rangle$ is a DMV -algebra $\mathcal{A}^- = \langle A, \oplus, \neg, 0, 1 \rangle$ such that $x, y \in A$, $n \in \mathbb{N}$

- (D_1) $n.\delta_n x = x$,
- (D_2) $\neg(\delta_n x) \oplus (n - 1).(\neg(\delta_n x)) = 1$,
- (Δ_1) $\Delta(x \Rightarrow y) \leq (\Delta(x) \Rightarrow \Delta(y))$,
- (Δ_2) $\Delta(x) \vee \neg\Delta(x) = 1$,
- (Δ_3) $\Delta(x) \leq x$,
- (Δ_4) $\Delta(x) \leq \Delta(\Delta(x))$,
- (Δ_5) $\Delta(x \vee y) = (\Delta(x) \vee \Delta(y))$

$$x \Rightarrow y, \neg x \oplus x \vee y, ((x \Rightarrow y) \Rightarrow y)$$

Definition 3. A DMV_Δ -algebra $\mathcal{A}$ is a DMV_Δ -algebra $\mathcal{A}$ such that R_Δ is a DMV_Δ -algebra $\mathcal{A}$ such that $e(\overline{0}) = 0$ and $e(\varphi) = 0$ for $\varphi \in \mathcal{A}$, $n \in \mathbb{N}$

$$\begin{aligned} e(\varphi \&\psi) &= \max\{0, e(\varphi) + e(\psi) - 1\}, \\ e(\varphi \rightarrow \psi) &= \min\{1, 1 - e(\varphi) + e(\psi)\}, \\ e(\Delta\varphi) &= \begin{cases} 1 & \text{if } e(\varphi) = 1 \\ 0 & \text{otherwise} \end{cases} \\ e(\delta_n\varphi) &= e(\varphi)/n \end{aligned}$$

A DMV_Δ -algebra $\mathcal{A}$ is a DMV_Δ -algebra $\mathcal{A}$ such that R_Δ is a DMV_Δ -algebra $\mathcal{A}$ such that $e(\overline{0}) = 0$ and $e(\varphi) = 0$ for $\varphi \in \mathcal{A}$, $n \in \mathbb{N}$. It can be shown that a standard completeness theorem holds for RL_Δ . Indeed it is shown in [9] that RL is standard complete. The introduction of the operator Δ in RL does not give rise to particular problems and thus a standard completeness theorem can be proved also for RL_Δ . Thus we have:

Theorem 3. () $\Gamma \vdash_{R\Delta} \varphi \rightarrow R\Delta \vdash \varphi$ $\Gamma \not\vdash_{R\Delta} \varphi$ DMV_{Δ} $\mathcal{A} = \langle [0, 1], \oplus, \neg, \{\delta_n\}_{n \in \mathbb{N}}, \Delta, 0, 1 \rangle$ $e(\gamma) = 1$ $\gamma \in \Gamma$ $e(\varphi) \neq 1$

Unfortunately RL_{Δ} has not the same expressive power of $L\Pi^1_{\frac{1}{2}}$, and then a fuzzy probabilistic logic over RL_{Δ} built in the usual way (that is obtained by adding only one modality P for ()) does not allow a treatment of conditional probability. In order to fill the gap brought about by the low expressive power of RL_{Δ} , we build a fuzzy probabilistic logic exploiting the notion of (), which was introduced in [4]. First of all a basic result:

Theorem 4. () $\mathcal{C} = \{\varphi_i \mid \psi_i\}_{i=1,\dots,n}$ $\mathcal{A}_1$ A_r $\mathcal{B}$ $\chi : P(\varphi_i \mid \psi_i) = \alpha_i, i = 1, \dots, n$ $\mathcal{C}$

- (i) χ
- (ii) k $\{P_1, \dots, P_k\}$ P_{α} $\mathcal{A}_{\alpha} \subseteq \mathcal{A}_1$ $\mathcal{A}_{\alpha'} \subset \mathcal{A}_{\alpha''}$ $\alpha' > \alpha''$ $P_{\alpha''}(A_r) = 0, A_r \in \mathcal{A}_{\alpha'}$
- (iii) $\gamma \in \mathcal{B}, \gamma \neq \emptyset$ P_{α} $\sum_{A_r \subseteq \gamma} P_{\alpha}(A_r) > 0$;
- (iv) $\varphi_i \mid \psi_i \in \mathcal{C}$ P_{β}
$$\sum_{A_r \subseteq \psi_i} P_{\beta}(A_r) > 0 \implies \alpha_i = \frac{\sum_{A_r \subseteq \varphi_i \wedge \psi_i} P_{\beta}(A_r)}{\sum_{A_r \subseteq \psi_i} P_{\beta}(A_r)}.$$

The concept of zero-layer arises from the non trivial structure of coherent conditional probability brought out by Theorem 4.

Definition 4. () $\mathcal{C} = \{\varphi_1 \mid \psi_1, \dots, \varphi_n \mid \psi_n\}$ P $\mathcal{C}$ $\mathcal{P} = \{P_1, \dots, P_k\}$ $\varphi \neq \emptyset$ $\varphi_1, \psi_1, \dots, \varphi_n, \psi_n$ zero-layer φ $\mathcal{P}$ β $\sum_{A_r \subseteq \varphi} P_{\beta}(A_r) > 0$

Thus, in order to have a treatment of conditional probability, we use the trick of introducing in RL_{Δ} k modalities $P_1, \dots, P_k$ (for a given $k \in \mathbb{N}$) and axioms assuring each P_{α} to be interpreted as a probability. In this way we define the fuzzy modal logic $FP_k(RL_{\Delta})$.

Definition 5. () $FP_k(R\Delta)$ $p_1, p_2, \dots$ $R\Delta$ $\&, \rightarrow, \delta_n$ $n \in \mathbb{N}$ Δ $\{P_1, \dots, P_k\}$ probably $\bar{r}$ $r \in [0, 1] \cap \mathbb{Q}$

$$\begin{aligned}
& \text{If } \Phi \text{ is a propositional constant } \bar{r}, \text{ then } \|\bar{r}\|_{\mathcal{M},w} = r, \\
& \|P_\alpha \varphi\|_{\mathcal{M},w} = \mu_\alpha(W_\varphi), \\
& \|\Phi \& \Psi\|_{\mathcal{M},w} = \max\{0, \|\Phi\|_{\mathcal{M},w} + \|\Psi\|_{\mathcal{M},w} - 1\}, \\
& \|\Phi \rightarrow \Psi\|_{\mathcal{M},w} = \min\{1, 1 - \|\Phi\|_{\mathcal{M},w} + \|\Psi\|_{\mathcal{M},w}\}, \\
& \|\Delta \Phi\|_{\mathcal{M},w} = \begin{cases} 1 & \text{if } \|\Phi\|_{\mathcal{M},w} = 1 \\ 0 & \text{otherwise.} \end{cases} \\
& \|\delta_n \Phi\|_{\mathcal{M},w} = \|\Phi\|_{\mathcal{M},w}/n.
\end{aligned}$$

If $\varphi \in BF$, then $\|\varphi\|_{\mathcal{M},w} = e(w, \varphi)$.

If Φ is a modal formula, then $\|\Phi\|_{\mathcal{M},w}$ is independent of w , so we will omit the subscript w . We say that $\mathcal{M}$ is a *model* of a formula Φ (and we write $\mathcal{M} \models \Phi$) if $\|\Phi\|_{\mathcal{M}} = 1$. If Γ is a set of modal formulas we say that $\mathcal{M}$ is a model of Γ ($\mathcal{M} \models \Gamma$) if $\mathcal{M} \models \Psi$ for every $\Psi \in \Gamma$. If Γ is a set of formulas and Φ is a modal formula, we write $\Gamma \models_{FP_k} \Phi$ to mean that for every Kripke model $\mathcal{M}$ for $FP_k(RL_\Delta)$ if $\mathcal{M} \models \Psi$ for every $\Psi \in \Gamma$, then $\mathcal{M} \models \Phi$.

Theorem 5. $FP_k(RL_\Delta) \vdash \Phi$ iff $\Gamma \models_{FP_k} \Phi$, where $\Gamma = \{ \Psi \mid \Gamma \vdash_{FP_k} \Psi \}$.

The right-to-left direction is straightforward. For the other direction, the argument is analogous to the argument used for $FP(RPL)$ in [11]. We sketch the part where the proofs are quite parallel and we add more details where the two proofs diverge.

For every modal formula Ξ , let Ξ^* be obtained from Ξ by replacing every occurrence of an atomic subformula of the form $P_\alpha \varphi$ by a new propositional variable p_φ^α . We write $p_\varphi^\alpha = (P_\alpha \varphi)^*$ and define $\bar{r}^* = \bar{r}$, $(\Psi \bullet \Lambda)^* = \Psi^* \bullet \Lambda^*$ for any $\bullet \in \{\&, \rightarrow\}$ and $(\circ \Psi)^* = \circ(\Psi^*)$ for any $\circ \in \{\delta_n, \Delta\}$. Let Γ^* be the theory (over RL_Δ) whose axioms are:

- Atoms p_φ^α for φ being a Boolean tautology and for $\alpha \in \{1, \dots, k\}$,
- Formulas $(P_\alpha 1)^*, \dots, (P_\alpha 3)^*$ for $(P_\alpha i)$ being an axioms of $FP_k(RL_\Delta)$ of the form (P) ,
- Formulas Ψ^* for $\Psi \in \Gamma$.

Then, along the lines of [11], it is easy to prove that for every modal formula Ψ ,

$$\Gamma \vdash_{FP_k} \Psi \text{ iff } \Gamma^* \vdash_{RL_\Delta} \Psi^*.$$

Hence if $\Gamma \not\vdash_{FP_k} \Phi$, then $\Gamma^* \not\vdash_{RL_\Delta} \Phi^*$ and from the completeness of RL_Δ w.r.t. the class of DMV_Δ -algebras it follows that there are an DMV_Δ -algebra $\mathcal{A}$ and an evaluation v in $\mathcal{A}$ such that $v(\Psi^*) = 1$ for each $\Psi^* \in \Gamma^*$ and $v(\Phi^*) \neq 1$. By the standard completeness of RL_Δ (Theorem 3) we may assume $\mathcal{A}$ being a standard DMV_Δ -algebra having truth values in the real unit interval $[0, 1]$.

Now let W be the set of all evaluation of Boolean formulas of $FP_k(RL_\Delta)$. Define, for every Boolean formula φ and for every $w \in W$, $e(w, \varphi) = w(\varphi)$.

Moreover, for every Boolean formula φ and for every $\alpha \in \{1, \dots, k\}$, we define $\mu_\alpha(W_\varphi) = v(p_\varphi^\alpha)$. Then $\mathcal{M} = \langle W, e, \mu_1, \dots, \mu_k \rangle$ is a probabilistic Kripke model for $FP_k(RL_\Delta)$. Now we prove by induction on Ψ that for every modal formula Ψ one has $\|\Psi\|_{\mathcal{M}} = v(\Psi^*)$:

- (i) Let Ψ be an atomic formula. Then, if $\Psi = P_\alpha \varphi$ for some $\varphi \in BF$, then $\|P_\alpha \varphi\|_{\mathcal{M}} = \mu_\alpha(W_\varphi) = v(p_\varphi^\alpha)$. Otherwise, if $\Psi = \bar{r}$ for some $r \in [0, 1] \cap \mathbb{Q}$, then $\bar{r}^* = \bar{r}$ and thus $\|\bar{r}\|_{\mathcal{M}} = r = v(\bar{r}^*)$.
- (ii) Let Ψ be a non atomic formula. Then the claim easily follows from Definition 6. We just quote an example: if $\Psi = \Sigma \rightarrow \Lambda$, then $\|\Sigma \rightarrow \Lambda\|_{\mathcal{M}} = \|\Sigma\|_{\mathcal{M}} \Rightarrow \|\Lambda\|_{\mathcal{M}}$. Now, by induction, we have $\|\Sigma\|_{\mathcal{M}} = v(\Sigma^*)$ and $\|\Lambda\|_{\mathcal{M}} = v(\Lambda^*)$, thus $\|\Sigma \rightarrow \Lambda\|_{\mathcal{M}} = v(\Sigma^*) \rightarrow v(\Lambda^*)$ that is $\|\Sigma \rightarrow \Lambda\|_{\mathcal{M}} = v(\Sigma^* \rightarrow \Lambda^*) = v((\Sigma \rightarrow \Lambda)^*)$.

Hence for all $\Psi \in \Gamma$ we have $\|\Psi\|_{\mathcal{M}} = 1$ and $\|\Phi\|_{\mathcal{M}} \neq 1$. This completes the proof.

3 Application to the Coherence Problem

What we are going to prove in this section is a logical characterization of probabilistic coherence by using the logic $FP_k(RL_\Delta)$. We start with a basic result:

Lemma 1. *Let $\{\varphi_1, \dots, \varphi_S\}$ be a finite set of formulas, $r \leq k$, $\Phi_r = \bigwedge_{j=1}^S \bigvee_{\alpha=1}^r \nabla P_\alpha \varphi_j$. Then, $FP_k(RL_\Delta) \vdash \Phi_r$ iff $\mu_\alpha(\varphi_j) > 0$ for every $j \in \{1, \dots, S\}$ and $\alpha \leq r$.*

$$\Phi_r : \bigwedge_{j=1}^S \bigvee_{\alpha=1}^r \nabla P_\alpha \varphi_j.$$

Proof. We know by Theorem 5 that $FP_k(RL_\Delta) \vdash \Phi_r$ iff for every Kripke model $\mathcal{M}$ for $FP_k(RL_\Delta)$, $\mathcal{M} \models \Phi_r$, that is $\|\Phi_r\|_{\mathcal{M}} = 1$. By Definition 6, for every Kripke model $\mathcal{N}$ for $FP_k(RL_\Delta)$ and for every modal formula Ψ we can split the evaluation $\|\Psi\|_{\mathcal{N}}$ by evaluating Ψ 's sub-formulas and then turn back to the evaluation of Ψ by using the truth functions which are the semantic for the connectives occurring in Ψ (for instance $\|\Gamma \rightarrow \Lambda\|_{\mathcal{N}} = \|\Gamma\|_{\mathcal{N}} \Rightarrow \|\Lambda\|_{\mathcal{N}}$). The standard semantics for $\wedge$ and $\vee$ is given by the operators $\inf$ and $\sup$ respectively. Therefore $\|\Phi_r\|_{\mathcal{M}} = 1$ iff

$$\inf_{j=1, \dots, S} \left(\sup_{\alpha=1, \dots, r} \|\nabla P_\alpha \varphi_j\|_{\mathcal{M}} = 1 \right).$$

For the same reason, $\|\nabla P_\alpha \varphi_j\|_{\mathcal{M}} = 1$ iff $\nabla(\|P_\alpha \varphi_j\|_{\mathcal{M}}) = 1$, that is, iff $\|P_\alpha \varphi_j\|_{\mathcal{M}} > 0$ iff $\mu_\alpha(\varphi_j) > 0$.

Theorem 6. Let $\mathcal{C} = \{\varphi_1 \mid \psi_1, \dots, \varphi_n \mid \psi_n\}$, $\mathcal{A} = \{A_1, \dots, A_M\}$, $\varphi_1, \psi_1, \dots, \varphi_n, \psi_n$ and $\chi: P(\varphi_i \mid \psi_i) = n_i/m_i$ be a formula in $\mathcal{C}$ such that $r \leq M$. Then Φ_r is a Kripke model for $FP_M(R_\Delta)$ iff $T_\chi \models \bar{0}$.

- (*) χ is a probability distribution on $\mathcal{A}$.
- (**) $FP_M(R_\Delta) \models T_\chi$.
- (T₁) Φ_M .
- (T₂) $\Phi_r \rightarrow (\bigwedge_{j=1}^M P_{r+1}(A_j) \leftrightarrow 1/M)$.
- (T₃) $(\neg \Phi_r \wedge \bigvee_{\alpha=1}^r \nabla P_\alpha(A_j)) \rightarrow \neg P_{r+1}(A_j)$.
- (T₄) $\bigwedge_{\alpha=1}^M \bigoplus_{j=1}^M P_\alpha(A_j)$.
- (T₅) $(\nabla P_\alpha \psi_i \wedge \bigwedge_{\beta=1}^{\alpha-1} \neg P_\beta \psi_i) \rightarrow (n_i.P_\alpha \psi_i \leftrightarrow m_i.P_\alpha(\varphi_i \wedge \psi_i))$.
- (*) $T_\chi \not\models \bar{0}$.

We divide the proof into two parts: in the first one we show, as in Lemma 1, which is the interpretation of the axioms (T₁) – (T₅) in a Kripke model $\mathcal{M}$ for $FP_k(RL_\Delta)$, while in the second part we prove that such interpretations are equivalent to the requirements expressed in Theorem 4 in order to ensure the coherence of χ .

Lemma 2. Let $\mathcal{M} = \langle W, e, \mu_1, \dots, \mu_M \rangle$ be a Kripke model for $FP_M(R_\Delta)$.

- (*) $FP_M(R_\Delta) \vdash (T_1)$ iff $\mu_\alpha(A_s) > 0$.
- (**) $FP_M(R_\Delta) \vdash (T_2)$ iff $\mu_\alpha(A_s) > 0$, $\mu_{r+1}(A_t) = 1/M$, $\mu_{r+1}(A_t) = 0$ for $r \neq M$.
- (***) $FP_M(R_\Delta) \vdash (T_3)$ iff $\mu_\alpha(A_s) = 0$, $\mu_\beta(A_t) > 0$ for $\beta \leq r$.
- (*) $FP_M(R_\Delta) \vdash (T_4)$ iff $\sum_{s=1}^M \mu_\alpha(A_s) = 1$.
- (*) $FP_M(R_\Delta) \vdash (T_5)$ iff $\mu_\alpha(\psi_i) > 0$, $\mu_\alpha(\varphi_i \wedge \psi_i) = n_i/m_i$.

$$\sum_{s=1}^M \mu_\alpha(A_s) = 1.$$

(*) $FP_M(R_\Delta) \vdash (T_5)$ iff $\mu_\alpha(\psi_i) > 0$, $\mu_\alpha(\varphi_i \wedge \psi_i) = n_i/m_i$.

$$\frac{\mu_\alpha(\varphi_i \wedge \psi_i)}{\mu_\alpha(\psi_i)} = n_i/m_i.$$

(iii) (). This proof proceeds exactly as in Lemma 1. We just prove (iii) in order to quote an example.

From completeness Theorem 5 it follows that $FP_M(RL_\Delta) \vdash (T_3)$ iff $\|(T_3)\|_{\mathcal{M}} = 1$, that is $\|\neg\Phi_r \wedge \bigvee_{\alpha=1}^r \nabla P_\alpha(A_j) \rightarrow \neg P_{r+1}(A_j)\|_{\mathcal{M}} = 1$. Hence, by Definition 6, we obtain that $FP_k(RL_\Delta) \vdash (T_3)$ iff the following holds:

$$\|\neg\Phi_r \wedge \bigvee_{\alpha=1}^r \nabla P_\alpha(A_j)\|_{\mathcal{M}} = 1 - \|P_{r+1}(A_j)\|_{\mathcal{M}}. \quad (1)$$

Now:

($\Leftarrow$) Let A_s be an atom such that for every $\alpha \leq r$ $\mu_\alpha(A_s) = 0$. Let A_t be an atom and let $\beta \leq r$ be an index such that $\mu_\beta(A_t) > 0$. Then, following the lines of Lemma 2, it is not difficult to prove that $\|\neg\Phi_r \wedge \bigwedge_{\alpha=1}^r \nabla P_\alpha(A_s)\|_{\mathcal{M}} = 1$. Now, if $\mu_{r+1}(A_t) = 0$, then $\|P_{r+1}(A_t)\|_{\mathcal{M}} = 0$ and thus, by (1), the claim follows.

($\Rightarrow$) If $RL_\Delta \vdash (T_3)$ and if there exists an atom A_s such that for every $\alpha \leq r$, $\mu_\alpha(A_s) = 0$ while $\mu_\beta(A_t) > 0$ for an atom A_t and an index β , then it is clear that $\|\neg\Phi_r\|_{\mathcal{M}} = 1$ and $\|\bigvee_{\alpha=1}^r \nabla P_\alpha(A_t)\|_{\mathcal{M}} = 1$ and thus $\|\neg\Phi_r \wedge \bigvee_{\alpha=1}^r \nabla P_\alpha(A_t)\|_{\mathcal{M}} = \min\{\|\neg\Phi_r\|_{\mathcal{M}}, \|\bigvee_{\alpha=1}^r \nabla P_\alpha(A_t)\|_{\mathcal{M}}\} = 1$. From (1) it follows $1 - \|P_{r+1}(A_t)\|_{\mathcal{M}} = 1$, that is $\|P_{r+1}\|_{\mathcal{M}} = 0$. Hence $\mu_{r+1}(A_t) = 0$ as desired.

Now we turn back to the proof of Theorem 6.

($\Leftarrow$) Let T_χ be coherent and let $\mathcal{M} = \langle W, e, \mu_1, \dots, \mu_M \rangle$ be a Kripke model for T_χ . Let $\mathcal{P} = \{P_1, \dots, P_k\}$ be inductively defined as follows:

- (l) $P_1(A_s) = \mu_1(A_s)$ for each $A_s \in \mathcal{A}$,
- (h) $P_h(A_s) = \mu_h(A_s)$ if $\mu_i(A_s) = 0$ for each $i \leq h$. Otherwise, if there is an $j \leq h$ such that $\mu_j(A_s) > 0$, then P_h is not defined on A_s .

Axiom (T_4) assures that, for every $i = 1, \dots, k$, P_i is a coherent assessment over the class of atoms $\mathcal{A}_i = \{A_s \in \mathcal{A} : P_{i-1}(A_s) = 0\}$.

Now, let $\mathcal{B}$ be the algebra generated by the atoms in $\mathcal{A}$. $\mathcal{M} \models T_\chi$ and in particular $\mathcal{M} \models (T_1)$. By Lemma 2 it follows that for each atom $A_s \in \mathcal{A}$, there exists a probability measure μ_α such that $\mu_\alpha(A_s) > 0$. By construction it follows that $P_{\alpha'}(A_s) > 0$ for some $\alpha' \leq \alpha$. Hence, for every $\gamma \in \mathcal{B} \setminus \{\emptyset\}$, there exists a P_β such that $\sum_{A_r \subseteq \gamma} P_\beta(A_r) > 0$ and thus condition (a) of Theorem 4 is satisfied.

Let $\varphi_i \mid \psi_i$ be a conditional event in $\mathcal{C}$. We know, by axiom (T_1) , that there exists (at least) an index β such that $\mu_\beta(\psi_i) > 0$. Let α be the minimum of all such index, that is α is the first index such that $\mu_\alpha(\psi_i) > 0$. Then axiom (T_5) assures that $n_i/m_i = \frac{\mu_\alpha(\varphi_i \wedge \psi_i)}{\mu_\alpha(\psi_i)}$ and thus

$$n_i/m_i = \frac{\sum_{A_r \subseteq \varphi_i \wedge \psi_i} P_\alpha(A_r)}{\sum_{A_r \subseteq \psi_i} P_\alpha(A_r)}.$$

Therefore also condition (b) of Theorem 4 is satisfied and thus χ is coherent.

($\Rightarrow$) Let χ be coherent and let $\mathcal{P} = \{P_1, \dots, P_k\}$ be the class of assessment satisfying condition (a) and (b) of Theorem 4. Now we build a Kripke model $\mathcal{M}$ for T_χ as follows: for every atom $A_s \in \mathcal{A}$ we define:

- (1) $\mu_1(A_s) = P_1(A_s)$ and
 (h) If $k < M$ and $h \leq k$ or if $k = M$, then $\mu_h(A_s) = P_h(A_s)$ if $P_h(A_s)$ is defined and $\mu_h(A_s) = 0$ otherwise.
 If $k < M$ and $M \geq h > k$, then we define $\mu_h(A_s) = 1/M$.

It is clear that for every $\alpha = 1, \dots, M$, μ_α is a coherent assessment over $\mathcal{A}$ and so every μ_α can be extended to a finitely additive probability measure defined on the algebra $\mathcal{B}$ generated by the atoms in $\mathcal{A}$. With an abuse we will use the same notation μ_α for both the assessment and the probability measure.

Let now $\mathcal{M}$ be the following system: $\mathcal{M} = \langle W, e, \mu_1, \dots, \mu_M \rangle$ where W is the set of all Boolean evaluation of formulas in $\mathcal{B}$ and e be defined, for every $w \in W$ and for every Boolean formula φ , by $e(w, \varphi) = w(\varphi)$. Thus $\mathcal{M}$ is a Kripke model for $FP_k(RL_\Delta)$ and $\mathcal{M} \models T_\chi$. In fact:

For every event $\gamma \in \mathcal{B}$ there exists an α such that $P_\alpha(\gamma) > 0$. In particular for every atom $A_r \in \mathcal{A}$ there exists an α' such that $P_{\alpha'}(A_r) > 0$. Then $\mu_{\alpha'}(A_r)$ and thus $\mathcal{M} \models (T_1)$.

That axioms (T_2) and (T_3) are satisfied by $\mathcal{M}$ is assured by the previous definition of the probabilities functions $\mu_1, \dots, \mu_M$.

By Theorem 4, each P_α is coherent and thus, for each α $\sum_{A_r \in \mathcal{A}_\alpha} P_\alpha(A_r) = 1$. It follows that for each α ,

$$\sum_{A_r \in \mathcal{A}} \mu_\alpha(A_r) = \sum_{A_r \in \mathcal{A}_\alpha} \mu_\alpha(A_r) + \sum_{A_s \in \mathcal{A} \setminus \mathcal{A}_\alpha} \mu_\alpha(A_s) = 1$$

being $\mu_\alpha(A_r) = P_\alpha(A_r)$ if $A_r \in \mathcal{A}_\alpha$ and $\mu_\alpha(A_s) = 0$ if $A_s \in \mathcal{A} \setminus \mathcal{A}_\alpha$. Then $\mathcal{M} \models (T_4)$.

By Theorem 4, if α is the zero-layer of ψ_i , then $\mu_\alpha(\psi_i) = \sum_{A_r \subseteq \psi_i} P_\alpha(A_r)$ and $\mu_\alpha(\varphi_i \wedge \psi_i) = \sum_{A_r \subseteq \varphi_i \wedge \psi_i} P_\alpha(A_r)$, thus

$$\frac{\mu_\alpha(\varphi_i \wedge \psi_i)}{\mu_\alpha(\psi_i)} = \frac{\sum_{A_r \subseteq \varphi_i \wedge \psi_i} P_\alpha(A_r)}{\sum_{A_r \subseteq \psi_i} P_\alpha(A_r)} = \frac{n_i}{m_i}.$$

Therefore $\mathcal{M} \models (T_5)$.

Thus $\mathcal{M} \models T_\chi$ and thus T_χ is coherent and Theorem 6 is completely proved.

4 Conclusion and Future Work

In this paper we presented the fuzzy probabilistic logic $FP_k(RL_\Delta)$ to reason about conditional probability. In particular we applied such logic in order to study the coherence of an assessment of conditional probability showing that there is a way of associating to every assessment of conditional probability a many-valued theory T_χ over $FP_k(RL_\Delta)$ which is logically coherent iff χ is coherent.

Here we presented such a logic but, unfortunately, we have been obliged to use not one but (finitely) n_i modalities in order to simulate zero-layers. The

theory T_χ has a number of axioms which (in the worst case) is exponential in the number of conditional events and thus we believe it is correct to claim that we have not found a good algorithm for checking the coherence of an assessment of conditional probability.

In the future, we plan to prove that the axioms of T_χ can be reduced to a polynomial number (in the number of the conditional events) of formulas with polynomial length. This result, together with a proof of co-NP containment for $FP_k(RL_\Delta)$ would give an NP-containment of the coherence problem of assessments of conditional probability.

Acknowledgments. The author is indebted to Franco Montagna for useful comments and precious suggestions as well as to anonymous referee for have led to considerable improvements in this final version.

References

1. P. CINTULA, *The LII and $LII\frac{1}{2}$ propositional and predicate logics*, Fuzzy Sets and Systems, **124**, (2001), 21-34.
2. P. CINTULA, *Advances in the LII and $LII\frac{1}{2}$ logics*, Archive for Mathematical Logic, **42**, (2003), 449-468.
3. R. CIGNOLI, D. MUNDICI, I. M. L. D'OTTAVIANO, *Algebraic Foundation of Many-valued Reasoning*, Kluwer, 2000.
4. G. COLETTI, R. SCOZZAFAVA, *Probabilistic Logic in a Coherent Setting*, Trends in Logic, Vol. **15** Kluwer, 2002.
5. F. ESTEVA, L. GODO, P. HÁJEK, *Reasoning about probability using fuzzy logic*, Neural Network World, **10**, No. 5 (2000), 811-824.
6. F. ESTEVA, L. GODO, F. MONTAGNA, *LII and $LII\frac{1}{2}$: two fuzzy logics joining Lukasiewicz and Product logics*, Archive for Mathematical Logic, **40** (2001), 39-67.
7. T. FLAMINIO, F. MONTAGNA, *A Logic and Algebraic Treatment of Conditional Probability*, Archive for Mathematical Logic, **44** (2005), 245-262.
8. B. GERLA, *Rational Lukasiewicz Logic and Divisible MV-algebras*. Neural Networks World, **10**, (2001).
9. B. GERLA, *Many-valued Logics of Continuous t-norms and Their Functional Representation*, Ph.D Thesis, University of Milan (2001).
10. L. GODO, E. MARCHIONI, *A logic for reasoning about coherent conditional probability: a fuzzy modal logic approach*. Lecture Notes in Artificial Intelligence, **3229**: 9th European Conference on Logics in Artificial Intelligence JELIA'04. Lisbon, Portugal, September 2004. José Júlio Alferes and Joao Leite (Eds). 213-225.
11. P. HÁJEK, *Metamathematics of fuzzy logic*, Kluwer, 1998.
12. P. HÁJEK, S. TULIPANI, *Complexity of Fuzzy Probabilistic Logics*, Fundamenta Informaticae, **45** (2001), 207-213.

A Logic with Coherent Conditional Probabilities^{*}

Nebojša Ikodinović¹ and Zoran Ognjanović²

¹ Faculty of Science - Department of Mathematics and Informatics,
R. Domanovića 12, 34000 Kragujevac, Serbia and Montenegro
ikodinovic@kg.ac.yu

² Mathematical Institute, Kneza Mihaila 35,
11000 Belgrade, Serbia and Montenegro
zorano@mi.sanu.ac.yu

Abstract. In this paper we investigate a probability logic which enriches propositional calculus with a class of conditional probability operators of de Finetti's type. The logic allows making formulas such as $CP_{\geq s}(\beta \mid \alpha)$, with the intended meaning "the conditional probability of β given α is at least s ". A possible-world approach is proposed to give semantics to such formulas. An infinitary axiomatic system for our logic which is sound and complete with respect to the mentioned class of models is given. We prove decidability of the presented logic.

1 Introduction

The idea to include probability as a part of logical systems is quite old. Some of the basic concepts of probabilistic treatment of propositions appeared even in Boole's book [1]. In the recent decades, there are two main approaches to formalization of probabilistic reasoning which perform different forms of reasoning: logic with probabilistic quantifiers (introduced by Keisler [11] and further developed in [10, 12, 21, 23]) and probabilistic systems with probabilistic operators of the form (in our notation) $P_{\geq s}$ with the intended meaning 'the probability is at least s ' [4, 5, 16, 17, 19, 20, 22]. The first approach is appropriate for so-called objective probability, i.e. for reasoning with statistical information (objective chance based on relative frequency). In the second case, absolute (unconditional) probability is interpreted as degree of belief. Many authors (N. Nilsson, for example, in a review [18] of work subsequent to his paper [17]) argue that the conditional probability of β given α reflects more accurately what we normally mean by the certainty of the rule 'if α than β ', then the probability of $\alpha \rightarrow \beta$ does. In this paper we continue the second line of the above mentioned research by presenting a formalism for reasoning about conditional probability.

^{*} This research was supported by Ministarstvo za nauku, tehnologije i razvoj Republike Srbije, through Matematički institut, under grant 1379.

There are also at least two approaches to the calculus of probability starting from different conceptual premisses. In the first one, which was given a firm mathematical foundations by Kolmogorov [13], probability is seen as a statistical, i. e. objective, notion, while in the second one, due to de Finetti [2, 3], only subjective probabilities exists. Beside the other consequences, these approaches result also in different understandings of conditional probability. In the objective approach conditional probability is defined via the unconditional probability by the formula $P(A \mid B) = \frac{P(A \cap B)}{P(B)}$ with the proviso that $P(B) > 0$. Kolmogorov's approach has become widely accepted so it is often referred to as the definition of conditional probability. On the other hand, under the subjective view, conditional probability is seen as more primitive concept than unconditional probability. Many examples from scientific and philosophical practice (see for example [2]) show that Kolmogorov's approach is not an adequate analysis of conditional probability, especially when the conditional probability $P(E \mid H)$ is considered having very small or even zero probability of the given conditioning event H .

Despite the fact that one of the crucial issue in the reasoning about probability is the notion of conditional probability, only a few papers discussing conditional probabilities from the logical point of view can be found in the literature. Hawthorne [8] described a range of nonmonotonic conditionals that behave like conditional probability functions at various levels of probabilistic supports. These conditional were defined as semantic relations on an object language for propositional logic. In [9] he extended the semantics of the most prominent family of these nonmonotonic conditionals to the first order language. In [4] conditional probabilities, in the sense of Kolmogorov, are defined syntactically. However, a complicated machinery of real closed fields is needed to obtain a corresponding sound and complete axiomatization. In [2], conditional probability is analyzed at the semantical level along the ideas proposed by de Finetti. Under the same view, a fuzzy modal logic is introduced in [15], such that for each pair of classical propositional formulas α and β , the probability of the conditional event " α given β " is taken as the truth-value of the (fuzzy) modal proposition $P(\alpha \mid \beta)$. In [24, 25] a logic, which contains several types of probabilistic operators (including operators of the form "the conditional probability of α given β is s "), is defined. The range of the probability functions is taken to be the unit interval of a recursive nonarchimedean field. Then, it is possible to define another probabilistic operator with the intended meaning "probabilities of $\alpha \wedge \beta$ and β are almost the same" which may be used to model default reasoning. In [6] a treatment of nonstandard conditional probability by means of fuzzy logic is given.

In this paper we consider a logic (denoted *LPCP*) which is suitable for reasoning about conditional probability that are based on de Finetti's approach. The corresponding probability language enriches propositional calculus with probability operators of the form $CP_{\geq s}$ (for every rational $s \in [0, 1]$). *LPCP* allows making statements such as $CP_{\geq s}(\alpha \mid \beta)$, with the intended meaning "the conditional probability of α given β is at least s ". We describe the corresponding class of Kripke models with probability measure defined over the words. These

models give semantics to the probability formulas so that interpreted formulas are either true or false. Following [16, 19, 20, 22, 25], we give an axiomatic system for which we prove the extended completeness theorem ('Every consistent set of formulas is satisfiable') with respect to the mentioned class of models. Since the satisfiability problem in our logic can be formulated in the form of checking coherence of conditional probability assessments, although our axiomatic system contains an infinitary inference rule, we can follow [2] and prove that the logic is decidable.

The rest of the paper is organized as follows. In Section 2 syntax of *LPCP* is given. Section 3 describes the corresponding class of models, while in Section 4 an axiomatic system for *LPCP* is introduced. A proof of Completeness theorem is presented in Section 5. Decidability of the logic is proved in Section 6. In Section 7 we summarize our results and compare *LPCP* to some of the related systems.

2 Syntax

The language of *LPCP* is a classical propositional language extended by a list of probability operators $CP_{\geq s}$, for every rational number s from $[0, 1]$. Let us denote the set of all classical propositional formulas by For_{LPCP}^C . If $\alpha, \beta \in For_{LPCP}^C$, where β is not a propositional contradiction (it can be checked in the well-known routine way) and s a rational number, $CP_{\geq s}(\alpha \mid \beta)$ is a basic probability formula. The set of all probability formulas is the least set For_{LPCP}^P containing all basic probability formulas, and closed under the formation rules: if $A, B \in For_{LPCP}^P$, then $\neg A, A \wedge B \in For_{LPCP}^P$. Let $For_{LPCP} = For_{LPCP}^C \cup For_{LPCP}^P$. We use the usual abbreviations for the other classical connectives, and also denote (if β is not a propositional contradiction):

- $CP_{< s}(\alpha \mid \beta) \stackrel{\text{def}}{=} \neg CP_{\geq s}(\alpha \mid \beta)$,
- $CP_{\leq s}(\alpha \mid \beta) \stackrel{\text{def}}{=} CP_{\geq 1-s}(\neg \alpha \mid \beta)$,
- $CP_{> s}(\alpha \mid \beta) \stackrel{\text{def}}{=} \neg CP_{\leq s}(\alpha \mid \beta)$, and
- $CP_{= s}(\alpha \mid \beta) \stackrel{\text{def}}{=} CP_{\geq s}(\alpha \mid \beta) \wedge CP_{\leq s}(\alpha \mid \beta)$.

In the sequel, elements from For_{LPCP}^C will be denoted by $\alpha, \beta, \dots$, elements from For_{LPCP}^P will be denoted by $A, B, \dots$, while we will use $\Phi, \Psi, \dots$, to denote formulas from For_{LPCP} .

3 Semantics

We use the possible-worlds approach to give semantics to formulas such that they remain either true or false.

Following [2], let Ω be a non empty set, H an algebra of subsets of Ω , and $H^0 = H \setminus \{\emptyset\}$. Then, $\mu : H \times H^0 \rightarrow [0, 1]$, is a conditional probability if the following holds:

- $\mu(A, A) = 1$, for every $A \in H^0$,
- $\mu(\cdot, A)$ is a finitely additive probability on H for any given $A \in H^0$, and
- $\mu(C \cap B, A) = \mu(B, A) \cdot \mu(C, B \cap A)$, for all $C \in H$ and $A, B, A \cap B \in H^0$.

An *LPCP*-model is a structure $\mathbf{M} = \langle W, H, \mu, v \rangle$ where:

- W is a nonempty set of objects called worlds,
- H is an algebra of subsets of W ,
- μ is a conditional probability, $\mu : H \times H^0 \rightarrow [0, 1]$,
- $v : W \times \{p_1, p_2, \dots\} \rightarrow \{true, false\}$ is a valuation which associates with every world $w \in W$ a truth assignment $v(w, \cdot)$ on the propositional letters.

Let $\mathbf{M} = \langle W, H, \mu, v \rangle$ be an *LPCP*-model, and $\alpha \in For_{LPCP}^C$. We use $[\alpha]_{\mathbf{M}}$ (or just $[\alpha]$, if $\mathbf{M}$ is clear from the context) to denote the set $\{w \in W : w \models \alpha\}$. An *LPCP*-model $\mathbf{M} = \langle W, H, \mu, v \rangle$ is measurable if $[\alpha] \in H$ for every formula $\alpha \in For_{LPCP}^C$. We denote the class of all measurable models by $LPCP_{Meas}$.

The satisfiability relation $\models \subseteq LPCP_{Meas} \times For_{LPCP}$ is defined by the following conditions for every measurable *LPCP*-model $\mathbf{M} = \langle W, H, \mu, v \rangle$:

- if $\alpha \in For_{LPCP}^C$, $\mathbf{M} \models \alpha$ iff $v(w, \alpha) = true$ for every world $w \in W$,
- $\alpha, \beta \in For_{LPCP}^C$, $\mathbf{M} \models CP_{\geq s}(\alpha \mid \beta)$ iff $\mu([\alpha], [\beta]) \geq s$.
- if $A \in For_{LPCP}^P$, $\mathbf{M} \models \neg A$ iff $\mathbf{M} \not\models A$,
- if $A, B \in For_{LPCP}^P$, $\mathbf{M} \models A \wedge B$ iff $\mathbf{M} \models A$ and $\mathbf{M} \models B$.

A formula $\Phi \in For_{LPCP}$ is satisfiable if there is an model $\mathbf{M}$ from $LPCP_{Meas}$ such that $\mathbf{M} \models \Phi$; Φ is valid if for every model $\mathbf{M}$ from $LPCP_{Meas}$, $\mathbf{M} \models \Phi$; a set of formulas is satisfiable if there is a model in which every formula from the set is satisfiable.

4 Axiomatic System

The axiomatic system Ax_{LPCP} for *LPCP* contains the following axiom schemata:

1. all For_{LPCP}^C -instances of classical propositional tautologies,
2. all For_{LPCP}^P -instances of classical propositional tautologies,
3. $CP_{\geq 0}(\alpha \mid \beta)$,
4. $CP_{\leq r}(\alpha \mid \beta) \rightarrow CP_{\leq s}(\alpha \mid \beta)$, $s > r$,
5. $CP_{\leq s}(\alpha \mid \beta) \rightarrow CP_{\leq s}(\alpha \mid \beta)$,
6. $(CP_{\geq r}(\alpha \mid \gamma) \wedge CP_{\geq s}(\beta \mid \gamma) \wedge CP_{\geq 1}(\neg \alpha \vee \neg \beta \mid \gamma)) \rightarrow CP_{\geq \min\{1, r+s\}}(\alpha \vee \beta \mid \gamma)$,
7. $(CP_{\leq r}(\alpha \mid \gamma) \wedge CP_{\leq s}(\beta \mid \gamma)) \rightarrow CP_{\leq r+s}(\alpha \vee \beta \mid \gamma)$, $r + s < 1$,
8. $CP_{\geq s}(\alpha \mid \gamma) \wedge CP_{\geq r}(\beta \mid \alpha \wedge \gamma) \rightarrow CP_{\geq s \cdot r}(\alpha \wedge \beta \mid \gamma)$,
9. $CP_{\geq 1}(\beta \mid \gamma) \wedge CP_{\geq s}(\alpha \wedge \beta \mid \gamma) \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma)$,

and inference rules:

1. from α and $\alpha \rightarrow \beta$ infer β ,
2. from $\alpha \rightarrow \beta$ infer $CP_{\geq 1}(\beta \mid \alpha)$,
3. from $A \rightarrow (CP_{\geq t}(\beta \mid \gamma) \rightarrow CP_{\geq s \cdot t}(\alpha \wedge \beta \mid \gamma))$, for every rational number t from $(0, 1)$, infer $A \rightarrow CP_{\geq s}(\alpha \mid \gamma \wedge \beta)$.

Note that Rule 3 is the only infinitary rule in Ax_{LPCP} . It corresponds to the last part in the definition of conditional probability.

A formula Φ is deducible from a set T of sentences ($T \vdash \Phi$) if there is an at most countable sequence of formulas $\Phi_0, \Phi_1, \dots, \Phi$, such that every Φ_i is an axiom or a formula from the set T , or it is derived from the preceding formulas by an inference rule. A formula Φ is a theorem ($\vdash \Phi$) if it is deducible from the empty set. A set T of sentences is consistent if there is at least one formula from For_{LPCP}^C , and at least one formula from For_{LPCP}^P that are not deducible from T , otherwise T is inconsistent. A consistent set T of sentences is said to be maximal consistent if the following holds:

- for every $\alpha \in For_{LPCP}^C$, if $T \vdash \alpha$, then $\alpha \in T$ and $P_{\geq 1}\alpha \in T$, and
- for every $A \in For_{LPCP}^P$, either $A \in T$ or $\neg A \in T$.

A set T is deductively closed if for every $\varphi \in For_{LPCP}$, if $T \vdash \varphi$, then $\varphi \in T$.

5 Completeness

Theorem 1 (Soundness Theorem). *The axiomatic system Ax_{LPCP} is sound with respect to the class of $LPCP_{Meas}$ -models.*

Proof. Soundness of our system follows from the soundness of propositional classical logic and from the properties of conditional probability. The arguments are of the type presented in the proof of Theorem 13 in [16]. $\square$

In order to prove Completeness theorem for $LPCP$, we show that every consistent set of formulas is satisfiable. We begin with some auxiliary statements. Then, we describe how a consistent set T of formulas can be extended to a suitable maximal consistent set, and how a canonical model can be constructed out of such maximal consistent set.

Theorem 2. 1. (Deduction Theorem) *If T is a set of formulas, Φ is a formula, and $T \cup \{\Phi\} \vdash \Psi$, then $T \vdash \Phi \rightarrow \Psi$, where Φ and Ψ are either both classical or both probability formulas.*

2. *Let α, β, γ be classical formulas, and γ is not a propositional contradiction.*

Then:

- (a) $\vdash CP_{\geq 1}(\alpha \rightarrow \beta \mid \gamma) \rightarrow (CP_{\geq s}(\alpha \mid \gamma) \rightarrow CP_{\geq s}(\beta \mid \gamma))$,
- (b) $\vdash CP_{\geq 1}(\alpha \rightarrow \beta \mid \gamma) \rightarrow (CP_{< s}(\beta \mid \gamma) \rightarrow CP_{< s}(\alpha \mid \gamma))$,
- (c) $\vdash CP_{\geq r}(\alpha \mid \gamma) \rightarrow CP_{\geq s}(\alpha \mid \gamma)$, $r > s$,
- (d) $\vdash CP_{\geq 1}(\alpha \wedge \beta \mid \gamma) \rightarrow (CP_{\geq 1}(\alpha \mid \gamma) \wedge CP_{\geq 1}(\beta \mid \gamma))$,
- (e) $\vdash (CP_{\geq 1}(\alpha \mid \gamma) \wedge CP_{\geq 1}(\beta \mid \gamma)) \rightarrow CP_{\geq 1}(\alpha \wedge \beta \mid \gamma)$,
- (f) $\vdash CP_{\geq 1}(\alpha \wedge \beta \mid \gamma) \leftrightarrow (CP_{\geq 1}(\alpha \mid \gamma) \wedge CP_{\geq 1}(\beta \mid \gamma))$,
- (g) $\vdash CP_{\geq 1}(\alpha \leftrightarrow \beta \mid \gamma) \rightarrow (CP_{\diamond s}(\alpha \mid \gamma) \leftrightarrow CP_{\diamond s}(\beta \mid \gamma))$, $\diamond \in \{\leq, \geq, <, >, =\}$,
- (h) $\vdash CP_{\geq s}(\alpha \mid \gamma) \leftrightarrow CP_{\geq s}(\alpha \mid \gamma \wedge \neg \perp)$,
- (i) $\vdash CP_{\geq s}(\alpha \mid \beta \wedge \gamma) \rightarrow (CP_{\geq t}(\beta \mid \gamma) \rightarrow CP_{\geq s \cdot t}(\alpha \wedge \beta \mid \gamma))$, if $\beta \wedge \gamma$ is not a propositional contradiction,
- (j) $\vdash CP_{\geq 1}(\beta \leftrightarrow \gamma \mid \neg \perp) \rightarrow (CP_{\geq s}(\alpha \mid \beta) \leftrightarrow CP_{\geq s}(\alpha \mid \gamma))$.

Proof. (1) We use the transfinite induction on the length of the proof of Ψ from $T \cup \{\Phi\}$. The classical cases follow as usual. Suppose that $\Phi, \Psi \in For_{LPCP}^P$, and $\Psi = A \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma)$ is obtained from $T \cup \{\Phi\}$ by application of the inference rule 3. Then:

$T \cup \{\Phi\} \vdash A \rightarrow (CP_{\geq t}(\beta \mid \gamma) \rightarrow CP_{\geq s, t}(\alpha \wedge \beta \mid \gamma))$, for all t ,
 $T \vdash \Phi \rightarrow (A \rightarrow (CP_{\geq t}(\beta \mid \gamma) \rightarrow CP_{\geq s, t}(\alpha \wedge \beta \mid \gamma)))$, for all t , by the induction hypothesis,
 $T \vdash (\Phi \wedge A) \rightarrow (CP_{\geq t}(\beta \mid \gamma) \rightarrow CP_{\geq s, t}(\alpha \wedge \beta \mid \gamma))$, for all t ,
 $T \vdash (\Phi \wedge A) \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma)$, by the inference rule 3,
 $T \vdash \Phi \rightarrow \Psi$.

(2) We prove some of the statements, while the other can be shown in a similar way.

(2d) The proof is:

$\vdash \gamma \rightarrow ((\alpha \wedge \beta) \rightarrow \alpha)$
 $\vdash \gamma \rightarrow ((\alpha \wedge \beta) \rightarrow \beta)$
 $\vdash CP_{\geq 1}((\alpha \wedge \beta) \rightarrow \alpha \mid \gamma)$
 $\vdash CP_{\geq 1}((\alpha \wedge \beta) \rightarrow \beta \mid \gamma)$
 $\vdash CP_{\geq 1}(\alpha \wedge \beta \mid \gamma) \rightarrow CP_{\geq 1}(\alpha \mid \gamma)$
 $\vdash CP_{\geq 1}(\alpha \wedge \beta \mid \gamma) \rightarrow CP_{\geq 1}(\beta \mid \gamma)$
 $\vdash CP_{\geq 1}(\alpha \wedge \beta \mid \gamma) \rightarrow (CP_{\geq 1}(\alpha \mid \gamma) \wedge CP_{\geq 1}(\beta \mid \gamma)).$

(2e) The proof is:

$\vdash \gamma \rightarrow (\alpha \rightarrow (\beta \rightarrow (\alpha \wedge \beta)))$
 $\vdash CP_{\geq 1}(\alpha \rightarrow (\beta \rightarrow (\alpha \wedge \beta)) \mid \gamma)$
 $\vdash CP_{\geq 1}(\alpha \mid \gamma) \rightarrow CP_{\geq 1}(\beta \rightarrow (\alpha \wedge \beta) \mid \gamma)$
 $\vdash CP_{\geq 1}(\alpha \mid \gamma) \rightarrow (CP_{\geq 1}(\beta \mid \gamma) \rightarrow CP_{\geq 1}(\alpha \wedge \beta \mid \gamma))$
 $\vdash (CP_{\geq 1}(\alpha \mid \gamma) \wedge CP_{\geq 1}(\beta \mid \gamma)) \rightarrow CP_{\geq 1}(\alpha \wedge \beta \mid \gamma).$

(2h) By an application of Rule 2, we obtain $\vdash CP_{\geq 1}(\neg \perp \mid \gamma)$, from $\vdash \gamma \rightarrow \neg \perp$. From (2g) we have $\vdash CP_{\geq s}(\alpha \mid \gamma) \leftrightarrow CP_{\geq s}(\alpha \wedge \neg \perp \mid \gamma)$. Now, by the axioms 8 and 9 we obtain $\vdash CP_{\geq s}(\alpha \mid \gamma) \leftrightarrow CP_{\geq s}(\alpha \mid \gamma \wedge \neg \perp)$.

(2j) If $\vdash CP_{\geq 1}(\beta \leftrightarrow \gamma \mid \neg \perp)$, then $\vdash CP_{\geq r}(\beta \mid \neg \perp) \leftrightarrow CP_{\geq r}(\gamma \mid \neg \perp)$ and $\vdash CP_{\geq r}(\alpha \wedge \beta \mid \neg \perp) \leftrightarrow CP_{\geq r}(\alpha \wedge \gamma \mid \neg \perp)$, for all r . If $\vdash CP_{\geq s}(\alpha \mid \beta)$ then $\vdash CP_{\geq s}(\alpha \mid \beta \wedge \neg \perp)$, and from (2i) we have $\vdash CP_{\geq s}(\alpha \mid \beta) \rightarrow (CP_{\geq t}(\beta \mid \neg \perp) \rightarrow CP_{\geq s, t}(\alpha \wedge \beta \mid \neg \perp))$, for all t , and from the above facts $\vdash CP_{\geq s}(\alpha \mid \beta) \rightarrow (CP_{\geq t}(\gamma \mid \neg \perp) \rightarrow CP_{\geq s, t}(\alpha \wedge \gamma \mid \neg \perp))$, for all t . Finally, by Rule 3, we obtain $\vdash CP_{\geq s}(\alpha \mid \beta) \rightarrow CP_{\geq s}(\alpha \mid \gamma \wedge \neg \perp)$. Thus, from (2h), $\vdash CP_{\geq s}(\alpha \mid \beta) \rightarrow CP_{\geq s}(\alpha \mid \gamma)$. Similarly, we can prove $\vdash CP_{\geq s}(\alpha \mid \gamma) \rightarrow CP_{\geq s}(\alpha \mid \beta)$. $\square$

Theorem 3. *Every consistent set of formulas can be extended to a maximal consistent set.*

Proof. Let T be a consistent set, $Con^C(T)$ the set of all classical formulas that are consequences of T , and $A_0, A_1, \dots$, an enumeration of For_{LPCP}^P . We define a sequence of sets T_i , $i = 0, 1, 2, \dots$, such that:

1. $T_0 = T \cup \text{Con}^C(T) \cup \{CP_{\geq 1}(\alpha \mid \beta) : \beta \rightarrow \alpha \in \text{Con}^C(T), \beta \not\vdash \perp\}$,
2. for every $i \geq 0$, if $T_i \cup \{A_i\}$ is consistent, then $T_{i+1} = T_i \cup \{A_i\}$,
3. for every $i \geq 0$, if $T_i \cup \{A_i\}$ is not consistent, then $T_{i+1} = T_i \cup \{\neg A_i\}$.
4. if the set T_{i+1} is obtained by adding a formula of the form $\neg(A \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma))$, then for some rational number $t \in (0, 1)$, $A \rightarrow \neg(CP_{\geq t}(\beta \mid \gamma) \rightarrow CP_{\geq s \cdot t}(\alpha \wedge \beta \mid \gamma)) \rightarrow CP_{\geq s \cdot t}(\alpha \wedge \beta \mid \gamma)$, is also added to T_{i+1} , so that T_{i+1} is consistent.

We have to prove that every T_i is a consistent set. The sets obtained by the steps 1-3 are obviously consistent. Consider the step 4. If $T_i \cup \{A \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma)\}$ is not consistent, then the set T_i can be consistently extended as it is described above. Suppose that it is not the case. Then:

1. $T_i, \neg(A \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma)), A \rightarrow \neg(CP_{\geq t}(\beta \mid \gamma) \rightarrow CP_{\geq s \cdot t}(\alpha \wedge \beta \mid \gamma)) \vdash \perp$, for all t ,
2. $T_i, \neg(A \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma)) \vdash \neg(A \rightarrow \neg(CP_{\geq t}(\beta \mid \gamma) \rightarrow CP_{\geq s \cdot t}(\alpha \wedge \beta \mid \gamma)))$, for all t , by the deduction theorem,
3. $T_i, \neg(A \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma)) \vdash A \rightarrow (CP_{\geq t}(\beta \mid \gamma) \rightarrow CP_{\geq s \cdot t}(\alpha \wedge \beta \mid \gamma))$, for all t , by the classical tautology $\neg(p \rightarrow q) \rightarrow (p \rightarrow \neg q)$,
4. $T_i, \neg(A \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma)) \vdash A \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma)$, by Rule 3.

We continue by showing that the set $\overline{T} = \cup_i T_i$ is maximal and consistent. First, we have to prove that $\overline{T}$ is deductively closed. The proof is along the line of the corresponding proof of Theorem 6 from [25], so we consider only the case when $A \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma)$ is obtained by the infinitary inference rule 3, assuming that all the premises $A \rightarrow (CP_{\geq t_j}(\beta \mid \gamma) \rightarrow CP_{\geq s \cdot t_j}(\alpha \wedge \beta \mid \gamma))$ belong to $\overline{T}$ ($t_1, t_2, \dots$ is an enumeration of all rational numbers from $[0, 1]$). If $A \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma) \notin \overline{T}$, by the construction of $\overline{T}$, there is some m , such that $A \rightarrow \neg(CP_{\geq t_m}(\beta \mid \gamma) \rightarrow CP_{\geq s \cdot t_m}(\alpha \wedge \beta \mid \gamma)) \in \overline{T}$. There is a set T_k which contains both $A \rightarrow (CP_{\geq t_m}(\beta \mid \gamma) \rightarrow CP_{\geq s \cdot t_m}(\alpha \wedge \beta \mid \gamma))$ and $A \rightarrow \neg(CP_{\geq t_m}(\beta \mid \gamma) \rightarrow CP_{\geq s \cdot t_m}(\alpha \wedge \beta \mid \gamma))$. It follows that $T_k \vdash A \rightarrow \perp$, which means that $A \rightarrow \perp \in \overline{T}$. However, it means that $A \rightarrow CP_{\geq s}(\alpha \mid \beta \wedge \gamma) \in \overline{T}$ as well. The above construction guarantees that for every formula ϕ , ϕ and $\neg\phi$ cannot be simultaneously in $\overline{T}$. Similarly, for every probability formula A it must be either $A \in \overline{T}$ or $\neg A \in \overline{T}$. It follows that $\overline{T}$ is a maximal consistent set. $\square$

Theorem 4. *Let $\overline{T}$ be a maximal consistent set of formulas. Let Φ and Ψ be either both classical or both probability formulas. Then the following hold:*

1. *If $\Phi \in \overline{T}$, then $\neg\Phi \notin \overline{T}$.*
2. *$\Phi \wedge \Psi \in \overline{T}$ iff $\Phi \in \overline{T}$ and $\Psi \in \overline{T}$.*
3. *If $\overline{T} \vdash \Phi$, then $\Phi \in \overline{T}$, i.e. $\overline{T}$ is deductively closed.*
4. *If $\Phi \in \overline{T}$ and $\Phi \rightarrow \Psi \in \overline{T}$, then $\Psi \in \overline{T}$.*
5. *If $CP_{\geq s}(\alpha \mid \beta) \in \overline{T}$, and $s \geq r$, then $CP_{\geq r}(\alpha \mid \beta) \in \overline{T}$.*
6. *For every rational $r = \sup\{s : CP_{\geq s}(\alpha \mid \beta) \in \overline{T}\}$, $CP_{\geq r}(\alpha \mid \beta) \in \overline{T}$.*

Proof. As an example we prove 6. Let $r = \sup\{s : CP_{\geq s}(\alpha \mid \beta) \in \overline{T}\}$. If $CP_{\geq r}(\alpha \mid \beta) \notin \overline{T}$, then $CP_{\geq r}(\alpha \mid \beta \wedge \neg\perp) \notin \overline{T}$. There is a rational number

$t_0 \in (0, 1)$, such that $\neg(CP_{\geq t_0}(\neg\perp \mid \beta) \rightarrow CP_{\geq r \cdot t_0}(\alpha \wedge \neg\perp \mid \beta)) \in \overline{T}$, i.e. $CP_{< r \cdot t_0}(\alpha \wedge \neg\perp \mid \beta) \wedge CP_{\geq t_0}(\neg\perp \mid \beta) \in \overline{T}$. So, we have $CP_{< r \cdot t_0}(\alpha \wedge \neg\perp \mid \beta) \in \overline{T}$ (and $CP_{\geq t_0}(\neg\perp \mid \beta) \in \overline{T}$), i.e. $CP_{< r \cdot t_0}(\alpha \mid \beta) \in \overline{T}$, a contradiction. $\square$

Theorem 5 (Completeness Theorem for $LPCP_{\text{Meas}}$). *Every consistent set T of formulas has an $LPCP_{\text{Meas}}$ -model.*

Proof. Using the maximal consistent set $\overline{T}$, we define a tuple $\mathbf{M} = \langle W, H, \mu, v \rangle$:

- $W = \{w \mid w \models \text{Con}^C(T)\}$ contains all interpretations that satisfy the set of all classical consequences T ,
- $[\alpha] = \{w \in W \mid w \models \alpha\}$ and $H = \{[\alpha] \mid \alpha \in \text{For}_{LPCP}^C\}$,
- $\mu : H \times H^0 \rightarrow [0, 1]$, such that $\mu([\alpha], [\beta]) = \sup\{s : CP_{\geq s}(\alpha \mid \beta) \in \overline{T}\}$,
- $v : W \times I \rightarrow \{\text{true}, \text{false}\}$ is an assignment such that for every world $w \in W$ and every propositional letter $p \in I$, $v(w, p) = \text{true}$ iff $p \in w$.

First we have to prove that $\mathbf{M}$ is an $LPCP_{\text{Meas}}$ -model. By the theorems 2 and 4, if $[\alpha] = [\beta]$ and $[\gamma] = [\delta]$, then $\mu([\alpha], [\beta]) = \mu([\gamma], [\delta])$, and hence μ is well-defined. Since $\vdash \alpha \rightarrow \alpha$ and, by inference rule 2, we have $\vdash CP_{\geq 1}(\alpha \mid \alpha)$, and hence $CP_{\geq 1}(\alpha \mid \alpha) \in \overline{T}$. Thus, $\mu([\alpha], [\alpha]) = 1$. Axioms guarantee that for every classical formula β , such that it is not a propositional contradiction, i.e. $[\beta] \neq \emptyset$, $\mu(\cdot, [\beta])$ is a finitely additive probability on H . In other words, for all α and γ we have:

1. $\mu(w)([\alpha], [\beta]) \geq 0$,
2. $\mu(w)([\alpha], [\beta]) = 1 - \mu(w)([\neg\alpha], [\beta])$, and
3. $\mu(w)([\alpha] \cup [\gamma], [\beta]) = \mu(w)([\alpha], [\beta]) + \mu(w)([\gamma], [\beta])$, for all sentence α and γ such that $[\alpha] \cap [\gamma] = \emptyset$.

Also, for all α, β, γ we have: $\mu(w)([\alpha] \cap [\beta], [\gamma]) = m(w)([\beta], [\gamma]) \cdot \mu(w)([\alpha], [\beta] \cap [\gamma])$. Let $\mu(w)([\beta], [\gamma]) = s$, $\mu(w)([\alpha], [\beta] \cap [\gamma]) = \mu(w)([\alpha], [\beta \wedge \gamma]) = t$, $s > 0$ and $t > 0$. If $s' \in [0, s]$ i $t' \in [0, t]$, then $CP_{\geq s'}(\beta \mid \gamma) \in w$ and $CP_{\geq t'}(\alpha \mid \beta \wedge \gamma) \in w$, and hence, by Axiom 8, $CP_{\geq s' \cdot t'}(\alpha \wedge \beta \mid \gamma) \in w$. Thus, $s \cdot t \leq \sup\{r : r \in CP_{\geq r}(\alpha \wedge \beta \mid \gamma) \in w\} = r_0$. If $s \cdot t < r_0$, then there is s' such that $s < s' < \frac{r_0}{t}$, and $CP_{< s'}(\alpha \mid \beta \wedge \gamma)$. By Rule 3 there is t_0 such that $CP_{< s' \cdot t_0}(\alpha \wedge \beta \mid \gamma) \wedge CP_{\geq t_0}(\beta \mid \gamma)$. From $CP_{\geq t_0}(\beta \mid \gamma)$, we obtain $t_0 \leq t$, and $s < s' < \frac{r_0}{t} \leq \frac{r_0}{t_0}$, i.e. $s' \cdot t_0 < r_0$. Thus $CP_{\geq s' \cdot t_0}(\alpha \wedge \beta \mid \gamma)$, a contradiction, and $\mathbf{M}$ is an $LPCP_{\text{Meas}}$ -model.

Finally, we have to prove that for every formula Φ , $\mathbf{M} \models \Phi$ iff $\Phi \in \overline{T}$. If $\Phi \in \text{For}_{LPCP}^C$ and $\Phi \in \overline{T}$, then certainly $\Phi \in \text{Con}^C(T)$, and for every $w \in W$, $w \models \Phi$, i.e. $\mathbf{M} \models \Phi$. If $\mathbf{M} \models \Phi$, then by the completeness of classical propositional logic $\Phi \in \text{Con}^C(T)$, and $\Phi \in \overline{T}$. Let $\Phi = CP_{\geq s}(\alpha \mid \beta)$. Suppose $CP_{\geq s}(\alpha \mid \beta) \in \overline{T}$. If $\Phi \in \overline{T}$, $\sup\{r : CP_{\geq r}(\psi \mid \theta) \in \overline{T}\} = \mu([\alpha], [\beta]) \geq s$, and $\mathbf{M} \models CP_{\geq s}(\psi \mid \theta)$. For the other direction, suppose that $\mathbf{M} \models CP_{\geq s}(\psi \mid \theta)$, i.e., that $\sup\{r : CP_{\geq r}(\psi \mid \theta) \in \overline{T}\} \geq s$. If $\mu([\alpha], [\beta]) > s$, then, by the well known property of supremum and monotonicity of μ , $CP_{\geq s}(\alpha \mid \beta) \in \overline{T}$. If $\mu([\alpha], [\beta]) = s$, then by Theorem 4.6, $CP_{\geq s}(\alpha \mid \beta) \in \overline{T}$. $\square$

6 Decidability

Since the satisfiability problem for classical propositional formulas is decidable, to prove decidability of our logic it is enough to show that the same holds for the satisfiability problem for probability formulas.

Let A be a probability formula and $p_1, \dots, p_n$ be a list of all propositional letters from A . An atom a of A is a formula $\pm p_1 \wedge \dots \wedge \pm p_n$, where $\pm p_i$ is either p_i , or $\neg p_i$. For different atoms a_i and a_j we have $\vdash a_i \rightarrow \neg a_j$. It is easy to show that A can be equivalently transformed to $DNF(A) = \bigvee_{i=1}^m \bigwedge_{j=1}^{k_i} CP^{i,j}(\alpha_{i,j} \mid \beta_{i,j})$ called a disjunctive normal form of A , where $CP^{i,j}$ is either $CP_{\geq r_{i,j}}$, or $CP_{< r_{i,j}}$. Obviously, A is satisfiable iff at least one of the disjuncts $D_i = \bigwedge_{j=1}^{k_i} CP^{i,j}(\alpha_{i,j} \mid \beta_{i,j})$ is satisfiable. However, that can be also seen as the problem of checking coherence of conditional probability assessments which is decidable [2]. Thus, we have:

Theorem 6. *The logic LPCP is decidable.*

7 Conclusions

We have presented a probability logic which is suitable for reasoning about conditional probability in the sense of de Finetti. It might be applied, for example, to default reasoning which can be formalized using conditional probabilities [7, 14, 24]. From that point of view, it will be useful to find an efficient algorithm for deciding satisfiability of formulas.

In the language of *LPCP* it is possible to define an inconsistent infinite set of formulas, every finite subset of which is consistent (e.g., $\{\neg CP_{>0}(\alpha, \beta)\} \cup \{CP_{<1/n}(\alpha, \beta) : n \text{ is a positive integer}\}$). Thus, the compactness theorem 'if every finite subset of a set T of formulas is satisfiable, then T is satisfiable' does not hold. Since that theorem follows easily from the extended completeness theorem, if one has a finitary axiomatization, providing an infinitary axiomatization is the only way to obtain the corresponding extended completeness theorem. As a consequence, any finitary axiomatization of conditional probability logic (in fact, all the other conditional probability logics that can be found in the literature are of that type) suffers from the serious logical problem that there must be unsatisfiable sets of formulas that are consistent with respect to that axiomatization.

Our approach can be seen as a counterpart of the fuzzy probability logic from [15], where the probability of the conditional event " φ given ψ " is considered as the truth-value of the fuzzy modal proposition $P(\varphi \mid \psi)$ which is read as " $\varphi \mid \psi$ is probable". Then, a fuzzy logic is built as a combination of the well-known Lukasiewicz and Product fuzzy logics. It includes Product implication $\rightarrow_\Pi$, Product conjunction $\oplus$ and Lukasiewicz implication $\rightarrow_L$. The obtained language allows statements like 'the conditional probability is equal to s (at least s , at most s , positive)' and 'the conditional event $\alpha_1 \mid \beta_1$ is at least as probable as the conditional event $\alpha_2 \mid \beta_2$ '. The former statements are directly

expressible in our logic, while the latter one can be modelled by a new operator $\triangleright(\alpha_1 \mid \beta_1, \alpha_2 \mid \beta_2)$ which is definable in our language by an infinite set of axioms:

$$\triangleright(\alpha_1 \mid \beta_1, \alpha_2 \mid \beta_2) \rightarrow (CP_{\geq s}(\alpha_2 \mid \beta_2) \rightarrow CP_{\geq s}(\alpha_1 \mid \beta_1)), \text{ for every } s,$$

and an infinitary inference rule:

$$\text{from } \{CP_{\geq s}(\alpha_2 \mid \beta_2) \rightarrow CP_{\geq s}(\alpha_1 \mid \beta_1)\}, \text{ for every } s, \text{ infer } \triangleright(\alpha_1 \mid \beta_1, \alpha_2 \mid \beta_2).$$

A logic related to [15] can be found in [6]. The authors concerns conditional probability in the framework of the fuzzy logic using non-standard probabilities, which is the well-known solution to overcome difficulties concerning conditional probability when dealing with zero probabilities. In that approach only the impossible event can take on probability 0, but non-impossible events can have an infinitesimal probability. Similar approach, starting from classical logic, is given in [24, 25].

One of the logics from [4] corresponds to the standard, Kolmogorov's, approach to conditional probability. In order to model conditional probabilities, the authors introduce so-called polynomial weight formulas. For example, $2w(\varphi_2 \mid \varphi_1) + w(\varphi_1 \mid \varphi_2) \geq 1$ is expressed as $2w(\varphi_1 \wedge \varphi_2) \cdot w(\varphi_2) + w(\varphi_1 \wedge \varphi_2) \cdot w(\varphi_1) \geq w(\varphi_1) \cdot w(\varphi_2)$, where $w(\varphi_2 \mid \varphi_1)$ denotes the conditional probability of φ_2 given φ_1 . In order to prove completeness and decidability of the logic, the theory of real closed field is introduced. In our approach, since the parts of field theory are moved to the meta theory, the axioms are rather simple. It would be interesting to see how the polynomial weight formulas can be included in *LPCP*. Also, an open problem is to formalize conditional probabilities in the sense of Kolmogorov by following our approach in axiomatization.

References

1. G. Boole, An investigation of the laws of thought on which are founded the mathematical theories of logic and probabilities, Walton and Maberley, London, 1854.
2. S. Coletti, R. Scozzafava, Probabilistic logic in a coherent setting, Kluwer Academic Press, Dordrecht, The Netherlands, 2002.
3. B. de Finetti, Theory of probability, Volume 1, John Wiley and Sons, 1974.
4. R. Fagin, J. Halpern, N. Megiddo, A logic for reasoning about probabilities, Information and Computation, vol. 87, no. 1/2, 78 – 128, 1990.
5. R. Fagin, J. Halpern, Reasoning about knowledge and probability, Journal of the ACM, vol. 41, no. 2, 340 – 367, 1994.
6. T. Flaminio, F. Montagna, A logical and algebraic treatment of conditional probability, in Proceedings of IPMU '04, (Edts. L.A. Zadeh), Perugia, Italy, July, 4-9, 2004, 493–500, 2004
7. A. Gilio, Probabilistic reasoning under coherence in System P^* , Annals of Mathematics and Artificial Intelligence 34: 5-34, 2002.
8. J. Hawthorne, On the logic of nonmonotonic conditionals and conditional probabilities, Journal of Philosophical Logic, v. 25, no. 1, 185–218, 1996.
9. J. Hawthorne, On the logic of nonmonotonic conditionals and conditional probabilities: Predicate logic, Journal of Philosophical Logic, v. 27, no. 1, 1–34, 1998.

10. D. N. Hoover, Probability logic, *Annals of mathematical logic* 14, 287 – 313, 1978.
11. H.J. Keisler, Hyperfinite model theory, in: *Logic Colloquium '76*, North-Holland, Amsterdam, 5 – 110, 1977.
12. H.J. Keisler, Probability quantifiers, in: *Model-theoretic logics*. eds. J. Barwise, S. Feferman, *Perspectives in Mathematical Logic*, Springer-Verlag, Berlin, 509 – 556, 1985.
13. A. N. Kolmogorov, *Grundbegriffe der Wahrscheinlichkeitrechnung*, *Ergebnisse Der Mathematik*, 1933; translated as *Foundations of Probability*, Chelsea Publishing Company, 1950.
14. T. Lukasiewicz, Probabilistic Default Reasoning with Conditional Constraints, *Annals of Mathematics and Artificial Intelligence* 34, 35-88, 2002.
15. E. Marchioni, L. Godo, A Logic for Reasoning about Coherent Conditional Probability: A Modal Fuzzy Logic Approach, *JELIA 2004*, *Lecture notes in artificial intelligence (LNCS/LNAI)* 3229, 213–225, 2004.
16. Z. Marković, Z. Ognjanović, M. Rašković, A Probabilistic Extension of Intuitionistic Logic, *Mathematical Logic Quarterly*, vol 49, no 5, 415–424, 2003.
17. N. Nilsson, Probabilistic logic, *Artificial intelligence*, no. 28, 71–87, 1986.
18. N. Nilsson, Probabilistic logic revised, *Artificial intelligence*, no. 59, 1993, 39–42.
19. Z. Ognjanović, M. Rašković, Some probability logics with new types of probability operators, *Journal of logic and Computation*, Volume 9, Issue 2, 181–195, 1999.
20. Z. Ognjanović, M. Rašković, Some first order probability logics, *Theoretical Computer Science*, Vol. 247, No. 1-2, 191 – 212, 2000.
21. M. Rašković, Completeness theorem for biprobability models, *Journal of Symbolic Logic* 51, 586–590, 1986.
22. M. Rašković, Classical logic with some probability operators, *Publication de l'Institut Math. (NS)* vol 53 (67), 1993, 1-3.
23. M. Rašković, R. Đorđević, Probability quantifiers and operators, *Vesta*, Beograd, 1996.
24. M. Rašković, Z. Ognjanović, Z. Marković, A probabilistic Approach to Default Reasoning, *10th International Workshop on Non-Monotonic Reasoning NMR2004*, Westin Whistler, Canada, June 6-8, 335–341, 2004.
25. M. Rašković, Z. Ognjanović, Z. Marković, A Logic with Conditional Probabilities, *9th European conference JELIA'04 Logics in Artificial Intelligence*, *Lecture notes in artificial intelligence (LNCS/LNAI)* 3229, 226-238, 2004.

Probabilistic Description Logic Programs

Thomas Lukasiewicz*

Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”,
Via Salaria 113, I-00198 Rome, Italy
lukasiewicz@dis.uniroma1.it

Abstract. Towards sophisticated representation and reasoning techniques that allow for probabilistic uncertainty in the Rules, Logic, and Proof layers of the Semantic Web, we present probabilistic description logic programs (or pdl-programs), which are a combination of description logic programs (or dl-programs) under the answer set semantics and the well-founded semantics with Poole’s independent choice logic. We show that query processing in such pdl-programs can be reduced to computing all answer sets of dl-programs and solving linear optimization problems, and to computing the well-founded model of dl-programs, respectively. Furthermore, we show that the answer set semantics of pdl-programs is a refinement of the well-founded semantics of pdl-programs.

1 Introduction

The *Semantic Web* initiative [5, 11] aims at an extension of the current World Wide Web by standards and technologies that help machines to understand the information on the Web so that they can support richer discovery, data integration, navigation, and automation of tasks. The main ideas behind it are to add a machine-readable meaning to Web pages, to use ontologies for a precise definition of shared terms in Web resources, to make use of KR technology for automated reasoning from Web resources, and to apply cooperative agent technology for processing the information of the Web.

The Semantic Web consists of several hierarchical layers, where the *Ontology layer*, in form of the *OWL Web Ontology Language* [30, 18] (recommended by the W3C), is currently the highest layer of sufficient maturity. OWL consists of three increasingly expressive sublanguages, namely *OWL Lite*, *OWL DL*, and *OWL Full*. OWL Lite and OWL DL are essentially very expressive description logics with an RDF syntax [18]. As shown in [16], ontology entailment in OWL Lite (resp., OWL DL) reduces to knowledge base (un)satisfiability in the description logic $SHIF(\mathbf{D})$ (resp., $SHOIN(\mathbf{D})$). On top of the Ontology layer, the *Rules*, *Logic*, and *Proof layers* of the Semantic Web will be developed next, which should offer sophisticated representation and reasoning capabilities. As a first effort in this direction, *RuleML* (Rule Markup Language) [6] is an XML-based markup language for rules and rule-based systems, whereas the OWL Rules Language [17] is a first proposal for extending OWL by Horn clause rules.

* Alternate address: Institut für Informationssysteme, Technische Universität Wien, Favoritenstraße 9-11, A-1040 Vienna, Austria; e-mail: lukasiewicz@kr.tuwien.ac.at.

A key requirement of the layered architecture of the Semantic Web is to integrate the Rules and the Ontology layer. In particular, it is crucial to allow for building rules on top of ontologies, that is, for rule-based systems that use vocabulary from ontology knowledge bases. Another type of combination is to build ontologies on top of rules, which means that ontological definitions are supplemented by rules or imported from rules. Towards this goal, the works [9, 10] have proposed *description logic programs* (or simply *dl-programs*), which are of the form $KB = (L, P)$, where L is a knowledge base in a description logic and P is a finite set of description logic rules (or simply *dl-rules*). Such dl-rules are similar to usual rules in logic programs with negation as failure, but may also contain *queries to L* in their bodies, which are given by special atoms (on which possibly default negation may apply). Another important feature of dl-rules is that queries to L also allow for specifying an input from P , and thus for a *flow of information from P to L* , besides the flow of information from L to P , given by any query to L . Hence, description logic programs allow for building rules on top of ontologies, but also (to some extent) building ontologies on top of rules. In this way, additional knowledge (gained in the program) can be supplied to L before querying. The semantics of dl-programs was defined in [9] and [10] as an extension of the answer set semantics by Gelfond and Lifschitz [13] and the well-founded semantics by Van Gelder, Ross, and Schlipf [29], respectively, which are the two most widely used semantics for non-monotonic logic programs. The description logic knowledge bases in dl-programs are specified in the well-known description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$.

In this paper, we continue this line of research. Towards sophisticated representation and reasoning techniques that also allow for modeling probabilistic uncertainty in the *Rules, Logic, and Proof layers* of the Semantic Web, we present *probabilistic description logic programs* (or simply *pdl-programs*), which generalize dl-programs under the answer set and the well-founded semantics by probabilistic uncertainty. This probabilistic generalization of dl-programs is developed as a combination of dl-programs with Poole's independent choice logic (ICL) [23]. Poole's ICL is a powerful (single- and also multi-agent) representation formalism for combining logic and probability, which can represent influence diagrams, Bayesian networks, Markov decision processes, and normal form games [23], and which also allows for natural notions of causes and explanations as in Pearl's structural causal models [12].

To my knowledge, this is the first work that combines description logic programs with probabilistic uncertainty. Related work by Straccia [26] combines description logic programs with *non-probabilistic uncertainty* using interval annotations. In [22], Nottelmann and Fuhr present pDAML+OIL, which is a *probabilistic generalization of the ontology language DAML+OIL*, and a mapping to stratified probabilistic datalog. In [14], Giugno and Lukasiewicz present a probabilistic generalization of the expressive description logic $\mathcal{SHOQ}(\mathbf{D})$ standing behind DAML+OIL. Related work on description logic programs (without uncertainty) is discussed in Section 4.

The main contributions of this paper can be summarized as follows:

- We present probabilistic description logic programs (or pdl-programs), which are a probabilistic generalization of dl-programs [9, 10]. They are obtained by combining dl-programs with Poole's independent choice logic (ICL) [23].

- We define a probabilistic answer set semantics of pdl-programs, which is a generalization of the (strong) answer set semantics of dl-programs in [9]. We show that query processing in pdl-programs under this semantics can be reduced to computing all answer sets of dl-programs and solving linear optimization problems.
- We define a probabilistic well-founded semantics of pdl-programs, which is a generalization of the well-founded semantics of dl-programs in [10]. We then show that query processing in pdl-programs under the well-founded semantics can be reduced to computing the well-founded semantics of dl-programs.
- We show that, like for the case of dl-programs, the answer set semantics of pdl-programs is a refinement of the well-founded semantics of pdl-programs. That is, whenever an answer to a query under the well-founded semantics is defined, it coincides with the answer to the query under the answer set semantics.

2 Preliminaries

In this section, we first recall the description logics $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$. We then recall *description logic programs* (or *dl-programs*) under the answer set and well-founded semantics [9, 10], which combine description logics and normal programs. They consist of a knowledge base L in a description logic and a finite set of description logic rules P . Such rules are similar to usual rules in logic programs with negation as failure, but may also contain *queries* to L , possibly default negated.

$\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$. We first describe $\mathcal{SHOIN}(\mathbf{D})$. We assume a set $\mathbf{D}$ of *elementary datatypes*. Each $d \in \mathbf{D}$ has a set of *data values*, called the *domain* of d , denoted $\text{dom}(d)$. Let $\text{dom}(\mathbf{D}) = \bigcup_{d \in \mathbf{D}} \text{dom}(d)$. A *datatype* is either an element of $\mathbf{D}$ or a subset of $\text{dom}(\mathbf{D})$ (called *datatype oneOf*). Let $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, and $\mathbf{I}$ be nonempty finite pairwise disjoint sets of *atomic concepts*, *abstract roles*, *datatype roles*, and *individuals*, respectively. Let $\mathbf{R}_A^-$ denote the set of all inverses R^- of abstract roles $R \in \mathbf{R}_A$.

A *role* is an element of $\mathbf{R}_A \cup \mathbf{R}_A^- \cup \mathbf{R}_D$. *Concepts* are inductively defined as follows. Every $C \in \mathbf{A}$ is a concept, and if $o_1, o_2, \dots \in \mathbf{I}$, then $\{o_1, o_2, \dots\}$ is a concept (called *oneOf*). If C and D are concepts and if $R \in \mathbf{R}_A \cup \mathbf{R}_A^-$, then $(C \sqcap D)$, $(C \sqcup D)$, and $\neg C$ are concepts (called *conjunction*, *disjunction*, and *negation*, respectively), as well as $\exists R.C$, $\forall R.C$, $\geq nR$, and $\leq nR$ (called *exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geq 0$. If $d \in \mathbf{D}$ and $U \in \mathbf{R}_D$, then $\exists U.d$, $\forall U.d$, $\geq nU$, and $\leq nU$ are concepts (called *datatype exists*, *value*, *atleast*, and *atmost restriction*, respectively) for an integer $n \geq 0$. We write $\top$ and $\perp$ to abbreviate $C \sqcup \neg C$ and $C \sqcap \neg C$, respectively, and we eliminate parentheses as usual.

An *axiom* is of one of the following forms: (1) $C \sqsubseteq D$, where C and D are concepts (*concept inclusion*); (2) $R \sqsubseteq S$, where either $R, S \in \mathbf{R}_A$ or $R, S \in \mathbf{R}_D$ (*role inclusion*); (3) $\text{Trans}(R)$, where $R \in \mathbf{R}_A$ (*transitivity*); (4) $C(a)$, where C is a concept and $a \in \mathbf{I}$ (*concept membership*); (5) $R(a, b)$ (resp., $U(a, v)$), where $R \in \mathbf{R}_A$ (resp., $U \in \mathbf{R}_D$) and $a, b \in \mathbf{I}$ (resp., $a \in \mathbf{I}$ and $v \in \text{dom}(\mathbf{D})$) (*role membership*); and (6) $a = b$ (resp., $a \neq b$), where $a, b \in \mathbf{I}$ (*equality* (resp., *inequality*)). A *knowledge base* L is a finite set of axioms. For decidability, number restrictions in L are restricted to simple $R \in \mathbf{R}_A$ [19].

The syntax of $\mathcal{SHIF}(\mathbf{D})$ is as the above syntax of $\mathcal{SHOIN}(\mathbf{D})$, but without the oneOf constructor and with the *atleast* and *atmost* constructors limited to 0 and 1.

For the semantics of $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, we refer the reader to [16].

Example 2.1. An online store (such as *amazon.com*) may use a description logic knowledge base to classify and characterize its products. For example, suppose that (1) textbooks are books, (2) personal computers and cameras are electronic products, (3) books and electronic products are products, (4) every product has at least one related product, (5) only products are related to each other, (6) *tb_ai* and *tb_hp* are textbooks, which are related to each other, (7) *pc_ibm* and *pc_hp* are personal computers, which are related to each other, and (8) *ibm* and *hp* are providers for *pc_ibm* and *pc_hp*, respectively. This knowledge is expressed by the following description logic knowledge base L_1 :

Textbook $\sqsubseteq$ *Book*; *PC* $\sqcup$ *Camera* $\sqsubseteq$ *Electronics*; *Book* $\sqcup$ *Electronics* $\sqsubseteq$ *Product*;
Product $\sqsubseteq$ ≥ 1 *related*; ≥ 1 *related* $\sqcup$ ≥ 1 *related*⁻ $\sqsubseteq$ *Product*;
Textbook(*tb_ai*); *Textbook*(*tb_hp*); *PC*(*pc_ibm*); *PC*(*pc_hp*);
related(*tb_ai*, *tb_hp*); *related*(*pc_ibm*, *pc_hp*);
provides(*ibm*, *pc_ibm*); *provides*(*hp*, *pc_hp*).

Syntax of Description Logic Programs. We assume a function-free first-order vocabulary Φ with nonempty finite sets of constant and predicate symbols, and a set $\mathcal{X}$ of variables. A *term* is a constant symbol from Φ or a variable from $\mathcal{X}$. If p is a predicate symbol of arity $k \geq 0$ from Φ and $t_1, \dots, t_k$ are terms, then $p(t_1, \dots, t_k)$ is an *atom*. A *classical literal* (or *literal*) l is an atom a or a negated atom $\neg a$. A *negation-as-failure (NAF) literal* is an atom a or a default-negated atom *not* a . A *normal rule* r is of form

$$a \leftarrow b_1, \dots, b_k, \text{not } b_{k+1}, \dots, \text{not } b_m, \quad m \geq k \geq 0, \quad (1)$$

where $a, b_1, \dots, b_m$ are atoms. We refer to a as the *head* of r , denoted $H(r)$, while the conjunction $b_1, \dots, b_k, \text{not } b_{k+1}, \dots, \text{not } b_m$ is the *body* of r ; its *positive* (resp., *negative*) part is $b_1, \dots, b_k$ (resp., $\text{not } b_{k+1}, \dots, \text{not } b_m$). We define $B(r) = B^+(r) \cup B^-(r)$, where $B^+(r) = \{b_1, \dots, b_k\}$ and $B^-(r) = \{b_{k+1}, \dots, b_m\}$. A *normal program* P is a finite set of normal rules. We say that P is *positive* iff no rule in P contains default-negated atoms. Informally, a dl-program consists of a description logic knowledge base L and a generalized normal program P , which may contain queries to L . In such a query, it is asked whether a certain description logic axiom or its negation logically follows from L or not. Formally, a *dl-query* $Q(\mathbf{t})$ is either

- (a) a concept inclusion axiom F or its negation $\neg F$; or
- (b) of the forms $C(\mathbf{t})$ or $\neg C(\mathbf{t})$, where C is a concept and $\mathbf{t}$ is a term; or
- (c) of the forms $R(\mathbf{t}_1, \mathbf{t}_2)$ or $\neg R(\mathbf{t}_1, \mathbf{t}_2)$, where R is a role and $\mathbf{t}_1, \mathbf{t}_2$ are terms.

A *dl-atom* has the form $DL[S_1 op_1 p_1, \dots, S_m op_m p_m; Q](\mathbf{t})$, where each S_i is a concept or role, $op_i \in \{\sqcup, \sqcup\}$, p_i is a unary resp. binary predicate symbol, $Q(\mathbf{t})$ is a dl-query, and $m \geq 0$. We call $p_1, \dots, p_m$ its *input predicate symbols*. Intuitively, $op_i = \sqcup$ (resp., $op_i = \sqcup$) increases S_i (resp., $\neg S_i$) by the extension of p_i . A *dl-rule* r is of form (1), where any $b \in B(r)$ may be a dl-atom. A *dl-program* $KB = (L, P)$ consists of a description logic knowledge base L and a finite set of dl-rules P . We say that

$KB = (L, P)$ is *positive* iff P is positive. *Ground terms, atoms, literals*, etc., are defined as usual. The *Herbrand base* of P , denoted HB_P , is the set of all ground atoms with standard predicate symbols that occur in P and constant symbols in Φ . We use $ground(P)$ to denote the set of all ground instances of dl-rules in P relative to HB_P .

Example 2.2. Consider the dl-program $KB_1 = (L_1, P_1)$, where L_1 is the description logic knowledge base from Example 2.1, and P_1 is the following set of dl-rules:

- (1) $pc(pc.1); pc(pc.2); pc(pc.3);$
- (2) $brand\_new(pc.1); brand\_new(pc.2);$
- (3) $vendor(dell, pc.1); vendor(dell, pc.2); vendor(dell, pc.3);$
- (4) $avoid(X) \leftarrow DL[Camera](X), not offer(X);$
- (5) $offer(X) \leftarrow DL[PC \sqcup pc; Electronics](X), not brand\_new(X);$
- (6) $provider(P) \leftarrow vendor(P, X), DL[PC \sqcup pc; Product](X);$
- (7) $provider(P) \leftarrow DL[provides](P, X), DL[PC \sqcup pc; Product](X);$
- (8) $similar(X, Y) \leftarrow DL[related](X, Y);$
- (9) $similar(X, Z) \leftarrow similar(X, Y), similar(Y, Z).$

The above dl-rules express that (1) $pc.1$, $pc.2$, and $pc.3$ are additional personal computers, (2) $pc.1$ and $pc.2$ are brand new, (3) $dell$ is the vendor of $pc.1$, $pc.2$, and $pc.3$, (4) a customer avoids all cameras that are not on offer, (5) all electronic products that are not brand new are on offer, (6) every vendor of a product is a provider, (7) every entity providing a product is a provider, (8) all related products are similar, and (9) the binary similarity relation on products is transitively closed.

Answer Set Semantics. In the sequel, let $KB = (L, P)$ be a dl-program. An *interpretation* I relative to P is any $I \subseteq HB_P$. We say that I is a *model* of $a \in HB_P$ under L , denoted $I \models_L a$, iff $a \in I$. We say that I is a *model* of a ground dl-atom $a = DL[S_1 op_1 p_1, \dots, S_m op_m p_m; Q](c)$ under L , denoted $I \models_L a$, iff $L \cup \bigcup_{i=1}^m A_i(I) \models Q(c)$, where $A_i(I) = \{S_i(e) \mid p_i(e) \in I\}$, for $op_i = \sqcup$; and $A_i(I) = \{\neg S_i(e) \mid p_i(e) \in I\}$, for $op_i = \sqcup$. A ground dl-atom a is *monotonic* relative to $KB = (L, P)$ iff $I \subseteq I' \subseteq HB_P$ implies that if $I \models_L a$ then $I' \models_L a$. In this paper, we consider only monotonic ground dl-atoms, but observe that one can also define dl-atoms that are not monotonic; see [9]. We say that I is a *model* of a ground dl-rule r iff $I \models_L H(r)$ whenever $I \models_L B(r)$, that is, $I \models_L a$ for all $a \in B^+(r)$ and $I \not\models_L a$ for all $a \in B^-(r)$. We say that I is a *model* of a dl-program $KB = (L, P)$, denoted $I \models KB$, iff $I \models_L r$ for all $r \in ground(P)$. We say that KB is *satisfiable* (resp., *unsatisfiable*) iff it has some (resp., no) model.

Like ordinary positive programs, every positive dl-program KB is satisfiable and has a unique least model that naturally characterizes its semantics. The *answer set semantics* of general dl-programs is then defined by a reduction to the least model semantics of positive ones as follows, using a transformation that removes all default-negated atoms in dl-rules and that generalizes the Gelfond-Lifschitz transformation [13]. For dl-programs $KB = (L, P)$, the *strong dl-transform* of P relative to L and an interpretation $I \subseteq HB_P$, denoted sP_L^I , is the set of all dl-rules obtained from $ground(P)$ by (i) deleting every dl-rule r such that $I \models_L a$ for some $a \in B^-(r)$, and (ii) deleting from each remaining dl-rule r the negative body. A (*strong*) *answer set* of KB is an interpretation $I \subseteq HB_P$ such that I is the unique least model of (L, sP_L^I) .

The answer set semantics of dl-programs $KB = (L, P)$ without dl-atoms coincides with the ordinary answer set semantics of P . Answer sets of a general dl-program KB are also minimal models of KB . Finally, positive and locally stratified dl-programs have exactly one answer set, which coincides with their canonical minimal model.

Well-Founded Semantics. In the sequel, let $KB = (L, P)$ be a dl-program. For literals $l = a$ (resp., $l = \neg a$), we use $\neg.l$ to denote $\neg a$ (resp., a), and for sets of literals S , we define $\neg.S = \{\neg.l \mid l \in S\}$ and $S^+ = \{a \in S \mid a \text{ is an atom}\}$. We define $Lit_P = HB_P \cup \neg.HB_P$. A set $S \subseteq Lit_P$ is *consistent* iff $S \cap \neg.S = \emptyset$. A *three-valued interpretation* relative to P is any consistent $I \subseteq Lit_P$. We define the well-founded semantics of KB by generalizing its standard definition based on unfounded sets [29].

We first define the notion of an unfounded set for dl-programs. Let $I \subseteq Lit_P$ be consistent. A set $U \subseteq HB_P$ is an *unfounded set* of KB w.r.t. I iff the following holds:

- (*) for every $a \in U$ and every $r \in \text{ground}(P)$ with $H(r) = a$, either (i) $\neg b \in I \cup \neg.U$ for some ordinary atom $b \in B^+(r)$, or (ii) $b \in I$ for some ordinary atom $b \in B^-(r)$, or (iii) for some dl-atom $b \in B^+(r)$, it holds that $S^+ \not\models_L b$ for every consistent $S \subseteq Lit_P$ with $I \cup \neg.U \subseteq S$, or (iv) $I^+ \models_L b$ for some dl-atom $b \in B^-(r)$.

For each dl-program KB and consistent $I \subseteq Lit_P$, the set of unfounded sets of KB relative to I is closed under union, and thus KB has a greatest unfounded set relative to I , denoted $U_{KB}(I)$. Intuitively, if I is compatible with KB , then all atoms in $U_{KB}(I)$ can be safely switched to false and the resulting interpretation is still compatible with KB . We define the operators T_{KB} and W_{KB} on all consistent $I \subseteq Lit_P$ by:

- $a \in T_{KB}(I)$ iff $a \in HB_P$ and some $r \in \text{ground}(P)$ exists such that (a) $H(r) = a$, (b) $I^+ \models_L b$ for all $b \in B^+(r)$, (c) $\neg b \in I$ for all ordinary atoms $b \in B^-(r)$, and (d) $S^+ \not\models_L b$ for each consistent $S \subseteq Lit_P$ with $I \subseteq S$, for all dl-atoms $b \in B^-(r)$;
- $W_{KB}(I) = T_{KB}(I) \cup \neg.U_{KB}(I)$.

The operators T_{KB} , U_{KB} , and W_{KB} are all monotonic. Thus, in particular, W_{KB} has a least fixpoint, denoted $\text{lfp}(W_{KB})$. The *well-founded semantics* of a dl-program $KB = (L, P)$, denoted $WFS(KB)$, is defined as $\text{lfp}(W_{KB})$. An atom $a \in HB_P$ is *well-founded* (resp., *unfounded*) relative to KB iff a (resp., $\neg a$) belongs to $WFS(KB)$. Intuitively, starting with $I = \emptyset$, rules are applied to obtain new positive (resp., negated) facts via $T_{KB}(I)$ (resp., $\neg.U_{KB}(I)$). This process is repeated until no longer possible.

The well-founded semantics of dl-programs $KB = (L, P)$ without dl-atoms coincides with the ordinary well-founded semantics of P . In the general case, $WFS(KB)$ is a partial model of KB . Here, a consistent $I \subseteq Lit_P$ is a *partial model* of KB iff it can be extended to a (two-valued) model $I' \subseteq HB_P$ of KB . Like in the case of ordinary normal programs, the well-founded semantics for positive and locally stratified dl-programs is total and coincides with their canonical minimal model. The well-founded semantics for dl-programs also approximates their answer set semantics. That is, every well-founded (resp., unfounded) atom $a \in HB_P$ is true (resp., false) in every answer set.

3 Probabilistic Description Logic Programs

In this section, we define probabilistic dl-programs as a combination of dl-programs with Poole's independent choice logic (ICL) [23]. Poole's ICL is based on ordinary acyclic logic programs under different "choices", where each choice along with an acyclic logic program produces a first-order model, and one then obtains a probability distribution over the set of first-order models by placing a distribution over the different choices. Here, we use dl-programs under the well-founded and the answer set semantics, instead of the above ordinary acyclic logic programs under their canonical model semantics (which coincides with their stable model and their answer set semantics).

Syntax. We assume a function-free first-order vocabulary Φ with nonempty finite sets of constant and predicate symbols, and a set of variables $\mathcal{X}$, as in Section 2. We use HB_Φ (resp., HU_Φ) to denote the Herbrand base (resp., universe) over Φ . In the sequel, we assume that HB_Φ is nonempty. We define *classical formulas* by induction as follows. The propositional constants *false* and *true*, denoted $\perp$ and $\top$, respectively, and all atoms are classical formulas. If ϕ and ψ are classical formulas, then also $\neg\phi$ and $(\phi \wedge \psi)$. A *conditional constraint* is of the form $(\psi|\phi)[l, u]$ with reals $l, u \in [0, 1]$ and classical formulas ϕ and ψ . We define *probabilistic formulas* inductively as follows. Every conditional constraint is a probabilistic formula. If F and G are probabilistic formulas, then also $\neg F$ and $(F \wedge G)$. We use $(F \vee G)$, $(F \Leftarrow G)$, and $(F \Leftrightarrow G)$ to abbreviate $\neg(\neg F \wedge \neg G)$, $\neg(\neg F \wedge G)$, and $(\neg(\neg F \wedge G) \wedge \neg(F \wedge \neg G))$, respectively, and adopt the usual conventions to eliminate parentheses. *Ground terms*, *ground formulas*, *substitutions*, and *ground instances* of probabilistic formulas are defined as usual.

A *choice space* C is a set of pairwise disjoint and nonempty sets $A \subseteq HB_\Phi$. Any member $A \in C$ is called an *alternative* of C and any element $a \in A$ an *atomic choice* of C . A *total choice* of C is a set $B \subseteq HB_\Phi$ such that $|B \cap A| = 1$ for all $A \in C$. A *probability* μ on a choice space C is a probability function on the set of all total choices of C . Since C and all its alternatives are finite, μ can be defined by (i) a mapping $\mu: \bigcup C \rightarrow [0, 1]$ such that $\sum_{a \in A} \mu(a) = 1$ for all $A \in C$, and (ii) $\mu(B) = \prod_{b \in B} \mu(b)$ for all total choices B of C . Intuitively, (i) associates a probability with each atomic choice of C , and (ii) assumes independence between the alternatives of C .

A *probabilistic dl-program* (or *pdl-program*) $KB = (L, P, C, \mu)$ consists of a dl-program (L, P) , a choice space C such that (i) $\bigcup C \subseteq HB_P$ and (ii) no atomic choice in C coincides with the head of any dl-rule in $\text{ground}(P)$, and a probability μ on C . A *probabilistic query* to KB has the form $?F$ or the form $?(\beta | \alpha) [L, U]$, where F is a probabilistic formula, β, α are classical formulas, and L, U are variables. The *correct answer* to $?F$ is the set of all substitutions θ such that $F\theta$ is a consequence of KB . The *tight answer* to $?(\beta | \alpha) [L, U]$ is the set of all substitutions θ such that $?(\beta | \alpha) [L, U]\theta$ is a tight consequence of KB . In the following paragraphs, we define the notions of *consequence* and *tight consequence* under the answer set and the well-founded semantics.

Example 3.1. Consider the pdl-program $KB_1 = (L_1, P_1, C_1, \mu_1)$, where L_1 is as in Example 2.1, and P_1 is as in Example 2.2 except that the dl-rules (4) and (5) are replaced by the dl-rules (4') and (5'), respectively, and the dl-rules (10) and (11) are added:

- (4') $avoid(X) \leftarrow DL[Camera](X), not\ offer(X), avoid\_pos;$
- (5') $offer(X) \leftarrow DL[PC \uplus pc; Electronics](X), not\ brand\_new(X), offer\_pos;$
- (10) $buy(C, X) \leftarrow needs(C, X), view(X), not\ avoid(X), v\_buy\_pos;$
- (11) $buy(C, X) \leftarrow needs(C, X), buy(C, Y), also\_buy(Y, X), a\_buy\_pos.$

Furthermore, let C_1 be given by $\{\{avoid\_pos, avoid\_neg\}, \{offer\_pos, offer\_neg\}, \{v\_buy\_pos, v\_buy\_neg\}, \{a\_buy\_pos, a\_buy\_neg\}\}$, and let $\mu_1(avoid\_pos) = 0.9$, $\mu_1(avoid\_neg) = 0.1$, $\mu_1(offer\_pos) = 0.9$, $\mu_1(offer\_neg) = 0.1$, $\mu_1(v\_buy\_pos) = 0.7$, $\mu_1(v\_buy\_neg) = 0.3$, $\mu_1(a\_buy\_pos) = 0.7$, and $\mu_1(a\_buy\_neg) = 0.3$.

Here, the new dl-rules (4') and (5') express that the dl-rules (4) and (5) actually only hold with the probability 0.9. Furthermore, (10) expresses that a customer buys a needed product that is viewed and not avoided with the probability 0.7, while (11) says that a customer buys a needed product x with probability 0.7, if she bought another product y , and every customer that previously had bought y also bought x .

In a probabilistic query, one may ask for the tight probability bounds that a customer c buys a needed product x , if (i) c bought another product y , (ii) every customer that previously had bought y also bought x , (iii) x is not avoided, and (iv) c has been shown product x (the result to this query may, e.g., help to decide whether it is useful to make a customer automatically also view product x when buying y):

$$?(buy(c, x) \mid needs(c, x) \wedge buy(c, y) \wedge also\_buy(y, x) \wedge view(x) \wedge not\ avoid(x))[L, U].$$

Answer Set Semantics. A *total world* I is a subset of HB_Φ . We use $\mathcal{I}_\Phi$ to denote the set of all total worlds over Φ . A *variable assignment* σ maps each variable $X \in \mathcal{X}$ to an element of HU_Φ . It is extended to all terms by $\sigma(c) = c$ for all constant symbols c from Φ . The *truth* of classical formulas ϕ in I under a variable assignment σ , denoted $I \models_\sigma \phi$ (or $I \models \phi$ when ϕ is ground), is inductively defined by:

- $I \models_\sigma p(t_1, \dots, t_k)$ iff $p(\sigma(t_1), \dots, \sigma(t_k)) \in I$;
- $I \models_\sigma \neg\phi$ iff not $I \models_\sigma \phi$; and $I \models_\sigma (\phi \wedge \psi)$ iff $I \models_\sigma \phi$ and $I \models_\sigma \psi$.

A *total probabilistic interpretation* Pr is a probability function on $\mathcal{I}_\Phi$ (that is, since $\mathcal{I}_\Phi$ is finite, a mapping $Pr: \mathcal{I}_\Phi \rightarrow [0, 1]$ such that all $Pr(I)$ with $I \in \mathcal{I}_\Phi$ sum up to 1). The *probability* of a classical formula ϕ in Pr under a variable assignment σ , denoted $Pr_\sigma(\phi)$ (or $Pr(\phi)$ when ϕ is ground), is defined as the sum of all $Pr(I)$ such that $I \in \mathcal{I}_\Phi$ and $I \models_\sigma \phi$. For classical formulas ϕ and ψ with $Pr_\sigma(\phi) > 0$, we use $Pr_\sigma(\psi|\phi)$ to abbreviate $Pr_\sigma(\psi \wedge \phi) / Pr_\sigma(\phi)$. The *truth* of probabilistic formulas F in Pr under a variable assignment σ , denoted $Pr \models_\sigma F$, is inductively defined as follows:

- $Pr \models_\sigma (\psi|\phi)[l, u]$ iff $Pr_\sigma(\phi) = 0$ or $Pr_\sigma(\psi|\phi) \in [l, u]$;
- $Pr \models_\sigma \neg F$ iff not $Pr \models_\sigma F$; and $Pr \models_\sigma (F \wedge G)$ iff $Pr \models_\sigma F$ and $Pr \models_\sigma G$.

A total probabilistic interpretation Pr is a *model* of a probabilistic formula F iff $Pr \models_\sigma F$ for every variable assignment σ . We say that Pr is a *model* of a pdl-program $KB = (L, P, C, \mu)$ iff (i) every total world $I \in \mathcal{I}_\Phi$ with $Pr(I) > 0$ is an answer set of $(L, P \cup \{p \leftarrow \mid p \in B\})$ for some total choice B of C , and (ii) $Pr(\bigwedge B) = Pr(\bigwedge_{p \in B} p) = \mu(B)$ for every total choice B of C . We say that KB is *consistent* iff it has a model Pr . A probabilistic formula F is an *answer set consequence* of KB , denoted $KB \models^{as} F$, iff every model of KB is also a model of F . A conditional constraint $(\psi|\phi)[l, u]$ is

$$\begin{aligned}
\sum_{r \in R, r \models \wedge B} -\mu(B) y_r + \sum_{r \in R, r \models \wedge B} (1 - \mu(B)) y_r &= 0 \quad (\text{for all total choices } B \text{ of } C) \\
\sum_{r \in R, r \models \alpha} y_r &= 1 \\
y_r &\geq 0 \quad (\text{for all } r \in R)
\end{aligned} \tag{2}$$

Fig. 1. System of linear constraints for Theorem 3.1

a *tight answer set consequence* of KB , denoted $KB \Vdash_{tight}^{as}(\psi|\phi)[l, u]$, iff l (resp., u) is the infimum (resp., supremum) of $Pr_\sigma(\psi|\phi)$ subject to all models Pr of KB and all variable assignments σ with $Pr_\sigma(\phi) > 0$. Here, we assume that $l = 1$ and $u = 0$, when $Pr_\sigma(\phi) = 0$ for all models Pr of KB and all variable assignment σ .

Clearly, deciding whether a pdl-program KB is consistent can be reduced to deciding whether dl-programs have an answer set. The following theorem shows that computing tight answers to queries $?(\beta | \alpha) [L, U]$ to KB , where β and α are ground, can be reduced to computing all answer sets of dl-programs and then solving two linear optimization problems. This theorem follows from a standard result on transforming linear fractional programs into equivalent linear programs by Charnes and Cooper.

Theorem 3.1. *Let $KB = (L, P, C, \mu)$ be a consistent pdl-program, and let β and α be ground classical formulas such that $Pr(\alpha) > 0$ for some model Pr of KB . Then, l (resp., u) such that $KB \Vdash_{tight}^{as}(\beta|\alpha)[l, u]$ is given by the optimal value of the following linear program over the variables y_r ($r \in R$), where R is the union of all sets of answer sets of $(L, P \cup \{p \leftarrow \mid p \in B\})$ for all total choices B of C :*

$$\text{minimize (resp., maximize) } \sum_{r \in R, r \models \beta \wedge \alpha} y_r \quad \text{subject to (2).}$$

Well-Founded Semantics. A *partial world* I is a consistent subset of $Lit_\Phi = HB_\Phi \cup \neg HB_\Phi$. We identify I with the three-valued interpretation $I: HB_\Phi \rightarrow \{\text{true}, \text{false}, \text{undefined}\}$ that is defined by $I(a) = \text{true}$ iff $a \in I$, $I(a) = \text{false}$ iff $\neg a \in I$, and $I(a) = \text{undefined}$ iff $I \cap \{a, \neg a\} = \emptyset$. We use $\mathcal{I}_\Phi^p$ to denote the set of all partial worlds over Φ . Every classical formula ϕ in a partial world I under a variable assignment σ is associated with a *three-valued truth value* from $\{\text{true}, \text{false}, \text{undefined}\}$, denoted $I_\sigma(\phi)$ (or simply $I(\phi)$ when ϕ is ground), which is inductively defined by:

- $I_\sigma(p(t_1, \dots, t_k)) = I(p(\sigma(t_1), \dots, \sigma(t_k)))$;
- $I_\sigma(\neg\phi) = \text{true}$ iff $I_\sigma(\phi) = \text{false}$, and $I_\sigma(\neg\phi) = \text{false}$ iff $I_\sigma(\phi) = \text{true}$;
- $I_\sigma(\phi \wedge \psi) = \text{true}$ iff $I_\sigma(\phi) = I_\sigma(\psi) = \text{true}$, and $I_\sigma(\phi \wedge \psi) = \text{false}$ iff $I_\sigma(\phi) = \text{false}$ or $I_\sigma(\psi) = \text{false}$.

A *partial probabilistic interpretation* Pr is a probability function on $\mathcal{I}_\Phi^p$. The *probability* of a classical formula ϕ in Pr under a variable assignment σ , denoted $Pr_\sigma(\phi)$ (or simply $Pr(\phi)$ when ϕ is ground), is **undefined**, if $I_\sigma(\phi)$ is **undefined** for some $I \in \mathcal{I}_\Phi^p$ with $Pr(I) > 0$; and $Pr_\sigma(\phi)$ is defined as the sum of all $Pr(I)$ such that $I \in \mathcal{I}_\Phi^p$

and $I_\sigma(\phi) = \mathbf{true}$, otherwise. For classical formulas ϕ and ψ with $Pr_\sigma(\phi) > 0$, we use $Pr_\sigma(\psi|\phi)$ to abbreviate $Pr_\sigma(\psi \wedge \phi) / Pr_\sigma(\phi)$. Every probabilistic formula F in Pr under σ is associated with a *three-valued truth value* from $\{\mathbf{true}, \mathbf{false}, \mathbf{undefined}\}$, denoted $Pr_\sigma(F)$, which is inductively defined as follows:

- $Pr_\sigma((\psi|\phi)[l, u]) = \mathbf{true}$ iff $Pr_\sigma(\phi) = 0$ or $Pr_\sigma(\psi|\phi) \in [l, u]$, and
 $Pr_\sigma((\psi|\phi)[l, u]) = \mathbf{false}$ iff $Pr_\sigma(\phi) > 0$ and $Pr_\sigma(\psi|\phi) \notin [l, u]$;
- $Pr_\sigma(\neg F) = \mathbf{true}$ iff $Pr_\sigma(F) = \mathbf{false}$, and $Pr_\sigma(\neg F) = \mathbf{false}$ iff $Pr_\sigma(F) = \mathbf{true}$;
- $Pr_\sigma(F \wedge G) = \mathbf{true}$ iff $Pr_\sigma(F) = Pr_\sigma(G) = \mathbf{true}$, and
 $Pr_\sigma(F \wedge G) = \mathbf{false}$ iff $Pr_\sigma(F) = \mathbf{false}$ or $Pr_\sigma(G) = \mathbf{false}$.

The *well-founded model* of a pdl-program $KB = (L, P, C, \mu)$, denoted Pr_{KB}^{wf} , is defined by (i) $Pr_{KB}^{wf}(I_B) = \mu(B)$, where I_B is the well-founded model of $(L, P \cup \{p \leftarrow \mid p \in B\})$, for every total choice B of C , and (ii) $Pr_{KB}^{wf}(I) = 0$ for all other $I \in \mathcal{I}_\Phi^p$. A probabilistic formula F is a *well-founded consequence* of KB , denoted $KB \Vdash^{wf} F$, iff F is **true** in Pr_{KB}^{wf} under every variable assignment σ . A conditional constraint $(\psi|\phi)[l, u]$ is a *tight well-founded consequence* of KB , denoted $KB \Vdash_{tight}^{wf} (\psi|\phi)[l, u]$, iff (i) $Pr_{KB}^{wf}(\phi)$ under every variable assignments σ is different from **undefined**, and (ii) l (resp., u) is the infimum (resp., supremum) of $Pr_\sigma(\psi|\phi)$ subject to $Pr = Pr_{KB}^{wf}$ and all variable assignments σ with $Pr_\sigma(\phi) > 0$.

As an advantage of the well-founded semantics, every pdl-program KB has a unique well-founded model, but not necessarily an answer set model. Furthermore, the unique well-founded model can be easily computed by fixpoint iteration [10]. As a drawback, the well-founded model associates only with *some* classical formulas under σ a probability, while every answer set model associates with *all* classical formulas under σ a probability. The following theorem shows that the answer set semantics is a refinement of the well-founded semantics. That is, if an answer to a query under the well-founded semantics is defined, then it coincides with the answer under the answer set semantics. The theorem follows from the result that the well-founded semantics of dl-programs approximates their answer set semantics [10]. The advantages of both semantics can thus be combined in query processing by first trying to compute the well-founded answer, and only if this does not exist the answer under the answer set semantics.

Theorem 3.2. *Let $KB = (L, P, C, \mu)$ be a consistent pdl-program, and let $(\psi|\phi)[l, u]$ be a ground conditional constraint. If $Pr_{KB}^{wf}(\phi) \neq \mathbf{undefined}$, then*

- (a) $KB \Vdash^{wf} (\psi|\phi)[l, u]$ iff $KB \Vdash^{as} (\psi|\phi)[l, u]$, and
- (b) $KB \Vdash_{tight}^{wf} (\psi|\phi)[l, u]$ iff $KB \Vdash_{tight}^{as} (\psi|\phi)[l, u]$.

4 Related Work

In this section, we discuss related work on the combination of description logics and logic programs, which can be divided into (a) hybrid approaches using description logics as input to logic programs, (b) approaches reducing description logics to logic programs, (c) combinations of description logics with default and defeasible logic, and

(d) approaches to rule-based well-founded reasoning in the Semantic Web. Below we give some representatives for (a)–(d). Further works and details are given in [9].

The works by Donini *et al.* [8], Levy and Rousset [21], and Rosati [25] are representatives of hybrid approaches using description logics as input. Donini *et al.* [8] introduce a combination of (disjunction-, negation-, and function-free) datalog with the description logic $\mathcal{ALC}$. An integrated knowledge base consists of a structural component in $\mathcal{ALC}$ and a relational component in datalog, where the integration of both components lies in using concepts from the structural component as constraints in rule bodies of the relational component. The closely related work by Levy and Rousset [21] presents a combination of Horn rules with the description logic $\mathcal{ALCN}\mathcal{R}$. In contrast to Donini *et al.* [8], Levy and Rousset also allow for roles as constraints in rule bodies, and do not require the safety condition that variables in constraints in the body of a rule r must also appear in ordinary atoms in the body of r . Finally, Rosati [25] presents a combination of disjunctive datalog (with classical and default negation, but without function symbols) with $\mathcal{ALC}$, which is based on a generalized answer set semantics.

Some approaches reducing description logic reasoning to logic programming are the works by Van Belleghem *et al.* [28], Alsaç and Baral [1], Swift [27], Grosz *et al.* [15], and Hufstadt *et al.* [20]. Early work on dealing with default information in description logics is the approach due to Baader and Hollunder [4], where Reiter's default logic is adapted to terminological knowledge bases. Antoniou [2] combines defeasible reasoning with description logics for the Semantic Web. In [3], Antoniou and Wagner summarize defeasible and strict reasoning in a single rule formalism.

An important approach to rule-based reasoning under the well-founded semantics for the Semantic Web is due to Damásio [7]. He aims at developing Prolog tools for implementing different semantics for RuleML [6]. So far, an XML parser library as well as a RuleML compiler have been developed, with routines to convert RuleML rule bases to Prolog and vice versa. The compiler supports paraconsistent well-founded semantics with explicit negation; it is planned to be extended to use XSB [24].

5 Summary and Outlook

We have presented probabilistic description logic programs (or pdl-programs), which are a combination of dl-programs under the answer set and the well-founded semantics with Poole's independent choice logic. We have shown that query processing in such pdl-programs can be reduced to computing all answer sets of dl-programs and solving linear optimization problems, and to computing the well-founded semantics of dl-programs, respectively. We have also shown that the answer set semantics of pdl-programs is a refinement of the well-founded semantics of pdl-programs.

An interesting topic of future research is to further enhance pdl-programs towards a possible use for Web Services. This may be done by exploiting and generalizing further features of Poole's ICL for dynamic and multi-agent systems [23]. Another interesting aspect is to analyze the computational complexity of query processing in pdl-programs under the answer set and the well-founded semantics.

Acknowledgments. This work has been supported by the Marie Curie Individual Fellowship HPMF-CT-2001-001286 of the European Union programme “Human Potential” (disclaimer: The author is solely responsible for information communicated and the European Commission is not responsible for any views or results expressed) and by a Heisenberg Professorship of the German Research Foundation. I thank the reviewers for their constructive comments, which helped to improve this work.

References

1. G. Alsaç and C. Baral. Reasoning in description logics using declarative logic programming. Report, Department of Computer Science and Engineering, Arizona State University, 2001.
2. G. Antoniou. Nonmonotonic rule systems on top of ontology layers. In *Proc. ISWC-2002*, LNCS 2342, pp. 394–398.
3. G. Antoniou and G. Wagner. Rules and defeasible reasoning on the Semantic Web. In *Proc. RuleML-2003*, LNCS 2876, pp. 111–120.
4. F. Baader and B. Hollunder. Embedding defaults into terminological representation systems. *J. Automated Reasoning*, 14:149–180, 1995.
5. T. Berners-Lee. *Weaving the Web*. Harper, San Francisco, CA, 1999.
6. H. Boley, S. Tabet, and G. Wagner. Design rationale for RuleML: A markup language for Semantic Web rules. In *Proc. SWWS-2001*, pp. 381–401.
7. C. V. Damásio. The W⁴ Project, 2002. <http://centria.di.fct.unl.pt/~cd/projectos/w4/index.htm>.
8. F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. $\mathcal{AL}$ -log: Integrating datalog and description logics. *Journal of Intelligent Information Systems (JIIS)*, 10(3):227–252, 1998.
9. T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the Semantic Web. In *Proc. KR-2004*, pp. 141–151.
10. T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Well-founded semantics for description logic programs in the Semantic Web. In *Proc. RuleML-2004*, LNCS 3323, pp. 81–97.
11. D. Fensel, W. Wahlster, H. Lieberman, and J. Hendler, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2002.
12. A. Finzi and T. Lukasiewicz. Structure-based causes and explanations in the independent choice logic. In *Proc. UAI-2003*, pp. 225–232.
13. M. Gelfond and V. Lifschitz. Classical negation in logic programs and deductive databases. *New Generation Computing*, 17:365–387, 1991.
14. R. Giugno and T. Lukasiewicz. P- $\mathcal{SHOQ}(\mathbf{D})$: A probabilistic extension of $\mathcal{SHOQ}(\mathbf{D})$ for probabilistic ontologies in the Semantic Web. In *Proc. JELIA-2002*, LNCS 2424, pp. 86–97.
15. B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: Combining logic programs with description logics. In *Proc. WWW-2003*, pp. 48–57.
16. I. Horrocks and P. F. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. In *Proc. ISWC-2003*, LNCS 2870, pp. 17–29.
17. I. Horrocks and P. F. Patel-Schneider. A proposal for an OWL Rules Language. In *Proc. WWW-2004*, pp. 723–731.
18. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.
19. I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In *Proc. LPAR-1999*, LNCS 1705, pp. 161–180.
20. U. Hufstadt, B. Motik, and U. Sattler. Reasoning for description logics around $\mathcal{SHIQ}$ in a resolution framework. Technical Report 3-8-04/04, FZI Karlsruhe, 2004.

21. A. Y. Levy and M.-C. Rousset. Combining Horn rules and description logics in CARIN. *Artif. Intell.*, 104(1-2):165–209, 1998.
22. H. Nottelmann and N. Fuhr. pDAML+OIL: A probabilistic extension to DAML+OIL based on probabilistic Datalog. In *Proc. IPMU-2004*.
23. D. Poole. The independent choice logic for modelling multiple agents under uncertainty. *Artif. Intell.*, 94:7–56, 1997.
24. P. Rao, K. Sagonas, T. Swift, D. S. Warren, and J. Freire. XSB: A system for efficiently computing WFS. In *Proc. LPNMR-1997, LNCS 1265*, pp. 430–440.
25. R. Rosati. Towards expressive KR systems integrating datalog and description logics: Preliminary report. In *Proc. DL-1999*, pp. 160–164.
26. U. Straccia. Uncertainty and description logic programs: A proposal for expressing rules and uncertainty on top of ontologies. Technical Report ISTI-2004-TR, CNR Pisa, 2004.
27. T. Swift. Deduction in ontologies via ASP. In *Proc. LPNMR-2004, LNCS 2923*, pp. 275–288.
28. K. Van Belleghem, M. Denecker, and D. De Schreye. A strong correspondence between description logics and open logic programming. In *Proc. ICLP-1997*, pp. 346–360.
29. A. Van Gelder, K. A. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, 1991.
30. W3C. OWL web ontology language overview, 2004. W3C Recommendation (10 February 2004). Available at www.w3.org/TR/2004/REC-owl-features-20040210/.

Coherent Restrictions of Vague Conditional Lower-Upper Probability Extensions

Andrea Capotorti¹ and Maroussa Zagoraiou²

¹ Dip. di Matematica e Informatica Università di Perugia - Italy
`capot@dipmat.unipg.it`

² Dip. di Scienze Statistiche “Paolo Fortunati” Università di Bologna - Italy
`maroussa@lycos.it`

Abstract. In this paper we propose a way to restrict extension bounds induced by coherent conditional lower-upper probability assessments. Such shrinkage turns out to be helpful whenever the natural bounds are too vague to be used. Since coherence of a conditional lower-upper probability assessment can be characterized through a class of conditional probability distributions, the idea is to take the intersection of the extension bounds induced by each single element of the class instead of the convex combination, as it is usually done. Coherence of such method is proved for extensions performed on both conditional events logical dependent and not-dependent on the initial domain.

Keywords: Coherent lower-upper conditional probability assessments, inference.

1 Introduction

Over the last decades the use of *imprecise* models to deal with uncertainty has dramatically increased. Especially in problems where the information at hand is not so fully detailed to allow us to adopt standard statistical tools. Initially outlined by the pioneering work of de Finetti [10] and latter mainly developed by Walley [14], imprecise models nowadays cover a wide range of subjects. (For a quite exhaustive view refer e.g. to the last ISIPTA symposia [12]). Among them, of particular interest are coherent lower-upper conditional probabilities assessments which have been developed in the work started in [6] and recently fully described in Coletti & Scozzafava’s book [9]. Their relevance relays mainly on their flexibility and potentiality to comprehend several different uncertainty approaches. Anyhow, if used in their full generality they could lead to vague inference conclusions. This can happen when they are applied to problems with really scarce information or whenever available data is compatible with a large set of models. In the present paper hence we propose a systematic procedure to shrink inference conclusions which will be identified with extension bounds. The result will profit from the coherence characterization of the existence of an agreeing class of conditional probabilities measures and from its analysis.

The rest of the paper is organized as follows. In the next three subsections we will give the basic notions about inference through coherent lower-upper conditional probability assessments. Even if everything is already fully described in [9], we will report what would suffice to let the contribution be as much as possible self-contained. In Section 2 the main result expressed by Proposition 1 is reported. This was in some way anticipated in [5], but here its proof is explicitly given and detailed. We conclude with Section 3 where a medical application illustrates a potential use of the results obtained.

1.1 Preliminaries

For the sake of simplicity we will use conditional and unconditional events, but everything can be easily generalized to (finite) random variables, conditional or not. (See for example what has been done with conditional previsions in [4]). The initial information, usually a knowledge “and/or” rule base, is represented through a conditional lower-upper probability assessment $(\mathcal{E}, \mathcal{L}_C, \mathbf{p})$. The first component of an assessment is a generic list of n conditional events $\mathcal{E} = (E_1|H_1, \dots, E_n|H_n)$. Note that some $E_i|H_i$ could be actually unconditional and in such case H_i will coincide with the sure event Ω . In the following we will also refer to the set $\mathcal{U}_\mathcal{E} = \{E_1, \dots, E_n, H_1, \dots, H_n\}$ of unconditional events appearing as components of the elements of $\mathcal{E}$.

Incompleteness of the information can have two origins: firstly the E_i ’s might not describe all possible combinations of situations; secondly the different circumstances H_i ’s might *overlap* or might not cover all possibilities. For this, it is crucial to know which are the relationships of incompatibility, implication, equivalence or whatever, among the events in $\mathcal{U}_\mathcal{E}$. In general, the list $\mathcal{L}_C$ of the logical constraints among $\mathcal{U}_\mathcal{E}$ will appear as the second component of an assessment. Such relationships $\mathcal{L}_C$ represent constraints that limit which are the possible *atoms*¹. The atoms A_r , with $r = 1, \dots, a \leq 2^{2^n}$, are elementary events (i.e. they form a partition) obtained by full combinations of affirmed or negated events in $\mathcal{U}_\mathcal{E}$. Hence a general atom A_r is obtained by an expression like $C_r = \widetilde{E_1} \wedge \dots \wedge \widetilde{E_n} \wedge \widetilde{H_1} \wedge \dots \wedge \widetilde{H_n}$, where each component, say $\widetilde{E_i}$, can be either the affirmed event E_i or its negation $\neg E_i$. Note that by the three valued logic associate to conditional events, each element $E_i|H_i \in \mathcal{E}$ generates a partition with only three elements $\{E_i H_i, \neg E_i H_i, \neg H_i\}$ so that the couples of atoms that present $E_i \neg H_i$ and $\neg E_i \neg H_i$ can be joined together leading to the more appropriate expression for the atoms

$$A_r = \widetilde{E_1|H_1} \wedge \dots \wedge \widetilde{E_n|H_n}, \quad (1)$$

where each $\widetilde{E_i|H_i}$ varies among $E_i H_i$, $\neg E_i H_i$, or $\neg H_i$. This will actually reduce to 3^n the upper bound for the number of atoms (for a detailed exposition on this subject refer e.g. to [1, 11]). Moreover, as it is usual in conditional contexts

¹ In some discipline atoms are called *possible worlds*, while in [13] they are referred as *realm*.

(see [8] and [9–§11.3]), we will refer only to atoms spanned by $\mathcal{U}_{\mathcal{E}}$ and inside the disjunction $\bigvee_{i=1}^n H_i$, because only elementary situations contemplated in some of the considered hypotheses must be involved to check the consistency of the assessment². Hence the proper upper bound for the number of atoms a is $3^n - 1$. The limitation imposed by $\mathcal{L}_{\mathcal{C}}$ consists on the inadmissibility of some atom, i.e. some expression like (1) turn out to coincide with the impossible event ϕ .

In the sequel we will also need to use the characteristic vectors of the events. These are vectors whose components are 1 or 0 depending on whether the corresponding atom implies the event or not. We will denote such vectors with the same letter as the event, but in boldface lower-cases. Hence, $\mathbf{e}_i$ and $\mathbf{h}_i$ will denote the characteristic vectors of E_i and H_i , respectively, while their juxtaposition $\mathbf{e}_i \mathbf{h}_i$ will represent the characteristic vector of the conjunction $E_i H_i$. (For the sake of simplicity in the following we will omit the usual conjunction operator $\wedge$.)

Introducing a vector of variables $\mathbf{x} = (x_1 \dots x_a)$, where each component x_r is associated with possible values for the probability of the atom A_r , it is possible to *rebuild* the possible values of probability for any event in $\mathcal{U}_{\mathcal{E}}$, say E_i , simply by $P(E_i) = \sum_{A_r \subseteq E_i} P(A_r) = \mathbf{e}_i \cdot \mathbf{x}$, where $\cdot$ represents the row-column matrix product.

Note that the atoms, and consequently the characteristic vectors, are not a part of the conditional probability assessment, i.e. they are not given directly by the analyst, but they are implicitly defined by the first two components $\mathcal{E}$ and $\mathcal{L}_{\mathcal{C}}$. Nonetheless, they are important because they are the main operational tool involved in the inferential process.

The last component of an assessment is represented by a vector of numerical ranges $\mathbf{p} = ([lb_1, ub_1], \dots, [lb_n, ub_n])$. Extreme points of each closed interval $[lb_i, ub_i]$ represent lower and upper bounds associated with probabilities for the corresponding conditional event $E_i | H_i$. These are usually estimated by expert beliefs, by literature reports or by collected data.

Note that some of the numerical ranges $[lb_i, ub_i]$ may degenerate to a single value p_i , representing a precise assessment.

1.2 Coherence

If we cannot adopt a unique probabilistic model for the assessment $(\mathcal{E}, \mathcal{L}_{\mathcal{C}}, \mathbf{p})$, it is still possible to search for the class $\mathcal{P}$ of full conditional probability distributions that are compatible with the assessments we can make. It is possible to ask various properties of $\mathcal{P}$: in the present paper we look for a class such that $\mathbf{p}$ coincides with the convex envelope of $\mathcal{P}$ restricted to $\mathcal{E}$, i.e. such that

$$\forall P \in \mathcal{P} \quad lb_i \leq P(E_i | H_i) \leq ub_i \quad \text{for all } E_i | H_i \in \mathcal{E}; \quad (2)$$

² For those familiar with Walley's notation, a similar motivation is used to introduce the consistency property of *Avoiding Uniform Loss* rather than the *Avoiding Sure Loss* for conditional previsions. See [14–§7.1.3] and [15–§2.1].

$$\forall E_i|H_i \in \mathcal{E} \quad \exists P', P'' \in \mathcal{P} \text{ s.t. } P'(E_i|H_i) = lb_i, P''(E_i|H_i) = ub_i. \quad (3)$$

Practically speaking, the third component $\mathbf{p}$ of the assessment represents a set of numerical constraints that all the admissible models (the conditional probabilities $P \in \mathcal{P}$) must satisfy (inequalities (2)). Such constraints must be tight enough so that their bounds can be actually reached by some of the admissible models (equalities (3)).

In the following, such a class $\mathcal{P}$ of probability distributions will be said to *agree* with the assessment $(\mathcal{E}, \mathcal{L}_C, \mathbf{p})$ and its existence guarantees about the *coherence* of the assessment.

As has been already stated in [7], and in particular in [9–§15.2], the existence of $\mathcal{P}$ can be checked *operationally* by the satisfiability of a *class of sequences* of linear systems $\{\mathcal{S}_\alpha^j\}$, with $j = 1, \dots, 2n$ and $\alpha = 1, \dots, \alpha_j$, having a common structure like

$$\begin{cases} (\mathbf{e}_j \mathbf{h}_j - p_j \mathbf{h}_j) \cdot \mathbf{x}^\alpha = 0 & \text{if } \mathbf{h}_j \cdot \mathbf{x}^{\alpha-1} = 0 \\ (\mathbf{e}_k \mathbf{h}_k - lb_k \mathbf{h}_k) \cdot \mathbf{x}^\alpha \geq 0 \\ (\mathbf{e}_k \mathbf{h}_k - ub_k \mathbf{h}_k) \cdot \mathbf{x}^\alpha \leq 0 \\ \mathbf{x}^\alpha \geq \mathbf{0}, \mathbf{x}^\alpha \neq \mathbf{0} \end{cases} \quad \forall E_k|H_k \neq E_j|H_j \text{ s.t. } \mathbf{h}_k \cdot \mathbf{x}^{\alpha-1} = 0, \quad (4)$$

where $\mathbf{0}$ is the null vector, $E_j|H_j$ equals $E_i|H_i$ for both consecutive odd and even indexes $j = 2i - 1$ and $j = 2i$ (or, conversely $i = \lfloor \frac{j+1}{2} \rfloor$), while the value p_j that appears in the first equation equals lb_i for the odd indexes $j = 2i - 1$ and ub_i for the even ones $j = 2i$.

Hence, for each event $E_i|H_i \in \mathcal{E}$ there will be associated two sequences of linear systems $\mathcal{S}_\alpha^{2i-1}$ and $\mathcal{S}_\alpha^{2i}$. This to ensure that, according to (3), the bounds lb_i and ub_i can be actually attained. Of course whenever the bounds degenerate to a single value p_i , the two sequences coincide.

Note that *sequences* of linear systems are necessary to allow conditioning events H_i 's to have induced probabilities that are not bounded away from 0. This procedure partitions $\mathcal{E}$ in different *zero layers*³ indexed by α .

Such linear systems reflect an attempt to determine unconditional probability distributions through which to construct the agreeing class $\mathcal{P}$ (i.e. a set of conditional probabilities satisfying (2) and (3)).

The set of all possible solutions to the class of linear systems sequence $\{\mathcal{S}_\alpha^j\}$ implicitly induces the searched class $\mathcal{P}$. Hence, if such set of solutions is not empty, the assessment $(\mathcal{E}, \mathcal{L}_C, \mathbf{p})$ is said to be coherent, otherwise not. Note that this coherence notion is almost the same as those usually adopted in imprecise probabilities frameworks. That is, $\mathbf{p}$ coincides with its *natural extension*. (See [14] and [15–§3.2] property (d).) The only difference occurs in the proper treatment of conditional events $E_i|H_i$ whose conditioning H_i can have probability not bounded away from zero.

³ Here we report only basic notions, for a deeper exposition of this aspect refer again to [9], in particular to § 12 and § 15.

1.3 Extension

In practical applications when information comes from different sources, it turns out that checking the coherence of the assessment $(\mathcal{E}, \mathcal{L}_C, \mathbf{p})$ is a compulsory step. Once coherence has been assured, it is possible to perform inference on any conditional event $K|F$ judged important to reach conclusions in the problem. Generally, K represents some hypothesis to judge whenever there should be some evidence F .

In this context, inference reduces to compute the coherent extension of $\mathbf{p}$ to $K|F$, obtainable as the closed interval $[lb_{K|F}, ub_{K|F}]$ with $lb_{K|F} = \min_{P \in \mathcal{P}} P(K_*|F_*)$ and $ub_{K|F} = \max_{P \in \mathcal{P}} P(K^*|F^*)$, where $K_*|F_*$ and $K^*|F^*$ are the greatest and the smallest conditional event logically dependent on $\mathcal{U}_{\mathcal{E}}$ contained and containing⁴ $K|F$, respectively.

Although this is theoretically simple, from the practical point of view it is more subtle. In fact, following a method similar of that depicted in the previous subsection, we are required to perform *sequences of optimizations* $\{\mathcal{O}_{\alpha}^j\}$. Thanks to the possibility of exploiting zero probabilities and thanks to proper normalization conditions, all the optimizations problems in $\{\mathcal{O}_{\alpha}^j\}$ are reduced to be linear programs⁵.

The number of sequences is the same as that of the linear systems (4): there are two sequences with $j = 2i - 1$ and $j = 2i$ for each conditional event $E_i|H_i \in \mathcal{E}$. Hence they are actually “at worst” $2n$, on account of the already stated consideration about the coincidence of the pair of sequences whenever coherence requires a precise value for $p_i \in \mathbf{p}$.

Each sequence is actually composed by two programs, one pertaining to $K_*|F_*$ and the other pertaining to $K^*|F^*$, that end with a pair of optimal values $lb_{K|F}^j$ and $ub_{K|F}^j$. These values represent the minimum and the maximum, respectively, for the searched probability $P(K|F)$ but under the specific equality constraint in (3) associated to $E_j|H_j$.

Each optimization program consists of two parts. In the first part there is the goal to reach the deepest zero-layer of the target $K_*|F_*$ (or of $K^*|F^*$). This is attained by taking a fictitious objective function, e.g. minimize $\mathbf{\Omega} \cdot \mathbf{x}$, while the constraint

$$\mathbf{f}_* \cdot \mathbf{x}^{\alpha} = 0 \quad (\mathbf{f}^* \cdot \mathbf{x}^{\alpha} = 0) \quad (5)$$

is imposed, together with those of (4). Till solutions exist, the program continues to pass from a zero-layer to the subsequent. Finally, either the set of constraints is empty so that $lb_{K|F}^j = 0$ ($ub_{K|F}^j = 1$), or it passes to the second part where an actual linear program is performed. This optimization step has as objective function

$$\text{minimize } \mathbf{k}_* \mathbf{f}_* \cdot \mathbf{x}^{\alpha} \quad (\text{maximize } \mathbf{k}^* \mathbf{f}^* \cdot \mathbf{x}^{\alpha}) \quad (6)$$

⁴ For the formal inclusion definition among conditional events refer to [9–§10.2].

⁵ For a full description see [9–§14.1]. A similar approach with dual programs in [15].

under the constraints (4) plus

$$\mathbf{f}_* \cdot \mathbf{x}^\alpha = 1 \quad (\mathbf{f}^* \cdot \mathbf{x}^\alpha = 1) \quad (7)$$

instead of (5).

The final coherent interval $[lb_{K|F}, ub_{K|F}]$ for $P(K|F)$ will result from the convex combination of all the intervals $[lb_{K|F}^j, ub_{K|F}^j]$, i.e. $lb_{K|F} = \min_{j \in \{1, \dots, 2n\}} lb_{K|F}^j$ and $ub_{K|F} = \max_{j \in \{1, \dots, 2n\}} ub_{K|F}^j$.

The main difficulty of such procedures is the usually huge number a of atoms. In fact, it has been show that already the problem of checking the coherence for unconditional precise assessments is NP-complete. Anyhow, thanks to a smart use of null probabilities and to the notion of *locally strong coherence*, in [3] the complexity problem has also been faced. Abstract problems have been solved with $O(n^3)$ logical satisfiability tests in place of solving the linear systems and the optimization problems that have $O(3^n)$ number of unknowns. Even with such promising results, a systematic complexity study of this last procedure is still missing.

2 Vague Extensions and Their Improvement

As already stated in the previous section, coherent bounds for a new event $K|F$

$$\underline{P}(K|F) = \min_{P \in \mathcal{P}} P(K|F) \text{ and } \overline{P}(K|F) = \max_{P \in \mathcal{P}} P(K|F) \quad (8)$$

are obtained by taking into account all the admissible distributions $\mathcal{P}$. They operationally results from the convex combination of all the single optimization bounds $\{lb_{K|F}^j, ub_{K|F}^j\}$, with $j \in \{1, \dots, 2n\}$, obtained in the different sequences of linear programs in $\{\mathcal{O}_\alpha^j\}$.

Unluckily, as it has been noticed in some practical applications, the range of values between the bounds in (8) could results very wide, till to reach the uninformative interval $[0, 1]$.

Anyhow, recall that differences among optimal values obtained in different sequences derive from the restrictions (3) applied each couple of sequences to a different conditional event $E_j|H_j \in \mathcal{E}$, and it could happen that some of the intervals $[lb_{K|F}^j, ub_{K|F}^j]$ are much narrower than the others. This sheds some light on the different roles played by the agreeing distributions inside $\mathcal{P}$. Hence it could be justified a *restrictive* attitude that takes from $\mathcal{P}$ only the most informative part. This realize by adopting the intersection of the intervals $[lb_{K|F}^j, ub_{K|F}^j]$ instead of their convex combination. We named this approach *restrictive* because it takes as good only inference values shared by *every* conditional distribution $P \in \mathcal{P}$, in opposition with the more *conservative* attitude of the coherence requirement for $(\mathcal{E}, \mathcal{L}_C, \mathbf{p})$ where it suffices that the bounds $\mathbf{p}$ are reached by *some* conditional distribution $P \in \mathcal{P}$. In fact the general extension method reported in the previous section, known also as *natural extension*, is a *least commitment* procedure, i.e. it produces nothing more and nothing less of what

can be obtained by a bald use of the available information. Anyhow, its generality does not guarantee the coherence of any restriction of the conclusion that could eventually be required by extra considerations. On the other hand, the more restrictive attitude we propose has the property that any restriction would result coherent. Coherence of such a choice is stated by the following proposition:

Proposition 1. *Let $(\mathcal{E}, \mathcal{L}_C, \mathbf{p})$ be a coherent conditional assessment, $K|F \notin \mathcal{E}$ a new conditional event and $\left\{lb_{K|F}^j, ub_{K|F}^j\right\}_{j=1}^{2n}$ the set of optimal values obtained by linear programs $\{\mathcal{O}_\alpha^j\}$ as described in Subsection 1.3 and performed on $K|F$.*

If the interval

$$[lb_{K|F}^*, ub_{K|F}^*] = \bigcap_{j=1}^{2n} [lb_{K|F}^j, ub_{K|F}^j]$$

is not empty then the conditional assessment $(\mathcal{E}^*, \mathcal{L}_C^*, \mathbf{p}^*)$ with

$$\begin{aligned} \mathcal{E}^* &= \mathcal{E} \cup \{K|F\} \\ \mathcal{L}_C^* &= \mathcal{L}_C \cup \{\text{logical constraints among } K, F \text{ and } \mathcal{U}_\mathcal{E}\}, \\ \mathbf{p}^* &= \mathbf{p} \cup \{[lb_{K|F}^*, ub_{K|F}^*]\} \end{aligned} \quad (9)$$

is coherent.

The rest of the section is devoted to prove Proposition 1 both when $K|F$ is logically dependent on $\mathcal{E}$ and when not. Before to face such an effort, note that the suggested range $[lb_{K|F}^*, ub_{K|F}^*]$, even being very restrictive with respect to $\mathcal{P}$, can be used without a modification of the initial numerical evaluation $\mathbf{p}$. This means that the choice of such restriction does not require a belief revision, but it implicitly induces structural constraints on the admissible distributions $\mathcal{P}$. Of course it cannot be used systematically but only when the natural extension produce unexpected wide results and one wants to profit from the *hints* suggested by the admissible distributions to reduce vagueness.

Proof (of proposition). We first prove Proposition 1 if $K|F$ is logically dependent on $\mathcal{E}$, i.e. whenever $K_* F_* = K^* F^* = K F$ and $F_* = F^* = F$. Since $K|F$ can be thought as the $n+1$ -th conditional event of $\mathcal{E}^*$, we need to prove the solvability of $2(n+1)$ linear systems sequences of type (4) that pertain to $(\mathcal{E}^*, \mathcal{L}_C^*, \mathbf{p}^*)$. Note that, in the case of logical dependence, the atoms spanned by $\mathcal{U}_{\mathcal{E}^*}$ are the same of those spanned by $\mathcal{U}_\mathcal{E}$, hence the unknowns involved in the linear systems to check the coherence of $(\mathcal{E}^*, \mathcal{L}_C^*, \mathbf{p}^*)$ will be the same of those used for the computation of the set of optimal values $\{lb_{K|F}^j, ub_{K|F}^j\}_{j=1}^{2n}$. In fact we will profit from the optimal solutions of the linear programs $\{\mathcal{O}_\alpha^j\}$ described in Subsect. 1.3 to build the searched solutions.

The solutions must satisfy (4) plus the constraints associated to $K|F$ that are

$$(\mathbf{k}\mathbf{f} - lb_{K|F}^* \mathbf{f}) \cdot \mathbf{x}^\alpha \geq 0 \quad \text{and} \quad (\mathbf{k}\mathbf{f} - ub_{K|F}^* \mathbf{f}) \cdot \mathbf{x}^\alpha \leq 0 \quad (10)$$

for the first $2n$ sequences, while they are replaced by

$$(\mathbf{k}\mathbf{f} - lb_{K|F}^* \mathbf{f}) \cdot \mathbf{x}^\alpha = 0 \quad \text{for } j = 2n+1, \quad (11)$$

and by

$$(\mathbf{k}\mathbf{f} - ub_{K|F}^* \mathbf{f}) \cdot \mathbf{x}^\alpha = 0 \quad \text{for } j = 2n + 2 . \quad (12)$$

About the first $2n$ sequences, fix an index $j \in \{1, \dots, 2n\}$. By hypothesis we know there are two optimization sequences $\mathcal{O}_\alpha^j$, one of minimization and the other of maximization, that bring to the optimal values $lb_{K|F}^j$ and $ub_{K|F}^j$, respectively. Anyhow, in the first part the two sequences coincide because $F_* = F^* = F$, while they differentiate only in the final optimization step. Since the optimal solutions $\mathbf{x}^\alpha$ of the first part satisfy the constraints (4) plus (5), and (5) trivially imply (10), they can be taken as solutions of first systems of the j -th sequence. On the other hand, the optimal solutions of the actual optimization steps, say $\mathbf{x}_l^\alpha$ and $\mathbf{x}_u^\alpha$, both satisfy the constraints (4) plus (7) so that we can write

$$lb_{K|F}^j = \frac{\mathbf{k}\mathbf{f} \cdot \mathbf{x}_l^\alpha}{\mathbf{f} \cdot \mathbf{x}_l^\alpha} \quad \text{and} \quad ub_{K|F}^j = \frac{\mathbf{k}\mathbf{f} \cdot \mathbf{x}_u^\alpha}{\mathbf{f} \cdot \mathbf{x}_u^\alpha} . \quad (13)$$

Denote by $m_{K|F}^* = \frac{lb_{K|F}^* + ub_{K|F}^*}{2}$ the midpoint between the values $lb_{K|F}^*$ and $ub_{K|F}^*$. By definition we have the following inequalities chain

$$lb_{K|F}^j \leq lb_{K|F}^* \leq m_{K|F}^* \leq ub_{K|F}^* \leq ub_{K|F}^j , \quad (14)$$

hence there exists a constant $\beta_j \in [0, 1]$ such that $m_{K|F}^* = \beta_j lb_{K|F}^j + (1 - \beta_j) ub_{K|F}^j$. Use now such constant β_j to built a convex combination of the two optimal solutions $\mathbf{x}_l^\alpha$ and $\mathbf{x}_u^\alpha$ as

$$\mathbf{x}_m^\alpha = \beta_j \mathbf{x}_l^\alpha + (1 - \beta_j) \mathbf{x}_u^\alpha . \quad (15)$$

It is trivial to check that $\mathbf{x}_m^\alpha$ satisfies (4) and (7) because $\mathbf{x}_l^\alpha$ and $\mathbf{x}_u^\alpha$ already did, while it also satisfies (10) because

$$\begin{aligned} (\mathbf{k}\mathbf{f} - lb_{K|F}^* \mathbf{f}) \cdot \mathbf{x}_m^\alpha &= \mathbf{k}\mathbf{f} \cdot \mathbf{x}_m^\alpha - lb_{K|F}^* \mathbf{f} \cdot \mathbf{x}_m^\alpha = \text{(by (7))} \\ &= \frac{\mathbf{k}\mathbf{f} \cdot \mathbf{x}_m^\alpha}{\mathbf{f} \cdot \mathbf{x}_m^\alpha} - lb_{K|F}^* = \text{(by (15) and (7))} \\ &= \beta_j \frac{\mathbf{k}\mathbf{f} \cdot \mathbf{x}_l^\alpha}{\mathbf{f} \cdot \mathbf{x}_l^\alpha} + (1 - \beta_j) \frac{\mathbf{k}\mathbf{f} \cdot \mathbf{x}_u^\alpha}{\mathbf{f} \cdot \mathbf{x}_u^\alpha} - lb_{K|F}^* = \\ &= \beta_j lb_{K|F}^j + (1 - \beta_j) ub_{K|F}^j - lb_{K|F}^* = \\ &= m_{K|F}^* - lb_{K|F}^* \geq 0 ; \\ (\mathbf{k}\mathbf{f} - ub_{K|F}^* \mathbf{f}) \cdot \mathbf{x}_m^\alpha &= \mathbf{k}\mathbf{f} \cdot \mathbf{x}_m^\alpha - ub_{K|F}^* \mathbf{f} \cdot \mathbf{x}_m^\alpha = \text{(by (7))} \\ &= \frac{\mathbf{k}\mathbf{f} \cdot \mathbf{x}_m^\alpha}{\mathbf{f} \cdot \mathbf{x}_m^\alpha} - ub_{K|F}^* = \text{(by (15) and (7))} \\ &= \beta_j \frac{\mathbf{k}\mathbf{f} \cdot \mathbf{x}_l^\alpha}{\mathbf{f} \cdot \mathbf{x}_l^\alpha} + (1 - \beta_j) \frac{\mathbf{k}\mathbf{f} \cdot \mathbf{x}_u^\alpha}{\mathbf{f} \cdot \mathbf{x}_u^\alpha} - ub_{K|F}^* = \\ &= \beta_j lb_{K|F}^j + (1 - \beta_j) ub_{K|F}^j - ub_{K|F}^* = \\ &= m_{K|F}^* - ub_{K|F}^* \leq 0 . \end{aligned}$$

Hence we have that $\mathbf{x}_m^\alpha$ is a solution of the last linear system of the j -th sequence, with $j = 1, \dots, 2n$. Let us face now the last two sequences for $j = 2n + 1$ and $j = 2n + 2$. Since, by hypothesis, $lb_{K|F}^*$ and $ub_{K|F}^*$ both belong to the set of optimal values $\{lb_{K|F}^j, ub_{K|F}^j\}_{j=1}^{2n}$, there exist two linear program sequences, say $\mathcal{O}_\alpha^l$ for $lb_{K|F}^*$ and $\mathcal{O}_\alpha^u$ for $ub_{K|F}^*$, where they are reached. The optimal solutions of $\mathcal{O}_\alpha^l$ satisfy, by definition of optimality, (4) plus (11), while those of $\mathcal{O}_\alpha^u$ satisfy (4) plus (12). This conclude the proof in the case of logical dependence. In the more general case we have that

$$K_* = \bigvee_{A_r \subseteq K} A_r \quad ; \quad F_* = \left(\bigvee_{A_r \subseteq K F} A_r \right) \vee \left(\bigvee_{A_r \neg K F \neq \phi} A_r \right) ; \quad (16)$$

$$K^* = \bigvee_{A_r, K \neq \phi} A_r \quad ; \quad F^* = \left(\bigvee_{A_r \subseteq \neg K F} A_r \right) \vee \left(\bigvee_{A_r, K F \neq \phi} A_r \right) . \quad (17)$$

Since $K_*|F_*$ and $K^*|F^*$ are logically dependent on $\mathcal{U}_\mathcal{E}$ and since the set of optimal values $\{lb_{K|F}^j, ub_{K|F}^j\}_{j=1}^{2n}$ are computed on them, from the first part of this proof we know the assessments $(\mathcal{E} \cup \{K_*|F_*\}, \mathcal{L}_\mathcal{C}, \mathbf{p} \cup \{lb_{K|F}^*\})$ and $(\mathcal{E} \cup \{K^*|F^*\}, \mathcal{L}_\mathcal{C}, \mathbf{p} \cup \{ub_{K|F}^*\})$ are coherent. Hence we know there exist solutions of linear system sequences with constraints (4) plus

$$(\mathbf{k}_* \mathbf{f}_* - lb_{K|F}^* \mathbf{f}_*) \cdot \mathbf{x}^\alpha = 0 \quad \text{or} \quad (\mathbf{k}^* \mathbf{f}^* - ub_{K|F}^* \mathbf{f}^*) \cdot \mathbf{x}^\alpha = 0 . \quad (18)$$

We need to show that such solutions can be modified to solve sequences of linear systems with constraints like (4) plus either (10), or (11), or (12), but with respect the assessment $(\mathcal{E}^*, \mathcal{L}_\mathcal{C}^*, \mathbf{p}^*)$. Note that now the atoms spanned by $\mathcal{U}_{\mathcal{E}^*}$ are a refinement of those spanned by $\mathcal{U}_\mathcal{E}$. In particular, each atom A_r spanned by $\mathcal{U}_\mathcal{E}$ could be split in the following potential⁶ three new atoms $B_{r_1} = A_r K F$, $B_{r_2} = A_r \neg K F$, and $B_{r_3} = A_r \neg F$. Hence we have now characteristic vectors with the triple of components. With a small abuse of notation we will maintain the same symbols for the old and the new characteristic vectors, with the understanding that the old ones will be associated to solution vectors denoted with $\mathbf{x}^\alpha$, while the new one with solution vectors denoted by $\mathbf{z}^\alpha$. The goal is to find a way to assign the numerical value of each old solution component x_r^α to three new values $z_{r_1}^\alpha$, $z_{r_2}^\alpha$ and $z_{r_3}^\alpha$. Such operation will be named in the following *re-distribution*.

Constraints of type (4) do not need particular attention because any re-distribution of the values x_r^α among $z_{r_1}^\alpha$, $z_{r_2}^\alpha$ and $z_{r_3}^\alpha$ would be fine since equalities $\mathbf{e}_i \mathbf{h}_i \cdot \mathbf{x}^\alpha = \mathbf{e}_i \mathbf{h}_i \cdot \mathbf{z}^\alpha$ and $\mathbf{h}_i \cdot \mathbf{x}^\alpha = \mathbf{h}_i \cdot \mathbf{z}^\alpha$ would hold for all $E_i | H_i \in \mathcal{E}$.

About the constraints of type (10), (11), or (12) we need to find proper re-distributions of the values x_r^α such that

$$(\mathbf{k}_* \mathbf{f}_* - lb_{K|F}^* \mathbf{f}_*) \cdot \mathbf{x}^\alpha = (\mathbf{k} \mathbf{f} - lb_{K|F}^* \mathbf{f}) \cdot \mathbf{z}^\alpha \quad (19)$$

⁶ Actually some of them could degenerate to the impossible event ϕ because the logical constraints among the elements of $\mathcal{U}_\mathcal{E}$ and F and K .

and

$$(\mathbf{k}^* \mathbf{f}^* - ub_{K|F}^* \mathbf{f}^*) \cdot \mathbf{x}^\alpha = (\mathbf{k} \mathbf{f} - ub_{K|F}^* \mathbf{f}) \cdot \mathbf{z}^\alpha . \quad (20)$$

By (16) it follows that $K^* F^* \subseteq KF$ and $\neg K^* F^* \supseteq \neg KF$. Hence it should suffice to find a re-distribution such that the values associated to $(KF) \setminus (K_* F_*)$ and to $(\neg K^* F^*) \setminus (\neg KF)$ will be null. In accordance with the atoms refinement, this can be obtained by the following assignments:

$$\begin{cases} z_{r_1}^\alpha = x_r^\alpha & B_{r_1} \neq \phi \\ z_{r_2}^\alpha = 0 & \text{if } B_{r_2} = \phi; \\ z_{r_3}^\alpha = 0 & B_{r_3} = \phi \end{cases} \begin{cases} z_{r_1}^\alpha = 0 & B_{r_2} \neq \phi \\ z_{r_2}^\alpha = x_r^\alpha & \text{if } B_{r_2} \neq \phi; \\ z_{r_3}^\alpha = 0 & B_{r_3} = \phi \end{cases} \begin{cases} z_{r_1}^\alpha = 0 \\ z_{r_2}^\alpha = 0 & \text{if } B_{r_3} \neq \phi \\ z_{r_3}^\alpha = x_r^\alpha \end{cases} . \quad (21)$$

With such re-distribution it easy to check that $\mathbf{k}_* \mathbf{f}_* \cdot \mathbf{x}^\alpha = \mathbf{k} \mathbf{f} \cdot \mathbf{z}^\alpha$ and $\mathbf{f}_* \cdot \mathbf{x}^\alpha = \mathbf{f} \cdot \mathbf{z}^\alpha$ so that also (19) holds.

About (20) the procedure is similar, but since by (17) we have $K^* F^* \supseteq KF$ and $\neg K^* F^* \subseteq \neg KF$, the proper re-distribution is

$$\begin{cases} z_{r_1}^\alpha = 0 & B_{r_1} = \phi \\ z_{r_2}^\alpha = x_r^\alpha & \text{if } B_{r_2} \neq \phi; \\ z_{r_3}^\alpha = 0 & B_{r_3} = \phi \end{cases} \begin{cases} z_{r_1}^\alpha = 0 & B_{r_1} = \phi \\ z_{r_2}^\alpha = 0 & \text{if } B_{r_2} \neq \phi; \\ z_{r_3}^\alpha = x_r^\alpha & B_{r_3} \neq \phi \end{cases} \begin{cases} z_{r_1}^\alpha = x_r^\alpha \\ z_{r_2}^\alpha = 0 & \text{if } B_{r_1} \neq \phi \\ z_{r_3}^\alpha = 0 \end{cases} . \quad (22)$$

This conclude the proof of Proposition 1. $\square$

From the previous proof it is easy to see that the result of the proposition could be strengthened by choosing any sub-interval of $[lb_{K|F}^*, ub_{K|F}^*]$. In fact the following lemma holds:

Lemma 1. *If $(\mathcal{E}^*, \mathcal{L}_C^*, \mathbf{p}^*)$ is a coherent extension restriction obtained as in Proposition 1, then any assessment $(\mathcal{E}^*, \mathcal{L}_C^*, \mathbf{p}^{**})$, where $\mathbf{p}^{**} = \mathbf{p} \cup \{[lb_{K|F}^{**}, ub_{K|F}^{**}]\}$ with $[lb_{K|F}^{**}, ub_{K|F}^{**}] \subseteq [lb_{K|F}^*, ub_{K|F}^*]$, is still coherent.*

Note that, even being stronger, the result of Lemma 1 is more debatable than that of Proposition 1 since the choice of the tighter interval $[lb_{K|F}^{**}, ub_{K|F}^{**}]$ is arbitrary, while that of $[lb_{K|F}^*, ub_{K|F}^*]$ is univocally induced by the class $\mathcal{P}$. In practical applications, as it will be shown in the next section, it is hence recommended to see first if Proposition 1 could bring some valuable hint how to reduce extension bounds, and only after to perform investigations on the inference target $K|F$ to choose a further improvement. Of course, whenever the intersection $[lb_{K|F}^*, ub_{K|F}^*]$ would be empty, Proposition 1 is meaningless. This does not mean that the natural extension $[lb_{K|F}, ub_{K|F}]$ cannot be shrunk, but only that we do not have a systematic indication suggested by the class $\mathcal{P}$. Anyhow, further investigations about the properties of the proposed procedure and about the characterization of the implicit structural constraints induced by it are needed.

3 An Example

In [2] a diagnostic process for *gastrointestinal stromal tumors* (GISTs) was analyzed. Only recently a new and reliable phenotypic marker (the KIT protein CD117) for these neoplasm has been introduced. The KIT protein is not adopted systematically with all the gastrointestinal lesions, but only to those that, after a first analysis, are suspected to belong to the GIST family. More specifically, the diagnosis path consists mainly of two stages: at first a histological analysis is done and later an immunohistochemical schema is adopted to confirm cases previously suspected to be GISTs. In the cited paper the quality of the first discrimination stage was numerically evaluated. Here we can focus only on the phase where results of Proposition 1 and Lemma 1 have been applied, even if at that time they were not systematically formalized and known as in the present paper. Events listed in Table 1 were selected as relevant for a lesion, where the

Table 1. Relevant events for the GIST diagnosis

<i>label</i>	<i>description</i>
SUSPECTED	lesion is histologically suspected to be a GIST
GIST	lesion is really a GIST
CD117	KIT protein expression
CD34	Hematopoietic progenitor cell antigen expression
SMA	Muscle actin expression
DESM	Desmin expression
S100	S-100 protein expression

first two distinguish the suspected tumors from those actually belonging to the GIST's family, while the others represent the positivity for specific immunohistochemical markers. Among them, only the logical restriction $CD117 \subseteq GIST$ holds and it is due to the extreme specificity of the KIT marker. By using very limited available data and after a belief revision induced by an original incoherence between the “knowledge base” and the “rule base”, the assessment reported in Table 2 was given. In a preliminary stage of the analysis, attention was focused on the “a priori” values of GIST's prevalence. Hence coherent extension bounds for $P(GIST)$ where determined. In this particular case there were 40 atoms and only one zero layer, so that the optimization task $\{\mathcal{O}_\alpha^j\}$ consisted of 10 single (i.e. sequences of length 1) couple of linear programs (each couple “associated” to the bounds of the five intervals in Table 2) with 18 constraints (the 17 induced by the assessment plus one of normalization). The result was⁷ $[lb_{GIST}, ub_{GIST}] = [.59, .97]$. Nonetheless, in line with Proposition 1, the intersection of extension intervals obtained by the 10 optimization programs led to the

⁷ Computations performed with the software “Check Coherence Interface” developed by the Research Group of Italian MIUR Cofin Project *PAID (Partial Information and Decision)* with directives reported in [3].

Table 2. The definitive assessment adopted in the GIST diagnosis reliability analysis (precise values estimated by data frequencies, interval values elicited by literature reports)

<i>statement</i>	<i>cond. prob. values or bounds</i>
SUSPECTED	.510
CD117 CD34 ¬DESM ¬S100 SUSPECTED	.380
¬SMA ¬CD117 CD34 DESM ¬S100 SUSPECTED	.077
SMA ¬CD117 CD34 ¬DESM ¬S100 SUSPECTED	.077
SMA CD117 ¬CD34 ¬S100 SUSPECTED	.077
SMA CD117 ¬CD34 ¬DESM S100 SUSPECTED	.077
¬SMA CD117 ¬CD34 ¬DESM ¬S100 SUSPECTED	.077
CD117 GIST	[.95, .99]
CD34 CD117	[.60 , .70]
SMA CD117	[.30 , .40]
S100 CD117	[.096 , .105]
DESM CD117	[.01 , .02]

more informative range $[lb^*_{GIST}, ub^*_{GIST}] = [.806, .931]$. This suggested a further pathologist’s investigation which induced to restrict the value of GIST’s prevalence to be around 81% finally. Hence the limitation $P(GIST) \in [.806, .815]$ was added to the assessment of Table 2 and it was checked to cohere with the rest, even if we know now that this was guaranteed by Lemma 1.

References

1. A. Capotorti. Generalized concept of atoms for conditional events, in *Mathematical Models for Handling Partial Knowledge in Artificial Intelligence*, G.Coletti *et al.* editors, Plenum Press, New York, (1995) 183–189.

2. A. Capotorti, S. Fagundes Leite. Reliability of GIST diagnosis based on partial information. in *Applied Bayesian Statistical Studies In Biology And Medicine*, (M. Di Bacco, G. D’Amore, F. Scalfari editors), Kluwer Academic Publishers, Boston/Dordrecht/London, (2003).

3. A. Capotorti, L. Galli and B. Vantaggi. How to use locally strong coherence in an inferential process based on upper-lower probabilities. *Soft Computing*, 7(5) (2003) 280–287.

4. A. Capotorti, T. Paneni. An operational view of coherent conditional previsions, in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Benferhat, S., Besnard, P. (eds): ECSQUARU 2001, LNAI 2143 (2001) 132–143.

5. A. Capotorti. Benefits of embedding structural constraints in coherent diagnostic processes, *International Journal of Approximate Reasoning* (2004) article in press (corrected proof available at <http://authors.elsevier.com/sd/article/S0888613X04001070>).

6. G. Coletti, R. Scozzafava. Characterization of Coherent Conditional Probabilities as a Tool for their Assessment and Extension. *Int. Journ. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 4(2) (1996) 103–127.

7. G. Coletti and R. Scozzafava: Conditional measures: old and new. In *New trends in Fuzzy Systems*, World Scientific, (1998) 107-120.
8. G. Coletti, R. Scozzafava. Conditioning and Inference in Intelligent Systems, *Soft Computing*, 3 (1999) 118-130.
9. G. Coletti, R. Scozzafava. *Probabilistic Logic in a Coherent Setting*, Dordrecht: Kluwer, Series "Trends in Logic", (2002).
10. B. de Finetti: *Teoria della probabilità*, Einaudi, Torino, 1970. (Engl. transl.: *Theory of Probability*, Vol.1 and 2. Wiley, Chichester, 1974).
11. A. Gilio. Algorithms for precise and imprecise conditional probability assessments, in *Mathematical Models for Handling Partial Knowledge in Artificial Intelligence*, G.Coletti *et al.* editors, Plenum Press, New York, (1995) 231-254.
12. *Proceedings of the International Symposia on Imprecise Probabilities and Their Applications.*, The International Society for Imprecise Probability Theory and Applications, (eletronic versions available at <http://www.sipta.org/isipta/>).
13. F. Lad. *Operational Subjective Statistical Methods: a mathematical, philosophical, and historical introduction*, New York: John Wiley, (1996).
14. P. Walley. *Statistical reasoning with Imprecise Probabilities*, Chapman and Hall, London, (1991).
15. P. Walley, R. Pelessoni and P. Vicig. Direct Algorithms for Checking Coherence and Making Inferences from Conditional Probability Assessments, *Journal of Statistical Planning and Inference*, 126(1) (2004) 119-151.

Type Uncertainty in Ontologically-Grounded Qualitative Probabilistic Matching

David Poole¹ and Clinton Smyth²

¹ Department of Computer Science, University of British Columbia
<http://www.cs.ubc.ca/spider/poole/>

² GeoReference Online Ltd., Vancouver, BC, Canada
<http://www.georeferenceonline.com>

Abstract. This paper is part of a project to match real-world descriptions of instances of objects to models of objects. We use a rich ontology to describe instances and models at multiple levels of detail and multiple levels of abstraction. The models are described using qualitative probabilities. This paper is about the problem of type uncertainty; what if we have a qualitative distribution over the types. For example allowing a model to specify that a meeting is always scheduled in a building, usually in a civic building, and never a shopping mall can help an agent find a meeting even if it is unsure about the address.

1 Introduction

In a recent paper [1], we described a system for matching instances and models of real-world phenomena. These instances and models have been described by different people using controlled vocabularies (using an ontology) which allow descriptions of model and instances at varied levels of abstraction (using more general or less general terms) and detail (describing objects in terms of parts and subparts or not).

In one practical domain, geological surveys of various countries or provinces publish descriptions of mineral occurrences (e.g., twelve thousand in British Columbia) at widely varying levels of abstraction and detail. Other people spend careers developing models of archetypical mineral deposits that can help determine where certain minerals can be found¹. These models are often at different levels of abstraction and detail from the mineral occurrence descriptions. The problem that we consider is to determine which mineral deposits fit which models, so that explorers can focus their exploration. This is a case where humans have to make decisions, but they are overwhelmed by the combinatorics. The aim of the computer system is to find the best fitting models to a mineral occurrence or to find the best fitting mineral occurrences to a model. As it is humans who are making the decisions, it is more important to have good explanation facilities and to return multiple potential matches than to return the “best” match

¹ See <http://www.yukonmineraltargets.com/> for examples using about 2000 mineral occurrences (“anomaly clusters”) in the Yukon, matched against 80 models. It shows the best matching instances for each model and the best matching models for each instance.

according to the computer. Computers help by narrowing down the search space and explaining and justifying the potential matches.

The main problems are the integration of (qualitative) probabilities with rich ontologies and reasoning about multiple objects with multiple subparts. For this paper we'll use the OWL [2] notation where appropriate.

In previous work [1], we make the assumption that different descriptions refer to different objects. This assumption is relaxed in this paper. In particular, we allow for uncertainty in the types and allow for qualitative distributions over hierarchically structured classes. By type we mean membership in a class, where the classes are organised hierarchically and are specified as part of an ontology.

This work is quite different to other work on combining probability and ontologies (e.g., P-Classic [3]) because we are using the ontologies to construct a rich hypothesis space rather than (or as well as) having probabilities over subclasses in the ontologies.

2 Qualitative Probabilistic Matching

The general problem is, given a model and an instance, to determine a qualitative value for $P(model|instance)$ that can be used by a human to make decisions. This section gives an overview of our previous paper [1].

We decided to use qualitative order-of-magnitude probabilities based on the kappa calculus [4, 5]. The kappa calculus can be described in terms of surprise; the kappa values correspond to the level of surprise. When probabilities multiply, the corresponding kappa values add, and summing in probability corresponds to minimisation in the kappa calculus.

The kappa calculus can be seen as a crude approximation to probability [6]. This is useful when the probabilities are not available as it gives a rough answer and leaves only a few of the possible matches for a human to evaluate; the implausible matches can be ignored by the human. Note that we use the kappa calculus for the first-level of approximation; we use some finer distinctions to distinguish matches that may have the same values in the kappa calculus. These are described when used.

2.1 Ontologies

We assume that we have an ontology that specifies the vocabulary and partly specifies the meaning of some of the terms (in terms of how some of the terms relate to each other and some restrictions on allowed values). For this paper, we assume that we are given an ontology that specifies:

- a class hierarchy. Formally a class is a set. The top class is “Thing”. We use *subClassOf* to denote the subclass relation. We assume that the class hierarchy is a tree; siblings in the class hierarchy form disjoint sets of individuals.
- a property hierarchy. Like OWL-DL, we assume that objects, classes and properties are distinct. Each property is either a *datatype* property or an *object* property.
- domains and ranges for properties.
- declarations that properties are functional.

Example 1. As part of the ontology, we can specify a class hierarchy, part of which may be written as tuples:

$$\begin{aligned} &\langle \text{boardroom}, \text{subClassOf}, \text{meeting\_room} \rangle \\ &\langle \text{meeting\_room}, \text{subClassOf}, \text{room} \rangle \\ &\langle \text{room}, \text{subClassOf}, \text{enclosed\_space} \rangle \end{aligned}$$

We can also specify properties and property attributes, such as:

$$\begin{aligned} &\langle \text{held\_in}, \text{subPropertyOf}, \text{has\_location\_in} \rangle \\ &\langle \text{held\_in}, \text{domain}, \text{meeting} \rangle \\ &\langle \text{held\_in}, \text{range}, \text{room} \rangle \end{aligned}$$

2.2 Describing Instances

Instances are things in the world that are described using the vocabulary of the ontology. Instances are given (internal) names.

We make the open world assumption: we do not assume that we are told everything about an individual, and so do not conclude anything from the lack of information. If we want to say that something is not true, we need to say explicitly that its value is absent, or say that there are no other values. Instances have property values that are marked as “present” or “absent”.

We use the term *object* to denote both an instance object and model object. The room at the North-East Corner of 123 Pretty St, Vancouver, may be an instance of a kitchen. A room in the model of houses that Joe likes may be a model of a kitchen.

Instances are described using tuples of the form:

$$\langle \text{Object}, \text{Property}, \text{Value}, \text{TruthStatus}, \text{Reference}, \text{Comment}, \text{EnteredBy} \rangle$$

where *Frequency* is either *present* or *absent*. *Reference*, *Comment* and *EnteredBy* specify the source of the information and human-readable comments. The rest of this paper ignores these fields, but it is important for us to attribute the source of all knowledge.

If *Property* is a datatype property, *Value* is a primitive data type (such as a string or a number or a pair of numbers representing a range). If *Property* is an object property, *Value* is (a reference to) an object.

Example 2. Assume we are using the ontology partly specified in Example 1. Consider the following instance of a meeting which we will call m_1 :

$$\begin{aligned} &\langle m_1, \text{held\_in}, \text{rmCS123}, \text{present} \rangle \\ &\langle m_1, \text{organised\_by}, \text{david}, \text{present} \rangle \\ &\langle m_1, \text{start\_time}, 5 : 00\text{pm}, \text{present} \rangle \\ &\langle m_1, \text{attended\_by}, \text{bill}, \text{absent} \rangle \end{aligned}$$

This meeting refers to a room *rmCS123*, which can similarly be described:

$$\begin{aligned} &\langle \text{rmCS123}, \text{type}, \text{boardroom}, \text{present} \rangle \\ &\langle \text{rmCS123}, \text{located\_in}, \text{CSbuilding}, \text{present} \rangle \\ &\langle \text{rmCS123}, \text{capacity}, 50, \text{present} \rangle \\ &\langle \text{rmCS123}, \text{contains}, \text{lectern}, \text{absent} \rangle \end{aligned}$$

2.3 Describing Models

Models are specified using tuples of the form:

$\langle \textit{Object}, \textit{Property}, \textit{Value}, \textit{Frequency}, \textit{Reference}, \textit{Comment}, \textit{EnteredBy} \rangle$

where the attributes are as before, and *Frequency* specifies the qualitative probability that the *Object* will have *Value* for *Property* in the current model. After feedback from domain experts, we use a 5-value frequency scale² to describe models. Suppose p represents the proposition “*Object* has *Value* for *Property*”, the frequency is one of:

- *always*: you are very surprised if p is false³.
- *usually*: you are somewhat surprised if p is false.
- *sometimes*: you aren’t surprised if p is true or if it’s false
- *rarely*: you are somewhat surprised if p is true.
- *never*: you are very surprised if p is true.

These provide a language for *inputting* uncertainty. We output a numerical value in the range [0,100] where 100 is the score for the best possible match and 0 is for the worst possible match. Internally we use a reasonably arbitrary numerical scale.

Example 3. We can specify a model of a particular type of meeting that we should attend. It is usually in a boardroom, always organised by an administrator, usually attended by a department head. This can be specified using tuples:

$\langle \textit{ImportantMeeting}, \textit{held\_in}, \textit{RoomOfImportantMeeting}, \textit{usually} \rangle$
 $\langle \textit{RoomOfImportantMeeting}, \textit{type}, \textit{boardroom}, \textit{always} \rangle$
 $\langle \textit{ImportantMeeting}, \textit{organised\_by}, \textit{OrganizerOfImportantMeeting}, \textit{always} \rangle$
 $\langle \textit{OrganizerOfImportantMeeting}, \textit{type}, \textit{administrator}, \textit{always} \rangle$
 $\langle \textit{ImportantMeeting}, \textit{attended\_by}, \textit{ADepartmentHead}, \textit{usually} \rangle$
 $\langle \textit{ADepartmentHead}, \textit{type}, \textit{department\_head}, \textit{always} \rangle$

2.4 Matching

If not for different levels of abstraction and different levels of detail, to compute the qualitative counterpart of $P(\textit{instance}|\textit{model})$, we add the *surprises* of the instance with respect to the qualitative probabilities specified in the model. The main contribution of [1] was to show how the kappa calculus could be combined with rich ontologies that let us describe models and instances at various levels of abstraction and detail.

As we are adding surprises, and returning the topmost (least surprising) match, the zero point is arbitrary. We can define zero to be the level of the empty match⁴ (i.e., a

² None of the theory or results in this paper depends on using this scale, but we will use it in all of our examples. In practice, we have found that experts are happy using this scale, and find it very natural.

³ This is *not* the always of modal logic. Even though our experts described things as “always” true, we allow for exceptions due to errors in descriptions.

⁴ This is for the case of the open-world assumption, where we don’t assume that a complete description is given. We do allow someone to specify that “silence implies absence”; that a part or property that is not described is false. In this case an empty description does not have value zero. It is positive as we expect nothing else and found nothing else.

match with an empty description), then we have positive rewards when there is a better match than this and negative rewards (penalties) for those matches that are worse than this.

For each model qualitative probability and for each instance value “present” or “absent”, we will have a numerical reward or penalty. Thus, for example, we will talk about the always-present reward (which gives the reward received when a model property that has qualitative probability “always” matches an corresponding instance property that is present) or a usually-absent penalty (when the model property is “usually” present, but it is absent in the instance). For example, if a model specifies a room that is always a bedroom and usually pink and we have an instance that is a bedroom that is not pink (i.e., bedroom is present and pink is absent), we get both the always-present reward and the usually-absent penalty.

Given an abstraction hierarchy of classes, it is important to distinguish the description of an instance from the instance itself. For example, if something is described as a building, it must be some sort of building (generic buildings don’t exist). One of the differences between an instance and a model is, when given a general concept, such as “building”, in an instance we *don’t know* what sort of building it is, but if the same term is used in a model, we *don’t care* what sort of building it is [1]. When we want the probability of an instance, we don’t want the probability of the description. The probability of a more abstract description is more likely than the that of a more specific instance. For or example, a house is a kind of building, so for any evidence e , $P(\text{building}|e) > P(\text{house}|e)$. However if the model specifies a house is always present, and instance 1 is described as a building and instance 2 is described as a house, then instance 2 definitely fits the model, but it is less likely that instance 1 fits the model.

Our previous paper [1] made two assumptions that we relax in this paper:

- The type of objects was known.
- In a single description, different descriptions of parts (or other property values) pertain to different parts (or property values).

3 Type Uncertainty

In many real examples, we may have uncertainty about the type of an object. We would like to have a qualitative distribution over which class an object is a member of. For example, the model may specify that a place that can take the role of a home office is always a room, usually a bedroom and rarely a master bedroom.

We also need to distinguish between one object satisfying multiple type restrictions and multiple objects satisfying them. Consider the following two contrasting examples:

Example 4. Suppose we had a a model of “a room that Sam likes” that says the room is “usually red and never pink”. This could mean either

- it has a single colour that is usually a non-pink shade of red, or
- there could be multiple colours; as long as one is red and one is non-pink they would be happy. So a blue and pink would be good; they just don’t want all-pink or no shade of red.

In the first case, the description is referring to a single colour and in the second to multiple colours.

Example 5. Suppose we have a model for a house that always contains a bathroom and always contains a room that is not a bathroom. In this case, we don't want this to refer to the same room (there is no room that is both a bathroom and is not a bathroom), but rather to two rooms.

We need a way to specify we are referring to a single colour or room or to multiple colours or rooms.

In summary, we need a representation to specify what objects are assumed to exist and what type distributions are over the objects. The same issues arise when specifying functional properties (as in colour above). We motivate the problems in terms of types, but then treat type as a (functional) property in the algorithm.

In order to understand the issues, we will give a detailed example of a specific example of matching the type description of an instance and a type description of model type. This example will be used as a prototype for the general case.

Example 6. Consider the class hierarchy of Figure 1. Some of the relations that are true include:

$\langle \text{House}, \text{subClassOf}, \text{ResidentialBuilding} \rangle$
 $\langle \text{ResidentialBuilding}, \text{subClassOf}, \text{Building} \rangle$
 $\langle \text{Building}, \text{subClassOf}, \text{EngineeredArtifact} \rangle$

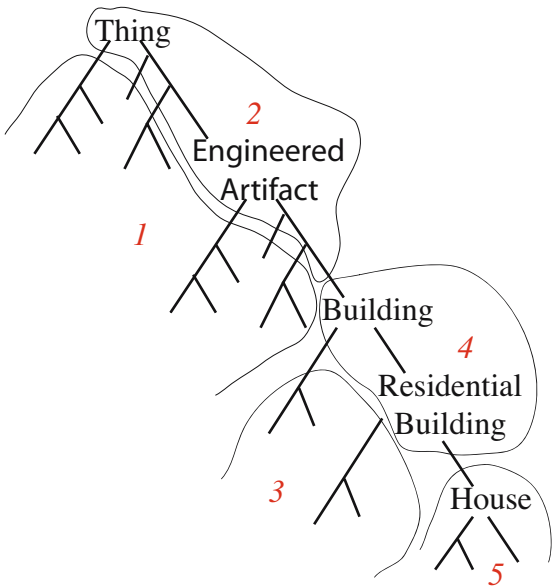


Fig. 1. A class hierarchy. Different exceptional regions for model M_1 are numbered (see Example 6)

For this discussion, we do not intend that these are immediate subclasses. There may be many intermediate classes (e.g., that are classes that are subclasses of “Engineered Artifact” and super classes of “Building”). At the root of the tree is *Thing* which is the topmost class.

Suppose a model that specifies that the location of some activity is:

always a Building and rarely a House.

Call this model M_1 .

Given a simple description of an instance where we only give a single class that is present, let’s determine how surprised we are that the location is at that instance. The description could be any position in the taxonomy, and the instance could be any leaf that is a descendant of the class that is said to be present. The figure shows the five qualitatively different regions of the taxonomy that the description of an instance could be in:

- Region 1. If the description is in region 1, the cousins of *Building* (the values in the same tree that are neither descendants nor ancestors of *Building*), the instance is not a *Building*.
- Region 2. The description is an ancestor of *Building* so the instance is perhaps a *Building*, but could be a non-building.
- Region 3. The instance is a *Building* and not a *House*.
- Region 4. The instance is a *Building* and perhaps a *House*.
- Region 5. The instance is a *House*.

Consider how surprised we would be that an object in each of these locations would be an instance of the model “*always a Building and rarely a House*”.

If the instance description is in region 1, the instance would receive the always-absent penalty. The model says the instance is always a building and the instance is not a building.

Suppose description d_2 of the instance is in region 2, for example it is the description *Engineered Artifact*. We don’t know if the instance is a *Building* or not. To understand the qualitative probability, consider the probability of the model M_1 given the description d_2 :

$$P(M_1|d_2) = P(M_1|Building \wedge d_2) * P(Building|d_2) \\ + P(M_1|\neg Building \wedge d_2) * P(\neg Building|d_2)$$

$P(M_1|Building \wedge d_2) = P(M_1|Building)$ as $Building \wedge d_2$ is logically equivalent to *Building*. $P(M_1|\neg Building \wedge d_2) = P(M_1|\neg Building)$ as the model doesn’t specify anything more general than *Building*. Thus

$$P(M_1|d_2) = P(M_1|Building) * P(Building|d_2) \\ + P(M_1|\neg Building) * P(\neg Building|d_2)$$

In terms of the kappa-calculus (taking logs, ignoring adding by zero, and minimising):

$$\kappa(M_1|d_2) = \min(\kappa(Building|d_2), \kappa(M_1|\neg Building))$$

assuming that

- $\kappa(\neg Building|d_2) = 0$; we are not surprised that the *Engineered Artifact* is not a building. We would be surprised if it is a building as there are many more sorts of engineered artifacts than buildings.
- $\kappa(M_1|Building) = 0$; we are not surprised that a building matches the model M_1 as the model M_1 specifies the object is always a building.

Thus the “surprise” that the engineered artifact fits the model comes from either the surprise that the *Engineered Artifact* is a building or the surprise that a non-building matches the model. Our level of surprise is the minimum of these two.

We could have surprise information as part of our ontology. For each element in the taxonomy, we would have a value of how surprising each child is. For example, we could infer the surprise of *Building* given the description *Engineered Artifact*.

If we didn’t have the information in the taxonomy, we can make some simplifying assumptions to estimate this value. Suppose the description d_2 is m levels in the hierarchy above *Building*, and suppose that the average branching factor of b , and that the children of any node have approximately equal probability. Then $P(Building|d_2) \approx (1/b)^m$. Taking logarithms, we see that the surprise that the instance is a *Building* should be linear in m .

We do not use the kappa calculus directly, as this would entail having a surprise that there is no description. We’d rather just ignore non-existent descriptions. This can be done by defining the surprise value of a empty description as zero. We get positive rewards for being less surprised than this and negative rewards (penalties) for being more surprised. Given that we know something exists, not specifying a value is the same as stating it has the top value (*Thing* in the above taxonomy). This then gives us a way to calibrate the surprise. The value is zero when the description is *Thing*. If the description is not *Thing*, but in region 2, then it should have a positive reward, as it is more likely a *Building* than if it were a *Thing*. Under the assumptions made above⁵, this should be linear in the depth.

If the assumption that children are approximately equal is not a reasonable assumption, it is possible to specify the surprise values for each child in the taxonomy as part of the ontology. For example, specifying how surprised you are that residential building is a house. Note that in this case it is possible to specify a model that is surprised by a normal condition; in this case, the model should also be surprised by a empty description. For example, if things are usually engineered artifacts, but a model specifies that fits to the model are rarely engineered artifacts, then the model should be surprised by a description of an object just as *Thing*.

If the instance is in region 5, it gets the rarely-present penalty. We know the instance is a type of house; the model specifies that we should be surprised the location is a house.

In order to understand the reward of an instance in region 3, it is instructive to consider some more models. Suppose model M_0 is “*always a Building*”, model M_1 is “*always a Building and rarely a house*” and model M_2 is “*always a Shopping Centre*”.

⁵ The main assumption is that surprises are not specified as part of the taxonomy and that all children are approximately equal. This means all children along the path from *Thing* are approximately equal, not that a child is equal to all of its sibling.

Then we have M_2 subsumes M_1 (given that *Shopping Centre* is a subclass of *Building* that is disjoint with *house*) and M_1 subsumes M_0 . If the instance is a *Strip Mall* (a subclass of *Shopping Centre*), it matches all three models. As M_0 and M_2 give the same match; an always-present reward, it seems reasonable to also give the match with M_1 the same reward. The match with M_1 should thus not also get a rarely-absent reward.

Instances in region 4, are known to be buildings and they could be houses. If we do a similar probabilistic analysis to region 2, with d_4 a description on region 4, we get:

$$\begin{aligned} P(M_1|d_4) &= P(M_1|House \wedge Building) * P(House|d_4) \\ &\quad + P(M_1|\neg House \wedge Building) * P(\neg House|d_4) \end{aligned}$$

(as $P(Building|d_4) = 1$). If you just consider the second part of the sum, you are not surprised that the model is true for a building that is not a house (it is “always” true), you are also not surprised that a building in region 4 is not a house. Thus in terms of the kappa values, this has kappa value 0. That is, $\kappa(M_1|d_4) = 0$.

However, this is considered to be a worse match than for an instance in region 3, and so gets a small penalty that is proportional to the depth of the description. This value is designed to be dominated by the kappa values so that it only distinguished instances that have the same kappa values.

4 Matching Algorithm

Under these assumptions, there is a canonical form for the value(s) of the type of an instance. Because the declarations are implicitly conjoined, you can assume there is exactly one “present” class for any functional property (including type) and a number of absent classes that are subclasses of the present class. This is because you can always assume that the top element is present, and if a class and a subclass are both present, you can remove the superclass as present and preserve the meaning. There can’t be two classes that are both “present” if one is not a subclass of the other if the hierarchies are trees (as there are no elements in common between the classes).

Similarly we can assume that for the value of a type of model object, there is at most one *always*, at most one *usually*, that frequencies go down in the hierarchy, there are no children of *never* or cousins of *always*. The only cousins of a *usually* are *nevers*. [Note that *sometimes* is used when we have a complete knowledge assumption; it will be ignored in this section.]

Figure 2 gives an algorithm to determine the score for matching the type descriptions of an instance object and a model object. Suppose I is an instance object that is to be matched with a model object M . We use the notation $I.present$ to be the position in the hierarchy of the value of the instance that is declared to be present. Similarly $M.always$ is the position in the hierarchy of the value declared to be always true in model M . Conditions involving $M.always$ are assumed to be false if nothing is declared to be always true in model M . Below and above refer to positions in the hierarchy (above is more general), and a node is below itself and is above itself.

A non-obvious aspect of this algorithm is when the model has an “always” above a “usually”. In this case, if an instance has *present* below the *always*, but a cousin

procedure scoreMatch**Inputs:**

Model Description M
 Instance Description I

Returns:

Score

begin

```

if ( $I.present$  is below a  $M.never$ )
  return never-present reward
else if ( $I.present$  is cousin of  $M.always$ )
  return never-present reward
else if ( $I.present$  is below a  $M.rarely$ )
  return rarely-present reward
else if ( $I.present$  is cousin of  $M.usually$ )
  return rarely-present reward
else if ( $I.absent$  is above  $M.always$ )
  return always-absent reward
else if ( $I.absent$  is above  $M.usually$ )
  return usually-absent reward
else if ( $I.present$  is below  $M.usually$ )
  return usually-present reward
else if ( $I.present$  is above  $M.usually$ )
  return  $\alpha \times$  usually-present reward
    where  $\alpha$  is depth of  $I.present$  / depth of  $M.usually$ 
else if ( $I.present$  is above  $M.always$ )
  return  $\alpha \times$  always-present reward
    where  $\alpha$  is depth of  $I.present$  / depth of  $M.always$ 

```

end

Fig. 2. Determining Reward from Type description of Instance I and Model M

of the *usually*, it gets just the usually-absent penalty, and no reward for the always-present. This is reasonable as, if the always was not there, this would be equivalent to having always at the top. If the instance has *present* above the *usually*, it has a reward that is linear in the depth of the *present* instance, independent of the position of the *always*.

5 Matching Parts

In order to be able to match complex description, we need to consider the case where the value may be an object that has proerties with values. In the case when the model and the instance both have have complex descriptions, we need to determine the correspondencies between these complex descriptions. If the instance has multiple instances of the property (and the property is not functional), we need to determine which model values correspond to which instance values.

Example 7. Consider the value of the *organized_by* property of M_1 in Figure 3. If there exists a person who fits the description of P_1 , there should be a reward, and if there doesn't there should be a penalty. If there are multiple people, we need to choose the most appropriate one to fill the role.

To understand how this works, it is constructive to consider a full probabilistic analysis of the probability of model M_1 given instance I_1 :

$$\begin{aligned} P(M_1|I_1) &= P(M_1|(\exists P_1) \wedge I_1)P((\exists P_1)|I_1) \\ &\quad + P(M_1|(\neg \exists P_1) \wedge I_1)P((\neg \exists P_1)|I_1) \\ &= P(M_1|(\exists P_1))P((\exists P_1)|I_1) \\ &\quad + P(M_1|(\neg \exists P_1))P((\neg \exists P_1)|I_1) \end{aligned}$$

where $\exists P_1$ is shorthand for there exists a P_1 that matches the description of P_1 in Figure 3. The model gives the qualitative value $P(M_1|(\exists P_1))$. To compute the qualitative values of $P((\exists P_1)|I_1)$ we find the best match of P_1 to the values of *organized_by* in I_1 .

Taking the qualitative version of this formula, we replace multiplication by addition and the addition by minimum in the kappa calculus or maximum in our system. The works if we expect $(\exists P_1)$ to be true (the frequency of *organized_by* in M_1 is always or usually) and we find that we have positive support for $(\exists P_1)$.

Unfortunately, the other cases are not as straightforward. We cannot readily compute $\kappa((\neg \exists P_1)|I_1)$ as if $(\exists P_1)$ has no surprise, then its negation has some surprise, but we don't know how much, and if $(\exists P_1)$ has some surprise, then its negation has no surprise. We have chosen a simple scheme that gives intuitive results, as follows.

If the model specifies we are surprised that $(\exists P_1)$ (the frequency is "rarely") and we are not surprised that it true in the instance (i.e., $(\exists P_1)|I_1$ is positive), we get the surprise of the rarely (the *rarely-present* reward).

If the qualitative probability of $(\exists P_1)$ in I_1 is negative, we give the appropriate *always-absent, usually-absent, ... never-absent* reward.

For example, if the instance had multiple organisers, we need to determine which one best fulfils the role specified in the model. In terms of the kappa calculus, the distribution over which instance fulfils the role becomes a minimisation of surprised (maximisation of scores). We do this by choosing the one with the highest score. Then we consider the model frequency and how surprised we are than an organiser of the appropriate type exists.

Object	Property	Value	Frequency
M_1	<i>has_location</i>	L_1	usually
M_1	<i>starts_at</i>	T_1	always
M_1	<i>organized_by</i>	P_1	always
...	...	...	...
P_1	<i>type</i>	<i>Admin_staff</i>	usually
P_1	<i>type</i>	<i>Dept_head</i>	rarely
P_1	<i>type</i>	<i>Financial_officer</i>	never

Fig. 3. An example model of a research meeting

6 Conclusion

This paper has grown out of a project to build knowledge-based decision tools in various domains such as minerals exploration, geological hazards (landslides, earthquakes), land-use planning. We needed qualitative reasoning and rich ontologies. We don't have the probabilistic knowledge or the utilities to do full decision theory, but have developed a system that uses a small but natural set of qualitative probabilities that can integrate with the ontologies being developed and with the sort of knowledge about instances and models that can be obtained. This paper outlined how we are handling cases where a functional relation has qualitative probabilistic constraints on what values it can take (some are more surprising than others). We have made some pragmatic choices that seem to work in practice, but there is much more theoretical and empirical work that needs to be carried out.

Acknowledgements

Thanks to Erica Huang for valuable feedback and comments on this paper.

References

1. Smyth, C., Poole, D.: Qualitative probabilistic matching with hierarchical descriptions. In: KR-04, Whistler, BC, Canada (2004)
2. McGuinness, D.L., van Harmelen, F.: Owl web ontology language overview. W3C Recommendation 10 February 2004, W3C (2004)
3. Koller, D., Levy, A., Pfeffer, A.: P-classic: A tractable probabilistic description logic. In: Proc. 14th National Conference on Artificial Intelligence, Providence, RI (1997) 390–397
4. Spohn, W.: A general non-probabilistic theory of inductive reasoning. In: Proc. Fourth Workshop on Uncertainty in Artificial Intelligence. (1988) 315–322
5. Pearl, J.: Probabilistic semantics for nonmonotonic reasoning: A survey. In Brachman, R.J., Levesque, H.J., Reiter, R., eds.: Proc. First International Conf. on Principles of Knowledge Representation and Reasoning, Toronto (1989) 505–516
6. Darwiche, A., Goldszmidt, M.: On the relation between kappa calculus and probabilistic reasoning. In: Proc. Tenth Conf. on Uncertainty in Artificial Intelligence (UAI-94). (1994) 145–153

Some Theoretical Properties of Conditional Probability Assessments

Veronica Biazzo¹ and Angelo Gilio²

¹ Dipartimento di Matematica e Informatica, Università degli Studi di Catania,
Città Universitaria, Viale A. Doria 6, 95152 Catania, Italy

`vbiazzo@dmi.unict.it`

² Dipartimento di Metodi e Modelli Matematici, Università “La Sapienza”,
Via A. Scarpa 16, 00161 Roma, Italy

`gilio@dmmm.uniroma1.it`

Abstract. We consider a finite family of conditional events and, among other results, we prove a connection property for the set of coherent assessments on such family. This property assures that, for every pair of coherent assessments on the family, there exists (at least) a continuous curve $\mathcal{C}$ whose points are intermediate coherent probability assessments. We also consider the compactness property for the set of coherent assessments. Then, as a corollary of connection and closure properties, we obtain the theorem of extension for coherent conditional probabilities.

Keywords: Conditional events, coherence, connection, compactness, theorem of extension.

1 Introduction

The probabilistic treatment of uncertainty is one of the relevant tools in many applications of Artificial Intelligence, such as uncertain reasoning. When facing real problems typically the set of uncertain quantities at hand has no particular algebraic structure. Then, a flexible approach to reasoning under uncertainty can be obtained by using precise and/or imprecise probabilities, based on the coherence principle of de Finetti and suitable generalizations of it, or on similar principles like that ones adopted for lower and upper probabilities ([1], [2], [3], [4], [5], [6], [9], [12], [14], [15], [16], [17], [18]).

In this paper we consider probability assessments on an arbitrary family $\mathcal{F}_n$ of n conditional events. After some preliminary results, we prove a connection property for the set Π_n of coherent probability assessments on $\mathcal{F}_n$. This property assures that, for every pair $\mathcal{P}' \in \Pi_n, \mathcal{P}'' \in \Pi_n$, there exists (at least) a continuous curve $\mathcal{C} \subseteq \Pi_n$ (from $\mathcal{P}'$ to $\mathcal{P}''$) whose points are intermediate coherent probability assessments between $\mathcal{P}'$ and $\mathcal{P}''$. We also give a simple proof of the compactness property for the set Π_n . Then, as a simple corollary of such results, we obtain the well known theorem of extension for coherent conditional probability assessments.

The paper is organized as follows: in Section 2 we recall some preliminary notions and results. In Section 3 we introduce the connection property by means of an example. In Section 4, after some preliminary results, we prove in general the connection property; moreover, we give a simple corollary. In Section 5 we examine the compactness property; then, exploiting the properties of connection and closure, we give a simple proof of the theorem of extension for coherent conditional probabilities. Finally, in Section 6 we give some conclusions.

2 Preliminaries

We recall some notions and results on coherence of conditional probability assessments. We denote by A^c the negation of A and by $A \vee B$ (resp., AB) the disjunction (resp., the conjunction) of A and B ; moreover, for each integer n , we set $J_n = \{1, 2, \dots, n\}$. Given a real function P defined on an arbitrary family of conditional events $\mathcal{K}$, let $\mathcal{F}_n = \{E_i|H_i, i \in J_n\}$ be a finite subfamily of $\mathcal{K}$ and $\mathcal{P}_n$ the vector $(p_i, i \in J_n)$, where $p_i = P(E_i|H_i)$. The constituents generated by $\mathcal{F}_n$ are obtained by expanding the expression

$$(E_1H_1 \vee E_1^cH_1 \vee H_1^c) \wedge \cdots \wedge (E_nH_n \vee E_n^cH_n \vee H_n^c) \quad (1)$$

and by eliminating all the conjunctions which coincide with the impossible event. We set $C_0 = H_1^c \cdots H_n^c$ (of course, it may be $C_0 = \emptyset$); moreover, we denote by $C_1, \dots, C_m$ the other constituents, which are all contained in $\mathcal{H}_n = H_1 \vee \cdots \vee H_n$. Then, with the pair $(\mathcal{F}_n, \mathcal{P}_n)$ we associate the random gain

$$G_n = \sum_{i \in J_n} s_i H_i (E_i - p_i), \quad (2)$$

where $s_1, \dots, s_n$ are arbitrary real numbers and E_i, H_i denote the indicators of the corresponding events. We denote by g_h the value of G_n corresponding to C_h and by $G_n|_{\mathcal{H}_n}$ the restriction of G_n to $\mathcal{H}_n$. Of course, $G_n|_{\mathcal{H}_n} \in \{g_1, \dots, g_m\}$. Then, using the *betting scheme*, we recall the following

Definition 1. The function P is said coherent if and only if

$$\max G_n|_{\mathcal{H}_n} \geq 0, \quad \forall n \geq 1, \quad \forall \mathcal{F}_n \subseteq \mathcal{K}, \quad \forall s_1, \dots, s_n \in \mathbb{R}. \quad (3)$$

Notice that, if P is coherent, then P satisfies all the axioms of a conditional probability; the converse is not true, as it can be shown by suitable counterexamples (see [10], example 13; [12], example 8).

In [10], to remark that a function P satisfying the axioms of a conditional probability may be not coherent, P is called a *weak conditional probability*.

Given any vector $\Lambda = (\lambda_r, r \in J_m)$, we denote by $\sum_{H_j} \lambda_r$ (resp., $\sum_{E_j H_j} \lambda_r$) the sum of the λ_r 's such that $C_r \subseteq H_j$ (resp., $C_r \subseteq E_j H_j$). Then, with the pair $(\mathcal{F}_n, \mathcal{P}_n)$ we associate the following system (with nonnegative unknowns $\lambda_1, \dots, \lambda_m$)

$$\sum_{E_j H_j} \lambda_r = p_j \sum_{H_j} \lambda_r, \quad j \in J_n; \quad \sum_{r \in J_m} \lambda_r = 1; \quad \lambda_r \geq 0, \quad r \in J_m. \quad (4)$$

We remark that, by a suitable alternative theorem ([11], Theorem 2.9), the set of solutions S of system (4) is non empty if and only if $\max G_n | \mathcal{H}_n \geq 0$.

We set

$$\Phi_j(\Lambda) = \sum_{H_j} \lambda_r, \quad j \in J_n; \quad I_0 = \{j \in J_n : \max_{\Lambda \in S} \Phi_j(\Lambda) = 0\}. \quad (5)$$

We remark that, for each solution Λ of system (4) it is

$$\sum_{j \in J_n} \Phi_j(\Lambda) = \sum_{j \in J_n} \sum_{H_j} \lambda_r \geq \sum_{r \in J_m} \lambda_r = 1. \quad (6)$$

Therefore, $\Phi_j(\Lambda) > 0$ for at least a subscript j and hence I_0 is a strict subset of J_n . Denoting by $\mathcal{P}_0$ the sub-assessment associated with I_0 , we have ([12]).

Theorem 1. The probability assessment $\mathcal{P}_n$ on $\mathcal{F}_n$ is coherent if and only if the following conditions are satisfied:

1. the system (4) is solvable;
2. if $I_0 \neq \emptyset$, then $\mathcal{P}_0$ is coherent.

3 An Example on Connection Property

In this section we illustrate the connection property of the set of coherent conditional probability assessments, by examining an example with three conditional events. Such property will be proved in general in next section.

Let be given the events A, B, H , with $AH \neq \emptyset$ (so that $H \neq \emptyset$ too); moreover, let us consider the family $\mathcal{F}_3 = \{A|H, B|AH, AB|H\}$. As is well known, the set of coherent probability assessments (x, y, z) on $\mathcal{F}_3$ is

$$\Pi_3 = \{(x, y, z) \in [0, 1]^3 : z = xy\}.$$

We observe that Π_3 is not convex; hence, given any pair of coherent probability assessments on $\mathcal{F}_3$, $\mathcal{P}_1 = (x_1, y_1, x_1y_1)$, $\mathcal{P}_2 = (x_2, y_2, x_2y_2)$, and denoting by $\mathcal{P} = (x, y, z)$ any point of the segment $\mathcal{P}_1\mathcal{P}_2$, in general one has $\mathcal{P} \notin \Pi_3$; that is, $\mathcal{P}$ is not coherent. However, if we are interested in making some (coherent) assessment $\mathcal{P} = (x, y, xy)$ which is "intermediate" between $\mathcal{P}_1$ and $\mathcal{P}_2$, i.e. such that

$$\min \{x_1, x_2\} \leq x \leq \max \{x_1, x_2\}, \quad \min \{y_1, y_2\} \leq y \leq \max \{y_1, y_2\},$$

$$\min \{x_1y_1, x_2y_2\} \leq xy \leq \max \{x_1y_1, x_2y_2\},$$

we can choose it in an infinite number of ways. For instance, assuming

$$x_1 \leq x_2, \quad y_1 \geq y_2, \quad x_1y_1 = c_1 \leq c_2 = x_2y_2,$$

any coherent assessment $\mathcal{P} = (x, y, xy)$, such that

$$x_1 \leq x \leq x_2 \frac{y_2}{y_1}, \quad \max \{y_1 \frac{x_1}{x}, y_2\} \leq y \leq y_1,$$

satisfies the inequalities

$$x_1 \leq x \leq x_2, \quad y_2 \leq y \leq y_1, \quad x_1 y_1 \leq xy \leq x_2 y_2;$$

hence, $\mathcal{P}$ is intermediate between $\mathcal{P}_1$ and $\mathcal{P}_2$.

In general, we can construct an infinite number of continuous curves $\mathcal{C}$ connecting $\mathcal{P}_1$ and $\mathcal{P}_2$, with $\mathcal{C} \subseteq \Pi_3$, as is shown by the following examples:

- (i) defining $\mathcal{P} = (x_2, y_1, x_2 y_1)$, the two segments

$$\mathcal{P}_1 \mathcal{P} = \{(x, y_1, xy_1) : x = x_1 + t(x_2 - x_1), 0 \leq t \leq 1\},$$

$$\mathcal{P} \mathcal{P}_2 = \{(x_2, y, x_2 y) : y = y_1 + t(y_2 - y_1), 0 \leq t \leq 1\},$$

belong to Π_3 . Then, the polygonal $\mathcal{C} = \mathcal{P}_1 \mathcal{P} \cup \mathcal{P} \mathcal{P}_2$ is contained in Π_3 and connects $\mathcal{P}_1, \mathcal{P}_2$.

- (ii) defining $\mathcal{P} = (x_1, y_2, x_1 y_2)$, the polygonal $\mathcal{C} = \mathcal{P}_1 \mathcal{P} \cup \mathcal{P} \mathcal{P}_2$ is contained in Π_3 and connects $\mathcal{P}_1, \mathcal{P}_2$.

- (iii) let be given, for suitable values a, b, c , the arc of parabola

$$\mathcal{A} = \{(x, y) \in [0, 1]^2 : \min \{x_1, x_2\} \leq x \leq \max \{x_1, x_2\}, y = ax^2 + bx + c\}.$$

Then, the curve $\mathcal{C} = \{(x, y, z) : (x, y) \in \mathcal{A}, z = xy = ax^3 + bx^2 + cx\}$ is contained in Π_3 and connects $\mathcal{P}_1, \mathcal{P}_2$.

- (iv) more in general, given a suitable function $f(x)$ such that

$$\Sigma = \{(x, y) : \min \{x_1, x_2\} \leq x \leq \max \{x_1, x_2\}, y = f(x)\} \subset [0, 1]^2,$$

the curve $\mathcal{C} = \{(x, y, z) : (x, y) \in \Sigma, z = xy = x f(x)\}$ is contained in Π_3 and connects $\mathcal{P}_1, \mathcal{P}_2$.

4 On the Connection Property of the Set of Conditional Probability Assessments

In this section we prove in general the connection property illustrated by the examples of Section 3. Given a family of n conditional events $\mathcal{F}_n$, we denote by Π_n the set of coherent probability assessments on $\mathcal{F}_n$. We prove that Π_n is connected. More specifically, we prove that, for each pair of points

$$\mathcal{P}' = (p'_i, i \in J_n) \in \Pi_n, \quad \mathcal{P}'' = (p''_i, i \in J_n) \in \Pi_n,$$

there exists a continuous curve $\mathcal{C}$, contained in Π_n , which connects $\mathcal{P}'$ with $\mathcal{P}''$ and is such that, for every $\mathcal{P} \in \mathcal{C}$, $\mathcal{P}$ is intermediate between $\mathcal{P}'$ and $\mathcal{P}''$ (in general, the number of curves of this kind is infinite). We set

$$\begin{aligned} p_i^m &= p'_i \wedge p''_i = \min \{p'_i, p''_i\}, & p_i^M &= p'_i \vee p''_i = \max \{p'_i, p''_i\}, & i &\in J_n; \\ \mathcal{P}^m &= \mathcal{P}' \wedge \mathcal{P}'' = (p_i^m, i \in J_n), & \mathcal{P}^M &= \mathcal{P}' \vee \mathcal{P}'' = (p_i^M, i \in J_n). \end{aligned} \tag{7}$$

Moreover, given a coherent assessment $\mathcal{P}_n$, we set $\Gamma_0 = J_n \setminus I_0$, with I_0 defined by (5); that is, Γ_0 is the subset of J_n such that, for each $i \in \Gamma_0$, there exists a solution $A_i = (\lambda_r^{(i)}, r \in J_m)$ of the system (4) such that $\Phi_i(A_i) = \sum_{H_i} \lambda_r^{(i)} > 0$. Before the main theorem, we give some preliminary results.

Theorem 2. Given a probability assessment $\mathcal{P}_n = (p_i, i \in J_n)$ on $\mathcal{F}_n = \{E_i | H_i, i \in J_n\}$, assume that the system (4) is solvable. Then, there exists a solution $\Lambda = (\lambda_r, r \in J_m)$ of (4) such that $\Phi_j(\Lambda) > 0, \forall j \in \Gamma_0$.

Proof. For each $i \in \Gamma_0$ it is $\max \Phi_i > 0$; hence there exists a subset of the set of solutions S , which we denote by $\{A_i = (\lambda_r^{(i)}, r \in J_m), i \in \Gamma_0\}$, such that $\Phi_i(A_i) > 0, i \in \Gamma_0$. Let n_0 be the cardinality of the set Γ_0 . Then, given a set of n_0 positive quantities $\{x_i, i \in \Gamma_0\}$, with $\sum_{i \in \Gamma_0} x_i = 1$, and defining $\Lambda = (\lambda_r, r \in J_m) = \sum_{i \in \Gamma_0} x_i A_i$, it is $\sum_{r \in J_m} \lambda_r = 1, \lambda_r \geq 0, \forall r \in J_m$. Of course, $\Phi_j(\Lambda) = \sum_{H_j} \lambda_r = 0 = \sum_{E_j H_j} \lambda_r, \forall j \in I_0$. Moreover, $\forall i \in \Gamma_0$, one has

$$\sum_{E_j H_j} \lambda_r^{(i)} = p_j \sum_{H_j} \lambda_r^{(i)}, j \in J_n; \quad \sum_{E_j H_j} x_i \lambda_r^{(i)} = p_j \sum_{H_j} x_i \lambda_r^{(i)}, j \in J_n.$$

Hence, for $j \in J_n$, one has

$$\sum_{E_j H_j} \left(\sum_{i \in \Gamma_0} x_i \lambda_r^{(i)} \right) = p_j \sum_{H_j} \left(\sum_{i \in \Gamma_0} x_i \lambda_r^{(i)} \right),$$

that is

$$\sum_{E_j H_j} \lambda_r = p_j \sum_{H_j} \lambda_r, j \in J_n; \quad (8)$$

therefore $\Lambda = (\lambda_r, r \in J_m)$ is a solution of system (4). Moreover,

$$\Phi_j(\Lambda) = \Phi_j \left(\sum_{i \in \Gamma_0} x_i A_i \right) = \sum_{i \in \Gamma_0} x_i \Phi_j(A_i) \geq x_j \Phi_j(A_j) > 0, \forall j \in \Gamma_0. \quad (9)$$

Theorem 3. Given a probability assessment $\mathcal{P}_n = (p_i, i \in J_n)$ on $\mathcal{F}_n = \{E_i | H_i, i \in J_n\}$, assume that the system (4) is solvable. Then, for every $\Gamma \subseteq \Gamma_0$, the sub-vector $\mathcal{P}_\Gamma = (p_i, i \in \Gamma)$ is a coherent probability assessment on the sub-family $\mathcal{F}_\Gamma$.

Proof. Of course, coherence of $\mathcal{P}_{\Gamma_0}$ implies coherence of $\mathcal{P}_\Gamma, \forall \Gamma \subset \Gamma_0$; so, we only need to prove coherence of $\mathcal{P}_{\Gamma_0}$. We distinguish two cases:

- (i) the sub-assessment $\mathcal{P}_0$ associated with I_0 is coherent;
- (ii) the sub-assessment $\mathcal{P}_0$ associated with I_0 is not coherent.

In the first case, by Theorem 1, $\mathcal{P}_n$ is coherent and hence $\mathcal{P}_{\Gamma_0}$ is coherent too.

In the second case, given any coherent sub-assessment $\mathcal{P}_0^* = (p_i^*, i \in I_0)$ on $\mathcal{F}_0$, by Theorem 1 the assessment $\mathcal{P}_n^* = (\mathcal{P}_{\Gamma_0}, \mathcal{P}_0^*) = (p_i, i \in \Gamma_0; p_i^*, i \in I_0)$ on $\mathcal{F}_n$ is coherent and hence $\mathcal{P}_{\Gamma_0}$ is coherent too.

Given a probability assessment $\mathcal{P}_n$ on $\mathcal{F}_n$ and assuming system (4) solvable, let S' be a subset of the set S of solutions of (4). Recalling that $\Phi_j(\Lambda) = \sum_{H_j} \lambda_r$, where $\Lambda = (\lambda_r, r \in J_m)$, we set

$$I_{S'} = \{j \in J_n : \Phi_j(\Lambda) = 0, \forall \Lambda \in S'\}, \quad \Gamma_{S'} = J_n \setminus I_{S'}. \quad (10)$$

We denote by $\mathcal{P}_{\Gamma_{S'}}$ (resp., $\mathcal{P}_{I_{S'}}$) the sub-vector of $\mathcal{P}_n$ associated with $\Gamma_{S'}$ (resp., $I_{S'}$). Obviously, $S' \subseteq S$ implies $\Gamma_{S'} \subseteq \Gamma_0$; hence, by Theorem 3, the sub-assessment $\mathcal{P}_{\Gamma_{S'}}$ is coherent. Notice that, by replacing the sub-vector $\mathcal{P}_{I_{S'}}$ with any sub-vector $\mathcal{P}_{I_{S'}}^*$, the set S' is also a subset of the set of solutions of the system (4) associated with the assessment $\mathcal{P}_n^* = (\mathcal{P}_{\Gamma_{S'}}, \mathcal{P}_{I_{S'}}^*)$. Of course, the same remark holds in the particular case $S' = \{\Lambda\}$. Then, we have

Theorem 4. Given a probability assessment $\mathcal{P}_n$ on $\mathcal{F}_n$, assume that system (4) is solvable. Then, given any subset $S' \subset S$ and any coherent assessment $\mathcal{P}_{I_{S'}}^*$ on the sub-family $\mathcal{F}_{I_{S'}}$, the assessment $\mathcal{P}_n^* = (\mathcal{P}_{\Gamma_{S'}}, \mathcal{P}_{I_{S'}}^*)$ on $\mathcal{F}_n$ is coherent.

Proof. We observe that $\mathcal{P}_n^*$ is obtained by $\mathcal{P}_n$ by replacing $\mathcal{P}_{I_{S'}}$ with $\mathcal{P}_{I_{S'}}^*$ and that S' is also a subset of the set of solutions of the system (4) associated with $\mathcal{P}_n^*$. Then, by applying Theorem 1 to the pair $(\mathcal{F}_n, \mathcal{P}_n^*)$, system (4) is solvable and $I_0 \subseteq I_{S'}$. Moreover $\mathcal{P}_0$, being a sub-vector of $\mathcal{P}_{I_{S'}}^*$, is coherent and hence $\mathcal{P}_n^*$ is coherent too.

In particular, by Theorem 4 it immediately follows that, if (4) is solvable and $\mathcal{P}_{I_{S'}}$ is coherent, then $\mathcal{P}_n$ is coherent. Based on the above results, we have

Theorem 5. Let $\mathcal{P}' = (p'_i, i \in J_n)$, $\mathcal{P}'' = (p''_i, i \in J_n)$ be two coherent probability assessments defined on $\mathcal{F}_n = \{E_i | H_i, i \in J_n\}$. There exists a continuous curve $\mathcal{C}$ with extreme points $\mathcal{P}', \mathcal{P}''$ such that:

$$(i) \quad \mathcal{P}^m \leq \mathcal{P} \leq \mathcal{P}^M, \quad \forall \mathcal{P} \in \mathcal{C}; \quad (ii) \quad \mathcal{C} \subseteq \Pi_n. \quad (11)$$

Proof. We denote by I'_0 and I''_0 the subsets defined by (5), associated respectively with $\mathcal{P}'$ and $\mathcal{P}''$. From coherence of $\mathcal{P}', \mathcal{P}''$ it follows that the systems like (4), associated with them, are solvable. Then, by Theorem 2, there exists two vectors $\Lambda' = (\lambda'_r, r \in J_m)$, $\Lambda'' = (\lambda''_r, r \in J_m)$ such that

$$\Phi_j(\Lambda') > 0, \quad \forall j \in \Gamma'_0 = J_n \setminus I'_0; \quad \Phi_j(\Lambda'') > 0, \quad \forall j \in \Gamma''_0 = J_n \setminus I''_0.$$

Given any number $\alpha_0 \in (0, 1)$, let us consider the vector

$$\Lambda = (1 - \alpha_0)\Lambda' + \alpha_0\Lambda'' = (\lambda_r, r \in J_m), \quad (12)$$

where $\lambda_r = (1 - \alpha_0)\lambda'_r + \alpha_0\lambda''_r, \forall r \in J_m$.

Defining $I^{(0)} = I'_0 \cap I''_0$, for each $j \in \Gamma^{(0)} = \Gamma'_0 \cup \Gamma''_0 = J_n \setminus I^{(0)}$ we have

$$\Phi_j(\Lambda) = \Phi_j[(1 - \alpha_0)\Lambda' + \alpha_0\Lambda''] = (1 - \alpha_0)\Phi_j(\Lambda') + \alpha_0\Phi_j(\Lambda'') > 0, \quad (13)$$

with $\Phi_j(A) = 0, \forall j \in I^{(0)} = J_n \setminus \Gamma^{(0)}$. Moreover, from coherence of $\mathcal{P}', \mathcal{P}''$, one has

$$\sum_{E_i H_i} \lambda'_r = p'_i \sum_{H_i} \lambda'_r, \quad \sum_{E_i H_i} \lambda''_r = p''_i \sum_{H_i} \lambda''_r, \quad i \in J_n. \quad (14)$$

Now, let us consider the point $\mathcal{P}_{\Gamma^{(0)}} = (p_i, i \in \Gamma^{(0)})$ defined as

$$p_i = (1 - \delta_i^0) p'_i + \delta_i^0 p''_i = p'_i + \delta_i^0 (p''_i - p'_i), \quad \forall i \in \Gamma^{(0)}, \quad (15)$$

where

$$\delta_i^0 = \frac{\alpha_0 \sum_{H_i} \lambda''_r}{(1 - \alpha_0) \sum_{H_i} \lambda'_r + \alpha_0 \sum_{H_i} \lambda''_r} = \frac{\alpha_0 \sum_{H_i} \lambda''_r}{\sum_{H_i} \lambda_r}, \quad \forall i \in \Gamma^{(0)}. \quad (16)$$

From (14) and (16), for each $i \in \Gamma^{(0)}$ we have

$$\begin{aligned} \sum_{E_i H_i} \lambda_r &= (1 - \alpha_0) \sum_{E_i H_i} \lambda'_r + \alpha_0 \sum_{E_i H_i} \lambda''_r = \\ &= (1 - \alpha_0) p'_i \sum_{H_i} \lambda'_r + \alpha_0 p''_i \sum_{H_i} \lambda''_r = \\ &= \left(\frac{(1 - \alpha_0) \sum_{H_i} \lambda'_r}{\sum_{H_i} \lambda_r} p'_i + \frac{\alpha_0 \sum_{H_i} \lambda''_r}{\sum_{H_i} \lambda_r} p''_i \right) \sum_{H_i} \lambda_r = [(1 - \delta_i^0) p'_i + \delta_i^0 p''_i] \sum_{H_i} \lambda_r; \end{aligned}$$

that is, recalling (15),

$$\sum_{E_i H_i} \lambda_r = p_i \sum_{H_i} \lambda_r, \quad \forall i \in \Gamma^{(0)}. \quad (17)$$

Now, given any quantities $\delta_i^0 \in [0, 1], i \in I^{(0)} = J_n \setminus \Gamma^{(0)}$, let us consider the assessment $\mathcal{P}_n = (p_i, i \in J_n)$, with p_i defined by (15) for all $i \in \Gamma^{(0)}$ and with $p_i = (1 - \delta_i^0) p'_i + \delta_i^0 p''_i$ for all $i \in I^{(0)}$. We have

$$\sum_{E_i H_i} \lambda_r = p_i \sum_{H_i} \lambda_r, \quad \forall i \in J_n; \quad (18)$$

hence Λ is a solution of system (4) and considering the set I_0 associated with $\mathcal{P}_n$, as defined by (5), we have $I_0 \subseteq I^{(0)}, \Gamma^{(0)} \subseteq I_0$; then, by Theorem 3, the probability assessment $\mathcal{P}_{I_0}$ on $\mathcal{F}_{I_0}$ is coherent (and hence $\mathcal{P}_{\Gamma^{(0)}}$ is coherent too). Notice that $\delta_i^0 \geq 0, 1 - \delta_i^0 \geq 0, \forall i \in I_0$, with $\delta_i^0 > 0, 1 - \delta_i^0 > 0, \forall i \in \Gamma^{(0)}$; hence $\mathcal{P}_{I_0}^m \subseteq \mathcal{P}_{I_0} \subseteq \mathcal{P}_{I_0}^M$, with $\mathcal{P}_{I_0}^m < \mathcal{P}_{\Gamma^{(0)}} < \mathcal{P}_{I_0}^M$; that is

$$\min \{p'_i, p''_i\} \leq p_i \leq \max \{p'_i, p''_i\}, \quad \forall i \in I_0, \quad (19)$$

with

$$\min \{p'_i, p''_i\} < p_i < \max \{p'_i, p''_i\}, \quad \forall i \in \Gamma^{(0)}. \quad (20)$$

We denote, respectively, by $\mathcal{P}'_0, \mathcal{P}''_0$, and $\mathcal{F}_0$, the sub-vectors of $\mathcal{P}', \mathcal{P}''$, and the sub-family of $\mathcal{F}_n$, associated with I_0 . Of course, from coherence of $\mathcal{P}', \mathcal{P}''$, it follows that $\mathcal{P}'_0, \mathcal{P}''_0$, defined on $\mathcal{F}_0$, are coherent too.

Moreover, we denote, respectively, by I'_1, I''_1 the subsets associated with $\mathcal{P}'_0, \mathcal{P}''_0$, as defined by (5). Then, by repeating the above procedure, for each $\alpha_1 \in (0, 1)$ we can determine a suitable coherent assessment $\mathcal{P}_{I_1}$ defined on $\mathcal{F}_{I_1}$, where

$$I_1 \supseteq \Gamma^{(1)} = I'_1 \cup I''_1 = I_0 \setminus I^{(1)} = I_0 \setminus (I'_1 \cap I''_1), \quad \mathcal{P}_{I_1}^m \leq \mathcal{P}_{I_1} \leq \mathcal{P}_{I_1}^M, \quad (21)$$

with $\mathcal{P}_{\Gamma^{(1)}}^m < \mathcal{P}_{\Gamma^{(1)}} < \mathcal{P}_{\Gamma^{(1)}}^M$. The coherence of the assessment $(\mathcal{P}_{I_0}, \mathcal{P}_{I_1})$ on $\mathcal{F}_{I_0} \cup \mathcal{F}_{I_1} = \mathcal{F}_{J_n \setminus I_1}$ is obtained by the following steps:

- (a) let $\mathcal{P}_1$ be any coherent assessment, on $\mathcal{F}_1$, associated with the subset I_1 ;
- (b) then, by Theorem 1, the assessment $(\mathcal{P}_{I_1}, \mathcal{P}_1)$ on $\mathcal{F}_0 = \mathcal{F}_{I_0} \cup \mathcal{F}_1$ is coherent;
- (c) then, by Theorem 1, the assessment $\mathcal{P}_n = (\mathcal{P}_{I_0}, \mathcal{P}_{I_1}, \mathcal{P}_1)$ on $\mathcal{F}_n = \mathcal{F}_{I_0} \cup \mathcal{F}_{I_1} \cup \mathcal{F}_1$ is coherent;
- (d) then, the sub-assessment $(\mathcal{P}_{I_0}, \mathcal{P}_{I_1})$ on $\mathcal{F}_{I_0} \cup \mathcal{F}_{I_1}$ is coherent.

By repeating the procedure for the triple $(\mathcal{P}'_1, \mathcal{P}''_1, \mathcal{F}_1)$ associated with I_1 , we determine a coherent probability assessment $\mathcal{P}_{I_2}$ defined on $\mathcal{F}_{I_2}$; and so on.

In this way, after $k + 1$ steps, with $k \leq n - 1$, we construct a probability assessment $\mathcal{P} = (\mathcal{P}_{I_0}, \mathcal{P}_{I_1}, \dots, \mathcal{P}_{I_k})$ on $\mathcal{F}_n$ which, by Theorem 1, is coherent.

We remark that each assessment $\mathcal{P}$ is obtained by using suitable continuous parameters $\alpha_j, \delta_i^j, i \in I_j, j = 0, 1, \dots, k$, and is such that $\mathcal{P}^m \leq \mathcal{P} \leq \mathcal{P}^M$; hence, $\mathcal{P}$ is intermediate between $\mathcal{P}'$ and $\mathcal{P}''$. Moreover, all the parameters $\alpha_0, \alpha_1, \dots, \alpha_k$ can assume all the values in $(0, 1)$; then, letting $\alpha_j \rightarrow 0, \delta_i^j \rightarrow 0$, and $\alpha_j \rightarrow 1, \delta_i^j \rightarrow 1, i \in I_j \setminus \Gamma^{(j)}, j = 0, 1, \dots, k$, we can construct a continuous curve $\mathcal{C} \subseteq \Pi_n$ connecting $\mathcal{P}', \mathcal{P}''$.

In particular, we could construct coherent assessments on $\mathcal{F}_n$ of the kind $\mathcal{P} = (\mathcal{P}_{\Gamma^{(0)}}, \mathcal{P}_{\Gamma^{(1)}}, \dots, \mathcal{P}_{\Gamma^{(h)}})$, by applying Theorem 4 with $S' = \{\Lambda^{(j)}\}, j = 0, 1, \dots, h$, where for each j the vector $\Lambda^{(j)}$ is obtained as in (12).

We observe that, given $\mathcal{P}' = (p'_i, i \in J_n) \in \Pi_n, \mathcal{P}'' = (p''_i, i \in J_n) \in \Pi_n$, with the pair $(\mathcal{P}^m, \mathcal{P}^M) = (\mathcal{P}' \wedge \mathcal{P}'', \mathcal{P}' \vee \mathcal{P}'')$ we can associate the interval

$$\mathcal{I} = [\mathcal{P}^m, \mathcal{P}^M] = [p_1^m, p_1^M] \times \dots \times [p_n^m, p_n^M] = \{\mathcal{P} : \mathcal{P}^m \leq \mathcal{P} \leq \mathcal{P}^M\}. \quad (22)$$

Of course, each curve $\mathcal{C}$ connecting $\mathcal{P}', \mathcal{P}''$ and constructed by the procedure above is contained in $\mathcal{I}$. Then, by Theorem 5, we obtain

Corollary 1. Given any quantities $p_1, \dots, p_{i-1}, l_i \leq u_i, p_{i+1}, \dots, p_n$, let us define $\mathcal{P}' = (p_1, \dots, p_{i-1}, l_i, p_{i+1}, \dots, p_n), \mathcal{P}'' = (p_1, \dots, p_{i-1}, u_i, p_{i+1}, \dots, p_n)$. Moreover, let $\mathcal{I} = \mathcal{P}'\mathcal{P}''$ be the segment $\{(p_1, \dots, p_i, \dots, p_n) : l_i \leq p_i \leq u_i\}$, with set of vertices $\mathcal{V} = \{\mathcal{P}', \mathcal{P}''\}$. Then: $\mathcal{I} \subseteq \Pi_n \iff \mathcal{V} \subseteq \Pi_n$.

Proof. Of course, $\mathcal{I} \subseteq \Pi_n$ implies $\mathcal{V} \subseteq \Pi_n$. Conversely, assuming $\mathcal{V} \subseteq \Pi_n$, as $\mathcal{P}^m = \mathcal{P}', \mathcal{P}^M = \mathcal{P}''$, the unique curve $\mathcal{C}$ contained in $\mathcal{I}$ and connecting $\mathcal{P}', \mathcal{P}''$ is the segment $\mathcal{I}$ itself. Then, by Theorem 5, $\mathcal{I} \subseteq \Pi_n$.

5 On the Compactness Property of the Set Π_n

In this section we examine the compactness property of the set Π_n ; we show that Π_n is closed; then, being a subset of $[0, 1]^n$, Π_n is compact. Finally, exploiting the connection and closure properties, we obtain the well known theorem of coherent extensions for conditional probability assessments.

Remark 1. We can represent Π_n as union of disjoint closed sets (each set being a segment). In particular, we have $\Pi_1 = [p'_1, p''_1]$, with

$$\begin{cases} p'_1 = p''_1 = 0, & \text{if } E_1 H_1 = \emptyset, \\ p'_1 = p''_1 = 1, & \text{if } E_1 H_1 = H_1, \\ p'_1 = 0, p''_1 = 1, & \text{if } \emptyset \subset E_1 H_1 \subset H_1. \end{cases} \quad (23)$$

In the case $n = 2$, with each assessment $p_1 \in [p'_1, p''_1]$ on $E_1|H_1$ it is associated a suitable closed interval $I_{p_1} \subseteq [0, 1]$ of coherent extensions on $E_2|H_2$. Then

$$\Pi_2 = \{(p_1, p_2) : p_1 \in \Pi_1, p_2 \in I_{p_1}\}. \quad (24)$$

We observe that, for each $p_1 \in \Pi_1$, the set $\{(p_1, p_2) : p_2 \in I_{p_1}\}$ is a segment containing its extreme points; hence it is closed. Moreover, $\forall p' \in \Pi_1, \forall p'' \in \Pi_1$, with $p' \neq p''$, the segments $\{(p', p_2) : p_2 \in I_{p'}\}, \{(p'', p_2) : p_2 \in I_{p''}\}$ are disjoint.

In general, with each assessment $(p_1, \dots, p_{n-1})$ contained in the set Π_{n-1} of coherent assessments on $\mathcal{F}_{n-1} = \{E_i|H_i, i \in J_{n-1}\}$ it is associated a suitable closed interval $I_{p_1, \dots, p_{n-1}} \subseteq [0, 1]$ of coherent extensions on $E_n|H_n$. Then

$$\Pi_n = \{(p_1, \dots, p_{n-1}, p_n) : (p_1, \dots, p_{n-1}) \in \Pi_{n-1}, p_n \in I_{p_1, \dots, p_{n-1}}\}. \quad (25)$$

We observe that, for each $(p_1, \dots, p_{n-1}) \in \Pi_{n-1}$, the set $\{(p_1, \dots, p_{n-1}, p_n) : p_n \in I_{p_1, \dots, p_{n-1}}\}$ is a segment containing its extreme points; hence it is closed. Moreover, for every $(p'_1, \dots, p'_{n-1}) \in \Pi_{n-1}, (p''_1, \dots, p''_{n-1}) \in \Pi_{n-1}$, with $(p'_1, \dots, p'_{n-1}) \neq (p''_1, \dots, p''_{n-1})$, the segments

$$\{(p'_1, \dots, p'_{n-1}, p_n) : p_n \in I_{p'_1, \dots, p'_{n-1}}\}, \quad \{(p''_1, \dots, p''_{n-1}, p_n) : p_n \in I_{p''_1, \dots, p''_{n-1}}\}$$

are disjoint. Hence, Π_n is representable as (in general, an uncountable) union of disjoint closed sets (segments).

The closure property has been obtained, based on betting scheme, in the general framework of conditional previsions in ([13], Lemma 2.5, Theorem 2.6), by proving that coherence is preserved in a passage to the limit (*continuity property*). We describe below a version of such result, adapted to the case of conditional probabilities. Let be $\lim_k \mathcal{P}^{(k)} = \mathcal{P} = (p_1, \dots, p_n)$. Then, for every $\varepsilon > 0$ there exists k_ε such that $|p_i - p_i^{(k)}| < \varepsilon, i \in J_n, \forall k > k_\varepsilon$. For each given subset $\Gamma \subseteq J_n$, we denote, respectively, by $\mathcal{P}_\Gamma, \mathcal{P}_\Gamma^{(k)}$ the corresponding subvectors of $\mathcal{P}, \mathcal{P}^{(k)}$, and by $\mathcal{G}_\Gamma = \sum_{i \in \Gamma} s_i H_i (E_i - p_i), \mathcal{G}_\Gamma^{(k)} = \sum_{i \in \Gamma} s_i H_i (E_i - p_i^{(k)})$ the associated random gains. Setting $\sigma = \sum_{i \in \Gamma} |s_i|$, for every $k > k_\varepsilon$ we have

$$\begin{aligned} |\mathcal{G}_\Gamma^{(k)} - \mathcal{G}_\Gamma| &= |\sum_{i \in \Gamma} s_i H_i (E_i - p_i^{(k)}) - \sum_{i \in \Gamma} s_i H_i (E_i - p_i)| = \\ &= |\sum_{i \in \Gamma} s_i H_i (p_i - p_i^{(k)})| \leq \sum_{i \in \Gamma} |s_i| |p_i - p_i^{(k)}| < \varepsilon \sigma. \end{aligned} \quad (26)$$

Hence, defining $\mathcal{H}_\Gamma = \bigvee_{i \in \Gamma} H_i$, $\mu_\Gamma = \max \mathcal{G}_\Gamma | \mathcal{H}_\Gamma$, $\mu_\Gamma^{(k)} = \max \mathcal{G}_\Gamma^{(k)} | \mathcal{H}_\Gamma$, one has $\mu_\Gamma - \varepsilon\sigma < \mu_\Gamma^{(k)} < \mu_\Gamma + \varepsilon\sigma$, $\forall k > k_\varepsilon$. If $\mathcal{P}$ were not coherent, for a suitable subset Γ and for suitable values $s_i, i \in \Gamma$, it should be $\mu_\Gamma < 0$. Then, for every $\varepsilon < -\frac{\mu_\Gamma}{\sigma}$ we would have $\mu_\Gamma^{(k)} < \mu_\Gamma + \varepsilon\sigma < 0$, which is absurd; therefore, $\mathcal{P}$ is coherent and hence Π_n is closed.

Exploiting Theorem 1, a simple proof of closure (and hence of compactness) property is given below.

Theorem 6. The set Π_n of coherent probability assessments on a family $\mathcal{F}_n$ of n conditional events is compact.

Proof. As Π_n is bounded, we only needs to verify that Π_n is closed. Let be

$$\lim_{k \rightarrow \infty} P^{(k)} = P = (p_1, \dots, p_n); \text{ i.e., } \lim_{k \rightarrow \infty} p_i^{(k)} = p_i, i \in J_n,$$

with $\{P^{(k)} = (p_1^{(k)}, \dots, p_n^{(k)}), k \in \mathbb{N}\} \subseteq \Pi_n$. By Theorem 1, for each k the system $\mathcal{S}^{(k)}$ associated with $\mathcal{P}^{(k)}$, defined as in (4), admits (at least) a solution $\Delta^{(k)} = (\lambda_r^{(k)}, r \in J_m)$. As $0 \leq \lambda_r^{(k)} \leq 1, \forall k, \forall r$, the m sequences $\{\lambda_r^{(k)}, k \in \mathbb{N}\}, r \in J_m$, are bounded; then, by a well known result, from each sequence $\{\lambda_r^{(k)}, k \in \mathbb{N}\}$ we can extract a convergent subsequence, say $\{\delta_r^{(k)}, k \in \mathbb{N}\}$. Then, by the same result, there exists a convergent subsequence

$$\{\Delta^{(k)}, k \in \mathbb{N}\} = \{(\delta_r^{(k)}, r \in J_m), k \in \mathbb{N}\},$$

where, for each k , the vector $\Delta^{(k)}$ is a solution of the system (4) associated with $P^{(k)}$; that is

$$\sum_{E_i H_i} \delta_r^{(k)} = p_i^{(k)} \sum_{H_i} \delta_r^{(k)}, \forall i \in J_n; \sum_{r \in J_m} \delta_r^{(k)} = 1; \delta_r^{(k)} \geq 0, \forall r \in J_m.$$

Setting $\lim_k \delta_r^{(k)} = \delta_r, r \in J_m$, we have

$$\sum_{E_i H_i} \delta_r^{(k)} \rightarrow \sum_{E_i H_i} \delta_r; \sum_{H_i} \delta_r^{(k)} \rightarrow \sum_{H_i} \delta_r; \sum_{r \in J_m} \delta_r = \lim_k \sum_{r \in J_m} \delta_r^{(k)} = 1. \quad (27)$$

Then, from $p_i = \lim_{k \rightarrow \infty} p_i^{(k)}, i \in J_n$, it follows that $\Delta = (\delta_1, \dots, \delta_m)$ is a solution of the system (4) associated with the limiting point $\mathcal{P}$. Thus, by Theorem 1, computing the set I_0 and denoting by $\{P_0^{(k)} = (p_i^{(k)}, i \in I_0), k \in \mathbb{N}\}$ its associated sequence, we can apply the same reasoning to the sub-assessment $\mathcal{P}_0 = (p_i, i \in I_0) = \lim_{k \rightarrow \infty} P_0^{(k)}$. After a finite number of steps, we have that $\mathcal{P}$ is coherent, i.e. $\mathcal{P} \in \Pi_n$; so that Π_n is closed.

We recall that the theorem of extension for coherent conditional probabilities (known as de Finetti's *fundamental theorem of probability* in the case of unconditional events) assures that each coherent probability assessment can be extended to further conditional events. As is shown below, such theoretical result can be directly obtained from Theorem 6 and Corollary 1.

Theorem 7. Given a coherent probability assessment $\mathcal{P}_n = (p_i, i \in J_n)$ on the family $\mathcal{F}_n = \{E_i|H_i, i \in J_n\}$ and a further conditional event $E_{n+1}|H_{n+1}$, there exists a suitable non empty interval $[p', p''] \subseteq [0, 1]$ such that the extension $P(E_{n+1}|H_{n+1}) = p_{n+1}$ is coherent if and only if $p_{n+1} \in [p', p'']$.

Proof. Denoting by Π the set of coherent extensions of $\mathcal{P}_n$ on $E_{n+1}|H_{n+1}$, we first verify that Π is not empty. Let $\mathcal{D} = \{D_1, \dots, D_s\}$ be the set of constituents generated by $\mathcal{F}_n \cup \{E_{n+1}|H_{n+1}\}$ and contained in $\mathcal{H}_{n+1} = H_1 \vee \dots \vee H_{n+1}$. Considering the constituents $C_1, \dots, C_m$ generated by $\mathcal{F}_n$ and contained in $\mathcal{H}_n = H_1 \vee \dots \vee H_n$, we observe that there exist disjoint subsets $\Gamma_1, \Gamma_2, \dots, \Gamma_m$ of $\mathcal{D}$, such that

$$C_r = \bigvee_{D_t \subseteq C_r} D_t, \quad r \in J_m. \quad (28)$$

Then, the system (4) with vector of unknowns $\Lambda = (\lambda_1, \dots, \lambda_m)$, associated with $C_1, \dots, C_m$, can be written as a system $\mathcal{S}'$ with vector of unknowns $\Delta = (\delta_1, \dots, \delta_s)$, associated with $D_1, \dots, D_s$, by replacing each λ_r by the sum $\sum_{D_t \subseteq C_r} \delta_t$. As it can be easily verified, system (4) is solvable if and only if $\mathcal{S}'$ is solvable. Then, from coherence of $\mathcal{P}_n$, the system $\mathcal{S}'$ is solvable; hence the following extended system $\mathcal{S}'_{n+1}$ (with a parameter $p \in [0, 1]$)

$$\sum_{E_{n+1}H_{n+1}} \delta_r = p \sum_{H_{n+1}} \delta_r, \quad \sum_{E_j H_j} \delta_r = p_j \sum_{H_j} \delta_r, \quad j \in J_n; \quad \sum_{r \in J_m} \delta_r = 1; \quad \delta_r \geq 0, \quad r \in J_s,$$

is solvable too. By suitably adapting the parameter p , each solution of the system $\mathcal{S}'$ is a solution of the system $\mathcal{S}'_{n+1}$ too. Let be $\Sigma^+ = \{\Delta \in \mathcal{S}' : \sum_{H_{n+1}} \delta_r > 0\}$. If $\Sigma^+ \neq \emptyset$, then for each $\Delta \in \Sigma^+$ by Theorem 1 the vector $\mathcal{P}' = (p_1, \dots, p_n, p)$, where $p = \frac{\sum_{E_{n+1}H_{n+1}} \delta_r}{\sum_{H_{n+1}} \delta_r}$, is a coherent extension of $\mathcal{P}_n$ on $\mathcal{F}_n \cup \{E_{n+1}|H_{n+1}\}$; hence $\Pi \neq \emptyset$. If $\Sigma^+ = \emptyset$, we determine the set $I_0 \subset J_n$ and the associated subfamily $\mathcal{F}_0 \subset \mathcal{F}_n$. If $I_0 = \emptyset$, then, by Theorem 1, $\Pi = [0, 1]$. If $I_0 \neq \emptyset$, we replace $(\mathcal{F}_n, \mathcal{P}_n)$ by $(\mathcal{F}_0, \mathcal{P}_0)$ by repeating the above reasoning. After a finite number of steps, we find $\Sigma^+ \neq \emptyset$, or $I_0 = \emptyset$; hence we conclude that $\Pi \neq \emptyset$.

Then, defining $p' = \inf \Pi$, $p'' = \sup \Pi$, there exists two sequences of coherent assessments on $\mathcal{F}_n \cup \{E_{n+1}|H_{n+1}\}$

$$P'_1, \dots, P'_k, \dots, \quad P''_1, \dots, P''_k, \dots,$$

with $P'_k = (p_1, \dots, p_n, l_{n+1}^{(k)})$, $P''_k = (p_1, \dots, p_n, u_{n+1}^{(k)})$, and with

$$\lim_{k \rightarrow \infty} P'_k = P' = (p_1, \dots, p_n, p'), \quad \lim_{k \rightarrow \infty} P''_k = P'' = (p_1, \dots, p_n, p'').$$

By Theorem 6, P' and P'' are coherent; hence by Corollary 1, for every $p \in [p', p'']$, the assessment $\mathcal{P} = (p_1, \dots, p_n, p)$ on $\mathcal{F}_n \cup \{E_{n+1}|H_{n+1}\}$ is coherent. Therefore, $\Pi = [p', p'']$.

6 Conclusions

We have considered probability assessments on a family $\mathcal{F}_n$ of n conditional events. Among other results, we have examined a connection property for the set Π_n of coherent assessments on $\mathcal{F}_n$. Such property assures that, for every $\mathcal{P}' \in \Pi_n, \mathcal{P}'' \in \Pi_n$, there exists (at least) a continuous curve $\mathcal{C}$ (from $\mathcal{P}'$ to $\mathcal{P}''$) whose points are coherent probability assessments on $\mathcal{F}_n$ intermediate between $\mathcal{P}'$ and $\mathcal{P}''$. We have also given a simple proof of the compactness for the set Π_n . Then, as a corollary, we have obtained the well known theorem of extension for coherent conditional probability assessments.

References

1. Biazzo V., and Gilio A., A generalization of the fundamental theorem of de Finetti for imprecise conditional probability assessments, *International Journal of Approximate Reasoning* 24, 251-272, 2000.
2. Biazzo V., and Gilio A., On the linear structure of betting criterion and the checking of coherence, *Annals of Mathematics and Artificial Intelligence* **35**: 83-106, 2002.
3. Biazzo V., Gilio A., Lukasiewicz T., and Sanfilippo G., Probabilistic Logic under Coherence, *Model-Theoretic Probabilistic Logic, and Default Reasoning in System P*, Journal of Applied Non-Classical Logics, Volume 12, No. 2, 189-213, 2002.
4. Biazzo V., Gilio A., and Sanfilippo G., Coherence Checking and Propagation of Lower Probability Bounds, *Soft Computing* 7, 310-320, 2003.
5. Biazzo V., Gilio A., and Sanfilippo G., On the checking of g-coherence of conditional probability bounds, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 11, Suppl. 2, 75-104, 2003.
6. Capotorti A., and Vantaggi B, Locally strong coherence in inferential processes, *Annals of Mathematics and Artificial Intelligence* **35**: 125-149, 2002.
7. Coletti G., Coherent numerical and ordinal probabilistic assessments, *IEEE Trans. on Systems, Man, and Cybernetics*, 24 (12), 1747-1754, 1994.
8. Coletti G., and Scozzafava R., Characterization of coherent conditional probabilities as a tool for their assessment and extension, *Journal of Uncertainty, Fuzziness and Knowledge-based Systems* 4 (2), 103-127, 1996.
9. Coletti G., and Scozzafava R., Conditioning and inference in intelligent systems, *Soft Computing*, 3/3, 118-130, 1999.
10. Coletti G., Scozzafava R., Probabilistic logic in a coherent setting, Kluwer Academic Publishers, 2002.
11. Gale D., The theory of linear economic models, McGraw-Hill, New York, 1960.
12. Gilio A., Algorithms for precise and imprecise conditional probability assessments, in *Mathematical Models for Handling Partial Knowledge in Artificial Intelligence* (Coletti, G.; Dubois, D.; and Scozzafava, R. eds.), New York: Plenum Press, 231-254, 1995.
13. Holzer S., On coherence and conditional prevision, *Boll. U.M.I., Serie VI, Vol. IV-C, N. 1*, 441-460, 1985.
14. Pelessoni R., and Vicig P., A consistency problem for imprecise conditional probability assessments, in *Proc. of the Seventh Int. Conf. on "Information Processing and Management of Uncertainty in Knowledge-Based Systems" (IPMU '98)*, Paris, France, 1478-1485, 1998.

15. Scozzafava R., Subjective conditional probability and coherence principles for handling partial information, *Mathware Soft Comput*, 3, 183-192.
16. Vicig P., An algorithm for imprecise conditional probability assessments in expert systems, in *Proc. of the Sixth Int. Conf. on "Information Processing and Management of Uncertainty in Knowledge-Based Systems" (IPMU '96)*, Granada, Spain, 61-66, 1996.
17. Walley P., Statistical reasoning with imprecise probabilities, Chapman and Hall, London, 1991.
18. Walley P., Pelesoni R., and Vicig P., Direct Algorithms for Checking Coherence and Making Inferences from Conditional Probability Assessments, *Journal of Statistical Planning and Inference*, 126(1), 119-151, 2004.

Unifying Logical and Probabilistic Reasoning

Rolf Haenni

Institute of Computer Science and Applied Mathematics,
University of Berne, CH-3012 Berne, Switzerland
`haenni@iam.unibe.ch`
`haenni.shorturl.com`

Abstract. Most formal techniques of automated reasoning are either rooted in logic or in probability theory. These areas have a long tradition in science, particularly among philosophers and mathematicians. More recently, computer scientists have discovered logic and probability theory to be the two key techniques for building intelligent systems which rely on reasoning as a central component. Despite numerous attempts to link logical and probabilistic reasoning, a satisfiable unified theory of reasoning is still missing. This paper analyses the connection between logical and probabilistic reasoning, it discusses their respective similarities and differences, and proposes a new unified theory of reasoning in which both logic and probability theory are contained as special cases.

1 Introduction

The goal of artificial intelligence (AI), from a modern perspective, is to construct *intelligent systems* (or *agents*) that behave in ways that would be called intelligent if a human were so behaving [28]. As a consequence, building intelligent systems requires almost always some sort of *reasoning component*. The aim of such a component is to provide means for drawing conclusions about hypotheses in the light of some given background knowledge. Such conclusions may then serve as a basis for rational decisions, which finally makes a system or an agent behave intelligently.

Most approaches for automated reasoning in AI are either built on *logic* or *probability theory*. Accordingly, we speak of *logical reasoning* and *probabilistic reasoning*, respectively. Logical reasoning deals with the *provability* of a hypothesis, whereas probabilistic reasoning is concerned with the *probability* of a hypothesis. These concepts seem to be closely related at first sight, the latter being some sort of generalization of the former, but as we will see later, they are in fact quite different. This complies with the observation that logical and probabilistic reasoning coexist as two independent and self-contained theories of reasoning within AI.

Because human reasoning seems not to make a strict distinction between various ways of reasoning, the distinction between logical and probabilistic reasoning seems to be arbitrary. On the other hand, it is obvious that human reasoning

contains aspects of both disciplines. It should therefore be of crucial importance to construct a unified theory of formal reasoning in which both logic and probability theory are contained as special cases.

1.1 Related Work

Integrating logic and probability is not a new idea at all, various philosophers and computer scientists have made their proposals. Usually, the starting point is to attach probabilities to logical statements rather than events. A prominent example of this stream of research is Nilsson's *probabilistic logic* [23], in which propositional sentences are weighed by probabilities. The same idea has been explored by Halpern and others for the case of first-order sentences with predicates [5, 13, 14]. A different answer offers the area of *objective Bayesianism* [36, 37], according to which probability theory generalizes deductive logic: given some background knowledge (set of premises), logic tells us which conclusions are certain, while probabilities tell us the extent to which one should believe a conclusion (certain conclusions being awarded full belief). A related viewpoint is Halpern's perspective of *probability logic*, in which probability theory is some sort of a many-valued logic [11, 12]. Other related viewpoints are the ones discussed in [1, 15].

The view proposed in this paper will be quite different from the ones described above. We will look at logical and probabilistic reasoning as opposite extreme cases in which the respective set of so-called *probabilistic variables* is either empty or complete, respectively. Models with arbitrary sets of probabilistic variables are then the general case, which will lead us to the unifying theory of probabilistic argumentation. This will require us to talk about the *probability of provability* (or *epistemic probability*) instead of provability in the case of logical reasoning and probability in the case of probabilistic reasoning [2, 4, 24, 26, 27, 30]. Probabilities of provability will be called *degrees of support*.

Conceptually, probabilistic argumentation is closely related to the areas of *interval-valued probabilities* [21, 22, 32, 38], the *Dempster-Shafer Theory* [3, 29], the *Transferable Belief Model* [31], the *Theory of Hints* [17], the *Evidentiary Value Model* [6], or *imprecise probabilities* in general [33, 34, 35]. What all these approaches have in common is that evaluating the truth of a statement yields two values, not just a single one like in classical probability theory.¹ Depending on the context, the first value is either a *lower probability bound*, a *measure of belief*, or a *degree of support*, respectively, whereas the second one is either an *upper probability bound*, a *measure of plausibility*, a *degree of possibility*. In all cases, it is possible to interpret the difference between the two values (i.e. the length of the corresponding interval) as a measure of ignorance relative to

¹ The idea of using two values to express uncertainty and ignorance (or corresponding belief states) goes back to Peirce's work on *Probability of Induction*. In [25] he writes "...to express the proper state of our belief, not one number but two are requisite, the first depending on the inferred probability, the second on the amount of knowledge on which that probability is based".

the hypothesis in question. In this paper, we will interpret the two values as sub-additive probabilities of provability.

1.2 Goals and Overview

The goal of this paper is to propose a unified theory of logical and probabilistic reasoning. It will be called *theory of probabilistic argumentation*. Probabilistic argumentation has been studied before, especially for the restricted case of propositional logic [8, 9, 16], but to see it as a general theory of reasoning is a relatively new viewpoint.

The paper has two main sections. In Section 2, we talk about the differences and similarities between logical and probabilistic reasoning. The unifying framework of probabilistic argumentation is then introduced in Section 3. We will formally prove that both logical and probabilistic reasoning are contained as opposite special cases.

2 Logical vs. Probabilistic Reasoning

Both logical and probabilistic reasoning have a long history in science. Logic dates back to the Greek philosopher Aristotle, whereas the roots of probability theory are in the 17th and 18th century (Pascal, Fermat, Bayes, Bernoulli, etc.). In both cases, the goal is to use the available knowledge to answer open questions about the world, or – in other words – to judge or evaluate *hypotheses* or *events*. In logic, knowledge is encoded by a set Σ of *sentences* of a formal language $\mathcal{L}_V$, whereas probability theory mainly deals with *probability distributions* $\mathbf{P}(V)$.² In both cases, the available knowledge refers to a set $V = \{X_1, \dots, X_n\}$ of variables, which describe possible states of the world.³

Logical Reasoning: Let H be a hypothesis to be evaluated in the light of some given background knowledge. In logic, a hypothesis is usually expressed by another formal sentence. Logical reasoning is a *qualitative* theory of reasoning which may come out with three different answers, depending on whether H and its complement $\neg H$ are *provable* with respect to Σ or not. The possible answers are: “ H is true”, “ H is false”, or “It is unknown whether H is true or false”. Note that the first case implies $\neg H$ to be false, whereas the second case implies $\neg H$ to be true. More formally, if Σ denotes the available knowledge and $\models$ the *entailment* or *consequence operator*, we have three possible situations:

² Alternatively, *density functions* are needed in the case of continuous variables. The discussion in this paper will be limited to discrete variables, but this not a conceptual restriction.

³ Although higher-order or special purpose logics will not be considered in this paper, speaking about logics that refer to sets of variables includes e.g. constraint systems or systems of (linear or non-linear) equations or inequalities.

(L1): $\Sigma \models H, \Sigma \not\models \neg H,$

(L2): $\Sigma \not\models H, \Sigma \models \neg H,$

(L3): $\Sigma \not\models H, \Sigma \not\models \neg H.$

The third case corresponds to a situation in which the available knowledge is insufficient to say anything about H , or – in other words – in which we are *ignorant* with regard to H . (L3) reflects thus an important aspect of logical reasoning, namely that the absence of information does not allow any meaningful conclusions. Note that without (L3), logical reasoning would be a trivial affair. Moreover, as we will see later in this section, it is due to (L3) that logical reasoning is not simply a special case of probabilistic reasoning.

Probabilistic Reasoning: The idea behind probabilistic reasoning is to indicate the possible truth of a hypothesis H by a corresponding *posterior probability* $P'(H) = P(H|E) \in [0, 1]$ for some observed *evidence* E . In probability theory, one usually thinks of a hypothesis as a set of possible world states. Because infinitely many outcomes are possible, we can look at probabilistic reasoning as a *quantitative* theory of reasoning. Posterior probabilities can be computed whenever a full prior distribution $\mathbf{P}(V)$ over the entire set V of variables is available. Prior distributions are often specified with the aid of *Bayesian networks* [24]. There are two extreme situations, namely

(P1): $P'(H) = 1,$

(P2): $P'(H) = 0,$

in which H is either true or false in the light of $\mathbf{P}(V)$ and E . Intermediate values in the spectrum between 0 and 1 represent all possible graded judgments between the two extremities. In the following, the case of such an intermediate value will be called (P3):

(P3): $0 < P'(H) < 1.$

Probability theory is based on Kolomogorov's axioms [20], which, among other things, stipulates *additivity*. This means that $P'(H) + P'(H^c) = 1$ for all possible hypotheses H . As a consequence, we have $P'(H^c) = 0$ for (P1) and $P'(H^c) = 1$ for (P2).

Differences and Similarities: The discussion above immediately identifies obvious one-to-one correspondences between cases (L1) and (P1) and between (L2) and (P2). On the other hand, (L3) seems not to be included in probabilistic reasoning, whereas logical reasoning does not include (P3). Without (L3), logical reasoning would be a special case of probabilistic reasoning, and vice versa without (P3). But because of (L3) and (P3), logical and probabilistic reasoning are different disciplines.

This observation is illustrated in Fig. 1, in which the two axes represent various degrees of truth for H and $\neg H$, respectively.⁴ Logical reasoning allows three possible outcomes, namely the upper left, lower left, and lower right corner of the unit square. In the case of probabilistic reasoning, because of the additivity axiom, the possible outcomes are all located on the diagonal from the upper left to the lower right corner. Because (L3) is located outside this diagonal, logical reasoning turns out to be a *sub-additive* theory of reasoning.

Another important difference comes from the fact that logical reasoning is *monotone* with respect to new information. This means that adding new sentences to Σ will never cause a transition from (L1) or (L2) back to (L3), from (L1) to (L2), or vice versa from (L2) to (L1). In other words, once a hypothesis H is accepted (or rejected) it will always be accepted (or rejected) at any future point in time. Note that an empty set Σ always implies (L3) for all non-trivial hypotheses.

Probabilistic reasoning, on the other hand, is *non-monotone*. This means that corresponding posterior probabilities (except for $P'(H|E) = 0$ and $P'(H|E) = 1$) may increase or decrease when new evidence is added. The picture in Fig. 2 illustrates the possible transitions for both logical and probabilistic reasoning.

To summarize the characteristics of logical and probabilistic reasoning, one can say that the former is qualitative, sub-additive, and monotone, whereas the latter is quantitative, additive, and non-monotone. Of course, we would expect a unifying theory to be at the same time quantitative, sub-additive, and non-monotone.

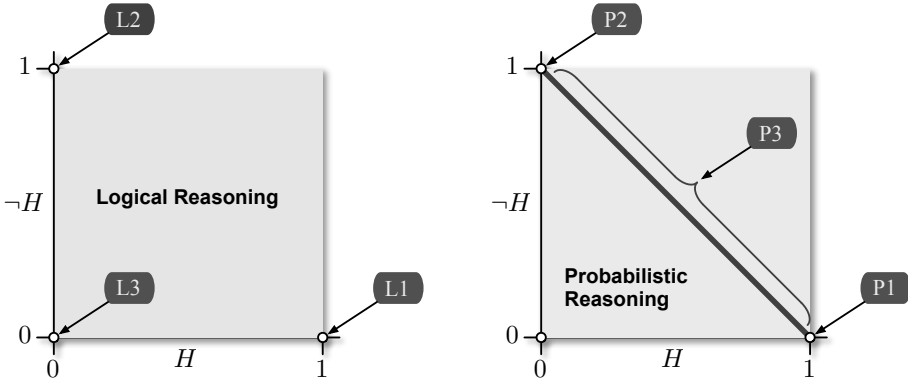


Fig. 1. Possible outcomes of logical and probabilistic reasoning

⁴ Because hypotheses are always denoted by the same symbol H , independently on whether we think of a logical sentence or a set of states, we will use $\neg H$ and H^c interchangeably for its complement.

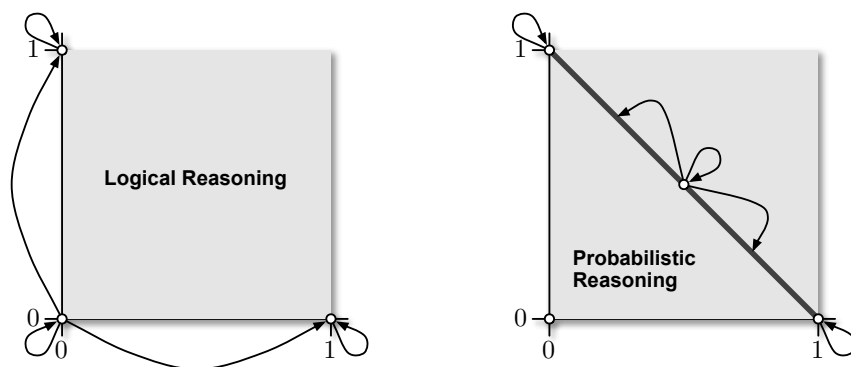


Fig. 2. Logical reasoning is monotone, probabilistic reasoning is non-monotone

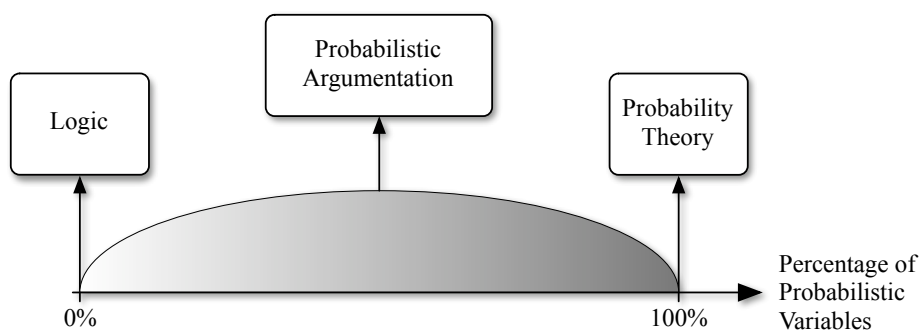


Fig. 3. The role of probabilistic variables

Probabilistic Variables: In order to build a unifying theory of reasoning, we must first try to better understand the origin of the differences between logical and probabilistic reasoning. The key point to realize is the following: probabilistic reasoning presupposes the existence of a probability distribution over *all* variables, whereas logical reasoning does not deal with probability distributions at all, i.e. it presupposes a probability distribution over *none* of the variables involved.

We will call the variables over which a probability distribution is known *probabilistic*. A probabilistic model contains then uniquely probabilistic variables, whereas a logical model contains no probabilistic variables. From this point of view, the main difference between logical and probabilistic reasoning is simply the number of probabilistic variables. And a unifying theory of reasoning will therefore allow an arbitrary number of probabilistic variables. This observation is illustrated in Fig. 3.

The above remark describes how the approach proposed in this paper differs from previous attempts of interconnecting logic and probability. Instead of

assigning probabilities to sentences [5, 13, 14, 23], we consider arbitrary sets of probabilistic variables. In this sense, our approach is an attempt to *unify* logical and probabilistic reasoning, whereas previous approaches tried to *combine* them. The next section will formally develop this theory.

3 Probabilistic Argumentation

As before, we consider models over a set of variables V , but now, some of the variables in V are *probabilistic*. $A \subseteq V$ denotes the corresponding subset and $\mathbf{P}(A)$ the given prior distribution over A . Logical reasoning corresponds to $A = \emptyset$ and probabilistic reasoning to $A = V$, but now we will analyze the general case of an arbitrary number of probabilistic variables. Note that we will not say anything about how to specify the prior distribution $\mathbf{P}(A)$. The prior distribution together with a set of sentences $\Sigma \subseteq \mathcal{L}_V$ encodes the available knowledge. We will call a quadruple $\mathcal{A} = (V, A, \mathbf{P}(A), \Sigma)$ a *probabilistic argumentation system*.

In the general context of probabilistic argumentation systems, we will talk about so-called *probabilities of provability*, or alternatively about *degrees of support*. The definition of degree of support will be based on the following observations:

1. If N_V denotes the set of all possible configurations (states of the world) with respect to V , then Σ determines a subset $E = N_V(\Sigma) \subseteq N_V$ of configurations, namely the ones that are consistent with Σ . In the light of Σ , the true but unknown state of the world must be an element of E .
2. One can also think of the set of possible configurations with respect to A . It is denoted by N_A and its elements are called *scenarios*. By projecting E from N_V to N_A , we get the set $E^{\downarrow A} \subseteq N_A$ of possible scenarios that are consistent with Σ . This means that exactly one of the elements of $E^{\downarrow A}$ is the *true scenario* that corresponds to the true state of the world. All other scenarios $\mathbf{s} \notin E^{\downarrow A}$ are impossible.
3. If $P(\mathbf{s})$ denotes the corresponding prior probability of a scenario $\mathbf{s} \in N_A$, we obtain the corresponding posterior probability $P'(\mathbf{s}) = P(\mathbf{s}|E^{\downarrow A})$ by conditioning $P(\mathbf{s})$ on $E^{\downarrow A}$. This means that the prior distribution $\mathbf{P}(A)$ is replaced by the posterior distribution $\mathbf{P}'(A) = \mathbf{P}(A|E^{\downarrow A})$. For the particular case of $V = \{X, Y\}$ and $A = \{X\}$, this is illustrated on the left hand side of Fig. 4.

Another observation goes in the other direction, that is from N_A to N_V . Let's assume that a certain scenario $\mathbf{s} \in N_A$ is the true scenario. This restricts the set of possible configurations with respect to V from E to

$$E|\mathbf{s} := \{\mathbf{x} \in E : \mathbf{x}^{\downarrow A} = \mathbf{s}\}. \quad (1)$$

This set contains all configurations of E that are compatible with $\mathbf{s}$. This idea is illustrated on the right hand side of Fig. 4 for $V = \{X, Y\}$, $A = \{X\}$, and with three scenarios $\mathbf{s}_0$, $\mathbf{s}_1$, and $\mathbf{s}_2$. Note that we have $E|\mathbf{s} \neq \emptyset$ for every consistent scenario $\mathbf{s} \in E^{\downarrow A}$, e.g. for $\mathbf{s}_1$ and $\mathbf{s}_2$. Otherwise, as in the case of $\mathbf{s}_0$, we have $E|\mathbf{s} = \emptyset$.

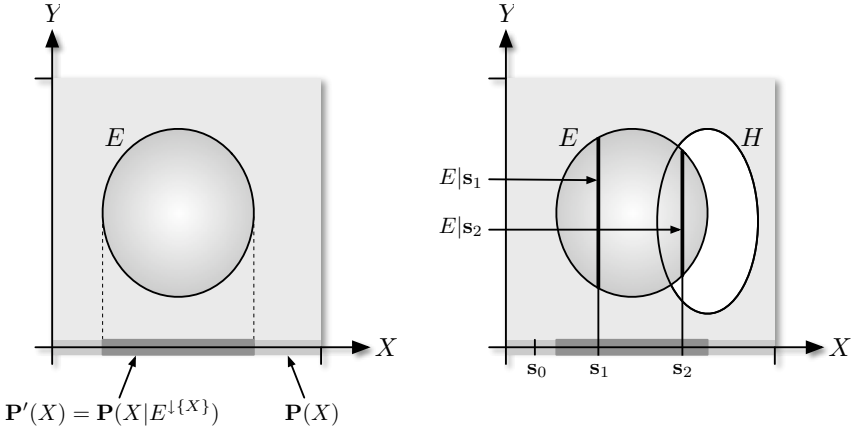


Fig. 4. The relationship between sets of configurations with respect to $V = \{X, Y\}$ and $A = \{X\}$

Arguments and Degrees of Support: Consider now a consistent scenario $\mathbf{s} \in E^{\perp A}$ for which $E|\mathbf{s} \subseteq H$. This means that H follows logically from $\mathbf{s}$ and E , and we can thus think of $\mathbf{s}$ as a *hypothetical proof* for H in the light of E . We must say *hypothetical*, because it is unknown whether $\mathbf{s}$ is the true scenario or not. In other words, H is only *supported* by $\mathbf{s}$, but not proved. The set of all supporting scenarios is denoted by

$$SP(H) := \{\mathbf{s} \in E^{\perp A} : E|\mathbf{s} \subseteq H\}. \quad (2)$$

The elements of $SP(H)$ and $SP(H^c)$ are also called *arguments* and *counter-arguments* of H , respectively. This is where the name of this theory comes from. Note that $SP(\emptyset) = \emptyset$ and $SP(N_V) = E^{\perp A}$. In the example shown on the right hand side of Fig. 4, s_2 is an argument for H , but s_0 or s_1 are not.

The set of supporting scenarios is the key notion for the definition of degree of support. In fact, because every supporting scenario $\mathbf{s} \in SP(H)$ contributes to the possible truth of H , we can measure the strength of such a contribution by the posterior probability $P'(\mathbf{s})$. Because the scenarios are exclusive, we can measure the total support for H by the sum

$$dsp(H) := P'(SP(H)) = \sum_{\mathbf{s} \in SP(H)} P'(\mathbf{s}) \quad (3)$$

over all elements of $SP(H)$. Note that $dsp(H)$ defines an ordinary (additive) probability measure in the classical sense of Kolmogorov, that is $P'(SP(H)) + P'(SP(H)^c) = 1$ holds for all hypotheses H . On the other hand, because the sets $SP(H)$ and $SP(H^c)$ are not necessarily complementary, we have

$$dsp(H) + dsp(H^c) \leq 1. \quad (4)$$

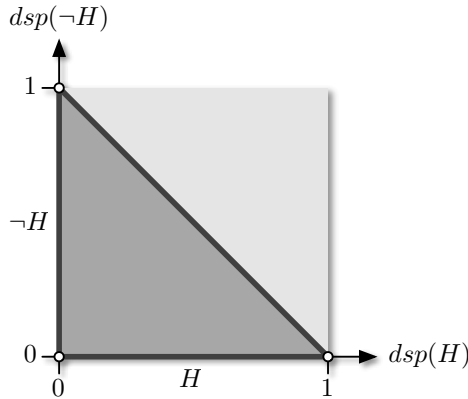


Fig. 5. The sub-additive nature of degree of support

This means the all points within the triangle shown in Fig. 5 are possible outcomes of applying this theory. Of course, it includes the cases (L1), (L2), and (L3) from logical reasoning, just as it includes (P1), (P2), and (P3) from probabilistic reasoning.

According to the above remarks, degrees of support should be understood as sub-additive *posterior probabilities of provability*. They are well-defined for all possible hypotheses H , that is even in cases in which the prior distribution $\mathbf{P}(A)$ does not cover all variables. This is a tremendous advantage over classical probabilistic reasoning which presupposes the existence of a prior distribution over all variables.

To complete this section, we will briefly investigate how the classical fields of logical and probabilistic reasoning fit into this general setting.

Logical reasoning is characterized by $A = \emptyset$. This has a number of consequences. First, it implies that the set of possible scenarios $N_A = \{\langle \rangle\}$ consists of a single element $\langle \rangle$, which represents the empty configuration. $P(\langle \rangle) = 1$ is thus the only possible prior distribution. Second, it means that $E^{\perp A} = \{\langle \rangle\} = N_A$ and thus $P'(\langle \rangle) = 1$. Therefore, we know that $E|\langle \rangle = E$, which allows us to rewrite (2) as

$$SP(H) = \begin{cases} \{\langle \rangle\}, & \text{for } E \subseteq H, \\ \emptyset, & \text{otherwise.} \end{cases} \quad (5)$$

Finally, $P(\langle \rangle) = 1$ implies $dsp(H) = 1$ for $E \subseteq H$ and $dsp(H) = 0$ otherwise. This corresponds to testing whether or not H is a logical consequence of E (respectively of Σ). Logical reasoning with its concept of provability is therefore a special case of probabilistic argumentation.

Probabilistic reasoning is characterized by $A = V$. This means that the sets N_A and N_V are identical, and it implies

$$E|\mathbf{s} = \begin{cases} \{\mathbf{s}\}, & \text{for } \mathbf{s} \in E, \\ \emptyset, & \text{otherwise.} \end{cases} \quad (6)$$

Note that (6) implies $SP(H) = H \cap E$, which allows us to rewrite (3) as

$$dsp(H) = \sum_{\mathbf{s} \in H \cap E} P'(\mathbf{s}) = \sum_{\mathbf{s} \in H} P'(\mathbf{s}) = P'(H). \quad (7)$$

This is exactly the definition of posterior probabilities in the case of probabilistic reasoning. As expected, probabilistic reasoning turns out to be another special case of probabilistic argumentation.

4 Conclusion

This paper introduces the theory of probabilistic argumentation as a unifying theory of reasoning. The key concept of the theory is the notion of degree of support. Degrees of support are sub-additive and non-monotone (posterior) probabilities of provability. This includes the notion of provability from logical reasoning, as well as the notion of probability from probabilistic reasoning. The two classical approaches to automated reasoning are thus included as special cases.

The parameter that makes logical and probabilistic reasoning distinct is the number of probabilistic variables. Probabilistic argumentation is more general in the sense that it allows any number of probabilistic variables. Note that probabilistic argumentation may also help to resolve the disputes between promoters of a pure Bayesian approach to uncertain reasoning and people devoted to other quantitative approaches such as Dempster-Shafer theory (see [10] for details).

The significance and the consequences of this paper are manifold. In the field of automated reasoning, we consider probabilistic argumentation as a new foundation that unifies the existing approaches of logical and probabilistic reasoning. This will have a great impact on the understanding and the numerous applications of automated reasoning within and beyond AI. An interesting application in the field of public-key cryptography has already been studied in depth [7].

An important basic requirement for further success is to develop a generalized *decision theory* that takes into account the possibility of lacking information or missing data. Because probabilistic argumentation helps to clarify the relationship between logical and probabilistic reasoning, it may also put some new light on topics from other areas such as philosophy, mathematics, or statistics. In fact, promising preliminary work shows that looking at statistical inference from the perspective of probabilistic argumentation helps to eliminate the discrepancies between the classical and the Bayesian approach to statistics [18, 19].

Acknowledgements

This research is supported by the Swiss National Science Foundation (Project No. PP002–102652).

References

1. E. W. Adams. *A Primer of Probability Logic*. CSLI Publications, Stanford, 1998.
2. L. J. Cohen. *The Probable and the Provable*. Clarendon Press, Oxford, 1977.
3. A. P. Dempster. A generalization of Bayesian inference. *Journal of the Royal Statistical Society*, 30:205–247, 1968.
4. R. Fagin and J. Y. Halpern. Uncertainty, belief, and probability. *Computational Intelligence*, 7(3):160–173, 1991.
5. R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Information and Computation*, 87(1/2):78–128, 1990.
6. P. Gärdenfors. Probabilistic reasoning and evidentiary value. In P. Gärdenfors, B. Hansson, and N. E. Sahlin, editors, *Evidentiary Value: Philosophical, Judicial and Psychological Aspects of a Theory*, pages 44–57. C. W. K. Gleerups, Lund, Sweden, 1983.
7. R. Haenni. Using probabilistic argumentation for key validation in public-key cryptography. *International Journal of Approximate Reasoning*, 38(3):355–376, 2005.
8. R. Haenni, B. Anrig, J. Kohlas, and N. Lehmann. A survey on probabilistic argumentation. In *ECSQARU'01, 6th European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty, Workshop on Adventures in Argumentation*, Toulouse, France, 2001.
9. R. Haenni, J. Kohlas, and N. Lehmann. Probabilistic argumentation systems. In J. Kohlas and S. Moral, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 5: Algorithms for Uncertainty and Defeasible Reasoning*, pages 221–288. Kluwer, Dordrecht, 2000.
10. R. Haenni and N. Lehmann. Probabilistic argumentation systems: a new perspective on Dempster-Shafer theory. *International Journal of Intelligent Systems (Special Issue: the Dempster-Shafer Theory of Evidence)*, 18(1):93–106, 2003.
11. T. Hailperin. Probability logic. *Notre Dame Journal of Formal Logic*, 25(3):198–212, 1984.
12. T. Hailperin. *Sentential Probability Logic: Origins, Development, Current Status, and Technical Applications*. Lehigh University Press, 1996.
13. J. Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46:311–350, 1990.
14. M. J. Hill, J. B. Paris, and G. M. Wilmers. Some observations on induction in predicate probabilistic reasoning. *Journal of Philosophical Logic*, 31:43–75, 2002.
15. C. Howson. Probability and logic. *Journal of Applied Logic*, 1(3–4):151–165, 2003.
16. J. Kohlas. Probabilistic argumentation systems: A new way to combine logic with probability. *Journal of Applied Logic*, 1(3–4):225–253, 2003.
17. J. Kohlas and P. A. Monney. *A Mathematical Theory of Hints. An Approach to the Dempster-Shafer Theory of Evidence*, volume 425 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, 1995.

18. J. Kohlas and P. A. Monney. Statistical information and assumption-based inference: Continuous models. Technical Report 04-08, Department of Informatics, University of Fribourg, 2004.
19. J. Kohlas and P. A. Monney. Statistical information and assumption-based inference: Discrete models. Technical Report 04-07, Department of Informatics, University of Fribourg, 2004.
20. A. N. Kolmogorov. *Foundations of the Theory of Probability*. Chelsea Publishing Company, New York, 1950.
21. H. E. Kyburg. Higher order probabilities and intervals. *International Journal of Approximate Reasoning*, 2:195-209, 1988.
22. H. E. Kyburg. Interval-valued probabilities. In G. de Cooman and P. Walley, editors, *The Imprecise Probabilities Project*. IPP Home Page, available at <http://ippserv.rug.ac.be>, 1998.
23. N. J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28(1):71-87, 1986.
24. J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.
25. C. S. Peirce. The probability of induction. *Popular Science Monthly*, 12:705-718, 1878.
26. E. Ruspini, J. Lowrance, and Th. Strat. Understanding evidential reasoning. *International Journal of Approximate Reasoning*, 6(3):401-424, 1992.
27. E. H. Ruspini. The logical foundations of evidential reasoning. Technical Report 408, AI Center, SRI International, Menlo Park, CA, 1986.
28. S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2nd edition, 2003.
29. G. Shafer. *The Mathematical Theory of Evidence*. Princeton University Press, 1976.
30. Ph. Smets. Probability of provability and belief functions. *Journal de la Logique et Analyse*, 133:177-195, 1991.
31. Ph. Smets and R. Kennes. The transferable belief model. *Artificial Intelligence*, 66:191-234, 1994.
32. B. Tessem. Interval probability propagation. *International Journal of Approximate Reasoning*, 7:95-120, 1992.
33. P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Monographs on Statistics and Applied Probability 42. Chapman and Hall, London, UK, 1991.
34. P. Walley. Towards a unified theory of imprecise probability. *International Journal of Approximate Reasoning*, 24(2-3):125-148, 2000.
35. K. Weichselberger. The theory of interval-probability as a unifying concept for uncertainty. *International Journal of Approximate Reasoning*, 24(2-3):149-170, 2000.
36. J. Williamson. Probability logic. In D. Gabbay, R. Johnson, H. J. Ohlbach, and J. Woods, editors, *Handbook of the Logic of Argument and Inference: the Turn Toward the Practical*, pages 397-424. Elsevier, Amsterdam, 2002.
37. J. Williamson. *Bayesian Nets and Causality: Philosophical and Computational Foundations*. Oxford University Press, 2005.
38. R. R. Yager and V. Kreinovich. Decision making under interval probabilities. *International Journal of Approximate Reasoning*, 22:195-215, 1999.

Possibility Theory for Reasoning About Uncertain Soft Constraints

Maria Silvia Pini, Francesca Rossi, and Brent Venable

Department of Pure and Applied Mathematics, University of Padova, Italy
{mpini, frossi, kvenable}@math.unipd.it

Abstract. Preferences and uncertainty occur in many real-life problems. The theory of possibility is one non-probabilistic way of dealing with uncertainty, which allows for easy integration with fuzzy preferences. In this paper we consider an existing technique to perform such an integration and, while following the same basic idea, we propose various alternative semantics which allow us to observe both the preference level and the robustness w.r.t. uncertainty of the complete instantiations. We then extend this technique to other classes of soft constraints, proving that certain desirable properties still hold.

1 Introduction

Preferences and uncertainty occur in many real-life problems. In this paper we are concerned with the coexistence of such concepts in the same problem. In particular, we consider uncertainty that comes from lack of data or imprecise knowledge and scenarios where probabilistic estimates are not available.

The theory of possibility [9, 14] is one non-probabilistic way of dealing with uncertainty, which allows for easy integration with fuzzy preferences [6]. In fact, both possibilities and fuzzy preferences are values between 0 and 1 associated to events and express the level of plausibility that the event will occur, or its preference.

In our context, we will describe a real-life problem as set of variables with finite domains and a set of soft constraints among subsets of the variables. A variable will be said to be uncertain if we cannot decide its value. In this case, we will associate a possibility degree to each value in its domain, which will tell how plausible it is that the variable will get that value.

Soft constraints allow to express preferences over the instantiations of the variables of the constraints. In particular, fuzzy preferences are values between 0 and 1, which are combined using the *min* operator, and are ordered in such a way that higher values denote better preferences.

In this paper we consider an existing technique to integrate fuzzy preferences and uncertainty, which uses possibility theory [6]. This technique allows one to handle uncertainty within a fuzzy optimization engine. However, we claim that the integration provided by this technique is too tight since the resulting ordering over complete assignments does not allow one to discriminate be-

tween solutions which are highly preferred but assume unlikely events and solutions which are not preferred but robust with respect to uncertainty. This is due to the fact that a single value, which summarizes the contributions of both the uncertain variables and the fuzzy preferences, is associated to each solution.

While following the same basic idea of translating uncertainty into fuzzy constraints, we propose various alternative semantics which allow us to observe separately the preference level and the robustness of the complete instantiations. More precisely, each solution will be associated to a pair of values between 0 and 1: one value will refer to the preference level, while the other one will refer to the robustness of the solution w.r.t. the uncertain variables. In this way, given a solution and the pair of values associated to it, we can see how preferred it is according to the constraints, by looking at the first value of its pair, and also how robust it is, by looking at the second value of its pair.

The desired ordering over such pairs will then be used to order the solutions. Thus, by choosing different orderings, we can reason in a more or less pessimistic way, giving more or less importance to the preferences w.r.t. the robustness of the problem. In this way, we define a class of different semantics.

2 Soft Constraints

Soft constraints [2] are a very general formalism to describe quantitative preferences. In general, a soft constraint is just pair $\langle def, con \rangle$, where con is the set of variables of the constraint (that is, its scope), and def is a function from the Cartesian product of the domains of the variables in con to a preference set, say A . Therefore def defines the constraint, by associating a level of preference from A to each assignment of values to the variables of the constraint.

Set A can be totally or partially ordered, and its ordering, denoted by $\leq$, can be used to order the assignments of values to variables: assignments corresponding to higher preferences are more preferred. Moreover, a combination operation $\times$ should be defined over A , to combine different constraints and generate the preference level of an assignment of values to variables which range over the scopes of several constraints. More precisely, A should have properties similar to a semiring. We will therefore say that a soft constraints is defined *over semiring* A . For more details on semiring-based soft constraints, see [2].

A soft constraint problem is usually denoted by a tuple $\langle S, V, C \rangle$ where S is a semiring, V is a set of variables, and C is a set of soft constraints over S whose scopes are subsets of these variables. An optimal solution of a soft constraint problem is an assignment of its variables which is optimal according to the ordering associated to the semiring.

This general description of soft constraints instantiates to several classes of concrete constraints:

- *Fuzzy constraints*: when $A = [0, 1]$, $\leq$ is derived by the *max* operator, and the combination operator is *min*. This means that a fuzzy constraint associates

an element between 0 and 1 to each instantiation of its variables, that values closer to 1 denote a higher preference, and that the preferences of two or more constraints are combined by taking their minimum value.

- *Hard constraints*: they can also be described by this framework, by just choosing $A = \{true, false\}$, $\leq$ derived by logical *or* (thus 1 is better than 0), and combination is logical *and*.
- *Weighted constraints*: they are soft constraints where each assignment of values to variables has a weight, and the goal is to minimize the sum of the weights: this can be cast by choosing A as the set of possible weights, by deriving the ordering by the *min* operator, and by using the *sum* as the combination operator.

The concept of fuzzy constraint, as defined above, was originally based on the notion of fuzzy set [14, 8, 11]. A fuzzy set A is a subset of a referential set U whose boundaries are gradual. More formally: the *membership function* μ_A of a fuzzy set A assigns to each element $u \in U$ its degree of membership $\mu_A(u)$ usually taking values in $[0, 1]$. If $\mu_A(u) = 1$, it means that u belongs to A , while $\mu_A(u) = 0$ means that u does not belong to A . If $\mu_A(u)$ is between 0 and 1, then it means that $u \in A$ with degree $\mu_A(u)$.

The *complement* of a fuzzy set A in U is denoted A^C and its membership function is $\mu_{A^C} = 1 - \mu_A$. The *union* and *intersection* of fuzzy sets are obtained by respectively taking the maximum and the minimum of membership degrees of each element of U in each of the fuzzy sets.

Fuzzy constraints use the notion of fuzzy sets to describe the level of preference of a certain assignment of values to variables. More precisely, a *soft fuzzy constraint* [6] C on variables $\{x_1, \dots, x_n\}$ is associated with a *fuzzy relation* R , i.e. a fuzzy subset of $D_1 \times \dots \times D_n$ of values that more or less satisfy C . A membership function μ_R is associated with relation R and specifies for each tuple $(d_1, \dots, d_n) \in D_1 \times \dots \times D_n$ the level of satisfaction $\mu_R(d_1, \dots, d_n)$ in a set L , which is totally ordered (e.g. $[0, 1]$). In particular, $\mu_R(d_1, \dots, d_n) = 1$ if tuple $(d_1, \dots, d_n)$ totally satisfies C , $\mu_R(d_1, \dots, d_n) = 0$ if it totally violates C , and $0 < \mu_R(d_1, \dots, d_n) < 1$ if it partially satisfies C . Moreover, $\mu_R(d_1, \dots, d_n) > \mu_R(d'_1, \dots, d'_n)$ means that tuple $(d_1, \dots, d_n)$ is better than tuple $(d'_1, \dots, d'_n)$.

In the following we will use two operations on fuzzy constraints [6]: projection and combination. The *projection* of a fuzzy constraint, represented by fuzzy relation R on variables $\{x_1, \dots, x_k\} \subseteq V(R) = \{x_1, \dots, x_n\}$, is a fuzzy relation $R^{\downarrow\{x_1, \dots, x_k\}}$ defined on $\{x_1, \dots, x_k\}$ such that: $\mu_{R^{\downarrow\{x_1, \dots, x_k\}}}(d_1, \dots, d_k) = \sup_{\{d=(d_1, \dots, d_n) \mid d^{\downarrow\{x_1, \dots, x_k\}}=(d_1, \dots, d_k)\}} \mu_R(d)$. The *conjunctive combination* of two fuzzy constraints, represented by fuzzy relations R_i and R_j , is a fuzzy relation $R_i \otimes R_j$ defined on variables $V(R_i) \cup V(R_j)$ such that: $\mu_{R_i \otimes R_j}(d_1, \dots, d_k) = \min(\mu_{R_i}(d_1, \dots, d_k)^{\downarrow V(R_i)}, \mu_{R_j}(d_1, \dots, d_k)^{\downarrow V(R_j)})$ where $\mu_{R_i \otimes R_j}(d_1, \dots, d_k)$ evaluates to what extent $(d_1, \dots, d_k)$ satisfies both C_i and C_j .

3 Possibility Theory

A *possibility distribution* [14] is the membership function of a fuzzy set A attached to a single-valued variable x . It is denoted $\pi_x = \mu_A$ and represents the set of more or less plausible, mutually exclusive values of x . A possibility distribution is similar to a probability density. However, $\pi_x(u) = 1$ only means that $x = u$ is a plausible situation, which cannot be excluded. Thus, a degree of possibility can be viewed as an upper bound of a degree of probability.

Possibility theory encodes incomplete knowledge while probability accounts for random and observed phenomena. In particular, the complete ignorance about x is expressed by $\pi_x(u) = 1$, for all $u \in U$, because in this case all values u are plausible for x and so it is impossible to exclude any of them. Whereas, $\pi_x(\bar{u}) = 1$ for a specific value $\bar{u}$ and $\pi_x(u) = 0$ otherwise, expresses the complete knowledge about x , because in this case only the value $\bar{u}$ is plausible for x .

The *possibility* of an event " $x \in E$ ", denoted by $\Pi(x \in E)$, is $\Pi(x \in E) = \sup_u \min(\pi_x(u), \mu_E(u)) = \sup_{u \in E} \pi_x(u)$.

If an event has possibility equal to 1, it means that it is totally possible. However, it could also not happen. Therefore it means that we are completely ignorant about its occurrence. On the contrary, having a possibility equal to 0 means that the event for sure will not happen.

The dual measure of *necessity* of " $x \in E$ ", denoted by $N(x \in E)$, evaluates the extent to which " $x \in E$ " is *certainly* true, $N(x \in E) = \inf_u \max(c(\pi_x(u)), \mu_E(u)) = \inf_{u \notin E} (c(\pi_x(u))) = 1 - \Pi(x \in E^C)$. where c is the order reversing map such that $c(p) = 1 - p$ and E^C is the complement of E in U .

$N(x \in E) = 1$ when it is certain that $x \in E$. On the contrary, having necessity equal to 0 means that the event is not necessary at all, although it may happen. In fact, $N(x \in E) = 0$ iff $P(x \in E^C) = 1$.

For example, if we have a possibility distribution π , attached to a variable x with domain $D_x = \{5, 6, 7, 8\}$, such that $\pi(5) = 0.9$, $\pi(6) = 0.4$, $\pi(7) = 0.7$, $\pi(8) = 0.5$, then, if $A = \{5, 6\}$ is a subset of D_x , the possibility degree of the event $x \in A$ is $\Pi(A) = \sup_{d \in A} \pi(d) = \sup\{0.9, 0.4\} = 0.9$, whereas the necessity degree of the same event, $x \in A$, is $N(A) = \inf_{d \notin A} c(\pi(d)) = \inf\{c(\pi(7)), c(\pi(8))\} = \inf\{c(0.7), c(0.5)\} = \inf\{0.3, 0.5\} = 0.3$. Computing $N(A)$ using the formula $N(A) = 1 - \Pi(\bar{A})$ is the same, in fact, $N(A) = 1 - \Pi(\bar{A}) = 1 - \sup_{d \in \bar{A}} \pi(d) = 1 - \sup\{0.7, 0.5\} = 1 - 0.7 = 0.3$.

4 Uncertainty in Soft Constraints

Whereas in usual soft constraint problems all the variables are assumed to be controllable, that is, their value can be decided according to the constraints which relate them to other variables, in many real-world problems uncertain parameters must be used. Such parameters are associated with variables which are not under the user's direct control and thus cannot be assigned. Only Nature will assign them.

Formally, we can define an uncertain soft constraint problem as a tuple $\langle S, V_c, V_u, C \rangle$, where S is a semiring, V_c is the set of controllable variables, V_u is the set of uncontrollable variables, and C is the set of soft constraints. The soft constraints in C may involve any subset of variables of $V_c \cup V_u$.

While in a classical soft constraint problem we can decide how to assign the variables to make the assignment optimal, in the presence of uncertain parameters we must assign values to the controllable variables guessing what Nature will do with the uncontrollable variables. So, in this paper an optimal solution for an uncertain soft constraint problem is an assignment of values to the variables in V_c such that, *whatever* Nature will decide for the variables in V_u , the overall assignment will be optimal. This is a pessimistic view and other definitions of solutions can be considered [1].

For example, we could be satisfied with finding an assignment of values to the variables in V_c such that, *for at least* one assignment decided by Nature for the variables in V_u , the overall assignment will be optimal. This definition follows an optimistic view. Other definitions can be between these two extremes.

Moreover, the uncontrollable variables can be equipped with additional information on the likelihood of their values. Such information can be given in several ways, depending on the amount and precision of knowledge we have. In this paper for expressing such information we will consider possibility distribution. This information can be used to infer new soft constraints over the controllable variables, expressing the compatibility of the controllable parts of the problem with the uncertain parameters, and can be used to change the notion of optimal solution.

5 Unifying Fuzzy Preferences and Uncertainty via Possibility Theory

Possibility theory [14] can be used to code some information about the uncertain variables in an uncertain soft constraint problem. In this section we will describe an existing approach for uncertain fuzzy soft constraints and later we will show how to modify it and extend it also to other classes of soft constraints.

In [6] it is shown how it is possible to replace a fuzzy constraint involving at least one uncontrollable variable with a fuzzy constraint over controllable variables only. Consider a fuzzy constraint C , represented by the fuzzy relation R , which relates a set of controllable variables $X = \{x_1, \dots, x_n\}$ to a set of uncertain parameters $Z = \{z_1, \dots, z_k\}$ with domains $A_1, \dots, A_k$. The knowledge of the uncertain parameters is modeled with the possibility distribution π_Z defined on $A_Z = A_1 \times \dots \times A_k$. The constraint C is considered satisfied by the assignment $d = (d_1, \dots, d_n) \in D_1 \times \dots \times D_n$ if, *whatever the values of* $z = (z_1, \dots, z_k)$, d is compatible with z , i.e., the set of possible values for z is included in $T = (R \otimes \{(d_1, \dots, d_n)\})^{\downarrow z}$. Obviously $\mu_T(a) = \mu_R(a, d)$ and $\mu'(d) = \mu'_R(d) = N(d \text{ satisfies } C) = N(z \in T) = \inf_{a \in A_Z} \max(\mu_T(a), c(\pi_Z(a))) = c(\sup_{a \in A_Z} \min(c(\mu_T(a)), \pi_Z(a)))$. If C is a hard constraint, then the formula above still applies, and becomes the following one: $N(d \text{ satisfies } C) = \inf_{a \notin T = (R \cap \{d\})^{\downarrow D_Z}} c(\pi_Z(a))$.

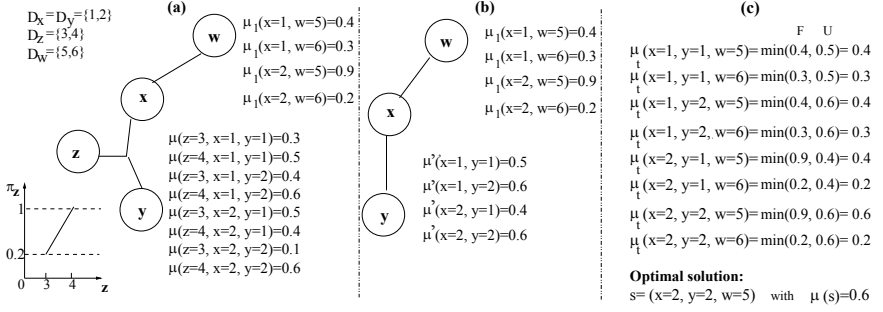


Fig. 1. An example of application of algorithm DFP

Notice that, when C is a soft constraint, μ' is computed by applying the *min* operator between preferences and possibilities. This can be done since their scales are equal assuming the *commensurability* between preferences and possibilities.

Summarizing, the method proposed in [6], which we call **Algorithm DFP** (by the name of the authors), for managing uncertainty in a fuzzy CSP, is the following:

1. It starts from an uncertain fuzzy CSP, say P .
2. P is reduced to a fuzzy constraint problem P' : all the constraints which link uncertain parameters to decision variables are replaced by fuzzy constraints only among the decision variables. The new preference levels of the decision variables in such new constraints are computed by applying the specific procedure given above in this section.
3. P' has only fuzzy constraints, therefore it can be solved by applying the usual method for solving fuzzy CSPs, i.e. using the *min* operator to combine the constraints and choosing the complete assignments with the highest preference.

An application of algorithm DFP to an uncertain fuzzy CSP is shown in Figure 1. Part (a) shows a fuzzy CSP with uncertainty. There are three decision variables (X, Y, W), one uncertain variable (Z), and two constraints: one, C_{XYZ} , among X, Y and Z with function μ and another one, C_{XW} , between X and W with function μ_1 . The constraint C_{XYZ} has membership function μ . The possibility distribution π_Z describes the plausibility of Z . Part (b) shows the fuzzy constraint problem on variables X and Y obtained by the one in part (a). Part (c) shows the complete assignments of the problem in part (b) with their preference degrees defined by μ_t .

In [6] the following property is given.

Property 1: $\mu'_R(d) \geq \alpha$ means that if it is taken for granted that the actual value of z has plausibility strictly greater than $c(\alpha)$, then it is sure that the decision d satisfies C at least at level α (i.e., $\mu'_R(d) \geq \alpha$ means that if $\pi_Z(a) > c(\alpha) \Rightarrow \mu_R(d, a) \geq \alpha$, where a is the actual value of z).

6 Derived Properties

Property 1 continues to hold also if the uncertain parameters are provided with a *probability distribution*, $prob_Z$, and not a possibility one. We have proved this recalling that a possibility is an upper bound to a probability. In this case, Property 1 becomes: $\mu'_R(d) \geq \alpha$ means that if $prob_Z(a) > c(\alpha) \Rightarrow \mu_R(d, a) \geq \alpha$, where a is the actual value of z .

Moreover we have proved the following two properties:

Property 1.1: Once the possibilities of uncertain parameters, defined by π_Z are fixed, if we consider an assignment d to the decision variables $X_1, \dots, X_k$ and an assignment d' such that, $\mu(d, a) \leq \mu(d', a) \forall a$, then the new preference incorporating also uncertainty μ' is such that $\mu'(d) \leq \mu'(d')$.

Property 1.2: Once the preferences $\mu(d, a)$ are fixed, where d is an assignment to the decision variables $X_1, \dots, X_k$ and a is the value of uncertain parameters, if π_1 and π_2 are two possibility distributions on uncertain parameters such that $\pi_1(a) \geq \pi_2(a) \forall a$, then the new preferences incorporating also uncertainty are such that $\mu'_1(d) \leq \mu'_2(d)$, where μ'_1 is the preference obtained considering possibility distribution π_1 and μ'_2 considering possibility distribution π_2 .

7 Separation and Projection

By using algorithm DFP, the preference of a complete assignment is the minimum value among all the preferences of the constraints, both the original fuzzy constraints and those obtained via the transformation which eliminates the uncontrollable variables. By looking again at Figure 1, we can see that the overall preference is $\min(F, U)$, where F is the minimum of the preferences in the initially given fuzzy constraints only on decision variables, and U is the minimum of the preferences of the new fuzzy constraints. This means, for example, that a low overall preference may be caused from a low preference in some of the new fuzzy constraints (when U is less than F), that is, a low compatibility with the uncertain events, or also from a low preference on some fuzzy constraint initially given only on decision variables (when F is less than U). In other words, some information is lost by passing from F and U to $\min(F, U)$.

In other words, according to algorithm DFP, an assignment d associated with the pair of preferences $\langle F, U \rangle$ is compared with another one d' associated with the pair $\langle F', U' \rangle$ by just comparing $\min(F, U)$ and $\min(F', U')$: d is better than d' iff $\min(F, U) > \min(F', U')$. Consider the following situations:

- $F = F'$ and $U > U'$: in this case, one would like to say that d is better than d' . However, this is the case only if $F > U'$, but not if $U' > F$, in which case d and d' are equally preferred. The same reasoning holds also in the dual case when $U = U'$.
- $F = U'$ and $U = F'$: in this case, d and d' are equally preferred, independently of the ordering of the two values. This means that the same

importance is given to both components of the pair, and there is no way to distinguish or consider differently the preference derived from the initial fuzzy constraints among decision variables (that is, F) and the measure of certainty of the problem (that is, U) w.r.t. uncertain parameters.

This problem is caused by the fact that, by using the *min* operator, one forgets about all the other elements, which are higher than the minimum. This is usually called the "drowning effect" [10].

To avoid this problem, it is important to keep separate these two components (F and U), rather than applying the *min* operator over them. Other approaches have gone in this direction. For example, in [1] there are two components which are indeed computed separately; however, solutions are then ordered only via the minimum or the maximum between the two values.

However, just keeping the two components separate will not give the desired ordering among the solutions. In fact, by replacing a fuzzy constraint c between decision variables X and uncertain parameters Y with a new constraint c' over X , and by computing the pair of preferences $\langle F, U \rangle$, it may happen that F is greater than all the preferences appearing in the constraint c . Thus the overall preference for a solution may be high even if this solution has a very low compatibility with all the values of the uncertain variables.

This can be solved by performing, for each constraint c involving both decision and uncertain variables (X and Z), a projection over the decision variables. This will create a new constraint c'' over X where, for each assignment of values to its variables, the preference is computed by assuming the best in the uncertain parameters. Since preferences are combined via the *min* operator, this new constraint will force the overall preference to be no higher than its best preference. Given an assignment to decision variables, we denote with P the minimum preference over these new *projection constraints*. Such value P , combined with F given by the initial constraints, defines the new preference value F_P .

We have proved that $\min(F, U) \leq P$, which implies that projections would be redundant in algorithm DFP, since it computes as final preference $\min(F, U)$.

We therefore propose the following algorithm, which we will call **algorithm SP** (from *separation* and *projection*), to handle uncertain fuzzy constraint problems:

1. It starts from an uncertain fuzzy CSP with fuzzy constraints C .
2. All the constraints which link uncertain parameters to decision variables are replaced by fuzzy constraints only among the decision variables. Let us call C_u such new constraints.
3. It computes the projection constraints, say C_p .
4. For each complete assignment, it computes its overall preference as a pair $\langle F_P, U \rangle$, where $F_P = \min(F, P)$ and F , P , and U are, respectively, the minimum preference over the constraints in C , C_p , and C_u .

Algorithm SP differs from algorithm DFP for points 3 and 4.

Let us consider the following example, where we have a complete assignment d with preference given by the pair $\langle F = 0.3, U = 0.9 \rangle$ and another one d' with

$\langle F = 0.9, U = 0.3 \rangle$. According to algorithm DFP, d is considered equal to d' since d and d' have the same preference $\min(F, U) = 0.3$. We will show that our method distinguishes among them. Since d is associated with the pair $\langle 0.3, 0.9 \rangle$ then the best preference that could be obtained is $F = 0.3$. Moreover there is at least a value $\bar{a}$ for the uncertain parameters such that the tuple $(d, \bar{a})$ has preference 0.3, since in this case $P \geq \min(F, U) = 0.3$ implies $F_P = F$. In addition certainty degree $U = 0.9$ implies if $\pi_Z(a) > 0.1$ then $\mu(d, a) \geq 0.9$, where a is the actual value of z . Thus the preference of (d, a) is < 0.3 only if $\pi_Z(a) \leq 0.1$. This means we have a high certainty to obtain as final preference $F_P = 0.3$.

On the contrary, the fact $F = 0.9$ for d' doesn't imply that such preference is obtained by any pair (a, d') . In fact, assume that P is 0.4. Then for any pair (a, d') the preference is ≤ 0.4 . This is an example of how the information in F can be misleading when the components of the preferences are kept separate and thus needs to be corrected combining it with P . Moreover $U = 0.3$ means that if $\pi_Z(a) > 0.7$ then $\mu(d, a) \geq 0.3$, where a is the actual value of z . Hence we can have a high possibility ≤ 0.7 to have a preference strictly less than 0.3, i.e., d' may have preference 0.9 if $P \geq 0.9$, but there is a high possibility (≤ 0.7) to have a preference < 0.3 .

As it can be seen for the reasoning above d and d' differ on both preference and robustness. This is why we believe it is reasonable to define distinct semantics ordering d and d' differently. In particular, if for example we assume that $P = 0.9$ for d' :

- We should prefer d over d' if we want be *safe*, because assignment d is more cautious than d' , showing a low preference similar to the one that can really happen.
- On the other hand, we should prefer d' over d if we want be *risky*, because assignment d' is more risky than d , showing the best preference that we can obtain, if we are very lucky, i.e., with a low possibility.
- We should not choose between d' and d if we want be *diplomatic*, i.e., we want to have a high preference with a high certainty.

In Figure 2 (a) there is the new fuzzy CSP obtained from the one in Figure 1 (a) after applying the *projection* of the ternary constraint C_{XYZ} on the decision variables X and Y , and the usual step 2 of the algorithm. In Figure 2 (b) all the complete assignments to decision variables of the FCSP. Each assignment, d , is associated with a tuple of three preference values: the first one (P) is the preference obtained by the projection constraints, the second one (F) is the preference given by the initial fuzzy constraint C_{XW} , and the third one is the value obtained by the uncertain parameter Z , which represents the *certainty* that d satisfies C_{XYZ} .

Given an assignment d to the decision variables and the pair $\langle F_P, U \rangle$ computed as described above, F_P tells us how much d is preferred by the constraints, while U represents to what extent it is impossible to have a possible value of the uncertain parameters violating the constraints involving uncertain parameters. This means that $1 - U$ gives an idea of the risk of hitting a value of uncertain

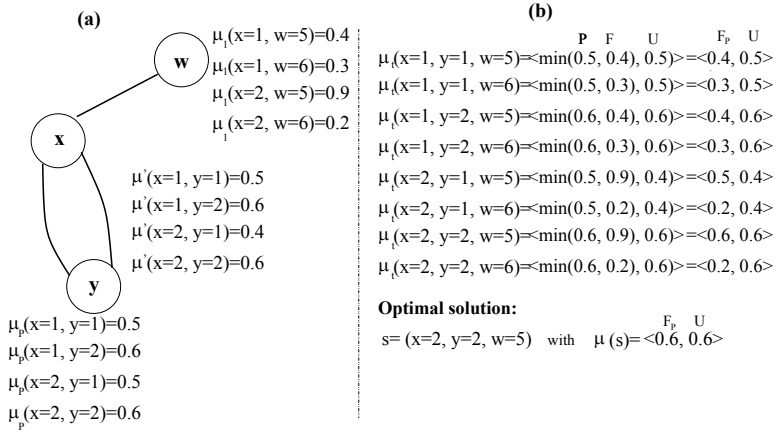


Fig. 2. Result of algorithm SP for the uncertain fuzzy CSP in Figure 1 (a)

parameter that is inconsistent with d , hence U can be seen as a measure of the *certainty* of d .

U is computed as in [6], so Properties 1, 1.1 and 1.2 still hold. We recall that these properties state that U can increase only in two cases: either when the possibilities of the uncertain parameters remain fixed and the preferences of the constraints involving them increase, or when preferences are fixed but possibilities decrease.

8 Three New Semantics

Consider two solutions d and d' and the corresponding pairs of values $\langle F_P(d), U(d) \rangle = \langle a_1, b_1 \rangle$ and $\langle F_P(d'), U(d') \rangle = \langle a_2, b_2 \rangle$.

The first semantics we propose, that we will call *Risky*, can be seen as a Lex ordering on pairs $\langle a_i, b_i \rangle$, with the first component as the most important feature. Hence

- if $a_1 > a_2$ then $\langle a_1, b_1 \rangle >_R \langle a_2, b_2 \rangle$ (and the opposite for $a_2 > a_1$);
- if $a_1 = a_2$ then
 - if $b_1 > b_2$ then $\langle a_1, b_1 \rangle >_R \langle a_2, b_2 \rangle$ (and the opposite for $b_2 > b_1$);
 - if $b_1 = b_2$ then $\langle a_1, b_1 \rangle = \langle a_2, b_2 \rangle$.

Informally, the idea is to give more importance to the preference level that can be reached in the best case (a higher a_i) considering less important a high risk of being inconsistent (a low certainty b_i).

The second semantics, called *Safe*, follows the opposite attitude with the respect to the previous one: it can be seen as a Lex ordering on pairs $\langle a_i, b_i \rangle$, with the second component as most important feature. Hence:

Table 1. Dubois et al. semantics compared to Risky, Safe and Diplomatic

Dubois et al.	Risky	Safe	Dipl.
=	$\langle, \rangle, =$	$\langle, \rangle, =$	$\langle, \rangle, =, \bowtie$
>	$\langle, \rangle$	$\langle, \rangle$	$\rangle, \bowtie$
<	$\langle, \rangle$	$\langle, \rangle$	$\langle, \bowtie$

- if $b_1 > b_2$ then $\langle a_1, b_1 \rangle >_S \langle a_2, b_2 \rangle$ (and the opposite for $b_2 > b_1$);
- if $b_1 = b_2$ then
 - if $a_1 > a_2$ then $\langle a_1, b_1 \rangle >_S \langle a_2, b_2 \rangle$ (and the opposite for $a_2 > a_1$);
 - if $a_1 = a_2$ then $\langle a_1, b_1 \rangle = \langle a_2, b_2 \rangle$.

Informally, the idea is to give more importance to the certainty level that can be reached (a higher b_i) considering less important having a high preference (a high a_i).

Our third semantics, called *Diplomatic*, aims at giving the same importance to the two aspects of a solution: preference and certainty. In order to do that, it is obtained via the Pareto ordering on pairs $\langle a_i, b_i \rangle$, where both components have the same importance. Hence:

- if $a_1 \leq a_2$ and $b_1 \leq b_2$ then $\langle a_1, b_1 \rangle \leq_D \langle a_2, b_2 \rangle$ (and the opposite for $a_2 \leq a_1$ and $b_2 \leq b_1$);
- if $a_1 = a_2$ and $b_1 = b_2$ then $\langle a_1, b_1 \rangle = \langle a_2, b_2 \rangle$;
- else $\langle a_1, b_1 \rangle \bowtie \langle a_2, b_2 \rangle$.

In this definition $\bowtie$ stands for incomparability. The idea is that a pair is to be preferred to another only if it wins both on preference and certainty, leaving incomparable all the pairs that have one component higher and the other lower. Contrarily to the diplomatic semantics, the other two semantics produce a total order over the solutions.

Figure 2 (b) shows a solution of the FCSP in Figure 1 which is optimal according to all semantics.

Let us now consider an example that explains the differences between our semantics and the approach of [6]. Suppose we have two complete assignments, d_1 and d_2 , with preference resp. 0.3 and 0.5, and certainty resp. 0.5 and 0.3. Then the method of [6] would say that they are equally good, since it would just consider the minimum of each pair, that is, 0.3. On the other hand, for our semantics we have the following ordering: $\langle 0.3, 0.5 \rangle <_R \langle 0.5, 0.3 \rangle$ according to Risky; $\langle 0.3, 0.5 \rangle >_S \langle 0.5, 0.3 \rangle$ according to Safe; $\langle 0.3, 0.5 \rangle \bowtie \langle 0.5, 0.3 \rangle$ according to Diplomatic.

In general, the comparison among the orders induced by our three semantics and the one of the method in [6] can be seen in Table 1.

9 Conclusion and Future Work

We defined a new way to deal with preference and uncertainty which assumes commensurability but does not mix preferences and compatibility with uncertain events. This allows us to obtain a solution ordering which better reflects the desirability and the robustness of a solution. Other approaches which do not mix these two aspects [12, 5, 3, 4] do not assume commensurability and thus cannot compare directly preferred assignments and uncertain events.

We plan to develop a solver that can handle problems with several classes of soft constraints, together with uncertainty expressed via possibility or probability distributions. The solver will be able to generate orderings according the three semantics proposed in this paper as well as others that we will define by following different optimistic or pessimistic approaches.

We plan also to extend the results of this paper to other classes of soft constraints (such as probabilistic and weighted) and also to probabilistic uncertainty.

References

1. L. Amgoud and H. Prade. Using arguments for making decisions: a possibilistic logic approach. Proc. UAI, 10-17, 2004.
2. S. Bistarelli, U. Montanari and F. Rossi. Semiring-based Constraint Solving and Optimization. Journal of the ACM, Vol.44, n.2, March 1997.
3. R. I. Brafman and M. Tennenholtz. On the Foundations of Qualitative Decision Theory. Proc. AAAI'96, 1291-1296, MIT Press, 1996.
4. R. I. Brafman and M. Tennenholtz. On the Axiomatization of Qualitative Decision Criteria. Proc. AAAI'97, 1997.
5. Hélène Fargier and Régis Sabbadin. Qualitative decision under uncertainty: back to expected utility. Proc. IJCAI'03, 303-308, 2003.
6. D. Dubois, H. Fargier and H. Prade. Possibility theory in constraint satisfaction problems: handling priority, preference and uncertainty. Applied Intelligence, 6, 287-309, 1996.
7. D. Dubois and H. Prade. Possibility theory: qualitative and quantitative aspects. Handbook of Defeasible Reasoning and Uncertainty Management Systems, Vol.1, 169-226, Kluwer, 1998.
8. D. Dubois and H. Prade. Fuzzy sets and Systems - Theory and Applications. Academic Press, 1980.
9. D. Dubois and H. Prade. Possibility theory. Plenum Press, 1988.
10. D. Dubois and H. Prade. Belief Revision and Updates in Numerical Formalisms: An Overview, with New Results for the Possibilistic Framework. In Proc. IJCAI 1993, 620-625, Chambery, France, 1993.
11. D. Dubois and H. Prade. Fundamentals of Fuzzy Sets. Handbooks of Fuzzy Sets Series, Kluwer, 2000.
12. D. Dubois, H. Fargier and P. Perny. On the limitations of ordinal approaches to decision making Proc. KR 2002, 133-144, 2002.
13. H. Fargier, J. Lang, T. Schiex. Mixed constraint satisfaction: a framework for decision problems under incomplete knowledge constraint satisfaction framework for decision under uncertainty. Proc. AAAI'96, 1996.
14. L.A. Zadeh. Fuzzy sets as a basis for the theory of possibility. Fuzzy Sets and Systems, 13-28, 1978.

About the Processing of Possibilistic and Probabilistic Queries

Patrick Bosc and Olivier Pivert

IRISA/ENSSAT, Technopole Anticipa BP 80518,
22305 Lannion Cedex, France
{Patrick.Bosc, Olivier.Pivert}@enssat.fr

Abstract. In this paper, the issue of querying databases that may contain ill-known values represented by disjunctive weighted sets (possibility or probability distributions) is considered. The queries dealt with are of the form: “to what extent is it possible (or probable, depending on the framework considered) that tuple t belongs to the result of query Q ”, where Q denotes a usual relational query. In the possibilistic database framework, some previous works resulted in the definition of an evaluation method that does not entail computing the different possible worlds of the database. In this paper, we show that this method cannot be used in the probabilistic database framework in general. On the other hand, we describe an alternative evaluation method that is suitable for probabilistic queries when Q complies with certain constraints.

1 Introduction

In different application domains, one can notice a growing need for information systems capable of dealing with ill-known data. It is the case in particular of data warehouses for which the gathering of information coming from different sources may induce imprecision (up to now, many such systems deal with this issue by means of usual “data cleaning” methods, at the expense of a loss of information). Possibility theory [6] proposes an ordinal model of uncertainty in which imprecision is coded by means of a preference relation that defines a total order over possible situations. This framework is strongly linked to fuzzy sets since the idea is to constrain the values that can be taken by a variable, by means of a normalized fuzzy set. A possibility distribution is an application π from a given domain into the unit interval $[0, 1]$ and $\pi(a)$ gives the possibility degree associated with the fact that the actual value of the variable is a . The normalization condition imposes that at least one value (a_0) is completely possible, i.e., $\pi(a_0) = 1$. This formalism is particularly suited to the expression of subjective uncertainties described by means of linguistic terms such as big, young, rather small, etc. When the domain is finite, a possibility distribution is denoted by: $\{\pi_1/a_1 + \dots + \pi_n/a_n\}$ where a_i is a candidate value and π_i is its possibility degree. In the context of relational databases, the possibilistic model offers a unified framework that enables to represent precise values as well as imprecise ones such as usual or fuzzy intervals, and also null values [2].

In the following, we call *imprecise database* a database where some attribute values may be ill-known and represented as disjunctive weighted sets, i.e., as possibility or probability distributions according to the framework chosen to model imprecision.

emp	#id	age	job
	1	$\{\rho_1/35, \rho_2/34, \rho_3/33\}$	cashier
	2	$\{\rho_4/38, \rho_5/39\}$	$\{\rho_6/\text{salesman}, \rho_7/\text{cashier}\}$
	3	$\{\rho_8/34, \rho_9/35\}$	salesman
	4	32	salesman

Let us recall that, from a semantic point of view, an imprecise database B can be interpreted as a set of usual databases (also called *worlds*) $B_1, \dots, B_p$ (denoted by $\text{rep}(B)$), each of these being more or less possible (or probable, depending on the framework used). Each B_i is obtained by choosing a candidate value in every distribution appearing in B , and one of these databases B_k is supposed to represent the actual state of the universe modeled. The degree assigned to a world is equal to the result of the aggregation by means of an operator $\otimes$ (the product in the probabilistic framework, a triangular norm – usually $\min$ – in the possibilistic one) of the degrees initially attached to the chosen candidates, under the assumption that the attributes are non-interactive (the choices of the candidate values are independent). In the following, we will also use the notion of an *interpretation*, defined as a world local to a given relation (the interpretations of a relation r thus correspond to the worlds attached to the database restricted to r). For instance, relation *emp* above corresponds to $24 (3 * 2 * 2 * 2)$ more or less possible (resp. probable) interpretations. In this relation, $\{\rho_6/\text{salesman}, \rho_7/\text{cashier}\}$ means that the job of employee 2 is either salesman or cashier, that the possibility (resp. probability) degree attached to the candidate “salesman” equals ρ_6 and that attached to “cashier” equals ρ_7 . The choices: i) 35 for age in the first tuple, ii) 38 for age and cashier for job in the second one, and iii) 35 for age in the third one lead to an interpretation whose degree equals $\otimes(\rho_1, \rho_4, \rho_7, \rho_9)$.

In this paper, we consider the issue of querying imprecise databases by means of specific queries, called possibilistic (resp. probabilistic) queries, whose general form is: “to what extent is it possible (resp. probable) that tuple t belongs to the answer to query Q ?”, where Q denotes a usual relational query. Such queries were initially studied by Abiteboul [1] in a context of databases involving null values. Our objective is to define an efficient processing strategy for such queries. The naïve strategy consisting in processing Q in every world of the database would obviously be unrealistic considering the number of worlds. In [3], we described a processing strategy, called “compact”, enabling to deal with possibilistic queries. The present contribution mainly aims at determining whether this strategy is also suited to the probabilistic database/query case.

The remainder of the paper is organized as follows. Section 2 recalls the principle of the “compact” evaluation strategy that was previously defined for processing possibilistic queries. Section 3 is devoted to the problems raised by the transposition of this method to a probabilistic database framework. These problems entail severe

restrictions as to the expression power of the query language, and an alternative strategy is studied in Section 4, based on the processing of the algebraic query underlying the probabilistic query considered. This approach entails extending the database model and redefining the relational operators in an appropriate way. Finally, the conclusion summarizes the contribution and draws some lines for future work.

2 About the Compact Processing of Possibilistic Queries

We first describe the general principle of the compact processing method, then we give the definitions of the compact operators of selection and join (due to space limitation, the other operators are not detailed here).

2.1 General Principle of the Compact Processing Method

As mentioned above, it is generally unrealistic to envisage processing a possibilistic query Q^p by evaluating Q in each world of the database considered.

It is thus crucial to have available a processing method that avoids computing the different worlds of the possibilistic database involved. However, even though the world-based approach is untractable, it still constitutes the semantic reference. The feasibility of a method that does not make the worlds explicit rests on the following remark. In a possibilistic query, one strives to determine the most possible world in which tuple t can be obtained in the result of Q , so it is not necessary to take into account individual worlds but only their union. It is thus sufficient to guarantee that the compact calculus produces all the tuples resulting from the evaluation of Q against the union of the worlds associated to the database B , and only those tuples.

Figure 1 describes the two aforementioned processing strategies: the world-based one (arrow 2) and the compact one (arrow 3). Arrow 1 corresponds to a preprocessing that uses the knowledge we have about the “target tuple” t appearing in Q^p in order to reduce the size of the database and thus the volume of intermediary results. The idea is to reduce the initial imprecise database B by discarding the attributes which do not

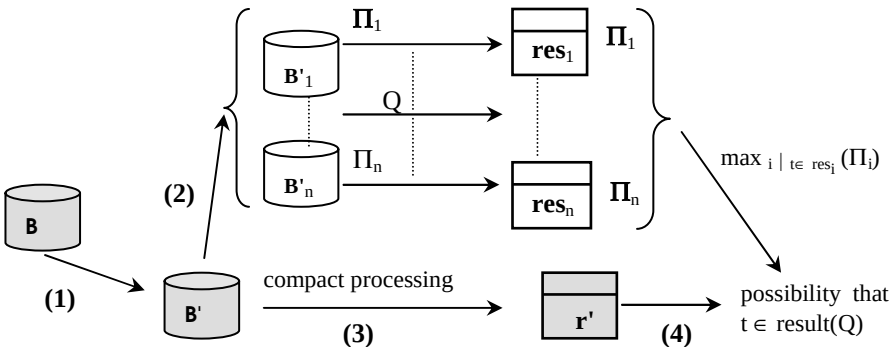


Fig. 1. Two processing strategies

appear in Q and by restricting the possibility distributions present in B to the sole candidate values that can generate the target tuple (which, in some cases, leads to the removal of some tuples). The reduced database obtained is denoted by B' . The step corresponding to arrow 4 aims at computing the degree that constitutes the answer to Q^p by scanning the tuples in the result $resQ$ of the compact processing of Q . By nature, $resQ$ only contains tuples whose attribute values correspond to those of the target tuple, and a degree is associated with each of them, which leads to:

$$resQ = \{t_i = \langle \pi_{i,1}/a_1, \dots, \pi_{i,p}/a_p \rangle \mid t = \langle a_1, \dots, a_p \rangle\}.$$

One then has to look for the most possible tuple among the t_i 's (cf. section 2.3).

2.2 Compact Possibilistic Operators

The operators that may appear in Q (i.e., those for which a compact processing is feasible) are the selection, the join, the projection, the intersection and the union. The compact versions of the first two ones are given hereafter.

2.2.1 Selection

Due to space constraints, we only deal with selection conditions of the form $A_i \theta v$ where A is an attribute, θ a comparison operator and v a constant. Let $R(A_1, \dots, A_n)$ be the schema of the relation r concerned. A tuple u of r produces a tuple in the result iff at least one candidate value $a_{i,k}$ from $u.A_i$ satisfies $(a_{i,k} \theta v)$. The resulting tuple u' has the same values as u for every attribute except A_i : $u'.A_i$ only contains the candidates $a_{i,k}$ of $u.A_i$ which satisfy $(a_{i,k} \theta v)$. This selection operation can be defined as:

$$select_{po}(r, A_i \theta v) = \{\langle u.A_1, \dots, u.A_{i-1}, restrict(a_i, P), u.A_{i+1}, \dots, u.A_n \rangle \mid u \in r\}$$

with: $restrict(a_i, P) = \{\pi_{i,k}/a_{i,k} \mid a_{i,k} \in scv(u.A_i) \text{ and } \pi_{i,k} = \pi_{u.A_i}(a_{i,k}) \text{ and } P(a_{i,k})\}$ where $P(x) = \text{true}$ if $x \theta v$, false otherwise, and $scv(u.A_i)$ denotes the set gathering the candidate values present in $u.A_i$.

2.2.2 Join

Let r and s be two relations of respective schemas $R(A, X)$ and $S(B, Y)$. Let us consider the join of r and s according to the criterion $A \theta B$. A pair of tuples (u, v) of $r \times s$ produces a tuple in the result iff at least one candidate value for $u.A$ is in relation θ with at least one candidate value for $v.B$. The principle consists in generating a tuple t_i for every satisfactory candidate value a_i from $u.A$: the value of $t_i.A$ is a_i along with its degree, the value of $t_i.B$ is the distribution $v.B$ restricted to the values b_j which satisfy $a_i \theta b_j$. This operation can be defined as:

$$join_{po}(r, s, A \theta B) = \{\langle \{\pi_i/a_i\}, u.X, restrict(v.B, P(a_i)), v.Y \rangle \mid \exists u \in r, \exists v \in s \text{ such that } u.A = \{\pi_1/a_1, \dots, \pi_i/a_i, \dots, \pi_n/a_n\}\}$$

where $restrict$ is defined the same way as for the selection (in this case, we have $P(a_i)(x) \equiv a_i \theta x$). In case of an equi-join, the expression above becomes:

$$equi-join_{po}(r, s, A = B) = \{\langle \{\min(\pi_i, \pi'_j)/a_i\}, u.X, v.Y \rangle \mid \exists u \in r, \exists v \in s \text{ such that } u.A = \{\pi_1/a_1, \dots, \pi_i/a_i, \dots, \pi_n/a_n\} \text{ and } v.B = \{\pi'_1/b_1, \dots, \pi'_j/b_j, \dots, \pi'_m/b_m\} \text{ and } a_i = b_j\}.$$

2.3 Example

Let us consider the two relations hereafter:

S	#i	type	date	location	L	i	type	date	location
	i ₁	{1/a ₂ , 0.5/a ₁ }	d ₁	{1/l ₂ , 0.8/l ₁ }		k ₁	{1/a ₄ , 0.6/a ₂ }	d ₃	{1/l ₁ , 0.7/l ₃ }
	i ₂	{1/a ₃ , 0.8/a ₂ }	d ₂	1/l ₄ + 0.7/l ₁		k ₂	{1/a ₂ , 0.8/a ₄ }	d ₁	{1/l ₁ , 0.5/l ₄ }
	i ₃	{1/a ₃ , 0.7/a ₂ }	d ₁	l ₁		k ₃	{1/a ₄ , 0.8/a ₂ }	d ₁	{1/l ₂ , 0.6/l ₁ }
	i ₄	{1/a ₄ , 0.2/a ₁ }	d ₁	l ₂		k ₄	{1/a ₅ , 0.7/a ₃ }	d ₁	l ₁

These relations describe two collections of images taken by two different satellites. Each image is supposed to represent one aircraft, whose type is generally ill-known due to the imprecision inherent in the recognition process. Let us consider the possibilistic query: “to which extent is it possible that an aircraft of type a₂ has been observed at the same place on the same date by both satellites?”. In other words, to which extent is it possible that a₂ belongs to the answer to:

project_{po}(select_{po}(equi-join_{po}(select_{po}(S, date = “d₁”),
select_{po}(L, date = “d₁”), {type}), location1 = location2), {type})).

The first step discards attribute #i from S and L, keeps in S and in L only those tuples which involve the candidate A-value a₂ and discards the other candidate A-values from the remaining tuples. One then performs the selections (date = “d₁”), and one gets:

S'	type	date	location	L'	type	date	location
	{1/a ₂ }	d ₁	{1/l ₂ , 0.8/l ₁ }		{1/a ₂ }	d ₁	{1/l ₁ , 0.5/l ₄ }
	{0.7/a ₂ }	d ₁	l ₁		{0.8/a ₂ }	d ₁	{1/l ₂ , 0.6/l ₁ }

The equi-join of S' et L' based on attribute “type” produces:

SL	type	date1	location1	date2	location2
	{1/a ₂ }	d ₁	{1/l ₂ , 0.8/l ₁ }	d ₁	{1/l ₁ , 0.5/l ₄ }
	{0.8/a ₂ }	d ₁	{1/l ₂ , 0.8/l ₁ }	d ₁	{1/l ₂ , 0.6/l ₁ }
	{0.7/a ₂ }	d ₁	l ₁	d ₁	{1/l ₁ , 0.5/l ₄ }
	{0.7/a ₂ }	d ₁	l ₁	d ₁	{1/l ₂ , 0.6/l ₁ }

The selection “location1 = location2” is computed as an equi-join and produces:

SL'	type	date1	location1	date2	location2
	{1/a ₂ }	d ₁	{0.8/l ₁ }	d ₁	{1/l ₁ }
	{0.8/a ₂ }	d ₁	{1/l ₂ }	d ₁	{1/l ₂ }
	{0.8/a ₂ }	d ₁	{0.8/l ₁ }	d ₁	{0.6/l ₁ }
	{0.7/a ₂ }	d ₁	l ₁	d ₁	{1/l ₁ }
	{0.7/a ₂ }	d ₁	l ₁	d ₁	{0.6/l ₁ }

The projection on attribute “type” involves the computation of a minimum locally to each tuple and produces: $\{<0.8/a_2>, <0.6/a_2>, <0.7/a_2>\}$. As to the last step, it yields the grade attached to the target value (a_2), i.e., $\max(0.8, 0.6, 0.7) = 0.8$.

3 On the Validity of the Method in a Probabilistic Context

This section aims at showing that the compact method described above is not suitable in general for probabilistic queries. Let us consider a probabilistic database involving the two relations r and s of respective schemas $R(A, B)$ et $s(B, C)$, the extensions:

r	A	B	s	B	C
	a_1	$\{0.9/b_1, 0.1/b_2\}$		$\{0.7/b_1, 0.3/b_2\}$	c_1

and the probabilistic query Q^p : “to what extent is it probable that a_1 belongs to the answer to $Q = \text{project}(\text{equi-join}(r, s, \{B\}), \{A\})$?”.

Processing Q against the different worlds of the database leads to:

M_1	r	<table><tr><th>A</th><th>B</th></tr><tr><td>a_1</td><td>b_1</td></tr></table>	A	B	a_1	b_1	<table><tr><th>B</th><th>C</th></tr><tr><td>b_1</td><td>c_1</td></tr></table>	B	C	b_1	c_1	$P = 0.63$ query result : $\langle a_1 \rangle$
A	B											
a_1	b_1											
B	C											
b_1	c_1											
M_2	r	<table><tr><th>A</th><th>B</th></tr><tr><td>a_1</td><td>b_1</td></tr></table>	A	B	a_1	b_1	<table><tr><th>B</th><th>C</th></tr><tr><td>b_2</td><td>c_1</td></tr></table>	B	C	b_2	c_1	$P = 0.27$ query result : $\emptyset$
A	B											
a_1	b_1											
B	C											
b_2	c_1											
M_3	r	<table><tr><th>A</th><th>B</th></tr><tr><td>a_1</td><td>b_2</td></tr></table>	A	B	a_1	b_2	<table><tr><th>B</th><th>C</th></tr><tr><td>b_1</td><td>c_1</td></tr></table>	B	C	b_1	c_1	$P = 0.07$ query result : $\emptyset$
A	B											
a_1	b_2											
B	C											
b_1	c_1											
M_4	r	<table><tr><th>A</th><th>B</th></tr><tr><td>a_1</td><td>b_2</td></tr></table>	A	B	a_1	b_2	<table><tr><th>B</th><th>C</th></tr><tr><td>b_2</td><td>c_1</td></tr></table>	B	C	b_2	c_1	$P = 0.03$ query result : $\langle a_1 \rangle$
A	B											
a_1	b_2											
B	C											
b_2	c_1											

The result of Q^p is then $0.63 + 0.03$ (since the two situations are independent) $= 0.66$.

Now, if we use a definition of the join similar to that presented in section 2.2.2, the compact processing leads to the relation $r' = \{<a_1, \{0.63/b_1\}, c_1>, <a_1, \{0.03/b_2, c_2>\}$. The probability for a tuple t to be present in at least one interpretation of a probabilistic relation s is equal to 1 minus the probability for t to be absent from every interpretation of s . It is thus the probability of the event: the first tuple of s does not represent t and ... and the last tuple of s does not represent t . However, this calculus is sound only if s has indeed the semantics of a relation, i.e., of a conjunctive set. Here, the calculus of the probability attached to the presence of a_1 in r' leads to:

$$1 - (1 - 0.63) * (1 - 0.03) = 1 - 0.37 * 0.97 = 1 - 0.3589 = 0.6411.$$

The erroneous nature of this result comes from the non-independence between the two tuples (which are in fact mutually exclusive). More generally, what makes the compact strategy valid in the possibilistic framework is that one looks for the best (in

the sense of a maximal degree) occurrence of an event. On the other hand, the probabilistic framework has an additive nature and in order to compute the probability of an event, one has to aggregate the partial results corresponding to the different worlds where this event occurs. Therefore, it is not sufficient to consider the union of the worlds since it is mandatory to distinguish between the case where an event occurs twice in a given world and the situation where it occurs once in two different worlds.

4 A Method Based on the Processing of the Underlying Algebraic Query

In order to overcome the previous difficulty, the idea, here, is to process Q and check whether tuple t belongs to its result. In order to do so, one must be able to represent the result of Q by means of a table of the model (i.e., in a “compact” way, not as a set of tables corresponding to the results obtained in the different worlds since, for efficiency reasons, the processing must not be based on the elicitation of worlds). Such a compact calculus is possible only if Q only contains operators for which the model constitutes a strong representation system [5], i.e., operators op such that:

$$op(rep(B)) = rep(op_c(B)) \quad (1)$$

where op_c is the “compact version” of op . As it will be seen in the following, this is possible only for a subset of the operators of relational algebra. Notice that with this strategy, the preprocessing evoked before (arrow 1 of Figure 1) still makes sense.

4.1 Database Model

Some selection criteria (and join conditions) induce dependencies between candidate values related to different attributes of a same tuple. This is not compatible with the hypothesis made in the initial model, which stated that the distributions of candidate values were independent. For instance, let us consider two attributes A and B that can take imprecise values and whose respective sets of candidate values in tuple t are $\{a_1, a_2\}$ and $\{b_1, b_2, b_3\}$. If, accordingly to a certain selection criterion, the only valid associations are (a_1, b_1) , (a_1, b_3) , (a_2, b_2) and (a_2, b_3) , it is impossible to represent them as a Cartesian product of subsets of $t.A$ and $t.B$. It is therefore necessary to represent the valid associations in an explicit manner, which entails using a model involving distributions built over several attributes. Such a distribution can be modeled by means of a nested relation and two points are worth being mentioned:

- the tuples of a nested relation are weighted since a distribution is a weighted set,
- a nested relation has a disjunctive semantics since a distribution gathers mutually exclusive candidates. Hence, level-one relations are conjunctive whereas nested ones are disjunctive. Consequently, there is no semantic ambiguity.

Moreover, let us mention that a single level of nesting suffices. In the following, $R(A_1, \dots, A_m, X_1(A_p, \dots, A_q), \dots, X_n(A_k, \dots, A_r))$ represents a schema where:

- $A_1, \dots, A_m$ are elementary attributes (also called level-one attributes) whose values are either precise or distributions,
- $X_i(A_h, \dots, A_j)$ denotes a structured attribute X_i whose values are distributions built over the attributes $A_h, \dots, A_j$, called nested attributes.

The interpretations of a relation whose schema involves nested relations are of course regular relations as usual. More details about the model can be found in [4].

Example 1. Let us consider the (probabilistic) intermediary relation whose schema is $R(A, B, X(C, D, E))$ and whose extension is given hereafter.

A	B	X			
		C	D	E	p
a ₁	{0.6/b ₁ , 0.4/b ₄ }	c ₂	d ₁	e ₃	1
a ₂	b ₃	c ₁	d ₁	e ₂	0.2
		c ₃	d ₂	e ₃	0.2
		c ₂	d ₄	e ₂	0.4
		c ₂	d ₁	e ₃	0.1

In this relation, the second tuple could also be denoted by:

$$\langle a_2, b_3, \{0.2/\langle c_1, d_1, e_2 \rangle, 0.2/\langle c_3, d_2, e_3 \rangle, 0.4/\langle c_2, d_4, e_2 \rangle, 0.1/\langle c_2, d_1, e_3 \rangle\} \rangle$$

which means that the nested relation is nothing but a distribution defined over the attributes C, D and E. Let us notice that this tuple has five interpretations, not four (the sum of the degrees is less than 1). Therefore, this probabilistic relation has ten interpretations. The degree attached to that containing the sole tuple $\langle a_1, b_4, c_2, d_1, e_3 \rangle$ equals: $(1 * 0.4 * 1) * (1 - 0.9)$. As to the one containing the two tuples $\langle a_1, b_1, c_2, d_1, e_3 \rangle$ and $\langle a_2, b_3, c_3, d_2, e_3 \rangle$, its degree is equal to $(1 * 0.6 * 1) * (1 * 1 * 0.2)$. In the first case, $(1 - 0.9)$ corresponds to the probability that the second tuple from the initial imprecise relation has no representative in a given world. ♦

4.2 About Binary Operators

Let us recall that it is not possible, in the relational framework, to define a strong representation system enabling to deal with the join in the presence of ill-known data. The problem is related to the coexistence of disjunctive and conjunctive tuples in the result. The same phenomenon can be observed for other operators and one can characterize the operations that are incompatible with the notion of a strong representation system in this context. Let us first categorize the binary operators the following way: type 1 (resp. type 2) operators are such that a tuple of a relation can participate in the production of at most one (resp. several) tuple(s) of the result.

An example of a type 1 operation is the selection. Operations of type 2 include intersection, difference, Cartesian product and join. However, in some special cases

related to the presence of keys among the arguments, a type 2 operation can behave as a type 1 one (this is notably the case of the fk-join which will be detailed later).

Proposition 1. In the relational framework, one cannot define a strong representation system enabling to deal with type 2 operations in the presence of ill-known values.

Justification. Let o be an operation, r and s its two operand relations and t a tuple of r . Let us assume that t involves an ill-known attribute value $t.A = \{p_1/a_1, \dots, p_n/a_n\}$. Let us also suppose that t participates in the production of two tuples u_1 and u_2 in the resulting relation res , so that $u_1.A = u_2.A = t.A$ (depending on operation o , $u_1.A$ and $u_2.A$ can be two different subsets of $t.A$ but this does not affect the reasoning). Every interpretation of res must be such that the same choice of candidates has been made in $u_1.A$ and $u_2.A$ (since u_1 and u_2 both stem from the same tuple t). Unfortunately, in a relational table, there is no way of representing such an inter-tuple dependency.

4.3 The Operators of the Language

In this section, we define the selection and the join in the model described above. The other valid operations (projection and union) are not detailed due to space limitation.

4.3.1 Selection

The basic idea is to keep the candidate values which satisfy the selection criterion. When a selection condition is of the form “attribute θ constante”, the structure of the result is the same as that of the input relation. If this schema is $R(A, B)$ and the selection concerns A , the operation is defined as:

$$\text{select}_{p_r}(r, \theta(A, v)) = \{ \langle \text{restrict}(t.A, \theta(A, v), t.B) \rangle \mid t \in r \}$$

with: $\text{restrict}(t.A, \theta(A, v)) = \{p_i/a_i \mid a_i \in \text{scv}(t.A) \text{ and } \theta(a_i, v) \text{ and } p_i = p_{t.A}(a_i)\}$. It is possible to prove that this definition is consistent with property 1.

Let us now consider a selection condition of the form $\theta(A_1, A_2)$, or a disjunctive one such as “ $c_1(A_1)$ or $c_2(A_2)$ ”. If A_1 and A_2 can take ill-known values, it is necessary to gather their candidate values in a nested relation so that only the correct pairs of candidate values are kept. If the schema of r is $R(A_1, A_2, B)$, the schema of the result is then $R'(X(A_1, A_2), B)$. The corresponding formal definition can be found in [4].

4.3.2 Fk-Join

Let us consider the relation schemas $R(X, Z)$ and $S(X, Y)$, a relation $r(R)$ where X and Z may take ill-known values, and a usual relation $s(S)$ in which the functional dependency $X \rightarrow Y$ is valid. Attribute X is thus a key of s and then a foreign key of r . The principle of the fk-join [4] is to complete every tuple of r by adding an X -component to it (it is thus a type 1 operation). By definition, this leads to a resulting relation of schema $((X, Y), Z)$, which involves a nested relation whose purpose is to link the pairs of candidate values over X and Y .

Let us assume that A is an elementary attribute. From two relations r and s of respective schemas $R(A, B)$ and $S(A, C)$, one gets a resulting relation of schema $(X(A, C), B)$ and the fk-join operation can be defined as:

$$\text{fk-join}_{\text{pr}}(r, s, (A, A)) = \{ \langle \{ \dots, p_k / \langle a_k, c \rangle, \dots \}, b \rangle \mid (\exists t \in r \text{ such that } t = \langle a, b \rangle \text{ with } a = \{ p_1 / a_1, \dots, p_n / a_n \}) \text{ and } (\exists u \in s \text{ such that } u = \langle a_k, c \rangle) \}.$$

Again, it can be proven that property 1 holds.

4.4 Example

Let us consider the relation schemas (#i, t-a, date, loc) and (t-a, lg, msp) (where msp stands for maximal speed) as well as the following extensions:

I	#i	type	date	loc
	i ₁	{0.8/a ₂ , 0.2/a ₁ }	{0.7/d ₃ , 0.3/d ₁ }	{0.5/p ₂ , 0.5/p ₁ }
	i ₂	{0.5/a ₃ , 0.3/a ₂ , 0.1/a ₄ , 0.1/a ₁ }	{0.8/d ₂ , 0.2/ d ₃ }	{0.7/p ₃ , 0.3/p ₄ }
	i ₃	{0.7/a ₃ , 0.3/a ₄ }	d ₁	p ₁
	i ₄	{0.8/a ₄ , 0.2/a ₁ }	{0.9/d ₁ , 0.1/ d ₃ }	p ₂
	i ₅	{0.6/a ₁ , 0.4/a ₄ }	{0.5/d ₃ , 0.5/ d ₁ }	{0.8/p ₃ , 0.2/p ₁ }

A	type	lg	msp
	a ₁	20	1200
	a ₂	20	900
	a ₃	18	800
	a ₄	20	1400

and the query “to what extent is it probable that (d₁, 20) belongs to the answer to:

project((fk-join(select(I, loc ≠ “p₂”), select(A, msp > 1000), {type}), {date, lg})).”.

In other terms, one looks for the probability that an aircraft of length 20 and of maximal speed over 1000 has been observed at a location different from p₂ on date d₁. The first step suppresses attribute #i from relation I, and filters the values of attribute date in I and lg in A. One then performs the selections and gets:

I'	type	date	loc
	{0.8/a ₂ , 0.2/a ₁ }	{0.3/d ₁ }	{0.5/p ₁ }
	{0.7/a ₃ , 0.3/a ₄ }	d ₁	p ₁
	{0.6/a ₁ , 0.4/a ₄ }	{0.5/ d ₁ }	{0.8/p ₃ , 0.2/p ₁ }

A'	type	lg	msp
	a ₁	20	1200
	a ₄	20	1400

The fk-join of relations I' and A' according to attribute type yields:

IA	X				date	loc
	type	lg	msp	p		
	a ₁	20	1200	0.2	{0.3/d ₁ }	{0.5/p ₁ }
	a ₄	20	1400	0.3	d ₁	p ₁
	a ₁	20	1200	0.6	{0.5/d ₁ }	{0.8/p ₃ , 0.2/p ₁ }
	a ₄	20	1400	0.4		

The projection works as in the classical case except that the degrees attached to the values of the suppressed attributes must be aggregated (by means of a product) to the degree attached to the value taken by one of the remaining attributes. Here, the projection onto attributes date and lg produces the final relation:

$$\{<\{0.03/d_1\}, 20>, <\{0.3/d_1\}, 20>, <\{0.5/d_1\}, 20>\}.$$

The final result is then: $1 - (1 - 0.03) * (1 - 0.3) * (1 - 0.5) = 1 - 0.3395 = 0.6605$.

5 Conclusion

This paper deals with the querying of possibilistic and probabilistic databases, where some attribute values are represented by disjunctive weighted sets (possibility or probability distributions). The queries considered have the general form: “to what extent is it possible (resp. probable) that tuple t belongs to the answer to query Q ?”, where Q denotes a usual relational query.

The objective is to process such queries without computing the worlds associated with the database. Such a method has been previously defined for possibilistic databases and a question was whether it could be used in the probabilistic context too. The answer is negative, due to the additive nature of the probabilistic framework, because then it is not sufficient to handle the union of the worlds corresponding to the database. Consequently, we have proposed to process a probabilistic query by computing the result of the underlying relational query, which implies to have available a strong representation system for the operators which may appear in this query. Such a model is presented here, which rests on nested relations and enables to deal with queries involving selections, projections, fk-joins and unions (providing that the input relations are independent, in the latter case). The constraints (in terms of authorized operators) on the underlying relational query are thus more restrictive in the probabilistic framework than in the possibilistic one, but they still correspond to a rather large range of queries.

It is important to mention that probabilistic and possibilistic queries are not the counterpart of each other. Indeed, the counterpart of a probabilistic query in the possibilistic context should be a *pair* of queries: “to what extent is it possible ...” and “to what extent is it certain ...”. It appears that the latter kind (certainty-based queries) cannot be processed in a compact way either (for different reasons than probabilistic queries, though) but one can use the same alternative strategy as described in Section 4 to process them, with the same constraints as to the operators that may appear in them. It is also possible to take advantage of the property that states that a Boolean event is totally uncertain when its possibility does not equal 1, to avoid processing the certainty-based query when the possibility-based one has issued a degree less than 1.

Several lines for future work can be thought of. On the one hand, it would be useful to study the computational complexity of the evaluation process more deeply. In particular, it would be worth assessing, by means of experimental measurements, the additional cost tied to the presence of ill-known values. On the other hand, an extension of this work would consist in studying analogous queries such as “to what extent is it possible (resp. probable) that the tuples $t_1, \dots, t_n$ belong *jointly* to the answer

to Q ", as well as queries which do not involve any target tuple such as: "to what extent is it possible (resp. probable) that the answer to Q is non-empty".

References

- [1] Abiteboul S., Kanellakis P., Grahne G., On the representation and querying of sets of possible worlds. *Theoretical Computer Science*, vol. 78, pp. 159-187, 1991.
- [2] Bosc P., Prade H., An introduction to fuzzy set and possibility theory-based approaches to the treatment of uncertainty and imprecision in data base management systems. In: *Uncertainty Management in Information Systems – From Needs to Solutions*, (Motro A. and Smets P. Eds.), Kluwer Academic Publishers, pp. 285-324, 1997.
- [3] Bosc P., Duval L., Pivert O., An initial approach to the evaluation of possibilistic queries addressed to possibilistic databases. *Fuzzy Sets and Systems*, vol. 140, pp. 151-166, 2003.
- [4] Bosc P., Pivert O., Towards an algebraic query language for possibilistic relations. *Proc. of the 12th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'2003)*, St. Louis, Missouri, USA, pp. 671-676, 2003.
- [5] Imielinski T., Lipski W., Incomplete information in relational databases. *Journal of the ACM*, vol. 31, pp. 115-143, 1984.
- [6] Zadeh L.A., Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, vol. 1, pp. 3-28, 1978.

Conditional Deduction Under Uncertainty

Audun Jøsang¹, Simon Pope¹, and Milan Daniel²

¹ DSTC* UQ Qld 4072, Australia

{ajosang, simon.pope}@dstc.edu.au

² Institute of Computer Science, AS CR** Prague, Czech Republic

milan.daniel@cs.cas.cz

Abstract. Conditional deduction in binary logic basically consists of deriving new statements from an existing set of statements and conditional rules. Modus Ponens, which is the classical example of a conditional deduction rule, expresses a conditional relationship between an antecedent and a consequent. A generalisation of Modus Ponens to probabilities in the form of probabilistic conditional inference is also well known. This paper describes a method for conditional deduction with beliefs which is a generalisation of probabilistic conditional inference and Modus Ponens. Meaningful conditional deduction requires a degree of relevance between the antecedent and the consequent, and this relevance can be explicitly expressed and measured with our method. Our belief representation has the advantage that it is possible to represent partial ignorance regarding the truth of statements, and is therefore suitable to model typical real life situations. Conditional deduction with beliefs thereby allows partial ignorance to be included in the analysis and deduction of statements and hypothesis.

1 Introduction

A conditional is for example a statement like “*If it rains, I will carry an umbrella*”, or “*If we continue releasing more CO₂ into the atmosphere, we will get global warming*”, which are of the form “IF x THEN y ” where x marks the antecedent and y the consequent. An equivalent way of expressing conditionals is through the concept of implication, so that “*If it rains, I will carry an umbrella*” is equivalent to “*The fact that it rains implies that I carry an umbrella*”. The statement “*It rains*” is here the antecedent, whereas “*I carry an umbrella*” is the consequent. The conditional is the statement that relates the antecedent and the consequent in a conditional fashion.

Consequents and antecedents are simple statements that in case of binary logic can be evaluated to TRUE or FALSE, in case of probability calculus be given a probability, or in case of belief calculus [7] be assigned belief values.

Conditionals are complex statements that can be assigned binary truth, probability and belief values in the same way as for simple statements. The binary logic interpre-

* The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government’s CRC Programme (Department of Education, Science, and Training).

** Partial support by the COST action 274 TARSKI acknowledged.

tation of conditional deduction is the Modus Ponens connector, meaning that a TRUE antecedent and a TRUE conditional necessarily will produce a TRUE consequent by deduction. Modus Ponens says nothing about the case when either the antecedent or the conditional, or both are false.

When assigning probabilities or beliefs to a conditional, the deduction process produces a probability or belief value that can be assigned to the consequent. In this case, the deduction can give a meaningful result even when the antecedent and conditional are not TRUE in a binary logic sense. The details of how this is done are described in Sec.3.

Because conditionals are not always true or relevant, it is common to hear utterings like: “*I don’t usually carry an umbrella, even when it rains*” which is contradicting the truth of the first conditional expressed above, or like: “*If we stop releasing more CO₂ into the atmosphere we will still have global warming*” which says that the second conditional expressed above is irrelevant. This can be nicely expressed with conditional beliefs, as described in Sec.3.

A conditional inference operator for beliefs that in special circumstances produced too high levels of uncertainty in the consequent belief, was presented in [8]. In the present paper we describe a new operator called *conditional deduction* that produces consequent beliefs with appropriate levels of uncertainty.

The advantage of the belief representation is that it can be used to model situations where the truth or probability values of the antecedent, the consequent and the conditionals are uncertain. Notice that probability and binary logic representations are special cases of, and therefore compatible with, our belief representation.

Sec.2 details our representation of uncertain beliefs. Sec.3 describes the conditional deduction operator, and Sec.4 describes an example of how the conditional deduction operator can be applied. Sec.5 provides a brief discussion on the theory of conditionals in standard logic and probability calculus. Sec.6 summarises the contribution of this paper.

2 Representing Uncertain Beliefs

This paper uses the bipolar belief representation called *opinion* [7], characterised by the use of separate variables pertaining to a given statement, and that bear some relationship to each other. In general, bipolarity in reasoning refers to the existence of positive and negative information to support an argument or the truth of a statement [1, 4].

In simplified terms, an opinion contains a variable representing the degree of belief that a statement is true, and a variable representing the degree of disbelief that the statement is true (i.e. the belief that the statement is false). The belief and disbelief values do not necessarily add up to 1, and the remaining belief mass is attributed to uncertainty. This representation can also be mapped to beta PDFs (probability density functions) [7, 9], which allows logic operators to be applied to beta PDFs. Subjective logic is a reasoning framework that uses the opinion representation and a set of logical connectors.

The bipolar belief representation in subjective logic is based on classic belief theory [13], where the *frame of discernment* Θ defines an exhaustive set of mutually exclusive atomic states. The power set 2^Θ is the set of all subsets of Θ .

A belief mass assignment¹ (called BMA hereafter) is a function m_Θ mapping 2^Θ to $[0, 1]$ (the real numbers between 0 and 1, inclusive) such that $\sum_{x \in 2^\Theta} m_\Theta(x) = 1$.

The BMA distributes a total belief mass of 1 amongst the subsets of Θ such that the belief mass for each subset is positive or zero. Each subset $x \subseteq \Theta$ such that $m_\Theta(x) > 0$ is called a focal element of m_Θ . In the case of total ignorance, $m_\Theta(\Theta) = 1$ and $m_\Theta(x) = 0$ for any proper subset x of Θ , and we speak about m_Θ being a *vacuous belief function*. If all the focal elements are atoms (i.e. one-element subsets of Θ) then we speak about *Bayesian belief functions*. A *dogmatic belief function* is defined by Smets[14] as a belief function for which $m_\Theta(\Theta) = 0$. Let us note that, trivially, every Bayesian belief function is dogmatic.

We are interested in expressing bipolar beliefs with respect to binary frames of discernment. In case Θ is larger than binary, this requires coarsening the original frame of discernment Θ to a binary frame of discernment. Let $x \in 2^\Theta$ be the element of interest for the coarsening and let $\bar{x}$ be the complement of x in Θ , then we can construct the binary frame of discernment $X = \{x, \bar{x}\}$. The coarsened belief mass assignment on 2^X can consist of maximum three belief masses, namely $m_X(x)$, $m_X(\bar{x})$ and $m_X(X)$, which we will denote by b_x , d_x and u_x because they represent *belief*, *disbelief* and *uncertainty* relative to x respectively. The *base rate*² of x can be determined by the relative size of the state x in the state space Θ , as defined by $a_x = \frac{|x|}{|\Theta|}$, or it can be determined on a subjective basis when no specific state space size information is known.

Coarsened belief masses can be computed e.g. with simple, normal or Dirichlet coarsening as defined in [8, 11, 9].

All the coarsenings have the property that b_x , d_x , u_x and a_x fall in the closed interval $[0, 1]$, and that

$$b_x + d_x + u_x = 1. \quad (1)$$

The expected probability of x is determined by: $E(\omega_x) = E(x) = b_x + a_x u_x$.

The ordered quadruple $\omega_x = (b_x, d_x, u_x, a_x)$, called the opinion about x , represents a bipolar belief function about x because it expresses positive belief in the form of b_x and negative belief in the form of d_x that are related by Eq.(1).

Although the coarsened frame of discernment X is binary, an opinion about $x \subset X$ carries information about the state space size of the original frame of discernment Θ through the base rate parameter a_x .

The base rate determined the probability expectation value when $u_x = 1$. In the absence of uncertainty, i.e. when $u_x = 0$, the base rate has no influence on the probability expectation value.

The opinion space can be mapped into the interior of an equal-sided triangle, where, for an opinion $\omega_x = (b_x, d_x, u_x, a_x)$, any two of the three parameters b_x , d_x and u_x determine the position of the point in the triangle representing the opinion.

Fig.1 illustrates an example where the opinion about a proposition x from a binary frame of discernment has the value $\omega_x = (0.7, 0.1, 0.2, 0.5)$.

¹ Called *basic probability assignment* in [13].

² Called relative atomicity in [7, 8].

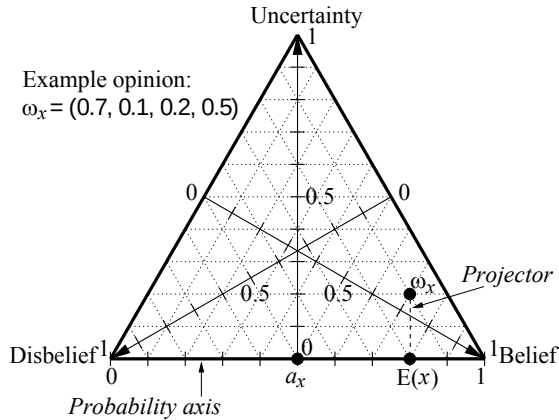


Fig. 1. Opinion triangle with example opinion

The top vertex of the triangle represents uncertainty, the bottom left vertex represents disbelief, and the bottom right vertex represents belief. The base line between the disbelief and belief vertices is the *probability axis*. The value of the base rate is indicated as a point on the probability axis.

Opinions on the probability axis have zero uncertainty and are equivalent to traditional probabilities. The distance from the probability axis to the opinion point can be interpreted as the uncertainty about the probability expectation value $E(x)$.

The *projector* is defined as the line going through the opinion point parallel to the line that joins the uncertainty vertex and the base rate point. The point at which the projector meets the probability axis determines the probability expectation value of the opinion, *i.e.* it coincides with the point corresponding to expectation value $E(x) = b_x + a_x u_x$.

Various visualisations of bipolar beliefs in the form of opinions are possible to facilitate human interpretation. For this, see <http://security.dstc.edu.au/spectrum/beliefvisual/>.

The next section describes a method for conditional deduction that takes bipolar beliefs in the form of opinions as input.

3 Conditional Deduction

A limitation of conditional propositions like ‘IF x THEN y ’ is that when the antecedent is false it is impossible to assert the truth value of the consequent.

What is needed is a complementary conditional that covers the case when the antecedent is false. One that is suitable in general is the conditional ‘IF NOT x THEN y ’. With this conditional it is now possible to determine the truth value of the consequent y in case the antecedent x is false.

Each conditional now provides a part of the picture and can therefore be called sub-conditionals. Together these sub-conditionals form a *complete conditional expres-*

sion that provides a complete description of the connection between the antecedent and the consequent. Complete conditional expressions have a two-dimensional truth value because they consist of two sub-conditionals that both have their own truth value.

We adopt the notation $y|x$ to express the sub-conditional ‘IF x THEN y ’, (this in accordance with Stalnaker’s [15] assumption that the probability of the proposition x implies y is equal to the probability of y given x) and $y|\bar{x}$ to express the sub-conditional ‘IF NOT x THEN y ’ and assume that it is meaningful to assign opinions (including probabilities) to these sub-conditionals. We also assume that the belief in the truth of the antecedent x and the consequent y can be expressed as opinions. The conditional inference with probabilities, which can be found in many text books, is described below.

Definition 1 (Probabilistic Conditional Inference). *Let x and y be two statements with arbitrary dependence, and let $\bar{x} = \text{NOT } x$. Let x , $\bar{x}$ and y be related through the conditional statements $y|x$ and $y|\bar{x}$, where x and $\bar{x}$ are antecedents and y is the consequent. Let $p(x)$, $p(y|x)$ and $p(y|\bar{x})$ be probability assessments of x , $y|x$ and $y|\bar{x}$ respectively. The probability $p(y||x)$ defined by:*

$$p(y||x) = p(x)p(y|x) + p(\bar{x})p(y|\bar{x}) = p(x)p(y|x) + (1 - p(x))p(y|\bar{x}). \quad (2)$$

is then the conditional probability of y as a function of the probabilities of the antecedent and the two sub-conditionals.

The purpose of the notation $y||x$ is to indicate that the truth or probability of the statement y is determined by the antecedent together with both the positive and the negative conditionals. The notation $y||x$ is this only meaningful in a probabilistic sense, i.e. so that $p(y||x)$ denotes the consequent probability. Below, this notation will also be used for beliefs, where $\omega_{y||x}$ denotes the consequent belief.

It can easily be seen that this definition of probabilistic deduction is a generalisation of Modus Ponens. Let for example x be TRUE (i.e. $p(x) = 1$) and $x \rightarrow y$ be TRUE (i.e. $p(y|x) = 1$), then it can be deduced that y is TRUE (i.e. $p(y||x) = 1$). In the case $p(x) = 1$, only the positive conditional counts, and in case $p(x) = 0$, only the negative conditional counts. In all other cases, both the positive and the negative conditionals are needed to determine the probability of y .

Conditional deduction with bipolar beliefs will be defined next. It is a generalisation of probabilistic conditional inference with probabilities. The definition is different from that of the conditional inference operator defined in [8], and the difference is explained in Sec.4.

Definition 2 (Conditional Deduction with Bipolar Beliefs). *Let $\Theta_X = \{x, \bar{x}\}$ and $\Theta_Y = \{y, \bar{y}\}$ be two frames of discernment with arbitrary mutual dependence. Let $\omega_x = (b_x, d_x, u_x, a_x)$, $\omega_{y|x} = (b_{y|x}, d_{y|x}, u_{y|x}, a_{y|x})$ and $\omega_{y|\bar{x}} = (b_{y|\bar{x}}, d_{y|\bar{x}}, u_{y|\bar{x}}, a_{y|\bar{x}})$ be an agent’s respective opinions about x being true, about y being true given that x is true and about y being true given that x is false. Let $\omega_{y||x} = (b_{y||x}, d_{y||x}, u_{y||x}, a_{y||x})$ be the opinion about y such that:*

$$\omega_{y||x} \text{ is defined by: } \begin{cases} b_{y||x} = b_y^I - a_y K \\ d_{y||x} = d_y^I - (1 - a_y) K \\ u_{y||x} = u_y^I + K \\ a_{y||x} = a_y \end{cases}$$

$$\text{where } \begin{cases} b_y^I = b_x b_{y|x} + d_x b_{y|\bar{x}} + u_x (b_{y|x} a_x + b_{y|\bar{x}} (1 - a_x)) \\ d_y^I = b_x d_{y|x} + d_x d_{y|\bar{x}} + u_x (d_{y|x} a_x + d_{y|\bar{x}} (1 - a_x)) \\ u_y^I = b_x u_{y|x} + d_x u_{y|\bar{x}} + u_x (u_{y|x} a_x + u_{y|\bar{x}} (1 - a_x)) \end{cases}$$

and K can be determined according to the following selection criteria:

Case I: $((b_{y|x} > b_{y|\bar{x}}) \wedge (d_{y|x} > d_{y|\bar{x}})) \vee ((b_{y|x} \leq b_{y|\bar{x}}) \wedge (d_{y|x} \leq d_{y|\bar{x}}))$
 $\implies K = 0.$

Case II.A.1: $((b_{y|x} > b_{y|\bar{x}}) \wedge (d_{y|x} \leq d_{y|\bar{x}}))$
 $\wedge (E(\omega_{y|\text{vac}(x)}) \leq (b_{y|\bar{x}} + a_y(1 - b_{y|\bar{x}} - d_{y|x})))$
 $\wedge (E(\omega_x) \leq a_x)$
 $\implies K = \frac{a_x u_x (b_y^I - b_{y|\bar{x}})}{(b_x + a_x u_x) a_y}.$

Case II.A.2: $((b_{y|x} > b_{y|\bar{x}}) \wedge (d_{y|x} \leq d_{y|\bar{x}}))$
 $\wedge (E(\omega_{y|\text{vac}(x)}) \leq (b_{y|\bar{x}} + a_y(1 - b_{y|\bar{x}} - d_{y|x})))$
 $\wedge (E(\omega_x) > a_x)$
 $\implies K = \frac{a_x u_x (d_y^I - d_{y|x})(b_{y|x} - b_{y|\bar{x}})}{(d_x + (1 - a_x) u_x) a_y (d_{y|\bar{x}} - d_{y|x})}.$

Case II.B.1: $((b_{y|x} > b_{y|\bar{x}}) \wedge (d_{y|x} \leq d_{y|\bar{x}}))$
 $\wedge (E(\omega_{y|\text{vac}(x)}) > (b_{y|\bar{x}} + a_y(1 - b_{y|\bar{x}} - d_{y|x})))$
 $\wedge (E(\omega_x) \leq a_x)$
 $\implies K = \frac{(1 - a_x) u_x (b_y^I - b_{y|\bar{x}})(d_{y|\bar{x}} - d_{y|x})}{(b_x + a_x u_x)(1 - a_y)(b_{y|x} - b_{y|\bar{x}})}.$

Case II.B.2: $((b_{y|x} > b_{y|\bar{x}}) \wedge (d_{y|x} \leq d_{y|\bar{x}}))$
 $\wedge (E(\omega_{y|\text{vac}(x)}) > (b_{y|\bar{x}} + a_y(1 - b_{y|\bar{x}} - d_{y|x})))$
 $\wedge (E(\omega_x) > a_x)$
 $\implies K = \frac{(1 - a_x) u_x (d_y^I - d_{y|x})}{(d_x + (1 - a_x) u_x)(1 - a_y)}.$

Case III.A.1: $(b_{y|x} \leq b_{y|\bar{x}}) \wedge (d_{y|x} > d_{y|\bar{x}})$
 $\wedge (E(\omega_{y|\text{vac}(x)}) \leq b_{y|x} + a_y(1 - b_{y|x} - d_{y|\bar{x}}))$
 $\wedge (E(\omega_x) \leq a_x)$
 $\implies K = \frac{(1 - a_x) u_x (d_y^I - d_{y|\bar{x}})(b_{y|\bar{x}} - b_{y|x})}{(b_x + a_x u_x) a_y (d_{y|x} - d_{y|\bar{x}})}.$

Case III.A.2: $(b_{y|x} \leq b_{y|\bar{x}}) \wedge (d_{y|x} > d_{y|\bar{x}})$
 $\wedge (E(\omega_{y|\text{vac}(x)}) \leq b_{y|x} + a_y(1 - b_{y|x} - d_{y|\bar{x}}))$
 $\wedge (E(\omega_x) > a_x)$
 $\implies K = \frac{(1 - a_x) u_x (b_y^I - b_{y|x})}{(d_x + (1 - a_x) u_x) a_y}.$

$$\begin{aligned}
\textbf{Case III.B.1: } & (b_{y|x} \leq b_{y|\bar{x}}) \wedge (d_{y|x} > d_{y|\bar{x}}) \\
& \wedge (E(\omega_{y|\text{vac}(x)}) > b_{y|x} + a_y(1 - b_{y|x} - d_{y|\bar{x}})) \\
& \wedge (E(\omega_x) \leq a_x) \\
\implies & K = \frac{a_x u_x (d_y^I - d_{y|\bar{x}})}{(b_x + a_x u_x)(1 - a_y)}.
\end{aligned}$$

$$\begin{aligned}
\textbf{Case III.B.2: } & (b_{y|x} \leq b_{y|\bar{x}}) \wedge (d_{y|x} > d_{y|\bar{x}}) \\
& \wedge (E(\omega_{y|\text{vac}(x)}) > b_{y|x} + a_y(1 - b_{y|x} - d_{y|\bar{x}})) \\
& \wedge (E(\omega_x) > a_x) \\
\implies & K = \frac{a_x u_x (b_y^I - b_{y|x})(d_{y|x} - d_{y|\bar{x}})}{(d_x + (1 - a_x)u_x)(1 - a_y)(b_{y|\bar{x}} - b_{y|x})}.
\end{aligned}$$

$$\begin{aligned}
\text{where } E(\omega_{y|\text{vac}(x)}) &= b_{y|x}a_x + b_{y|\bar{x}}(1 - a_x) + a_y(u_{y|x}a_x + u_{y|\bar{x}}(1 - a_x)) \\
\text{and } E(\omega_x) &= b_x + a_x u_x.
\end{aligned}$$

Then $\omega_{y||x}$ is called the conditionally deduced opinion of ω_x by $\omega_{y|x}$ and $\omega_{y|\bar{x}}$. The opinion $\omega_{y||x}$ expresses the belief in y being true as a function of the beliefs in x and the two sub-conditionals $y|x$ and $y|\bar{x}$. The conditional deduction operator is a ternary operator, and by using the function symbol ‘ $\odot$ ’ to designate this operator, we define $\omega_{y||x} = \omega_x \odot (\omega_{y|x}, \omega_{y|\bar{x}})$.

3.1 Justification

The expressions for conditional inference is relatively complex, and the best justification can be found in its geometrical interpretation.

The image space of the consequent opinion is a subtriangle where the two sub-conditionals $\omega_{y|x}$ and $\omega_{y|\bar{x}}$ form the two bottom vertices. The third vertex of the subtriangle is the consequent opinion resulting from a vacuous antecedent. This particular consequent opinion, denoted by $\omega_{y|\text{vac}(x)}$, is determined by the base rates of x and y as well as the horizontal distance between the sub-conditionals. The antecedent opinion then determines the actual position of the consequent within that subtriangle.

For example, when the antecedent is believed to be TRUE, i.e. $\omega_x = (1, 0, 0, a_x)$, the consequent opinion is $\omega_{y||x} = \omega_{y|x}$, when the antecedent is believed to be FALSE, i.e. $\omega_x = (0, 1, 0, a_x)$, the consequent opinion is $\omega_{y||x} = \omega_{y|\bar{x}}$, and when the antecedent opinion is vacuous, i.e. $\omega_x = (0, 0, 1, a_x)$, the consequent opinion is $\omega_{y||x} = \omega_{y|\text{vac}(x)}$. For all other opinion values of the antecedent, the consequent opinion is determined by linear mapping from a point in the antecedent triangle to a point in the consequent subtriangle according to Def.2.

It can be noticed that when $\omega_{y|x} = \omega_{y|\bar{x}}$, the consequent subtriangle is reduced to a point, so that it is necessary that $\omega_{y||x} = \omega_{y|x} = \omega_{y|\bar{x}} = \omega_{y|\text{vac}(x)}$ in this case. This means that there is no relevance relationship between antecedent and consequent, as will be explained in Sec.5.

Fig.2 illustrates an example of a consequent image defined by the subtriangle with vertices $\omega_{y|x} = (0.90, 0.02, 0.08, 0.50)$, $\omega_{y|\bar{x}} = (0.40, 0.52, 0.08, 0.50)$ and $\omega_{y|\text{vac}(x)} = (0.40, 0.02, 0.58, 0.50)$.

Let for example the opinion about the antecedent be $\omega_x = (0.00, 0.38, 0.62, 0.50)$. The opinion of the consequent $\omega_{y||x} = (0.40, 0.21, 0.39, 0.50)$ can then be obtained by mapping the position of the antecedent ω_x in the main triangle onto a position that

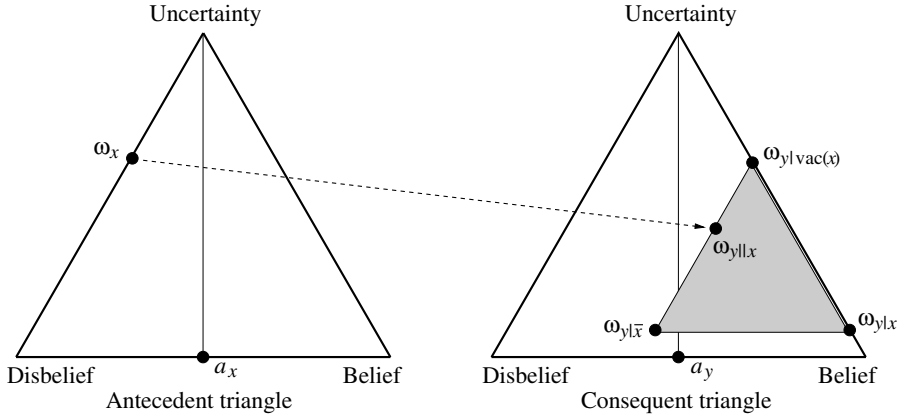


Fig. 2. Mapping from antecedent triangle to consequent subtriangle

relatively seen has the same belief, disbelief and uncertainty components in the subtriangle (shaded area).

In the general case, the consequent image subtriangle is not equal sided as in this example. By setting base rates of x and y different from 0.5, and by defining subconditionals with different uncertainty, the consequent image subtriangle will be skewed, and it is even possible that the uncertainty of $\omega_{y|vac(x)}$ is less than that of $\omega_{x|y}$ or $\omega_{x|\bar{y}}$.

4 Example

Let us divide the weather into 3 the exclusive types “sunny”, “overcast” and “rainy”, and assume that we are interested in knowing whether I carry umbrella when it rains. To define the conditionals, we need the beliefs in the statement y : “I carry an umbrella” in case the antecedent x : “It rains” is TRUE, as well in case it is FALSE. Let the opinion values of the antecedent and the two sub-conditionals, as well as their rough fuzzy verbal descriptions be defined as:

$$\omega_{y|x} = (0.72, 0.18, 0.10, 0.50) : \text{quite likely but somewhat uncertain,}$$

$$\omega_{y|\bar{x}} = (0.13, 0.57, 0.30, 0.50) : \text{quite unlikely but rather uncertain,} \quad (3)$$

$$\omega_x = (0.70, 0.00, 0.30, 0.33) : \text{quite likely but rather uncertain.}$$

The opinion about the consequent $y||x$ can be deduced with the conditional deduction operator expressed by $\omega_{y||x} = \omega_x \odot (\omega_{y|x}, \omega_{y|\bar{x}})$. Case II.A.2 of Def.2 is invoked in this case. This produces:

$$\omega_{y||x} = (0.54, 0.20, 0.26, 0.50) : \text{somewhat likely but rather uncertain.} \quad (4)$$

This example is visualised in Fig.3, where the dots represent the opinion values. The dot in the left triangle represents the opinion about the antecedent x . The middle triangle

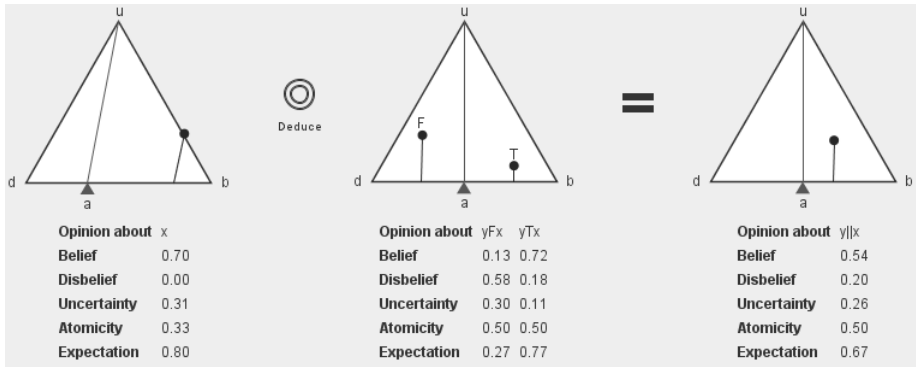


Fig. 3. Conditional deduction example

shows the conditionals, where the dot labelled “T” (TRUE) represents the opinion of $y|x$, and the dot labelled “F” (FALSE) represents the opinion of $y|\bar{x}$. The dot in the right hand triangle represents the opinion about the consequent $y||x$.

The consequent opinion value produced by the conditional deduction operator in this example contains slightly less uncertainty than the conditional inference operator defined in [8] would have produced. The simple conditional inference operator would typically produce too high uncertainty in case of state spaces different from $\frac{1}{2}$. More specifically, $\omega_{y|vac(x)}$ would not necessarily be a vertex in the consequent subtriangle in case of the simple conditional inference operator, whereas this is always the case for the deduction operator defined here. In the example of Fig.3, the antecedent state space was deliberately set to $\frac{1}{3}$ to illustrate that $\omega_{y|vac(x)}$ is the third vertex in the subtriangle. The conditional deduction operator defined here behaves well with any state space size, and it can be mentioned that $\omega_{y|vac(x)} = (0.13, 0.25, 0.62, 0.50)$, which is determined by Case II.A of Def.2 in this example.

The influence that the base rate has on the result increases as a function of the uncertainty. In the extreme case of a dogmatic antecedent opinion ($u_x = 0$), the base rate a_x has no influence on the result, and in the case of a vacuous antecedent opinion ($u_x = 1$), the consequent belief is fully conditioned by the base rate.

An online interactive demonstration of the conditional deduction operator can be accessed at <http://security.dstc.edu.au/spectrum/trustengine/>. Fig.3 is a screen shot taken from that demonstrator.

5 Discussion

The idea of having a conditional connection between the antecedent and the consequent can be traced back to Ramsey [12] who articulated what has become known as Ramsey’s Test: *To decide whether you believe a conditional, provisionally or hypothetically add the antecedent to your stock of beliefs, and consider whether to believe the consequent.* By introducing Ramsey’s test there has been a switch from truth and truth-functions to

belief and whether to believe which can also be expressed in terms of probability and conditional probability. This idea was articulated by Stalnaker [15] and expressed by the so-called Stalnaker's Hypothesis as: $p(\text{IF } x \text{ THEN } y) = p(y|x)$.

However, Lewis [10] argues that conditionals do not have truth-values and that they do not express propositions. In mathematical terms this means that given any propositions x and y , there is no proposition z for which $p(z) = p(y|x)$, so the conditional probability can not be the same as the probability of conditionals. Without going into detail we believe in Stalnaker's Hypothesis, and would argue against Lewis by simply saying that it is meaningful to assign a probability to a sub-conditional statement like " $y|x$ ", which is defined in case x is true, and undefined in case x is false.

A meaningful conditional deduction requires that the antecedent is relevant to the consequent, or in other words that the consequent depends on the antecedent, as explicitly expressed in relevance logics [5]. Conditionals that are based on the dependence between consequent and antecedent are considered to be universally valid (and not truth functional), and are called *logical conditionals* [3]. Deduction with logical conditionals reflect human intuitive conditional reasoning, and do not lead to any of the paradoxes of material implication.

Material implication, defined as $(x \rightarrow y) = (\bar{x} \vee y)$, is counterintuitive and riddled with paradoxes. Material implication, which is purely truth functional, ignores any relevance connection between antecedent x and the consequent y , and attempts to determine the truth value of the conditional as a function of the truth values of the antecedent and consequent alone. Material implication does not lend itself to any meaningful interpretation, and should never have been introduced into the theory of logic in the first place.

We will now show that it is possible to express the relevance between the antecedent and the consequent as a function of the conditionals.

For probabilistic conditional deduction, the relevance denoted as $R(x, y)$ can be defined as:

$$R(x, y) = |p(y|x) - p(y|\bar{x})|. \quad (5)$$

It can be seen that $R(x, y) \in [0, 1]$, where $R(x, y) = 0$ expresses total irrelevance/independence, and $R(x, y) = 1$ expresses total relevance/dependence between x and y . For belief conditionals, the same type of relevance can be defined as:

$$R(x, y) = |E(\omega_{y|x}) - E(\omega_{y|\bar{x}})|. \quad (6)$$

For belief conditionals, a second order *uncertainty relevance*, denoted as $R_u(x, y)$, can be defined:

$$R_u(x, y) = |u_{y|x} - u_{y|\bar{x}}|. \quad (7)$$

In case $R(x, y) = 0$, there can thus still exist a relevance which can make conditional deduction meaningful regarding the certainty in the consequent belief.

In the example of Fig.3, the relevance $R(x, y)$ is visualised as the horizontal distance between the probability expectations of the conditionals (i.e. where the projectors

intersect with the base line) in the middle triangle. The uncertainty relevance $R_u(x, y)$ is visualised as the vertical distance between the two dots representing the conditionals in the middle triangle.

Our approach to conditional deduction can be compared to that of conditional event algebras[6] where the set of events e.g. x, y in the probability space is augmented to include so-called class conditional events denoted by $y|x$. The primary objective in doing this is to define the conditional events in such a way that $p((y|x)) = p(y|x)$, that is so that the probability of the conditional event $y|x$ agrees with the conditional probability of y given x . There are a number of established conditional event algebras, each with their own advantages and disadvantages. In particular, one approach[2] used to construct them has been to employ a ternary truth system with values true, false and undefined, which corresponds well with the belief, disbelief and uncertainty components of bipolar beliefs.

Modus Ponens and probabilistic conditional inference are sub-cases of conditional deduction with bipolar beliefs. It can easily be seen that Def.2 collapses to Def.1 when the argument opinions are all dogmatic, i.e. when the opinions contain zero uncertainty. It can further be seen that Def.1 collapses to Modus Ponens when the arguments can only take probability values 0 or 1. It can also be seen that the probability expectation value of the deduced opinions of Def.2 is equal to the deduced probabilities of Def.1 when the input values are the probability expectation values of the original opinion arguments. This is formally expressed below:

$$E(\omega_{y||x}) = E(\omega_x)E(\omega_{y|x}) + E(\omega_{\bar{x}})E(\omega_{y|\bar{x}}) . \quad (8)$$

By using the mapping between opinions and beta PDFs described in [7, 9], it is also possible to perform conditional deduction when antecedents and conditionals are expressed in the form of beta PDFs. It would be impossible to do conditional deduction with beta PDFs algebraically, although numerical methods are probably possible. This conditional deduction operator defined here is therefore an approximation to an ideal case. The advantages are simple expressions and fast computation.

6 Conclusion

The subjective logic operator for conditional deduction with beliefs described here represents a generalisation of the binary logic Modus Ponens rule and of probabilistic conditional inference. The advantage of our approach is that it is possible to perform conditional deduction under uncertainty and see the effect it has on the result. When considering that subjective logic opinions can be interpreted as probability density functions, this operator allows conditional deduction to be performed on conditionals and antecedents represented in the form of probability density functions. Purely analytical conditional inference with probability density functions would normally be too complex to be practical. Our approach can be seen as a good approximation in this regard, and provides a bridge between belief theory on the one hand, and binary logic and probability theory on the other.

References

1. L. Amgoud, C. Cayrol, and M.C. Lagasque-Schieux. On the bipolarity in argumentation frameworks. In *Proceedings of NMR Workshop*, Whistler, Canada, June 2004.
2. P.G. Calabrese. Reasoning with Uncertainty using Conditional Logic and Probability. In Bila M. Ayyub, editor, *Proceedings of the First International Symposium on Uncertainty Modelling and Analysis*, pages 682–8. IEEE Computer Society Press, 1990.
3. M.R. Diaz. *Topics in the Logic of Relevance*. Philosophia Verlag, München, 1981.
4. D. Dubois, S. Kaci, and H. Prade. Bipolarity in reasoning and decision - An introduction. The case of the possibility theory framework. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty (IPMU2004)*. Springer, Perugia, July 2004.
5. J.K. Dunn and G. Restall. Relevance Logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic, 2nd Edition*, volume 6, pages 1–128. Kluwer, 2002.
6. I.R. Goodman, H.T. Nguyen, and R. Mahler. *The Mathematics of Data Fusion*, volume 37 of *Theory and Decision Library, Series B, Mathematical and Statistical Methods*. Kluwer Press, Amsterdam, 1997.
7. A. Jøsang. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001.
8. A. Jøsang and T. Grandison. Conditional Inference in Subjective Logic. In Xuezhong Wang, editor, *Proceedings of the 6th International Conference on Information Fusion*, 2003.
9. A. Jøsang, D. McAnally, and S. Pope. Interpreting Bipolar Beliefs as Probability Density Functions. (Submitted to) *Fuzzy Sets and Systems*, 00(0):0–0, 2005. Working paper at: <http://security.dstc.edu.au/staff/ajosang/papers.html>.
10. David Lewis. Probabilities of Conditionals and Conditional Probabilities. *The Philosophical Review*, 85(3), 1976.
11. D. McAnally and A. Jøsang. Addition and Subtraction of Beliefs. In *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2004)*, Perugia, July 2004.
12. Frank Ramsey. *The foundations of mathematics, and other logical essays*. London, edited by R.B. Braithwaite, Paul, Trench and Trubner, 1931. Reprinted 1950, Humanities Press, New York.
13. G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
14. Ph. Smets. Belief Functions. In Ph. Smets et al., editors, *Non-Standard Logics for Automated Reasoning*, pages 253–286. Academic Press, 1988.
15. R. Stalnaker. Probability and conditionals. In W.L. Harper, R. Stalnaker, and G. Pearce, editors, *The University of Western Ontario Series in Philosophy of Science*, pages 107–128. D.Riedel Publishing Company, Dordrecht, Holland, 1981.

Heterogeneous Spatial Reasoning

Haibin Sun and Wenhui Li

Key Laboratory of Symbol Computation and Knowledge
Engineering of the Ministry of Education,
College of Computer Science and Technology,
Jilin University, Changchun 130012, China
`Offer_sun@hotmail.com`

Abstract. In this article, we investigate the problem of checking consistency in a hybrid formalism, which combines two essential formalisms in qualitative spatial reasoning: topological formalism and cardinal direction formalism. Instead of using conventional composition tables, we investigate the interactions between topological and cardinal directional relations with the aid of rules that are used efficiently in many research fields such as content-based image retrieval. These rules are shown to be sound, i.e. the deductions are logically correct. Based on these rules, an improved constraint propagation algorithm is introduced to enforce the path consistency. The results of computational complexity of checking consistency for constraint satisfaction problems based on various subsets of this hybrid formalism are presented at the end of this article.

1 Introduction

Combining and integrating different kinds of knowledge is an emerging and challenging issue in Qualitative Spatial Reasoning (QSR), content-based image retrieval and computer vision, etc. Gerevini and Renz [1] has dealt with the combination of topological knowledge and metric size knowledge in QSR, and Isli et al. [2] has combined the cardinal direction knowledge and the relative orientation knowledge.

To combine topological and directional relations, Sharma [3] represented topological and cardinal relations as interval relations along two axes, e.g., horizontal and vertical axes. Based on Allen's composition table [4] for temporal interval relations, Sharma identifies all of the composition tables combining topological and directional relations. But his model approximated regions with Minimal Boundary Rectangles (MBRs), and if a more precise model (e.g., in this paper) is used, his composition tables will not be correct. We base our work on the same topological model as Sharma's, and a different directional model from his, which is more general and thereby, is more practical.

In this paper, we detail various interaction rules between two formalisms and we are also devoted to investigating the computational problems in the formalism combining topological and cardinal directional relations.

In the next section, we give the background for this paper. The interaction rules are introduced in section 3, which are used to implement our new path

consistency algorithm in section 5 after some definitions and terminologies are prepared in section 4. In section 6, the computational complexity of consistency checking is analyzed, which is followed by the conclusion section.

2 Background

We first introduce the two formalisms of topological and cardinal directional relations, respectively. The region considered in this paper is a point-set homeomorphic to a unit disk in Euclidean space $\mathbb{R}^2$.

2.1 Topology Formalism

Topology is perhaps the most fundamental aspect of space. Topological relations are invariant under topological transformations, such as translation, scaling, and rotation. Examples are terms like DC , EC , PO , EQ , TPP , $NTPP$, $TPPi$, $NTPPi$, with the meaning of DisConnected, Extensionally Connected, Partial Overlap, Equal, Tangential Proper Part, Non-Tangential Proper Part, and their converses (see Fig.1). Exactly one of these relations holds between any two spatial regions. In this paper, we will focus on RCC8 formalism.

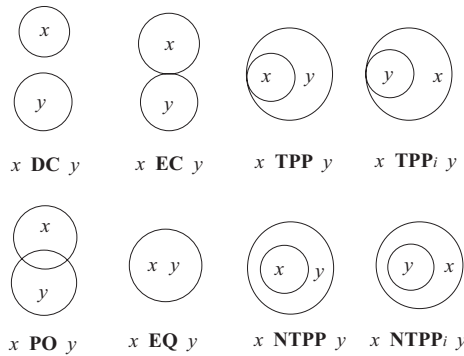


Fig. 1. Two-dimensional examples for the eight basic relations of RCC8

2.2 Cardinal Direction Formalism

Goyal and Egenhofer [8] introduced a direction-relation model for extended spatial objects that considers the influence of the objects' shapes. It uses the projection-based direction partitions and an extrinsic reference system, and considers the exact representation of the target object with respect to the reference frame. The reference frame with a polygon as reference object has nine direction tiles: north (N_A), northeast (NE_A), east (E_A), southeast

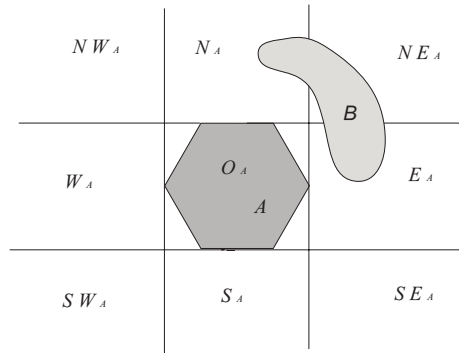


Fig. 2. Capturing the cardinal direction relation between two polygons, A and B, through the projection-based partitions around A as the reference object

(SE_A), south (S_A), southwest (SW_A), west (W_A), northwest (NW_A), and same (O_A , i.e., the minimum bounding rectangle) (see Fig.2). The cardinal direction from the reference object to a target is described by recording those tiles into which at least one part of the target object falls. We call the relations where the target object occupies one tile of the reference object $\{N, NE, E, SE, S, SW, W, NW, O\}$ relations, and others $\{N, NE, E, SE, S, SW, W, NW, O\}$ relations. We denote this formalism by CDF(Cardinal Direction Formalism) for brevity. It should be noted that Sharma [3] did not consider the kind of $\{N, NE, E, SE, S, SW, W, NW, O\}$ relation and the intermediate relations, i.e., $\{N, NE, E, SE, S, SW, W, NW, O\}$ and $\{N, NE, E, SE, S, SW, W, NW, O\}$.

3 Interaction Rules Between RCC8 and CDF

The internal operations, including converse and composition, on RCC8 can be found in [10]. The internal operations on CDF have been investigated in [9] and [11]. In order to integrate these two formalisms, we must investigate interaction rules between them. These rules are very useful to improve the spatial reasoning and can be the complement of the present composition tables. The spatial reasoning based on rules is more efficient and extended easily in the future as Sistla et al. [5] indicated.

The notation and representation of these rules are similar to [5], i.e. each rule will be written as $r :: r_1, r_2, \dots, r_k$, where r is called the head of the rule, which is deduced by the list $r_1, r_2, \dots, r_k$ called the body of the rule.

To facilitate the representation of the interaction rules, we denote a basic cardinal direction (i.e., single-tile or multi-tile relation) relation by a set $\{N, NE, E, SE, S, SW, W, NW, O\}$, which includes at most nine elements, i.e. the nine single-tile cardinal direction relations. For example, a relation $\{N, NE, E, SE, S, SW, W, NW, O\}$ (multi-tile relation) can be denoted by $\{N, NE, E, SE, S, SW, W, NW, O\}$. The general cardinal direction relation (i.e., a basic cardinal direction relation or the disjunction of basic cardinal direction relations) can be regarded as a superset $\{N, NE, E, SE, S, SW, W, NW, O\}$, whose element is the kind of set

. So we have the relation: $SB \in GB$. The universal relation is the set $BIN = \{O, N, NE, E, SE, S, SW, W, NW\}$, and the universe, i.e. the set of all possible cardinal relations, is denoted by GB .

Let A be a region. The x -axis (respectively y -axis) of the projection of region A on the x -axis (respectively y -axis) is denoted by $inf_x(A)$ (respectively $inf_y(A)$). The x -axis (respectively y -axis) of the projection of region A on the x -axis (respectively y -axis) is denoted by $sup_x(A)$ (respectively $sup_y(A)$). The $MBB(A)$ of a region A , denoted by $MBB(A)$, is the box formed by the straight lines $x = inf_x(A)$, $x = sup_x(A)$, $y = inf_y(A)$ and $y = sup_y(A)$. Based on these symbols, Skiadopoulos and Koubarakis [9] formally defined the cardinal directional relations.

Now, we present a system of rules for deducing new spatial relations from existing ones.

3.1 Rules for Deducing CDF Relations from RCC8 Relations ($RCC8 \rightarrow CDF$)

Assume that there exists some RCC8 relation between two regions A and B and we want to know the potential cardinal direction relations between them, we show the deduction rules in three cases and give their proofs if necessary.

Case 1. From the RCC8 relation $A \text{ } B$, we can not specify the CDF relation between them, i.e.,

$$A \cup B :: A \text{ } B, \quad (1)$$

This rule is obvious, because the $\text{ } B$ relation is the least restricted relation between two regions.

Case 2. Let x denote any relation symbol in $\{EC, PO, TPPi, NTPPi\}$. We have the following rule for each x . Because this rule is difficult to represent, we adopt first-order logic and the notations for CDF.

$$\forall SB \in GB(A, B), O \in SB :: A \text{ } B \quad (2)$$

According to definitions for $EC, PO, TPPi$ and $NTPPi$ [7], A and B must have a common part. From $B \subseteq (B)$, it follows that A and (B) must have a common part (i.e., $A \cap (B) \neq \emptyset$). According to the definitions for relation O and multi-tile relation [9], region A must have a part which satisfies the relation O with respect to B . $\square$

Case 3. Let x denote any of the relation symbols in $\{ \text{ } \}$. We have the following rule for each such x .

$$A \text{ } B :: A \text{ } B \quad (3)$$

From the relation $A \times B$, we have $A \subseteq B$. Hence $A \subseteq (B)$. According to the definition for CDF relation O [9], we conclude that the relation $A \ O \ B$ holds. $\square$

3.2 Rules for Deducing RCC8 Relations from CDF Relations (CDF $\rightarrow$ RCC8)

In this section, we will investigate the rules deducing RCC8 relation between any two regions A and B from the CDF relation between them in three cases.

Case 1. Let y denote any relation symbol in $\{DC, EC, PO, TPP, NTPP, EQ, TPPi\}$ (i.e., $\overline{NTPPi}$). We have the following rule.

$$A \ y \ B :: A \ O \ B \quad (4)$$

From the relation $A \ O \ B$ and the definition in [9], we have $A \subseteq (B)$. we can construct a scenario where $A \subseteq (B)$ and $A \ y \ B$ are simultaneously satisfied. We now prove the relation $A \ y \ B$ is impossible if $A \ O \ B$ holds. According to definition for y in [7], it is clear that there must be a part belonging to A which is outside of (B) . Hence the CDF relation between A and B must be a multi-tile one according to definition for multi-tile relation in [9]. So there is a contradiction. $\square$

Case 2. Let x denote a cardinal direction relation which is a multi-tile relation at least including O and another single-tile relation, for example $\{ \text{North} \}$. Let y denote the relation set $\{ \text{North}, \text{South}, \text{East}, \text{West}, \text{North-East}, \text{North-West}, \text{South-East}, \text{South-West} \}$, which means y can be anyone of these relations. We have the rule below.

$$A \ y \ B :: A \ x \ B \quad (5)$$

From the relation x , we know there must be a part of A in (B) , and another outside it. So any of the RCC8 relations $\{ \text{North}, \text{South}, \text{East}, \text{West}, \text{North-East}, \text{North-West}, \text{South-East}, \text{South-West} \}$ is impossible, because, if so, A will be contained in (B) . $\square$

Case 3. Let x denote any of the cardinal direction relations which do not contain O . Another rule can be described as follows.

$$A \ DC \ B :: A \ x \ B \quad (6)$$

This rule is obvious. Because x does not contain relation O , we have $A \cap (B) = \emptyset$. Hence $A \cap B = \emptyset$, it follows $A \ DC \ B$ according to definition for RCC8 relation DC [7]. $\square$

3.3 Rules for Deducing Relations from the Composition of RCC8 and CDF Relations ($RCC8 \circ CDF$)

We will discuss these rules in three cases.

Case 1. Let x denote any of the relation symbols in $\{TPP, NTPP\}$, y any CDF relation and z the induced CDF relation. The rule is described as follows.

$$A z C :: A x B, B y C, \quad (7)$$

Proof. From $A \{ \dots \} B$, We know $A \subseteq B$. Hence, if B satisfies a single-tile CDF relation with respect to C , A must also satisfy it. Then it follows that $A y C$ holds. We now consider the situation where y is a multi-tile CDF relation. According to definition for multi-tile relations [9], B can be regarded as consisting of several subregions which satisfy single-tile relations in y with respect to C , respectively. So region A can be one of, or consist several of these subregions. It follows that the relation z can be any subset of y . $\square$

Case 2. This rule is similar to the above except that x is anyone of the relation symbols in $\{ \dots \}$. So we have the relation $A \supseteq B$. It follows that the rule can be described as follows.

$$A z C :: A x B, B y C, \quad (8)$$

Proof. From $A \{ \dots \} B$, We know $A \supseteq B$. Hence, if B satisfies a single-tile CDF relation with respect to C , A must also satisfy it. Then it follows that $A y C$ holds. We now consider the situation where y is a multi-tile CDF relation. According to definition for multi-tile relations [9], B can be regarded as consisting of several subregions which satisfy single-tile relations in y with respect to C , respectively. So region A can be one of, or consist several of these subregions. It follows that the relation z can be any subset of y . $\square$

Case 3. This rule is obvious, so we present it directly.

$$A y C :: A EQ B, B y C \quad (9)$$

The rules for deducing RCC8 relations from the composition of RCC8 and CDF relations can be derived by combining the above rules (7)-(9) and rules (4)-(6).

3.4 Rules for Deducing Relations from the Composition of CDF and RCC8 Relations ($CDF \circ RCC8$)

The rules are presented in three cases as follows

Case 1. Let x denote any single-tile CDF relation and y denote the deduced CDF relation. The rule is described as follows.

$$A y C :: A x B, C \{TPP, NTPP\} B, \quad (10)$$

Proof. From $A x B$, We know $A \subseteq B$. Hence, if B satisfies a single-tile CDF relation with respect to C , A must also satisfy it. Then it follows that $A y C$ holds. We now consider the situation where y is a multi-tile CDF relation. According to definition for multi-tile relations [9], B can be regarded as consisting of several subregions which satisfy single-tile relations in y with respect to C , respectively. So region A can be one of, or consist several of these subregions. It follows that the relation z can be any subset of y . $\square$

Definition 2.

Let $\mathcal{R}$ be a binary relation over $\mathbb{R}^2$. The n -th power of $\mathcal{R}$, denoted by $\mathcal{R}^n$, is defined as follows: $\mathcal{R}^n = \{(x, y) \mid \exists x_1, \dots, x_{n-1} \in \mathbb{R}^2, (x, x_1) \in \mathcal{R}, (x_1, x_2) \in \mathcal{R}, \dots, (x_{n-1}, y) \in \mathcal{R}\}$.

Grigni et al.[16] identified two notions of satisfiability in a BCSP: relational consistency and realizability. In this paper, we focus on the relational consistency in a RDF-BCSP, i.e. whether it is possible to assign regions to all the variables in a RDF-BCSP such that all the specified relations among these variables hold.

A binary constraint problem with n variables and universe U can be simply viewed as an n -by- n matrix M of binary relations over U : the relation M_{ij} (in row i , column j) is the constraint on $\langle x_i, x_j \rangle$.

Let M and N be n -by- n matrices of binary relations. We have definitions as follows:

Definition 3.

$$(M \circ N)_{ij} = (M_{i0} \circ N_{0j}) \cap (M_{i1} \circ N_{1j}) \cap \dots \cap (M_{in-1} \circ N_{n-1j}) = \bigcap_{k < n} M_{ik} \circ N_{kj}.$$

Let $M^2 = M \circ M$.

Definition 4. A binary constraint problem $\mathcal{P}$ is *path-consistent* if $M \leq M^2$.

$\mathcal{P}$ is path-consistent just in case $M_{ij} \subseteq M_{ik} \circ M_{kj}$. We must note that path consistency is the necessary, but not sufficient, condition for the consistency of a BCSP.

5 Path Consistency in RDF-BCSP

To enforce the path consistency in RDF-BCSP, we must consider the interactions between the RCC8 component and CDF component in RDF-BCSP in addition to the internal path consistency in RCC8-BCSP and CDF-BCSP, respectively.

We devise a constraint propagation procedure $\text{PC}()$ for enforcing path consistency in RDF-BCSP, which is adapted from the path consistency algorithm described in [4]. Our algorithm employs two queues RCC8-Queue and CDF-Queue, which are initialized to all pairs (x, y) of the RCC8-BCSP and CDF-BCSP variables, respectively, verifying $x \leq y$ (the variables are supposed to be ordered). The algorithm removes pairs of variables from the two queues in parallel or in turn. When a pair $\langle X, Y \rangle$ of variables of RCC8-BCSP (respectively

CDF-BCSP) is removed from RCC8-Queue (respectively CDF-Queue), firstly the RCC8 (respectively CDF) relation on $\langle X, Y \rangle$ is converted to the CDF (respectively RCC8) relation on $\langle X, Y \rangle$ according to the rules (1)-(3) (respectively (4)-(6)). If the resulting CDF (respectively RCC8) relation on $\langle X, Y \rangle$ is different from the original relation on $\langle X, Y \rangle$, the pair of variables will be entered to the CDF-Queue (respectively RCC8-Queue); Then this CDF (respectively RCC8) relation on the pair $\langle X, Y \rangle$ is used to update the CDF (respectively RCC8) relations on the neighboring pairs of variables (pairs sharing at least one variable) according to the prerequisites in the rules provided by section 3. If a pair is successfully updated, it is entered into RCC8-Queue (respectively CDF-Queue), if it is not already there, in order to be considered at a future stage for propagation. This propagation procedure is common with Allen's algorithm, what's different is that the RCC8 (respectively CDF) relation on every pair of variables will be used to refine the relevant relations according to these rules provide by section 3.

The algorithm loops until it terminates if the empty relation, indicating inconsistency, is detected, or if RCC8-Queue and CDF-Queue become empty, indicating that a fixed point has been reached and the input RDF-BCSP is made path consistent.

Theorem 1. *The number of variable pairs in RCC8-Queue and CDF-Queue is $O(n^3)$.*

Proof. The number of variable pairs is $O(n^2)$. A pair of variables may be placed in queue at most a constant number of times (8 for a pair of RCC8 variables, which is the total number of RCC8 atoms; and 218 for a pair of CDF variables, which is the total number of CDF basic cardinal direction relations. Every time a pair is removed from queue for propagation, the procedure performs $O(n)$ operations. $\square$

Example 1. Suppose we have RCC8 relations A (NTPP) B and B (DC) C, and cardinal direction relation B (NW) C. First we run the constraint propagation algorithm within the RCC8-BCSP, so based on the relations A (NTPP) B and B (DC) C, we derive A (DC) C from the composition operation on them. But we know there is only one cardinal direction relation B (NW) C within the CDF-BCSP. When we consider the interaction rules (R1) and (R3), we can deduce the following cardinal direction relations: A (B) B, A (U) C and B (U) C. Based on the interaction rule (R6), we can derive the RCC8 relations B (DC) C from B (NW) C. Given the composition rule (R7) and the relations A (NTPP) B and B (NW) C, a new cardinal direction relation A (NW) C can be derived. After these operations, we have the RCC8 relations A (NTPP) B, B (DC) C and A (DC) C, and the cardinal direction relations A (B) B, B (NW) C and A (NW) C. Obviously the RDF-BCSP combining RCC8-BCSP and CDF-BCSP is consistent and we also get new useful information easily. This is a simple instance, but this method can be generalized to handle complicated ones.

6 Complexity of Consistency Checking in RDF-BCSP

We use T to denote the set of general RCC8 relations, T_b the set of basic RCC8 relations including universal relation, C the set of general CDF relations and C_b the set of basic CDF relations in

Renz and Nebel [13] and Renz [14] identified three maximal tractable subsets, i.e. $\mathcal{C}_8$, $\mathcal{Q}_8$ and $\mathcal{H}_8$, of the relations in RCC8 that contains all basic relations and showed that path-consistency is sufficient for deciding consistency for BCSPs based on these subsets. Skiadopoulos and Koubarakis [15] has presented the first algorithm for checking the consistency of a set of cardinal direction constraints and proved that the consistency checking of a set of basic cardinal direction constraints can be performed in $O(n^5)$ time while the consistency checking of an unrestricted set of cardinal direction constraints is NP-complete.

Theorem 2. *Let b be a positive integer and C_b be the b -th cyclotomic polynomial. Then*

Checking the consistency of RCC8-BCSP based on T_b is polynomial [13], and checking the consistency of CDF-BCSP based on C_b has been shown to be $O(n^5)$ [15]. From the rules (4), (5) and (6), all possible CDF basic relations can only entail = , $\vee$, $\vee \dots \vee$, $\vee \dots \vee \dots \vee$, $\vee \dots \vee \dots \vee \dots \vee$, or $\text{}$ relations, which belong to the maximal tractable subset $\hat{\mathcal{H}}_8$ (see Appendix B of [13]) of RCC8. So checking the consistency of RDF-BCSP based on the union of T_b and C_b is polynomial. First we can run the improved constraint propagation algorithm to enforce the path consistency, and then the algorithm in [15] will be employed to check the consistency for CDF-BCSP component. Obviously the complexity is $O(n^5)$. $\square$

Because checking the consistency of RCC8-BCSP based on the set $\mathcal{C}$ is **NP**-Complete (see theorem 22 of [13]) and checking the consistency of CDF-BCSP based on the set C is **NP**-Complete (see theorem 3 of [15]), we have the following corollary:

Corollary 1. *Let $S = T \cup C_{b_1} \cup \dots \cup C_{b_k} \cup \dots \cup C_{b_n}$ be a set of strings in $\mathbf{NP}$. Then*

7 Conclusions

In this paper, we have combined two essential formalisms in qualitative spatial reasoning, i.e., RCC8 and cardinal direction formalism. The interaction rules have been given and they can be embedded into the propagation algorithm to enforce the consistency of BCSP based on the new hybrid formalism, and then the results for the complexity of checking consistency based on various subsets of this new formalism are given. The complexities for other combinations of formalisms in QSR should be investigated in the future, and the modeling and computational problems in Fuzzy QSR should be also interesting.

References

1. A. Gerevini and J. Renz. Combining Topological and Size Constraints for Spatial Reasoning. *Artificial Intelligence (AIJ)*, vol. 137(1-2): 1-42, 2002
2. A Isli, V Haarslev and R Moller. Combining cardinal direction relations and relative orientation relations in Qualitative Spatial Reasoning. Fachbereich Informatik, University Hamburg, Technical report FBI-HH-M-304/01, 2001
3. J. Sharma. Integrated spatial reasoning in geographic information systems: combining topology and direction. Ph.D. Thesis, Department of Spatial Information Science and Engineering, University of Maine, Orono, ME, 1996
4. James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, vol. 26(11): 832-843, November, 1983
5. A. Prasad Sistla, Clement T. Yu and R. Haddad. Reasoning About Spatial Relations in Picture Retrieval Systems. In: *20th International Conference on Very Large Data Bases*, pp. 570-581, Morgan Kaufmann, 1994
6. M. Egenhofer. A Formal Definition of Binary Topological Relations. In *Third International Conference on Foundations of Data Organization and Algorithms (FODO)*, Vol. 367, pp. 457-472, Paris, France: Lecture Notes in Computer Science, Springer-Verlag, 1989
7. Randell D, Cui Z and Cohn A. A spatial logic based on regions and connection. In: Nebel B, Rich C, Swartout W, (eds.) *Proc. of the Knowledge Representation and Reasoning*, pp. 165~176, San Mateo: Morgan Kaufmann, 1992
8. R. Goyal and M. Egenhofer. Cardinal Directions between Extended Spatial Objects. *IEEE Transactions on Knowledge and Data Engineering* (to be published), 2000
9. S. Skiadopoulos and M. Koubarakis. Composing cardinal direction relations. *Artificial Intelligence*, vol. 152(2): 143—171, 2004
10. D. A. Randell, A. G. Cohn and Z. Cui. Computing Transitivity Tables: A Challenge For Automated Theorem Provers. In *11th International Conference on Automated Deduction*, pp.786-790, Berlin: Springer Verlag, 1992
11. Serafino Cicerone and Paolino Di Felice. Cardinal directions between spatial objects: the pairwise-consistency problem. *Information Sciences*, vol. 164: 165-188, 2004
12. A. Prasad Sistla, Clement T. Yu, and R. Haddad. Reasoning About Spatial Relations in Picture Retrieval Systems. *International Journal on Very Large Databases (VLDB)*, vol. 3(4): 570-581, 1994
13. J. Renz and B. Nebel. On the Complexity of Qualitative Spatial Reasoning: A Maximal Tractable Fragment of the Region Connection Calculus. *Artificial Intelligence (AIJ)*, vol.108(1-2): 69-123, 1999
14. J. Renz. Maximal Tractable Fragments of the Region Connection Calculus: A Complete Analysis. In: *16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, pp. 448-455, Stockholm, Sweden, August, 1999
15. S. Skiadopoulos and M. Koubarakis. Qualitative spatial reasoning with cardinal directions. In: *7th International Conference on Principles and Practice of Constraint Programming (CP'02)*, Vol.2470, pp. 341-355, in Lecture Notes in Comput. Sci., Springer, Berlin, 2002
16. Michelangelo Grigni, Dimitris Papadias, and Christos Papadimitriou. Topological inference. In: *C. Mellish, editor, Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 1, pp. 901-906. Morgan Kaufmann, 1995.

A Notion of Comparative Probabilistic Entropy Based on the Possibilistic Specificity Ordering

Didier Dubois¹ and Eyke Hüllermeier²

¹ Institut de Recherche en Informatique de Toulouse, France
dubois@irit.fr

² Faculty of Computer Science, University of Magdeburg, Germany
eyke.huellermeier@iti.cs.uni-magdeburg.de

Abstract. In this paper, we reconsider the problem of deciding whether one probability distribution is more informative (in the sense of representing a less indeterminate situation) than another one. Instead of using well-established information measures such as the Shannon entropy, however, we take up the idea of comparing probability distributions in a qualitative way. More specifically, we focus on a natural partial ordering induced by what is called the “peakedness” of a distribution. Moreover, there is a close connection between this ordering between probability distributions and the standard specificity ordering on possibility distributions that can be constructed from them. The main result of the paper is a proof showing that possibilistic specificity is consistent with probabilistic entropy in the sense that the (total) ordering defined by the latter refines the (partial) ordering defined by the former.

1 Introduction

The principle of maximum entropy plays an important role in probability theory, especially in the case of incomplete probabilistic models (see e.g. Paris [14]). It is instrumental in selecting a probability distribution in agreement with the available constraints, preserving as much indeterminateness as possible and verifying as many independence assumptions as possible. More precisely, entropy faithfully accounts for existing dependencies and only assumes independence where no justification to the contrary can be found. There are axiomatic characterizations of the Shannon entropy function, and Paris [14] has strongly advocated the selection of the maximum entropy probability as being a reasonable default choice under basic principles.

In possibility theory, a similar kind of “least commitment” information principle exists (e.g. Dubois et al. [8]): When a set of constraints delimits a set of possibility distributions, the least committed choice is the minimally specific distribution. The underlying idea is to consider any situation as possible as long it is not explicitly ruled out by the constraints. This principle obviously suggests a natural ordering between possibility degrees.

There also exists a natural, qualitative information ordering between possibility distributions, called the *possibilistic specificity ordering*. This ordering is based on fuzzy set

inclusion: If a possibility distribution $\pi : X \rightarrow [0, 1]$ is pointwisely dominated by another distribution $\pi' : X \rightarrow [0, 1]$, i.e. $\pi(x) \leq \pi'(x)$ for all $x \in X$, the former is said to be more specific than the latter (and strictly more specific if $\pi(x) < \pi'(x)$ for at least one $x \in X$). The natural measure of non-specificity in agreement with this partial ordering is the sum of the possibility degrees.¹

There is a close connection between maximal entropy and minimal specificity principles, especially in the light of the Laplace indifference principle: In the possibilistic framework, the case of complete ignorance is adequately represented by the uniform distribution $\pi \equiv 1$ (all x are completely possible). Likewise, if a unique probability distribution must be picked, the aforementioned indifference principle suggests selecting the uniform distribution $p \equiv |X|^{-1}$. For these distributions, the Shannon entropy and the additive possibilistic measure of non-specificity coincide with the Hartley entropy of a set (Higashi and Klir [10]), that is the logarithm of the number of elements in the set. These authors use an additive index of possibilistic non-specificity that looks like Shannon entropy.

The temptation to compare specificity and entropy is great. For instance, Klir [11] has tried to equate numerical entropy and (additive) non-specificity indices for the purpose of transforming possibility distributions into probability distributions and conversely. This is debatable, however, because the entropy scale and the specificity scale are not commensurate. Maung [12] has tried to justify the principle of minimal specificity by adapting Paris' rationality axioms to the possibilistic setting.

Regarding the information-based comparison of distributions, there is an important difference between the probability and possibility settings. In the uncertainty literature, the comparison between probability distributions is always based on a type of entropy index without reference to an underlying, *a priori*, ordering, which is directly defined between probability distributions. There are actually several entropy indices and similar ones (such as the Gini index) but they never stem from a partial ordering that decides if a probability measure is more informative than another one.

Interestingly, it turns out that an old paper by Birnbaum [1] suggested such a qualitative comparison of probabilities on the real line in terms of what is called their *peakedness*, independently of the notion of entropy. It basically consists of checking the nestedness of confidence intervals of various confidence levels extracted from the probability distribution. Interestingly, the nestedness property of confidence intervals strongly suggests a similarity between the relative peakedness of probability distributions and the relative specificity of possibility distributions. On the other hand, the more peaked a probability distribution, the less indeterminate it is, and the lower its entropy should be.

The aim of this paper is to prove that these intuitions are mathematically valid, thereby bridging the gap between Birnbaum peakedness adapted to the finite setting, and Shannon entropy. We show that the former is a partial ordering

¹ Of course, here we assume the domain X to be finite or at least countable. Otherwise, the sum must be replaced by an integral.

on probabilities compatible with the latter index (and other related information measures). The connection uses possibility theory because checking the peakedness relation between two probability distributions comes down to comparing, in terms of specificity, possibility distributions whose cuts are the confidence intervals of the original probability distributions. These possibility distributions are in fact the most specific transforms from probability to possibility, already proposed by Dubois and Prade [3], and Delgado and Moral [2] in the eighties.

The paper thus establishes a new link between possibilistic and probabilistic traditions, and proposes a qualitative comparison test between probability distributions that may arguably be considered as the natural information ordering between probability functions, something that, to the best of our knowledge, is apparently missing in the uncertainty literature.

The next section introduces the basic notions of possibilistic specificity and probabilistic peakedness and discusses a particular type of probability-possibility transformation. Our main result, establishing the consistency between the possibilistic specificity ordering and the probabilistic entropy measure, is stated and proved in section 3. A discussion of related works follows in section 4. The paper concludes with a summary and an outlook in section 5.

2 Specificity, Peakedness and Probability-Possibility Transforms

Consider two probability distributions (probability vectors) $\alpha = (\alpha_1 \dots \alpha_n)$ and $\beta = (\beta_1 \dots \beta_n)$ defined over a finite domain X of cardinality n ; α_i resp. β_i denote the probability $\Pr(x_i)$ of the i -th element x_i . We denote by $a = O(\alpha)$ the probability vector obtained from the vector α by rearranging the probability degrees α_i in a non-increasing order. That is,

$$a = (a_1 \dots a_n) = (\alpha_{\sigma(1)} \dots \alpha_{\sigma(n)}),$$

where σ is a permutation of $\{1 \dots n\}$ such that $\alpha_{\sigma(i)} \geq \alpha_{\sigma(j)}$ for $i < j$. Likewise, we denote by $b = (b_1 \dots b_n) = O(\beta)$ the ordered probability vector associated with β . Since both a and b are still probability vectors, they do of course satisfy the characteristic properties $a, b \geq 0$, $\sum_{j=1}^n a_j = \sum_{j=1}^n b_j = 1$.

A possibility distribution π is a mapping from X to the unit interval such that $\pi(x) = 1$ for some $x \in X$. A possibility degree $\pi(x)$ expresses the absence of surprise about x being the actual state of the world, and can be viewed as an upper bound of a probability degree [6]. Let $\pi = T(a)$ be the possibility distribution derived from the (ordered) probability vector a according to the following probability-possibility transformation suggested by Dubois and Prade [3]:

$$\pi_i = \sum_{j=i}^n a_j, \quad i = 1 \dots n. \quad (1)$$

Obviously, $1 = \pi_1 \geq \dots \geq \pi_n$. We note that this possibility function is also a (de-)cumulative distribution function with respect to the ordering induced by

the probability values. Moreover, the possibility measure Π associated with π dominates the corresponding probability measure $\Pr$, that is,

$$\forall A \subseteq X : \Pi(A) \geq \Pr(A)$$

with $\Pi(A) = \max_{x_i \in A} \pi_i$ and $\Pr(A) = \sum_{x_i \in A} a_i$.

In the following definition, we recall a basic notion from possibility theory (e.g. Dubois et al. [8]) already mentioned in the introduction.

Definition 1. Let π and ρ be two possibility measures on $X = \{x_1, \dots, x_n\}$. π is more specific than ρ (written $\pi \geq \rho$) if and only if $\pi_i \leq \rho_i$ for all $i \in \{1, \dots, n\}$. If $\pi_i < \rho_i$ for at least one $i \in \{1, \dots, n\}$, then π is strictly more specific than ρ .

Clearly, the more specific π , the more informative it is. If $\pi(x_i) = 1$ for some i and $\pi(x_j) = 0$ for all $j \neq i$, then π is maximally specific (full knowledge); if $\pi(x_i) = 1$ for all i , then π is minimally specific (no information).

It turns out that $T(a)$ is a maximally specific element of the family of possibility measures that dominate the probability function $\Pr$ induced by the distribution a (see Dubois and Prade [3], and Delgado and Moral [2]). Moreover, if the ordering induced by a on X is linear (i.e. $a_i \neq a_j$ for all $i \neq j$) then $T(a)$ is the unique maximally specific dominating possibility distribution and respecting the ordering induced by the probability assignment. When there are elements of equal probability, the uniqueness of the maximally specific dominating possibility distribution can be recovered if the ordering induced by π on X is requested to be the same as the ordering induced by a (but then the equation defining $T(a)$ must be adjusted accordingly).

Probability-possibility transformations have been extended to the real line by Dubois et al. [7] (see also Dubois et al. [9]). Let p be a unimodal continuous probability density. It is first proved that the most narrow prediction interval I such that $\Pr(I) \geq \lambda$, where λ is a fixed confidence level, is of the form $I_\lambda = \{x \mid p(x) \geq \theta\}$ for some threshold θ . Then the most specific possibility transform (inducing the same ordering as p on the real line) is $\pi = T(p)$ such that

$$\forall x \in R : \pi(x) = \pi(y) = 1 - \Pr([x, y]),$$

where $[x, y] = I_{p(x)}$.

In 1948, Birnbaum dealt with what he called the peakedness of a probability distribution, referring to its peakedness. Considering that the fourth moment of a distribution is not an appropriate measure of peakedness he proposed a definition of the relative peakedness of distributions as follows:

Definition 2. Let Y and Z be two continuous random variables with probability density functions p_Y and p_Z respectively. Let y_1 and z_1 be two fixed points in the support of Y and Z respectively. Then the relative peakedness of Y with respect to Z is defined as

$$\Pr(|Y - y_1| \geq t) \leq \Pr(|Z - z_1| \geq t)$$

for all $t \geq 0$.

It is clear that the function

$$\pi_y(y_1 - t) = \pi_y(y_1 + t) = \Pr(|x - y_1| \geq t) = 1 - \Pr([y_1 - t, y_1 + t])$$

is a possibility distribution, and easy to show that for any choice of y_1 , its possibility measure dominates $\Pr$ (see Dubois et al. [9]). In this paper, we shall adapt this definition in two ways: First, the results on the probability-possibility transforms clearly indicate that for unimodal densities, choosing y_1 as the mode of the distribution is reasonable. Moreover, Birnbaum [1] considers intervals whose common midpoint is y_1 , yielding a symmetric possibility distribution even if the density is not symmetric by itself. Instead of intervals of the form $[y_1 - t, y_1 + t]$, we shall use intervals of the form $\{x | p(x) \geq \theta\}$, since they lead to a possibility distribution of the same shape as the probability density (and peakedness refers to the shape of this density anyway). This change enables peakedness to be defined for any referential set, not just the reals. Indeed, the set $\{x | p(x) \geq \theta\}$ makes sense in general, if measurability is ensured, while $[y_1 - t, y_1 + t]$ assumes the real line as an underlying domain. Here, we nevertheless restrict ourselves to the case of a finite referential set, because entropy indices are usually applied to such domains.

Now, for $\pi = T(a)$ it is clear that $\pi_i = 1 - \Pr(\{x | \Pr(\{x\}) \geq \theta\})$ if $a_{i-1} \geq \theta > a_i$, so that the above considerations motivate the following variant of the original peakedness relation due to Birnbaum.

Definition 3. Let $\pi = T(a)$ and $\rho = T(\beta)$ be two possibility distributions on a finite set X . We say that π is more peaked than ρ (written $\pi \geq \rho$) if and only if $\pi_i \leq \rho_i$ for all $i \in \{1, \dots, n\}$. If $\pi \geq \rho$ and $\pi \neq \rho$, we say that π is strictly more peaked than ρ (written $\pi > \rho$).

Subsequently, the peakedness relation is understood in the sense of this definition. It is clear that it compares probability distributions by means of the specificity relation applied to their optimal possibility transforms. The relation is obviously invariant under permutations of the involved probability vectors. Therefore, we restrict our attention to ordered probability or possibility vectors in the next section.

For the two probability distributions specified by the probability vectors

$$\alpha = (.05 .20 .25 .25 .20 .05), \\ \beta = (.30 .15 .05 .05 .15 .30)$$

(see Fig. 1 for a graphical illustration) we obtain

$$\pi = (1.0 .75 .50 .30 .10 .05), \\ \rho = (1.0 .70 .40 .25 .10 .05).$$

Since $\pi \geq \rho$ (and $\pi_2 > \rho_2$), α is (strictly) less peaked than β .

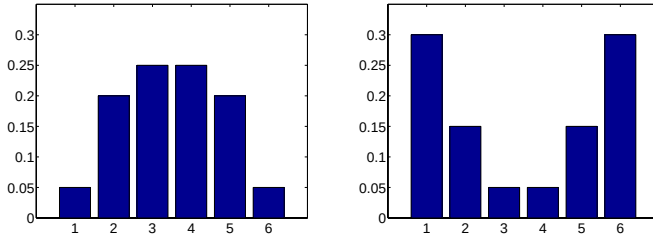


Fig. 1. The probability distribution on the left is (strictly) less peaked than the one on the right

3 From Peakedness to Entropy

The aim of this section is to prove that the peakedness relation, which is expressed in terms of possibilistic specificity, is consistent with the ordering of probability distributions induced by Shannon entropy.

Definition 4. The entropy of a probability vector $a = (a_1, \dots, a_n)$ is defined as

$$E(a) = - \sum_{j=1}^n a_j \cdot \log a_j. \quad (2)$$

The main result of this paper claims that the entropy ordering refines the peakedness relation.

Theorem 1. Let $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$ be two probability vectors. If $E(a) \geq E(b)$ and a is strictly less peaked than b , then $E(a) > E(b)$.

Below, we shall prove this theorem in the following way: We construct a sequence of probability vectors $a^0, a^1, \dots, a^m$ such that $a^0 = a$, $a^m = b$ and a^{k+1} is more peaked than a^k . Moreover, this sequence will satisfy $E(a^k) \geq E(a^{k+1})$ (resp. $E(a^k) > E(a^{k+1})$) for all $1 \leq k \leq m-1$.

Simple counterexamples can be constructed showing that an implication in the other direction, namely that $E(a) \geq E(b)$ implies a to be less peaked than b , does not hold. In fact, such an implication cannot be expected since the entropy measure induces a total ordering on the class of probability measures, whereas the peakedness relation defines only a partial ordering. In other words, the former ordering is a proper refinement of the latter one.

3.1 Auxiliary Result

Let a and b denote two (ordered) probability vectors such that a is strictly less peaked than b . Starting with $a^0 = a$, a distribution a^{k+1} will be obtained from

a distribution a^k by shifting a part of the probability mass a_j^k to a_i^k for appropriately defined indices $j > i$. More generally, a shifting operation $S(a, i, j, c)$ will transform an ordered vector $a = (a_1 \dots a_i \dots a_j \dots a_n)$ into the ordered vector

$$a^c = (a_1 \dots a_i + c \dots a_j - c \dots a_n).$$

Note that if $\pi = T(a)$ and $\pi^c = T(a^c)$ denote, respectively, the possibilistic transforms of a and a^c , then

$$\pi_k^c = \begin{cases} \pi_k & \text{if } k \leq i \\ \pi_k & \text{if } j < k \\ \pi_k - c & \text{if } i < k \leq j \end{cases} \quad (3)$$

Thus, $\pi^c \leq \pi$ does obviously hold true, and a^c is strictly more peaked than a in the case where $c > 0$.

To guarantee a shifting operation $S(a, i, j, c)$ to be valid in the scope of turning a into b , the choice of c must satisfy the following conditions:

- (i.) Proper ordering : $a_{i-1} \geq a_i + c$ and $a_j - c \geq a_{j+1}$
- (ii.) Limited increase of specificity: $\pi^c \geq \rho$

Recalling (3), the latter item means that

$$\pi_k^c = \sum_{i=k}^n a_i - c \geq \sum_{i=k}^n b_i = \rho_k$$

for all $i < k \leq j$. Define $d_k = a_k - b_k$. Since $\pi = T(a) \geq T(b) = \rho$ by assumption, we have $\sum_{m=k}^n d_m \geq 0$ for all $1 \leq k \leq n$. The condition $\pi^c \geq \rho$ can thus be written as

$$\forall i < k \leq j : c \leq \sum_{m=k}^n d_m.$$

To satisfy both (i.) and (ii.), we hence need

$$c \leq \min \left(\min_{i < k \leq j} \sum_{m=k}^n d_m, a_{i-1} - a_i, a_j - a_{j+1} \right). \quad (4)$$

Since $a \neq b$, there exists $j = \max\{k \mid a_k \neq b_k\}$. Of course, $a_j > b_j$ since $\pi \geq \rho$. By definition, we also have $d_j = a_j - b_j = \pi_j - \rho_j$. Since a and b are probability distributions, there must be some $i < j$ such that $b_i > a_i$. So, let

$$i = \max\{k \mid 1 < k \leq n, b_k > a_k \text{ and } a_{k-1} > a_k\} \quad (5)$$

if the set on the right-hand side is not empty (as will be assumed for the time being).

In order to simplify the upper bound on the number c , we first derive a lower bound on the quantity $\min_{i < k \leq j} \sum_{m=k}^n d_m$ that appears as the first argument on the right-hand side of (4).

Lemma 1. $\min_{i < k \leq j} \sum_{m=k}^n d_m \geq \min\{a_j - b_j, b_i - a_i\}$

Proof: Define $D(k) = \sum_{m=k}^n d_m$ and $D = \min_{i < k \leq j} D(k)$. We consider two cases:

(a) $D = D(j)$. In this case, the lemma obviously holds, since $d_m = a_m - b_m = 0$ for $m > j$ and hence $D(j) = a_j - b_j$.

(b) $D < D(j)$. In this case, there must be an index k_0 with $i < k_0 < j$ and such that $D(k_0) < D(k_0 + 1)$. We claim that

$$D(i+1) < D(i+2) < \dots < D(k_0). \quad (6)$$

In fact, since $D(k_0) < D(k_0 + 1)$ we have $a_{k_0} < b_{k_0}$. Thus, either $i = k_0 - 1$ (in which case (6) does trivially hold), or $a_{k_0-1} = a_{k_0}$ (since if $a_{k_0-1} > a_{k_0}$ and $a_{k_0} < b_{k_0}$, the index k_0 is a potential candidate for the choice of i). In the latter case, $a_{k_0-1} = a_{k_0} < b_{k_0} \leq b_{k_0-1}$ and therefore

$$D(k_0 - 1) = D(k_0) + (a_{k_0-1} - b_{k_0-1}) < D(k_0).$$

This argument can be repeated, showing that (6) does indeed hold. This in turn shows that $D = D(i+1)$. Moreover, we then have

$$D = D(i+1) = D(i) - (a_i - b_i) = D(i) + (b_i - a_i) \geq b_i - a_i$$

since $D(i) \geq 0$.

Overall, we get $D \geq a_j - b_j$ in case (a) and $D \geq b_i - a_i$ in case (b). Thus, the lemma does indeed hold. Q.E.D.

Now, if we let

$$c = \min(a_j - b_j, b_i - a_i, a_{i-1} - a_i) \quad (7)$$

then the above results and the fact that

$$a_j - a_{j+1} = (a_j - b_j) + (b_j - a_{j+1}) \geq (a_j - b_j) + (b_{j+1} - a_{j+1}) = (a_j - b_j)$$

guarantee that (4) is satisfied. Moreover, the constant c is strictly positive, since $a_{i-1} - a_i > 0$, $a_j - b_j > 0$, $b_i - a_i > 0$ by construction.

Let us now turn to the case where the right-hand side of (5) is empty.

Lemma 2. $\dots, \dots, a_{\dots}, \dots, b_{\dots}, \dots, \dots$
 $(\dots, \dots, b_1 > a_1$

Proof: Suppose that a is less peaked than b . There is some $k < j$ such that $b_k > a_k$. Since the right-hand side on (5) is empty, it holds that $b_u > a_u$ implies $a_u = a_{u-1}$ for all $u < j$. Moreover, since $b_k > a_k$, this implies in turn $b_{k-1} \geq b_k > a_{k-1}$. The fact that $b_1 > a_1$ follows immediately by repeating this argument. Q.E.D.

Regarding the choice of c in the case of an empty right-hand side in (5), the only difference concerns the condition $a_{i-1}^c \geq a_i^c$ which simply becomes unnecessary. Hence, one can define

$$c = \min(a_j - b_j, b_1 - a_1) \quad (8)$$

and apply the shifting operation $S(a, 1, j, c)$ in the same way as before.

3.2 Proof of the Main Result

Obviously, if the quantity c as defined in (7) (resp. (8)) is shifted from position j to position i (resp. position 1), then either $a_j^c = b_j$ or $a_i^c = b_i$ or $a_i^c = a_{i-1}$. In any case, at least one of the indices i or j will have a smaller value in the next iteration. Hence, the process of repeating the shifting operation, with i , j , and c as specified above, is well-defined, admissible and turns a into b in a finite number of steps.

Given the above results, Theorem 1 follows immediately from the next lemma (recall that in each step of our iterative procedure, the constant c shifted from index j to index i is strictly positive):

Lemma 3. . . . $E(a) = -\sum_{j=1}^n a_j \cdot \log(a_j)$. . . $E(a) > E(a^c)$. . . $c > 0$

Proof: It is easy to see that $E(a) > E(a^c)$ is equivalent to

$$(a_i + c) \log(a_i + c) - a_i \log(a_i) > a_j \log(a_j) - (a_j - c) \log(a_j - c).$$

Noting that $a_i > a_j$, this inequality can be secured by showing that the function $x \mapsto x \log(x)$ is strictly convex on $(0, 1)$. This is indeed the case, since the second derivative of this function is given by $x \mapsto 1/x$. Q.E.D.

Let us finally note that Theorem 1 can be generalized to informativeness measures other than the standard entropy. In fact, it is easily verified that the logarithm $\log(\cdot)$ in (2) can be replaced by any monotone increasing function $F(\cdot)$ the second derivative $F''(\cdot)$ of which exists on $(0, 1)$ and satisfies $F''(x)/F'(x) > -2/x$ for all $0 < x < 1$ (where $F'(\cdot)$ denotes the first derivative).

As an example, consider the case of the well-known Gini measure

$$G(a) = \sum_{j=1}^n (a_j)^2.$$

Since $G(\cdot)$ thus defined is an informativeness index rather than a measure of indeterminateness (such as entropy), we actually have to consider its negation $-G(a) = -\sum_{j=1}^n (a_j)^2 = -\sum_{j=1}^n a_j F(a_j)$ with $F : x \mapsto x$. Here, we have

$$a \text{ (strictly) less peaked than } b \Rightarrow -G(a)(>) \geq -G(b)$$

since $F'' \equiv 0$.

4 Related Work

Even though the proposed notion of relative informativeness, based on possibilistic specificity and Birnbaum peakedness, seems to be unknown in the uncertainty literature, there is a subfield of the social sciences where similar notions have apparently been developed for some twenty years or so:² The study of social welfare orderings.

² The authors are grateful to Jérôme Lang for pointing out this connection.

We refer to the book by Moulin [13]. In this framework, X is a set of agents, whose welfare under some life conditions is measured by a utility function over X . The problem is to compare the quality of utility vectors $(u_1 \dots u_n)$ from the standpoint of social welfare. Under an egalitarian program of redistribution from the rich to the poor, the so-called Pigou-Dalton principle of transfer states that transferring some utility from one agent to an other one so as to reduce inequalities of utility values improves the social welfare of the population.³

Formally, the transformation of a vector a into a vector a^c as in section 3.1 is known as a Pigou-Dalton transfer. The sequence of transformations we propose here is also used in this literature. Moreover, the role of entropy is played by so-called inequality indices. The counterpart to the possibility transform of a probability vector is called the Lorentz curve of the utility vector, and the counterpart of the peakedness ordering is called the Lorentz dominance relation.

It seems that counterparts to our main results already exist in this literature, and this point would be worth studying in more detail. One difference is that utility vectors do not sum to 1. But Lorentz dominance is precisely making sense for the comparison of utility vectors with equal sum. Note that it would not be the first time that possibility-probability transformations find counterparts in the social sciences. For instance, a transformation from a belief function to a probability measure (obtained by generalizing the Laplace indifference principle) introduced in [3] and called *Shapley value* by Smets [16] is known in social science as the Shapley value of cooperative games (see again Moulin [13]).

5 Conclusions and Perspectives

The contribution of this paper is mainly to lay bare a notion of relative information content that can decide if a probability distribution represents more or less uncertainty than another one (or whether the two distributions are not directly comparable). The test we offer appears to be natural in the sense that it exactly captures the notion of relative peakedness of distributions, thus meeting our intuition. The fact that Shannon entropy as well as the Gini index (and many other ones, potentially) refine the peakedness relation corroborates this intuition. It sheds light on the meaning of these indices, that were sometimes dogmatically proposed as natural ones, even if axioms or properties that justify the entropy index were proposed in order to its use for uncertain reasoning more transparent. The peakedness ordering offers a minimal robust foundation for probabilistic information indices. The surprise is that it comes down to comparing two possibility distributions in the sense of their relative specificity (using fuzzy set inclusion!). Finding an extension of these results to continuous probability distributions, using differential entropy for instance, is an obvious next task.

Our discussion also shows that there is a range of arbitrariness in the choice of these indices, namely in the case of two distributions that cannot be compared

³ This principle does not seem to be popular nowadays.

by the peakedness relation but are ranked in opposite orders by, say, the entropy and the Gini index. This point needs further study. We note, however, that the situation is the same with the specificity relation in possibility theory where several non-specificity indices have been proposed (Higashi and Klir [10], Dubois and Prade [4], Yager [17], Ramer [15]) that disagree with each other. The same difficulty can be observed in the case of belief functions (Dubois and Prade [5]). Besides, the close relationship between peakedness and Lorentz dominance also comforts the legitimacy of the proposed relative probabilistic informativeness notion.

In his book [14], Jeff Paris advocates the use of conditional probability statements as a natural means for expressing knowledge and the maximal entropy principle as a natural tool for selecting a reasonable default probabilistic model of this knowledge. The above results suggest that the maximal entropy principle can be replaced by a minimal peakedness principle in problems with incompletely specified probability distributions. Of course, the minimally peaked distribution in agreement with the constraints may fail to be unique, and the issue of choosing between them is an intriguing one. Anyway, the peakedness relation can be used in all reasoning problems where the information content of a distribution is relevant, for example in machine learning techniques à la decision tree induction where measures of that kind are used for selecting (hopefully) optimal attributes according to which the data is partitioned in a recursive manner. The notion of peakedness is easy to understand, but, compared to entropy and other numerical indices, quite weak and its efficiency in probabilistic reasoning and decision making is still unclear. These issues constitute interesting topics of future research.

Acknowledgements. The authors are grateful to Jürgen Beringer and Jérôme Lang for helpful comments.

References

1. Birnbaum Z. W. On random variables with comparable peakedness, *Annals of Mathematical Statistics*, 19, 1948, 76-81.
2. Delgado M. and Moral S. On the concept of possibility-probability consistency, *Fuzzy Sets and Systems*, 21, 1987 311-318.
3. Dubois D. and Prade H. On several representations of an uncertain body of evidence, in *Fuzzy Information and Decision Processes*, M.M. Gupta, and E. Sanchez, Eds., North-Holland, Amsterdam, 1982, pp. 167-181.
4. Dubois D. and Prade H. A note on measures of specificity for fuzzy sets, *Int. J. of General Systems*, 10, 1985, 279-283.
5. Dubois D. and Prade H.: The principle of minimum specificity as a basis for evidential reasoning, In: *Uncertainty in Knowledge-Based Systems* (B. Bouchon, R.R. Yager, eds.), Springer Verlag, 1987, 75-84.
6. Dubois D. and Prade H. When upper probabilities are possibility measures, *Fuzzy Sets and Systems*, 49, 1992 65-74.
7. Dubois D., Prade H. and Sandri S. On possibility/probability transformations. In: *Fuzzy Logic. State of the Art*, (R. Lowen, M. Roubens, eds.), Kluwer Acad. Publ., Dordrecht, 1993, 103-112.

8. Dubois D., Nguyen H. T., Prade H. Possibility theory, probability and fuzzy sets: misunderstandings, bridges and gaps. In: *Fundamentals of Fuzzy Sets*, (Dubois, D. Prade, H., Eds.), Kluwer, Boston, Mass., The Handbooks of Fuzzy Sets Series, 2000 343-438.
9. Dubois D., Foulloy L., Mauris G., Prade H. Possibility/probability transformations, triangular fuzzy sets, and probabilistic inequalities. *Reliable Computing* 10, 2004, 273-297.
10. Higashi and Klir G. Measures of uncertainty and information based on possibility distributions, *Int. J. General Systems*, 8, 1982, 43-58.
11. Klir G. A principle of uncertainty and information invariance, *Int. J. of General Systems*, 17, 1990, 249-275.
12. Maung I. Two characterizations of a minimum-information principle in possibilistic reasoning *Int. J. of Approximate Reasoning*, 12, 1995, 133-156.
13. H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, Cambridge, MA, 1988.
14. Paris J. *The Uncertain Reasoner's Companion*. Cambridge University Press, Cambridge, UK, 1994.
15. Ramer A. Possibilistic information metrics and distances: Characterizations of structure, *Int. J. of General Systems*, 18, 1990, 1-10.
16. Smets P. Constructing the pignistic probability function in a context of uncertainty, *Uncertainty in Artificial Intelligence* 5 (Henrion M. et al., Eds.), North-Holland, Amsterdam, 1990, 29-39.
17. Yager R.R. On the specificity of a possibility distribution, *Fuzzy Sets and Systems*, 50, 1992, 279-292.

Consonant Random Sets: Structure and Properties^{*}

Enrique Miranda

Rey Juan Carlos University, Department of Informatics,
Statistics and Telematics. C-Tulipán, s/n 28933 Móstoles, Spain
`enrique.miranda@urjc.es`

Abstract. In this paper, we investigate consonant random sets from the point of view of lattice theory. We introduce a new definition of consonancy and study its relationship with possibility measures as upper probabilities. This allows us to improve a number of results from the literature. Finally, we study the suitability of consonant random sets as models of the imprecise observation of random variables.

Keywords: Consonant random sets, lattice theory, possibility measures, upper probabilities, measurable selections.

1 Introduction

Random sets, or measurable multi-valued mappings, have gained a lot of attention in the past decades. They have been studied for instance within stochastic geometry ([16]), economy ([13]), or from the measure-theoretic point of view ([12]). Within random sets, those which are *consonant* constitute a subclass of particular interest, as the works in [4, 10, 17, 19] testify. In spite of all this work, there is not a unique definition of consonant random set; on the contrary, the term ‘consonancy’ has been used whenever there is some relationship of nestedness between the images of the multi-valued mapping. The different levels of this relationship, as well as other hypotheses that can be imposed on the random set, such as the initial and final spaces, or the topological characteristics of the images, have made of the term consonant random set a rather vague one.

In this paper, we try to get to the core of the notion of consonancy: we study this property not from the point of view of the order that we can consider in the images of the random set, but from the one we can induce on the elements of the initial space. We study the properties of this order within lattice theory, and use them to investigate a number of features of consonant random sets. Our main subject of interest is the relationship between consonant random sets and

^{*} The research in this paper has been partially supported by MEC-DGI, grant numbers MTM2004-01269 and TSI2004-06801-C04-01. The scientific responsibility rests with the author.

possibility measures. This relationship has been thoroughly studied in the literature, but the point of view we adopt in this paper allows us to easily characterise this property, generalising along the way some results from the literature. This is detailed in Sections 2 and 3. In Section 4 we investigate whether a consonant random set can be used to model the imprecise observation of a random variable, and which would be the best tool in that case. Finally, Section 5 contains some conclusions and open problems on the matter at hand.

2 Consonant Random Sets and Lattice Theory

Let us introduce some basic concepts from random set theory.

Definition 1. Let $(\Omega, \mathcal{A}, P)$ be a probability space, $(X, \mathcal{A}')$ a measurable space and let $\Gamma : \Omega \rightarrow \mathcal{P}(X)$ be a multi-valued mapping. Given $A \in \mathcal{A}'$, its **upper inverse** by Γ is $\Gamma^*(A) = \{\omega \in \Omega : \Gamma(\omega) \cap A \neq \emptyset\}$. Γ is said to be **strongly measurable** (or a **random set**) when $\Gamma^*(A)$ belongs to $\mathcal{A}$ for all $A \in \mathcal{A}'$.

There are other conditions of measurability that can be considered on multi-valued mappings (see a review in [14]). We have chosen the strong measurability ([22]), also called sometimes $\mathcal{B}$ -measurability, because it allows us to define the *upper probability* of the random set, which will be one of the main points of interest in this paper.

Definition 2. [5] Let $(\Omega, \mathcal{A}, P)$ be a probability space, $(X, \mathcal{A}')$ a measurable space and consider a random set $\Gamma : \Omega \rightarrow \mathcal{P}(X)$. Given $A \in \mathcal{A}'$, its **upper probability** is given by $P_\Gamma^*(A) = \frac{P(\Gamma^*(A))}{P(\Gamma^*(X))}$.

When there is no ambiguity about the random set we are working with, we shall denote $P^* = P_\Gamma^*$. The upper probability induced by a random set is ∞ -alternating and lower continuous ([22]).

Let us introduce next some notions of consonancy that can be considered on a random set. The idea underlying consonant random sets is the existence of some order in the set of images, normally by means of the inclusion operator. This order provides a common background to the images of the different elements of the initial space, so there is not contradiction between them (hence the term consonant). Although there are other conditions (see for instance [17, 19]), the ones we recall here are the strongest and the most interesting ones for the purposes of this paper.

Definition 3. A random set $\Gamma : \Omega \rightarrow \mathcal{P}(X)$ is said to be

- **antitone** if $(\Omega, \mathcal{A}, P) = ([0, 1], \beta_{[0, 1]}, \lambda_{[0, 1]})$, where $\beta_{[0, 1]}$ and $\lambda_{[0, 1]}$ denote respectively the Borel σ -field and the Lebesgue measure on $[0, 1]$, and $x \leq y \in [0, 1] \Rightarrow \Gamma(x) \supseteq \Gamma(y)$.
- **C1**, if for any $\omega_1, \omega_2 \in \Omega$, it is $\Gamma(\omega_1) \subseteq \Gamma(\omega_2)$ or $\Gamma(\omega_2) \subseteq \Gamma(\omega_1)$.
- **C2**, if the previous relation holds for any ω_1, ω_2 on $\Omega \setminus N$, where N is a null subset of Ω .

An *isotone* random set is one defined on $[0, 1]$ such that the natural order is the same as we have in the images, that is, such that $x \leq y \Rightarrow \Gamma(x) \subseteq \Gamma(y)$. The dual notion of antitone random set allows us to relate consonant random sets to the α -cuts of a fuzzy number [6] (but see also [21]). On the other hand, the more general condition C1 only requires the set of images to be totally ordered by the inclusion relationship, but it does not make any assumption on the initial space. The introduction of C2 random sets is due to the fact that the behaviour of a random set on a null subset of the initial space does not affect its upper probability (this will become clearer in Section 3). By Definition 3, it is clear that an antitone random set is in particular C1, and that a C1 random set is C2. Next, we are going to introduce an intermediate notion of consonancy that comes between C1 and antitone random sets. We shall denote this condition C0. It involves not only the existence of a total order on the initial space, but also some kind of ‘continuity’ in the way the images of the random set are nested.

Definition 4. A random set $\Gamma : \Omega \rightarrow \mathcal{P}(X)$ is said to be **C0** if it satisfies the following two hypotheses:

- For any $\omega_1, \omega_2 \in \Omega$, either $\Gamma(\omega_1) \subseteq \Gamma(\omega_2)$ or $\Gamma(\omega_2) \subseteq \Gamma(\omega_1)$.
- For any $A \subseteq \Omega$ there exists $B \subseteq A$ countable s.t. $\cap_{\omega \in A} \Gamma(\omega) = \cap_{\omega \in B} \Gamma(\omega)$.

The remainder of this paper is devoted to the study of the properties of C0 random sets, and the results we obtain will serve as a justification of their introduction. We shall see in particular that, even though a C0 random set is in particular C1, the converse is not true.

First, we are going to study the representation of consonant random sets in terms of lattices. We refer to [2, 9] for the definitions of the different concepts in lattice theory we shall use. Let Γ be a C1 random set, and let us define, for any $\omega \in \Omega$, the class $[\omega] := \{\omega' \in \Omega \mid \Gamma(\omega') = \Gamma(\omega)\}$, and denote $\Omega^* := \{[\omega] \mid \omega \in \Omega\}$. Let us define the relation $\preceq_\Gamma$ on Ω^* by

$$[\omega_1] \preceq_\Gamma [\omega_2] \Leftrightarrow \Gamma(\omega_1) \subseteq \Gamma(\omega_2)$$

Then, it is easy to see that $(\Omega^*, \preceq_\Gamma)$ is totally ordered (i.e., a *chain*). In fact, there is a one-to-one correspondence between chains and lattices induced by C1 random sets: given a chain $(\Omega, \preceq)$, it suffices to consider the multi-valued mapping $\Gamma : \Omega \rightarrow \mathcal{P}(\Omega)$ given by $\Gamma(\omega) = \{\omega' \in \Omega : \omega' \preceq \omega\}$ ¹. Then, the chain $(\Omega^*, \preceq_\Gamma)$ induced by Γ coincides with $(\Omega, \preceq)$. We deduce from this correspondence that the lattice induced by a C1 random set is not complete in general. Nevertheless, when it is complete, it is a *continuous* lattice.

Definition 5. [9] Given a lattice $(\Omega, \preceq)$, $\omega_1, \omega_2 \in \Omega$, we say that ω_1 is **way below** ω_2 and denote $\omega_1 \ll \omega_2$ when $\forall D \subseteq \Omega$ s.t. $\omega_2 \preceq \sup D, \exists d \in D$ s.t. $\omega_1 \preceq d$. $(\Omega, \preceq)$ is **continuous** if it is complete and $\omega = \sup\{\omega' \ll \omega\}$ for any $\omega \in \Omega$.

¹ We can easily make Γ a random set by considering $\mathcal{A} = \mathcal{A}' = \mathcal{P}(\Omega)$ and P a degenerate probability distribution on some $\omega_0 \in \Omega$.

Proposition 1. *If $(\Omega^*, \preceq_\Gamma)$ is a complete lattice, then it is continuous.*

Proof. Consider $\omega \in \Omega^*$. If $\omega \ll \omega$, the condition holds trivially. Otherwise, given $\omega' \prec_\Gamma \omega$, it follows from the definition that $\omega' \ll \omega$. We deduce that $\{\omega' \prec_\Gamma \omega\} = \{\omega' \ll \omega\}$, and then it is clear that $\omega = \sup\{\omega' \ll \omega\}$. ■

The random set Γ induces a multi-valued mapping $\Gamma' : \Omega^* \rightarrow \mathcal{P}(X)$ by $\Gamma'([\omega]) = \Gamma(\omega)$. If we consider on $\mathcal{P}(X)$ the partial order given by the inclusion, then Γ' is an homomorphism, and it is an isomorphism if we restrict the final space to $\text{Im}(\Gamma) = \text{Im}(\Gamma')$. Moreover, the class of the upper inverses of the elements of $\mathcal{A}'$ can be characterised in terms of filters of Ω^* . Let us consider $\mathcal{H} = \{[\omega] \in \Omega^* : \Gamma(\omega) = \cup_{[\omega'] \prec_\Gamma [\omega]} \Gamma(\omega')\}$ and let us define the sets of filters

$$\mathcal{H}_1 := \{[\omega'] \succeq_\Gamma [\omega] : [\omega] \in \mathcal{H}\}, \quad \mathcal{H}_2 := \{B \subseteq \Omega^* : B \text{ is a filter}, B \notin \mathcal{H}_1\}$$

Proposition 2. *For any $A \in \mathcal{A}'$, $\Gamma'^*(A) \in \mathcal{H}_2$. If moreover $\mathcal{A}' = \mathcal{P}(X)$, then $\{\Gamma'^*(A) : A \subseteq X\} = \mathcal{H}_2$.*

Proof. Consider $A \in \mathcal{A}'$. Given $[\omega] \in \Gamma'^*(A)$, $[\omega'] \succ_\Gamma [\omega]$, it is $\emptyset \neq \Gamma(\omega) \cap A \subseteq \Gamma(\omega') \cap A$, whence $[\omega'] \in \Gamma'^*(A)$. Since $(\Omega^*, \preceq_\Gamma)$ is totally ordered, we deduce that $\Gamma'^*(A)$ is a filter. Assume ex-absurdo that $\Gamma'^*(A) = \{[\omega] \succeq_\Gamma [\omega_0]\}$ for some $[\omega_0] \in \mathcal{H}$. Then, there exists $x \in \Gamma(\omega_0) \cap A = (\cup_{[\omega] \prec_\Gamma [\omega_0]} \Gamma(\omega)) \cap A$, whence there exists $[\omega_1] \prec_\Gamma [\omega_0]$ s.t. $x \in \Gamma(\omega_1)$. This means that $\{[\omega] \succeq_\Gamma [\omega_1]\} \subseteq \Gamma'^*(\{x\}) \subseteq \Gamma'^*(A) = \{[\omega] \succeq_\Gamma [\omega_0]\} \subsetneq \{[\omega] \succeq_\Gamma [\omega_1]\}$, a contradiction.

Assume now $\mathcal{A}' = \mathcal{P}(X)$, and consider $B \in \mathcal{H}_2$. Let $A = (\cup_{[\omega] \notin B} \Gamma(\omega))^c$. Given $[\omega] \notin B$, $\Gamma'([\omega]) \cap A = \Gamma(\omega) \cap (\cup_{[\omega'] \notin B} \Gamma(\omega'))^c = \emptyset$, whence $\Gamma'^*(A) \subseteq B$. Now, if there exists $[\omega_0] \in B \setminus \Gamma'^*(A)$,

$$\begin{aligned} A \subseteq \Gamma(\omega_0)^c \subseteq \cap_{[\omega] \notin B} \Gamma(\omega)^c = A &\Rightarrow \Gamma(\omega_0) = \cup_{[\omega] \notin B} \Gamma(\omega) \\ &\Rightarrow \Gamma(\omega_0) = \cup_{[\omega] \prec_\Gamma [\omega_0]} \Gamma(\omega) \Rightarrow [\omega_0] \in \mathcal{H}. \end{aligned}$$

Hence, $B = \{[\omega] \succeq_\Gamma [\omega_0]\} \in \mathcal{H}_1$, a contradiction. Consequently, $\Gamma'^*(A) = B$, and we deduce the desired equality. ■

This proposition will be useful in the next section.

3 Consonant Random Sets and Possibility Measures

One of the main features of consonant random sets is their connection with supremum-preserving set functions. These set functions have appeared in the literature under a number of different names (see [23, 26, 27]), although since Zadeh's paper [28] they have been mostly referred to with the term possibility measure. This designation is due to their usefulness for modelling linguistic uncertainty ([7, 28]).

Definition 6. Let $(X, \mathcal{A})$ be a measurable space. A monotone and normalised set function $\Pi : \mathcal{A} \rightarrow [0, 1]$ is called **maxitive** if for any $A_1, \dots, A_n \in \mathcal{A}$, $\Pi(\cup_{i=1}^n A_i) = \max_{i=1, \dots, n} \Pi(A_i)$, and it is said to be a **possibility measure** when for any family $(A_i)_{i \in I}$ of elements of $\mathcal{A}$ such that $\cup_{i \in I} A_i \in \mathcal{A}$, it is $\Pi(\cup_{i \in I} A_i) = \sup_{i \in I} \Pi(A_i)$.

When all the singletons belong to the σ -field $\mathcal{A}$, a possibility measure Π is characterised by its **possibility distribution** $\pi : X \rightarrow [0, 1]$, which is given by $\pi(x) = \Pi(\{x\})$. Then, it is $\Pi(A) = \sup_{x \in A} \pi(x)$ for any $A \in \mathcal{A}$.

The relationship between possibility measures and consonant random sets is rather intuitive if we restrict ourselves to finite spaces: a monotone and normalized set function on a finite space is a possibility measure if and only if its focal elements are **nested** ([24]); on the other hand, the focal elements of the upper probability of a random set are those subsets of the final space whose inverses have probability non-zero. Taking this into account, it is easy to prove the following:

Proposition 3. [8, 17] Let $(\Omega, \mathcal{A}, P)$ be a probability space, $(X, \mathcal{P}(X))$ a finite space and let $\Gamma : \Omega \rightarrow \mathcal{P}(X)$ be a random set. Then, P^* is a possibility measure if and only if Γ is C2.

There are many works in the literature devoted to the investigation of this equality in the infinite case; that is, whether the upper probability of a consonant random set is always a possibility measure and whether when the upper probability is a possibility measure the random set is necessarily consonant. In [17], we showed that a C1 random set does not induce in general a possibility measure, and that a random set inducing a possibility measure need not be C2. Nevertheless, there exist a number of sufficient conditions for these implications. We summarise the most important ones in the following theorem:

Theorem 1. 1. ([4]) If Γ is antitone, then P^* is a possibility measure.
 2. ([19]) If Γ is C2, P^* is maxitive.
 3. ([19]) If Γ is compact on a Polish space, or closed on a σ -compact metric space, then P^* is a possibility measure $\iff \Gamma$ is C2.

In this paper, we are going to study this problem for the new condition of consonancy we have introduced, and we are going to take advantage of the representation of consonant random sets in terms of lattices made in the previous section. Let us define $\mathcal{A}'_1 = \{A \in \mathcal{A}' : \Gamma^*(A) \neq \{[\omega'] \succeq_\Gamma [\omega]\} \forall [\omega] \in \Omega^*\}$. The following fairly straightforward result reduces the problem of studying if the upper probability is a possibility measure to arbitrary unions in $\mathcal{A}'_1$.

Proposition 4. P^* is a possibility measure if and only if for any $(A_i)_{i \in I}$ s.t. $\cup_{i \in I} A_i \in \mathcal{A}'_1$, $P^*(\cup_{i \in I} A_i) = \sup_{i \in I} P^*(A_i)$.

Proof. Consider $(A_i)_{i \in I}$ in $\mathcal{A}'$ s.t. $A := \cup_{i \in I} A_i$ belongs to $\mathcal{A}' \setminus \mathcal{A}'_1$. Then, there exists $[\omega_0]$ s.t. $\Gamma^*(A) = \{[\omega] \succeq_\Gamma [\omega_0]\}$. Now, $\Gamma(\omega_0) \cap A \neq \emptyset \Rightarrow \exists i_0 \in I$ s.t. $\Gamma(\omega_0) \cap A_{i_0} \neq \emptyset$, whence $\Gamma^*(A) = \Gamma^*(A_{i_0})$ and $P^*(A) = \sup_{i \in I} P^*(A_i)$. We conclude that P^* is a possibility measure if and only if it is supremum-preserving for arbitrary collections of elements of $\mathcal{A}'$ whose union belongs to $\mathcal{A}'_1$. ■

Taking this proposition into account, we can establish the following theorem:

Theorem 2. *Let $(\Omega, \mathcal{A}, P)$ be a probability space, $(X, \mathcal{P}(X))$ a measurable space and $\Gamma : \Omega \rightarrow \mathcal{P}(X)$ a C1 random set s.t. $(\Omega^*, \preceq_\Gamma)$ is a complete chain. Then, P^* is a possibility measure if and only if $\forall A \in \mathcal{A}'_1$ s.t. $\cap_{[\omega] \in \Gamma'^*(A)} \Gamma(\omega) \cap A = \emptyset$, there exists some countable $\{\omega_n\}_n \subseteq \Gamma'^*(A)$ s.t. $P^*(A) = P(\cup_n \{[\omega] \succeq_\Gamma [\omega_n]\})$.*

Proof. Since $(\Omega^*, \preceq_\Gamma)$ is complete, given $A \in \mathcal{A}'_1$, there exists some $[\omega_A] \in \Omega^*$ s.t. $\Gamma'^*(A) = \{[\omega] \succ_\Gamma [\omega_A]\}$. Now, if there is some $x \in \cap_{[\omega] \succ_\Gamma [\omega_A]} \Gamma(\omega) \cap A$, then trivially $\Gamma'^*({x}) = \Gamma'^*(A)$ and $P^*(A) = P^*({x})$. Hence, P^* will be a possibility measure if and only if $P^*(A) = \sup_{x \in A} P^*({x})$ when $A \in \mathcal{A}'_1$ and $\cap_{[\omega] \succ_\Gamma [\omega_A]} \Gamma(\omega) \cap A = \emptyset$.

($\Rightarrow$) If there exists $x \in A$ s.t. $P^*({x}) = \sup_{y \in A} P^*({y}) = P^*(A)$ then, since $(\Omega^*, \preceq_\Gamma)$ is complete and $\Gamma'^*({x})$ is a filter, there exists $[\omega_x] \succeq_\Gamma [\omega_A]$ s.t. $\{[\omega'] \succ_\Gamma [\omega_x]\} \subseteq \Gamma'^*({x}) \subseteq \{[\omega'] \succeq_\Gamma [\omega_x]\}$. There are three possibilities: if $[\omega_x] \in \Gamma'^*({x})$, the result holds. If $[\omega_x] \notin \Gamma'^*({x})$ but $[\omega_x] \succ_\Gamma [\omega_A]$, then we have $P^*(A) = P(\{[\omega'] \succ_\Gamma [\omega_A]\}) = P(\{[\omega'] \succeq_\Gamma [\omega_x]\})$, and the result holds. And finally, if $[\omega_x] = [\omega_A]$, then $x \in \Gamma(\omega)$ for any $[\omega] \succ_\Gamma [\omega_A]$, a contradiction with $\cap_{[\omega] \succ_\Gamma [\omega_A]} \Gamma(\omega) = \emptyset$.

Assume now that $P^*({x}) < P^*(A) \forall x \in A$; then there exists a sequence $\{x_n\}_n$ s.t. $P^*({x_n})_n$ converges to $\sup_{y \in A} P^*({y}) = P^*(A)$, and such that $P^*({x_n}) < P^*({x_{n+1}}) \forall n$. For any n , there exists some $[\omega_n] \succ_\Gamma [\omega_A]$ s.t. $\{[\omega'] \succ_\Gamma [\omega_n]\} \subseteq \Gamma'^*({x_n}) \subseteq \{[\omega'] \succeq_\Gamma [\omega_n]\}$, and then $P^*(A) = P(\{[\omega] \succ_\Gamma [\omega_A]\}) = P^*({x_n}) = P(\cup_n \{[\omega] \succeq_\Gamma [\omega_n]\})$.

($\Leftarrow$) Consider $A \in \mathcal{A}'_1$, and let $\{\omega_n\}_n \subseteq \Gamma'^*(A)$ s.t. $P^*(A) = P(\cup_n \{[\omega] \succeq_\Gamma [\omega_n]\})$. For any natural number n , take $x_n \in \Gamma(\omega_n) \cap A$. Then, $P^*({x_n}) \geq P(\cup_n \{[\omega] \succeq_\Gamma [\omega_n]\}) = P^*(A)$, and since Theorem 1 implies that P^* is maxitive, $P^*(A) = \sup_n P^*({x_n})$. ■

Next, we use the ideas in this theorem to establish the main result of this paper. It establishes some relationships between different conditions of consonancy and possibility measures as upper probabilities, under more general conditions than the ones in Theorem 2: note that we do not require here neither the chain on the initial space to be complete nor the final σ -field to agree with $\mathcal{P}(X)$.

Theorem 3. *Let Γ be a C1 random set. Then, each of the following hypotheses implies the next:*

1. Γ is antitone.
2. Γ is C0.
3. For any $A \in \mathcal{A}'_1$ there exists some $\{\omega_n\}_n \subseteq \Gamma'^*(A)$ such that $\Gamma'^*(A) = \cup_n \{[\omega] \succeq_\Gamma [\omega_n]\}$.
4. P^* is a possibility measure.

Proof. We start showing that $(1 \Rightarrow 2)$. Let Γ be antitone, and consider $A \subseteq [0, 1]$. If $\inf A$ belongs to A , then $\cap_{\omega \in A} \Gamma(\omega) = \Gamma(\inf A)$. If $\inf A \notin A$, let $\{\omega_n\}_n$ be a

sequence of elements of A that converges to $\inf A$. Then, $\cap_{\omega \in A} \Gamma(\omega) = \cap_n \Gamma(\omega_n)$. Since any antitone random set is C1, we conclude that Γ is C0.

Let us show next that $(2 \Rightarrow 3)$. Consider $A \in \mathcal{A}'_1$. Then, there exists $\{[\omega_n]\}_n \subseteq \Gamma'^*(A)$ s.t. $\cap_{[\omega] \in \Gamma'^*(A)} \Gamma(\omega) = \cap_n \Gamma(\omega_n)$. It is clear that $\cup_n \{[\omega] \succeq_\Gamma [\omega_n]\} \subseteq \Gamma'^*(A)$, because this is a filter from Proposition 2. Assume ex-absurdo that there exists $[\omega_0] \in \Gamma'^*(A)$ s.t. $[\omega_0] \prec_\Gamma [\omega_n] \forall n$. Then, $\cap_{[\omega] \in \Gamma'^*(A)} \Gamma(\omega) \subseteq \Gamma(\omega_0) \subseteq \cap_n \Gamma(\omega_n) = \cap_{[\omega] \in \Gamma'^*(A)} \Gamma(\omega)$, whence $\cap_{[\omega] \in \Gamma'^*(A)} \Gamma(\omega) = \Gamma(\omega_0)$. This means that $\Gamma'^*(A) = \{[\omega'] \succeq_\Gamma [\omega_0]\}$, which contradicts $A \in \mathcal{A}'_1$. Hence, $\Gamma'^*(A) = \cup_n \{[\omega'] \succeq_\Gamma [\omega_n]\}$.

Finally, we prove $(3 \Rightarrow 4)$. Consider $(A_i)_{i \in I} \in \mathcal{A}'$ s.t. $\cup_{i \in I} A_i = A \in \mathcal{A}'_1$. Then, condition (3) implies the existence of $\{[\omega_n]\}_n \subseteq \Gamma'^*(A)$ s.t. $\Gamma'^*(A) = \cup_n \{[\omega] \succeq_\Gamma [\omega_n]\}$. For every n , there exists some $A_n \in (A_i)_{i \in I}$ s.t. $\Gamma(\omega_n) \cap A_n \neq \emptyset$, whence $\Gamma'^*(A) = \cup_n \{[\omega] \succeq_\Gamma [\omega_n]\} = \cup_n \Gamma'^*(A_n)$. As a consequence, we also have $\Gamma^*(A) = \cup_n \Gamma^*(A_n)$. Since P^* is maxitive from Theorem 1 and lower continuous, we conclude that $P^*(A) = \sup_n P^*(A_n) = \sup_{i \in I} P^*(A_i)$. Applying Proposition 4, P^* is a possibility measure. ■

Example 1. It is easy to construct examples showing that the converses of these implications are not true in general: for the first, consider $\Gamma : [1, 2] \rightarrow \mathcal{P}([1, 2])$ given by $\Gamma(\omega) = [\omega, 2]$; the second and third counterexamples can be obtained by suitably modifying the C1 random set Γ in [17–Example 5] s.t. P^* is not a possibility measure: for the first, consider $\mathcal{A}' = \{\emptyset, \mathcal{P}(X)\}$, and $\Gamma_1 : [0, 1] \rightarrow \mathcal{P}([0, 1])$ given by $\Gamma_1(\omega) = \Gamma(\omega) \cup \{0\} \forall \omega \neq 0, \Gamma_1(0) = \{0\}$; for the second, take $\Gamma_2 = \Gamma$ but with a degenerate probability measure on the initial space. ♦

We deduce that an antitone random set is not necessarily C0, and consequently Theorem 3 generalises the first point of Theorem 1. On the other hand, since a C1 random set does not induce a possibility measure ([17]), we deduce that this condition of consonancy is weaker than that of C0.

As we said before, a random set inducing a possibility measure is not necessarily C2 [17], and, consequently, it is not C0 either. Nevertheless, it may be useful to study the representability of a possibility measure in terms of a consonant random set. Goodman proved in [10] that for any possibility measure Π on a measurable space $(X, \mathcal{P}(X))$ there exists an antitone random set whose upper probability is Π . In [19], we considered the problem of the representability when we fix also the initial space. We proved that for any random set Γ inducing a possibility measure there is a C1 random set Γ_1 defined between the same spaces and with the same upper probability. We show next that we may even require Γ_1 to be C0:

Proposition 5. *Let $(\Omega, \mathcal{A}, P)$ be a probability space, $(X, \mathcal{P}(X))$ a measurable space and $\Gamma : \Omega \rightarrow \mathcal{P}(X)$ a random set such that P_Γ^* is a possibility measure. Then, there exists a C0 random set $\Gamma_1 : \Omega \rightarrow \mathcal{P}(X)$ such that $P_{\Gamma_1}^* = P_\Gamma^*$.*

Proof. Let us define $C_x := \{y | P^*(\{y\}) \geq P^*(\{x\})\}$ for any $x \in X$, and $\Gamma_1 : \Omega \rightarrow \mathcal{P}(X)$ by $\Gamma_1(\omega) = \cup_{x \in \Gamma(\omega)} C_x$. We check in [19–Theorem 4.7] that Γ_1 is strongly

measurable, C1 and that $P_{\Gamma_1}^* = P_{\Gamma}^*$. It remains then to verify that it is also C0. Let us consider $A \subseteq \Omega$, and let us denote $z_A = \sup_{\omega \in A} \inf_{x \in \Gamma(\omega)} P^*(\{x\})$. From the definition of Γ_1 we deduce that there are only two alternatives: either $\cap_{\omega \in A} \Gamma_1(\omega) = \{y : P^*(\{y\}) \geq z_A\}$, and then given a sequence $\{\omega_n\}_n$ of elements of A such that $z_n = \inf_{x \in \Gamma(\omega_n)} P^*(\{x\})$ converges to z_A , it is $\cap_n \Gamma_1(\omega_n) \subseteq \cap_n \{y | P^*(\{y\}) \geq z_n\} = \{y | P^*(\{y\}) \geq z_A\} = \cap_{\omega \in A} \Gamma_1(\omega)$; or $\cap_{\omega \in A} \Gamma_1(\omega) = \{y : P^*(\{y\}) > z_A\} \subsetneq \{y : P^*(\{y\}) \geq z_A\}$. In that case, given y_0 s.t. $P^*(\{y_0\}) = z_A$, there exists $\omega_0 \in A$ s.t. $y_0 \notin \Gamma_1(\omega_0)$, whence $\Gamma_1(\omega_0) = \{y | P^*(\{y\}) > z_A\} = \cap_{\omega \in A} \Gamma_1(\omega)$. We conclude in both cases that Γ_1 is C0. ■

To conclude this section, we represent in Figure 1 the relationships between the different conditions of consonancy we have considered and possibility and maxitive measures as upper probabilities. It follows from Example 1 and the examples in [17, 19] that none of the converses of these implications holds in general.

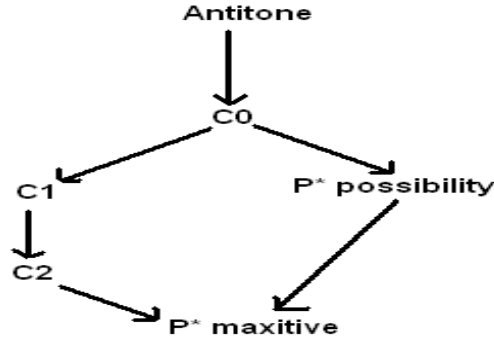


Fig. 1. Relationships between consonancy, P^* possibility and P^* maxitive

4 Consonant Random Sets as Imprecise Random Variables

Among the different interpretations given to random sets, one of the most important in the framework of uncertainty modelling is that of imprecise observations of random variables. This goes back to Kruse and Meyer [15]: we assume the existence of a measurable mapping $U_0 : \Omega \rightarrow X$ which is observed with some imprecision, so that for any ω in the initial space all we know about $U_0(\omega)$ is that it belongs to some subset $\Gamma(\omega)$ of the final space ². We obtain then a multi-valued mapping $\Gamma : \Omega \rightarrow \mathcal{P}(X)$, which, in case it satisfies the condition of strong measurability, is a random set.

² Hence, we will assume in this section that $\Gamma(\omega)$ is non-empty for all $\omega \in \Omega$.

Under such interpretation, our interest lies in the information we can recover about the ‘original’ random variable, U_0 . All we know is that it belongs to the class

$$S(\Gamma) := \{U : \Omega \rightarrow X \text{ measurable}, U(\omega) \in \Gamma(\omega) \forall \omega\}$$

of **measurable selections** of Γ , and consequently its distribution belongs to

$$P(\Gamma) := \{P_U : U \in S(\Gamma)\}.$$

In this section, we are going to study whether this interpretation is compatible with the one we have given to consonant random sets. For this, we must determine first if a consonant random set possesses measurable selections. This is what we prove in the following theorem:

Proposition 6. *Let $(\Omega, \mathcal{A}, P)$ be a probability space, $(X, \mathcal{A}')$ a measurable space s.t. $\mathcal{A}'$ contains the singletons and let $\Gamma : \Omega \rightarrow \mathcal{P}(X)$ be a C0 random set. Then, $S(\Gamma) \neq \emptyset$.*

Proof. If there is some $x \in \cap_{\omega \in \Omega} \Gamma(\omega)$, then the constant mapping on x is trivially a measurable selection of Γ . Assume then that $\cap_{\omega \in \Omega} \Gamma(\omega) = \emptyset$. Since Γ is C0, there exists a countable set $\{\omega_n\}_n$ such that $\cap_n \Gamma(\omega_n) = \cap_{\omega \in \Omega} \Gamma(\omega)$, and we may assume without loss of generality that $\Gamma(\omega_n) \subsetneq \Gamma(\omega_{n-1})$ for all $n \geq 2$. Let us consider $x_n \in \Gamma(\omega_n) \setminus \Gamma(\omega_{n+1})$ for every $n \geq 1$, and define $U := \sum_n x_n I_{\Gamma^*(\{x_n\}) \setminus \Gamma^*(\{x_{n+1}\})}$. It can be checked that this random variable is well defined (that is, $U(\omega) \neq \emptyset \forall \omega$), and this implies that U is a measurable selection of Γ . ■

As far as we know, a similar result for C1 random sets hasn’t been established. On the other hand, it is easy to check that a random set inducing a possibility measure has an *almost sure* measurable selection (i.e., there exists $U : \Omega \rightarrow X$ measurable such that $U(\omega) \in \Gamma(\omega)$ for all but a null subset of Ω). These selections are sometimes used instead of everywhere selections (see for instance [12]); however, when we interpret a random set as a model of the imprecise observation of a random variable, we need to consider measurable mappings which are selections on all the elements of the initial space, and not just on a subset of probability one.

As we said before, if a random set Γ models the imprecise observation of U_0 , our information about P_{U_0} is given by the class $P(\Gamma)$; although this is the most precise class we can consider, it may be more useful for practical purposes to work with the class

$$M(P^*) = \{Q : \mathcal{A}' \rightarrow [0, 1] \text{ probability s.t. } Q(A) \leq P^*(A) \forall A \in \mathcal{A}'\}$$

of probability distributions dominated by the upper probability P^* : this class is convex and is uniquely determined by P^* , and in some cases, it is even determined by the values of P^* on some classes of sets (see the discussion on this subject in [20]). It becomes then interesting to investigate the relationship between these two classes, so that we can decide if the use of P^* for modelling the information about P_{U_0} causes an important loss of precision.

The relationship between $P(\Gamma)$ and $M(P^*)$ has been studied by a number of authors ([1, 3, 11, 12, 20]), under different hypotheses on the images of the random set and on its initial or final spaces. We are going to study here the situation for consonant random sets. First, we investigate whether the bound given by $P^*(A)$ is tightest we can give for the value $P_{U_0}(A)$ for some arbitrary set A in the final σ -field. We will use the condition of condensability, whose definition can be found in [25].

Proposition 7. *Let $(\Omega, \mathcal{A}, P)$ be a probability space, $(X, \mathcal{A}')$ a measurable space s.t. $\mathcal{A}'$ includes the singletons and $\Gamma : \Omega \rightarrow \mathcal{P}(X)$ a random set. If $S(\Gamma) \neq \emptyset$ and P^* is condensable, then $P^*(A) = \max_{Q \in P(\Gamma)} Q(A) \forall A \in \mathcal{A}'$.*

Proof. Let $A \in \mathcal{A}'$. Then, the condensability of P^* implies the existence of a countable set $\{x_n\}_n \subseteq A$ s.t. $P^*(\{x_n\}_n) = P^*(A)$. Take $U \in S(\Gamma)$, and let us define $V := \sum_n x_n I_{\Gamma^*(\{x_n\}) \setminus \Gamma^*(\{x_1, \dots, x_{n-1}\})} + UI_{(\Gamma^*(\{x_n\}_n))^c}$. Then, V is a measurable selection of Γ and moreover $P^*(A) = P^*(\{x_n\}_n) = P_V(\{x_n\}_n) \leq \max_{Q \in P(\Gamma)} Q(\{x_n\}_n) \leq \max_{Q \in P(\Gamma)} Q(A) \leq P^*(A)$. Consequently $P^*(A) = \max_{Q \in P(\Gamma)} Q(A)$. ■

Note that the result holds in particular for those random sets Γ inducing a possibility measure and such that $S(\Gamma)$ is non-empty. Using this proposition, we deduce the following:

Corollary 1. *Let $(\Omega, \mathcal{A}, P)$ be a probability space, (X, d) be a separable metric space and let $\Gamma : \Omega \rightarrow \mathcal{P}(X)$ be a C0 random set. Then, under the weak topology, we have*

1. $cl(M(P^*)) = cl(Conv(P(\Gamma)))$.
2. If moreover $(\Omega, \mathcal{A}, P)$ is non-atomic, then $cl(P(\Gamma)) = cl(M(P^*))$.

Proof. From Proposition 6, Γ possesses measurable selections, and from Theorem 3, P^* is a possibility measure. Applying [19–Theorem 2.4], P^* is condensable, and Proposition 7 implies then that $P^*(A) = \max_{Q \in P(\Gamma)} Q(A)$ for any A in the final σ -field. The result follows now from [20–Theorems 4.4 and 4.7]. ■

This shows that, in the case of C0 random sets, the upper probability can be used to model the information about P_{U_0} without causing a big loss of precision. Note that the hypothesis of non-atomicity of $(\Omega, \mathcal{A}, P)$ is not too restrictive: it holds for instance in the particular case of antitone random sets, or when we know that the probability distribution of U_0 is continuous. We must warn the reader, however, that the sets $P(\Gamma)$ and $M(P^*)$ do not necessarily agree for C0 random sets, as the following example shows:

Example 2. [21–Example 3.3] Let us consider the antitone random set $\Gamma : [0, 1] \rightarrow \mathcal{P}([0, 1])$ given by $\Gamma(\omega) = [-\omega, \omega]$. Then, the uniform probability distribution on $[-1, 1]$ belongs to $M(P^*) \setminus P(\Gamma)$. ♦

Therefore, the use of the upper probability in a C0 random set may cause some loss of information respect to the class of the probability distributions

of the measurable selections. We would like to know if under some additional conditions we can guarantee the equality $P(\Gamma) = M(P^*)$. In [21], we give a number of sufficient conditions for this equality when Γ is a random interval. Although one of those conditions (namely, that $\Gamma = [0, B]$ for some non-negative random variable B) is compatible with C0 random sets, the most important one (that $\Gamma = [A, B]$ with A, B strictly comonotone) will only be compatible with the C0 condition when A and B are constant. More specifically, in the particular case where X is a finite space, we have that $P(\Gamma) = M(P^*)$ whenever the initial probability space is non-atomic, regardless of the characteristics of the images of Γ ([18]). We conclude from this that C0 random sets are not specially suited, when compared to other types of random sets, for modelling the imprecise observation of a random variable.

5 Conclusions

The approximation to consonant random sets we have considered in this paper has allowed us to prove a number of results in a fairly straightforward manner. It allows us moreover to consider consonant random sets defined between arbitrary spaces, and not necessarily antitone, because in our opinion the core of the notion of consonancy is the order we can establish in the initial space. In this respect, it would be interesting to make a deeper study of the properties of this order. We would like in particular to see if the completeness of the chain induced by a consonant random set is related to some additional condition on its images.

Concerning the different definitions of consonancy considered in this paper, we think that C0 random sets are sufficiently general and have moreover a number of interesting properties that other weaker notions, such as C1 and C2 random sets, do not possess in general. As an open problem from this paper, we propose to study the relationship between C0 and C1 random sets, and if a C1 random set inducing a possibility measure is always C0.

Finally, regarding the use of consonant random sets as a model for the imprecise observation of random variables, we still have to determine whether C1 random sets or random sets inducing a possibility measure possess measurable selections; this existence would allow us to derive a number of relationships between the class of probability distributions of these selections and those dominated by the upper probability, in the vein of Corollary 1. We wonder if in this case the study of the chain induced on the initial space will also be helpful.

References

- [1] Z. Arstein and S. Hart. Law of large numbers for random sets and allocation processes. *Mathematics of Operations Research*, 6(4):485–492, 1981.
- [2] G. Birkhoff. *Lattice theory*. AMS Colloquium Publications 25, 1967.
- [3] A. Castaldo, F. Macceroni and M. Marinacci. Random correspondences as bundles of random variables. *Sankhya* 66(3):409–427, 2004.

- [4] G. de Cooman and D. Aeyels. A random set description of a possibility measure and its natural extension. *IEEE Transactions on Systems, Man and Cybernetics*, 30(2):124–130, 2000.
- [5] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38:325–339, 1967.
- [6] D. Dubois and H. Prade. The mean value of a fuzzy number. *Fuzzy Sets and Systems*, 24(3):279–300, 1987.
- [7] D. Dubois and H. Prade. *Possibility theory*. Plenum Press, New York, 1988.
- [8] D. Dubois and H. Prade. When upper probabilities are possibility measures. *Fuzzy Sets and Systems*, 49(1):65–74, 1992.
- [9] G. Gierz, K. Hofmann, K. Keimel, J. Lawson, M. Mislove and D. Scott. *A compendium of continuous lattices*. Springer, Berlin, 1980.
- [10] I. R. Goodman. Fuzzy sets as equivalence classes of possibility random sets. In *Fuzzy Sets and Possibility Theory: Recent Developments* (R. R. Yager, ed.), 327–343. Pergamon, Oxford, 1982.
- [11] S. Hart and E. Köhlberg. Equally distributed correspondences. *Journal of Mathematical Economics*, 1(2):167–174, 1974.
- [12] C. Hess. The distribution of unbounded random sets and the multivalued strong law of large numbers in nonreflexive Banach spaces. *Journal of Convex Analysis*, 6(1):163–182, 1999.
- [13] W. Hildenbrand. *Core and Equilibria of a Large Economy*. Princeton University Press, Princeton, 1974.
- [14] C.J. Himmelberg. Measurable relations. *Fundamenta Mathematicae*, 87:53–72, 1975.
- [15] R. Kruse and K. D. Meyer. *Statistics with vague data*. D. Reidel Publishing Company, Dordrecht, 1987.
- [16] G. Mathéron. *Random sets and integral geometry*. Wiley, New York, 1975.
- [17] E. Miranda, I. Couso and P. Gil. Relationships between possibility measures and nested random sets. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(1):1–15, 2002.
- [18] E. Miranda, I. Couso and P. Gil. Upper probabilities and selectors of random sets. In *Soft Methods in Probability, Statistics and Data Analysis* (P. Grzegorzewski, O. Hryniewicz and M. A. Gil, eds.), 126–133, Physica-Verlag, 2002.
- [19] E. Miranda, I. Couso and P. Gil. A random set characterisation of possibility measures. *Information Sciences*, 168(1-4):51–75, 2004.
- [20] E. Miranda, I. Couso and P. Gil. Random sets as imprecise random variables. *Journal of Mathematical Analysis and Applications*, 2005, in press.
- [21] E. Miranda, I. Couso and P. Gil. Random intervals as a model for imprecise information. *Fuzzy Sets and Systems*, 2005, in press.
- [22] H. T. Nguyen. On random sets and belief functions. *Journal of Mathematical Analysis and Applications*, 65(3):531–542, 1978.
- [23] G. L. S. Shackle. *Decision, Order and Time in Human Affairs*. Cambridge University Press, Cambridge, 1961.
- [24] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, New Jersey, 1976.
- [25] G. Shafer. Allocations of probability. *Annals of Probability*, 7(5):827–839, 1979.
- [26] N. Shilkret. Maxitive measures and integration. *Indagationes Mathematicae*, 33:109–116, 1971.
- [27] M. Sugeno. Theory of fuzzy integrals and its applications, PhD Thesis, Tokyo Institute of Technology, 1974.
- [28] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3–28, 1978.

Comparative Conditional Possibilities

Giulianella Coletti¹ and Barbara Vantaggi²

¹ Dip. Matematica e Informatica,
Università di Perugia, 06123 Perugia, Italy
`coletti@dipmat.unipg.it`

² Dip. Metodi e Modelli Matematici,
Università “La Sapienza” Roma, 00161 Roma, Italy
`vantaggi@dmmm.uniroma1.it`

Abstract. Any dynamic decision model or procedure for acquisition of knowledge must deal with conditional events and should refer to (not necessarily structured) domains containing only the elements and the information of interest. We consider conditional possibility theory as numerical reference model to handle uncertainty and to study binary relations, defined on an arbitrary set of conditional events expressing the idea of “no more possible than”. We give the necessary conditions for the representability of a relation by a T -conditional possibility, for any triangular norm T , and we provide a complete characterization in terms of necessary and sufficient conditions for the representability by a conditional possibility (i.e. when T is the minimum).

1 Introduction

In the relevant literature, following the kolmogorovian probabilistic model, a conditional measure is usually defined starting from an unconditional one. But this is a very restrictive view of conditioning, trivially corresponding to just a modification of the “world”. It is instead essential to regard conditioning events as “variables” or, in other words, as uncertain events which can be either true or false. This point of view gives the opportunity to the decision maker or the field expert to take into account at the same time all the possible scenarios (represented by the conditioning events of interest).

On the other hand, starting from probability [6, 11] many models based on a *direct* definition of conditional measures have been recently given in literature (see, for instance, [5, 7]): these conditional measures are directly defined on a set (with a suitable algebraic structure) of conditional events, in such a way that $\Pi(E|H)$ makes sense for any pair of events E and H , with $H \neq \emptyset$, and it satisfies *suitable axioms*.

But in a situation of partial knowledge, it is clearly very significant (especially from the point of view of any real application) not assuming that the chosen family of conditional events, on which the conditional measure must be assessed, had any specific algebraic structure. To be able to handle assessments relative to

an arbitrary set of conditional events, it is necessary to avoid the introduction of “arbitrary” information, which is often also cause of inconsistencies.

However, in many situations the field expert or the decision maker, due to his partial knowledge, is not able or interested to give a numerical evaluation “even if partial”. In these situations, we are content with getting (from the decision maker) an ordinal evaluation (i.e. a comparative degree of belief among conditional events) comparing only some uncertain alternatives.

In this case, given a numerical model of reference (e.g. probabilistic, possibilistic framework) it necessary to determine the conditions characterizing ordinal relations $\preceq$, which are representable by a function (e.g. probability, possibility measures) belonging to the numerical reference model.

In literature it is well-known that conditional relations are usually derived from an unconditional one simply by assuming that an event H (strictly preferred to the impossible event) occurs [4, 16, 17]. Some authors just consider ordinal relation $\preceq_H$ among conditional events conditioned to the same event H .

However, it is enough to consider the following example to understand that this view is restrictive: let us consider the following events H = “the daily variation of MIB TEL tomorrow will be less than -0.1% ”, E = “I buy the portfolio A whose return depends on MIB TEL index” and G = “the Italian main electric company Enel will buy a share of two Rumanian companies, which supply electric power”, and we can take the ordinal relation $E|H \prec E|H^c$, $H|G \sim H$, which compares conditional events with different conditioning events.

Then, a conditional model needs to deal with ordinal relations $\preceq$ defined on an arbitrary set of conditional events. This topic has been faced in [9] by using as reference model conditional probability and generalized decomposable measures.

In this paper, we will adopt as numerical model of reference conditional possibilities (which are decomposable measures), following the definition proposed in [2] and we study “partial” numerical assessments that can be seen as the restriction of a conditional possibility.

The main aim of this paper is to study ordinal relations on conditional events and to characterize them by taking possibility theory as numerical reference model. Then, we give necessary conditions for representability of a relation on a finite set of conditional events by a T -conditional possibility (with T a triangular norm).

Moreover, we provide a complete characterization in the particular case of conditional possibility (when the triangular norm T is the minimum) by giving necessary and sufficient conditions for the representability of a relation by a coherent conditional possibility.

2 Conditional Possibility

We recall that given a Boolean algebra $\mathcal{B}$, a function

$$\Pi : \mathcal{B} \rightarrow [0, 1]$$

is a possibility measure if $\Pi(\Omega) = 1$, $\Pi(\emptyset) = 0$ and for every $E_i, E_j \in \mathcal{B}$ one has

$$\Pi(E_i \vee E_j) = \max\{\Pi(E_i), \Pi(E_j)\}.$$

In the relevant literature conditional possibility are mostly presented as a derived notion of the unconditional ones: given a possibility $\Pi(\cdot)$ on $\mathcal{B}$ and a triangular norm T (i.e. a function from $[0, 1] \times [0, 1]$ into $[0, 1]$ commutative, associative, increasing and satisfying the boundary condition $T(1, x) = x$, for any x [15]), for any $H \in \mathcal{B} \setminus \{\emptyset\}$, a T -conditional possibility $\Pi(\cdot|H)$ on E is actually defined as any solution of the equation

$$\Pi(E \wedge H) = T(x, \Pi(H)). \quad (1)$$

For example, when the t-norm (which stands for triangular norm) $T = \min$, the possible solutions of the relevant equation (1) (see [14]) are

$$\begin{aligned} \Pi(A|B) &= \Pi(A \wedge B) & \text{if } \Pi(A \wedge B) < \Pi(B) \\ \Pi(A \wedge B) &\leq \Pi(A|B) \leq 1 & \text{if } \Pi(A \wedge B) = \Pi(B). \end{aligned}$$

Note that an arbitrary solution needs not be a normalized possibility: it happens, for example, choosing Zadeh's conditioning rule [18] where the conditional possibility $\Pi(A|B)$ is defined as $\Pi(A \wedge B)$: in this case $\Pi(\cdot|B)$ is simply defined as the projection of the joint possibility.

To avoid such a problem of getting not normalized (conditional) possibility different conditions have been proposed: it is well known the following given in [13], which consists in taking the greatest solution, i.e.

$$\Pi(A|B) = \begin{cases} \Pi(A \wedge B) & \text{if } \Pi(A \wedge B) < \Pi(B) \\ 1 & \text{if } \Pi(A \wedge B) = \Pi(B). \end{cases} \quad (2)$$

According to this definition any conditional possibility is normalized, but, we can obtain the following counter-intuitive situation: let A and B be two incompatible (i.e. $A \wedge B = \emptyset$) events and $\Pi(B) = 0$, then $\Pi(A|B)$ is equal to 1 (according to (2)) instead of 0 as it would be natural (being $A|B = \emptyset|B$). Note that for such conditioning events B the function $\Pi(\cdot|B)$ is not a possibility.

Moreover, for any event A such that $\Pi(A \wedge B) = \Pi(B)$ it is not convincing to chose as $\Pi(A|B)$ a unique value inside $[\Pi(B), 1]$; however the choice of value 1 for (at least) an atom $C \subseteq B$ is necessary (see [1, 13]) to get a normalized possibility.

Here, we adopt a different model introduced in [2], whose "primitive" concept is the conditional possibility, that is a function Π defined on a set of conditional events satisfying a set of axioms:

Definition 1. Let $\mathcal{E} = \mathcal{B} \times \mathcal{H}$ be a finite set of conditional events $E|H$ such that $\mathcal{B}$ is a Boolean algebra and $\mathcal{H}$ an additive set (i.e. closed with respect to logical sums), with $\mathcal{H} \subset \mathcal{B}$ and $\emptyset \notin \mathcal{H}$. A function $\Pi : \mathcal{E} \rightarrow [0, 1]$ is a T -conditional possibility if it satisfies the following properties:

1. $\Pi(E|H) = \Pi(E \wedge H|H)$, for every $E \in \mathcal{B}$ and $H \in \mathcal{H}$;
2. $\Pi(\cdot|H)$ is a possibility measure, for any $H \in \mathcal{H}$;
3. for any $H, E \wedge H \in \mathcal{H}$ and $E, F \in \mathcal{B}$

$$\Pi(E \wedge F|H) = T(\Pi(E|H), \Pi(F|E \wedge H)).$$

Notice that condition 2 requires that, for any conditioning event $H \in \mathcal{H}$, the function $\Pi(\cdot|H)$ is a possibility, which implies that the function is normalized.

According to Definition 1, a T -conditional possibility cannot always be derived by just one “unconditional” possibility except in the trivial case when $\Pi(E \wedge H) < \Pi(H)$ for any $H \in \mathcal{H}$ and $E \in \mathcal{B}$ (with $E \wedge H \neq H$).

Moreover, condition 3 implies that the conditional possibility $\Pi(\cdot|H)$ is not singled-out by the possibility of its conditioning event H , but its value is ruled by the values of other possibilities $\Pi(\cdot|E \wedge H)$, for suitable events E .

Actually, (see [2]) the above axiomatic definition includes for any pair of logical independent events A, B the definitions of T -conditional possibilities $\Pi(A|B)$ given in [10, 13].

2.1 Coherence

We note that the above set of axioms (of Definition 1) works if the set $\mathcal{E}$ is well structured by a logical point of view, but the axioms become not enough strong when the set of conditional events is arbitrary (more precisely when the function is not defined on the whole product of a Boolean algebra and an additive set). To handle general situations we introduce the concept of coherence.

Definition 2. *Given an arbitrary set of conditional events $\mathcal{F}$, a real function Π on $\mathcal{F}$ is a coherent T -conditional possibility assessment if, there exists $\mathcal{E} \supseteq \mathcal{F}$ with $\mathcal{E} = \mathcal{B} \times \mathcal{H}$, such that there exists a T -conditional possibility $\Pi'(\cdot|\cdot)$ on $\mathcal{E}$ extending Π .*

Obviously, in the unconditional case a function on $\mathcal{F} = \{E_1, \dots, E_n\}$ into $[0, 1]$ is a coherent possibility assessment if it can be extended on the algebra $\mathcal{B}$, spanned by $\mathcal{F}$, as a possibility measure.

Remark 1. If Π on $\mathcal{F}$ is a coherent T -conditional possibility, then it can be extended on any set $\mathcal{E} \supseteq \mathcal{F}$, in particular also in $\mathcal{B} \times \mathcal{B}^o$, with $\mathcal{B}$ the algebra generated by the set $\{E, H : E|H \in \mathcal{F}\}$ and $\mathcal{B}^o = \mathcal{B} \setminus \{\emptyset\}$.

We recall a characterization of coherent conditional possibilities (i.e. $T = \min$) given in [8].

Definition 3. *Let $\mathcal{B}$ be a finite algebra and $\mathcal{C}_o$ the set of atoms of $\mathcal{B}$. The class $\mathcal{P} = \{\Pi_o, \dots, \Pi_k\}$ of possibilities defined on $\mathcal{B}$ is said nested if, denoting (for $j = 1, \dots, k$) by $\mathcal{C}_j = \{C \in \mathcal{C}_{j-1} : \Pi_{j-1}(C) < 1\}$ and*

$$\mathcal{H}_j = \{C_i \in \mathcal{C}_j : \nexists C \in \mathcal{C}_o \text{ s.t. } \Pi_{j-1}(C) > \Pi_{j-1}(C_i)\},$$

the following conditions hold:

1. $\Pi_j(C) = \Pi_{j-1}(C)$ if $C \in \mathcal{C}_j \setminus \mathcal{H}_j$ ($j > 0$);
2. $\Pi_{j-1}(C) \leq \Pi_j(C) \leq 1$ if $C \in \mathcal{H}_j$ ($j > 0$);
3. $\Pi_j(C) = 0$ for all the atoms $C \in \mathcal{C}_o \setminus \mathcal{C}_j$;
4. for any $C \in \mathcal{C}_o$ there exists a (unique) $j = 0, \dots, k$ such that $\Pi_j(C) = 1$.

Obviously, $\mathcal{H}_j \subseteq \mathcal{C}_j$ and $\mathcal{C}_j \subset \mathcal{C}_{j+1}$, moreover $\mathcal{H}_j$ (with $j > 0$) is a subset of atoms in $\mathcal{C}_j$, or better it includes those atoms with the “highest” possibility under Π_{j-1} , which potentially can have possibility equal to 1 under Π_j .

Note that since Π_j 's ($j = 0, \dots, k$) are possibilities, so there is at least an atom such that $\Pi_j(C) = 1$, then k must be less than the number of atoms in $\mathcal{B}$.

Theorem 1. *Let $\mathcal{F} = \{E_1|H_1, \dots, E_n|H_n\}$ be a finite set of conditional events, $\mathcal{C}_o$ and $\mathcal{B}$ denote, respectively, the set of atoms and the algebra generated by $\{E_1, H_1, \dots, E_n, H_n\}$.*

For a real function $\Pi : \mathcal{F} \rightarrow [0, 1]$, the following two statements are equivalent:

- a) *Π is a coherent conditional possibility on $\mathcal{F}$;*
- b) *there exists (at least) a nested class $\mathcal{P} = \{\Pi_o, \dots, \Pi_k\}$ of possibilities on $\mathcal{B}$, such that for any $E_i|H_i \in \mathcal{F}$ there exists a unique Π_α with $\Pi_\alpha(H_i) = 1$ and $\Pi(E_i|H_i)$ is the unique solution of the equation*

$$\Pi_\alpha(E_i \wedge H_i) = \max\{x, \Pi_\alpha(H_i)\}, \quad (3)$$

and it is solution of any equation

$$\Pi_\beta(E_i \wedge H_i) = \min\{x, \Pi_\beta(H_i)\} \quad (4)$$

with $\beta \leq \alpha$;

- c) *there exists a sequence of compatible systems*

$$S_\alpha = \begin{cases} \max_{C_r \subseteq E_i \wedge H_i} x_r^\alpha = \min\{\Pi(E_i|H_i), \max_{C_r \subseteq H_i} x_r^\alpha\} & \text{if } \max_{C_r \subseteq H_i} \mathbf{x}_r^{\alpha-1} < 1 \\ x_r^\alpha \geq \mathbf{x}_r^{\alpha-1} & \text{if } C_r \in \mathcal{H}^\alpha \\ x_r^\alpha = \mathbf{x}_r^{\alpha-1} & \text{if } C_r \in \mathcal{C}_\alpha \setminus \mathcal{H}^\alpha \\ \max_{C_r \in \mathcal{C}_\alpha} x_r = 1 \\ x_r^\alpha \geq 0 & \forall C_r \in \mathcal{C}_\alpha \end{cases}$$

with $\alpha = 0, \dots, k$, where $\mathbf{x}^\alpha$ (with r -th component $\mathbf{x}_r^\alpha$) is the solution of the system S_α and $\mathcal{C}_\alpha = \{C_r : \mathbf{x}_r^\alpha < 1\}$, $\mathcal{H}^\alpha = \{C_r : C_r \in \mathcal{C}_\alpha, \mathbf{x}_r^{\alpha-1} = \max_{C_j \in \mathcal{C}_\alpha} \mathbf{x}_j^{\alpha-1}\}$,

moreover $\mathbf{x}_r^{-1} = 0$ for any C_r in $\mathcal{C}_o$.

Any class $\{\Pi_\alpha\}$ of possibilities on $\mathcal{B}$ is said *agreeing* with a coherent conditional possibility $\Pi(\cdot|\cdot)$ if it is nested and satisfies condition b) of Theorem 1.

Remark 2. Any given conditional possibility on $\mathcal{B} \times \mathcal{B}^o$ generates a unique class $\{\Pi_\alpha\}$ agreeing with Π ; vice versa a nested class gives rise to a unique conditional possibility on $\mathcal{B} \times \mathcal{B}^o$ (obviously agreeing). Then, any conditional possibility can be “represented” by means of a suitable class of possibilities instead of just one possibility measure [8].

Moreover, for a coherent conditional possibility Π there are many agreeing classes agreeing with Π : this happens also when the conditional possibility Π is defined on $\mathcal{B} \times \mathcal{H}$ with $\mathcal{H} \subset \mathcal{B}^o$.

3 Qualitative Model

In [12] comparative possibilities have been introduced: given a finite algebra $\mathcal{B}$, a binary relation $\preceq$ on $\mathcal{B}$ is a comparative possibility if $\preceq$ is a total preorder (i.e. reflexive, transitive and defined for any pair $E, F \in \mathcal{B}$) satisfying non-triviality condition (i.e. $\emptyset \prec \Omega$) and, for any $A, B, C \in \mathcal{B}$

$$(PO) \quad A \preceq B \implies (A \vee C) \preceq (B \vee C).$$

The relation $\preceq$ is composed by an asymmetric part denoted by $\prec$ and a symmetric one denoted as $\sim$.

As proved in [12] the condition (PO) is equivalent to the following condition

$$A \preceq B \implies (A \vee B) \sim B \text{ for any } A, B \in \mathcal{B}, \quad (5)$$

and moreover, any comparative possibility is representable by a possibility measure. More precisely, $\preceq$ is a comparative possibility if and only if there exists a possibility Π such that

$$E \preceq F \implies \Pi(E) \leq \Pi(F)$$

$$E \prec F \implies \Pi(E) < \Pi(F).$$

The main aim of this paper is to study binary relations defined on a finite set of conditional events, and to detect which are the necessary and sufficient conditions for representing a relation $\preceq$ by a coherent conditional possibility.

Let $\mathcal{E}$ be a set of conditional events and $\preceq$ a binary relation defined on $\mathcal{E}$, a function f from $\mathcal{E}$ to $\mathbf{R}^+$ represents $\preceq$ if

$$E|H \preceq F|K \implies f(E|H) \leq f(F|K),$$

$$E|H \prec F|K \implies f(E|H) < f(F|K).$$

We introduce for a relation $\preceq$ defined on $\mathcal{B} \times \mathcal{H}$ the following basic axioms, which are necessary and sufficient (see [9]) for the representation of $\preceq$ by a generalized decomposable conditional measure, note that conditional possibilities and conditional probabilities are just specific cases (see [5]).

(A0) the relation $\preceq$ is a total preorder, (i.e. it is reflexive, transitive and defined for every pair $E|H, F|K$);

(A1) for all $H \in \mathcal{H}$ it holds $\emptyset|H \prec H|H$;

(A2) for all $H, K \in \mathcal{H}$ it holds $\emptyset|H \sim \emptyset|K$ and $H|H \sim K|K$;

(M) for all $A, B \in \mathcal{B}$ and $H \in \mathcal{H}$, if $(A \wedge H) \subseteq (B \wedge H)$, then $A|H \preceq B|H$;

(A3) for all $A_i, B_i \in \mathcal{B}$ and $B_i \wedge H_i, H_i \in \mathcal{H}$, ($i = 1, 2$)

$A_1|B_1 \wedge H_1 \preceq A_2|B_2 \wedge H_2$ and $B_1|H_1 \preceq B_2|H_2 \implies A_1 \wedge B_1|H_1 \preceq A_2 \wedge B_2|H_2$,
and

$A_1|B_1 \wedge H_1 \preceq B_2|H_2$ and $B_1|H_1 \preceq A_2|B_2 \wedge H_2 \implies A_1 \wedge B_1|H_1 \preceq A_2 \wedge B_2|H_2$.

Conditions (A0), (A1) and (M) just refer to a given conditioning event H . In fact, considering the relation $\preceq_H$ (with $H \in \mathcal{H}$) defined (as the restriction of $\preceq$) through

$$A \preceq_H B \Leftrightarrow A|H \preceq B|H$$

for any $A, B \in \mathcal{B}$; requiring the above three conditions, $\preceq_H$ is representable by a capacity.

The condition (A2) is a first constraint between the different conditioning events.

Axiom (A3) links the relations related to conditional events with different conditioning; it has the same role of axiom 3 (of Definition 1) in the numerical framework.

Note that axiom (A3) implies that for any $A \in \mathcal{B}$ and $B, H \in \mathcal{H}$ with $A \subset B \subset H$

$$A|H \preceq A|B, \quad (6)$$

in fact, taking $A_1 = A = A_2$, $H_1 = H$, $B_1 = B$ and $H_2 = B \wedge H = B_2$ and substituting in condition (A3), one gets (6). Moreover, applying also (A2) and transitivity one has that

$$B|H \sim H|H \Rightarrow A|B \sim A|H. \quad (7)$$

Analogously, it follows

$$A|B \sim B|B \Rightarrow A|H \sim B|H. \quad (8)$$

Moreover, replacing condition (M) with the following stronger one (essentially that proposed by Dubois in [12] reread on the hypothesis H , i.e. by putting H as conditioning event instead of Ω)

(CPO) for all $A, B, C \in \mathcal{B}$ and $H \in \mathcal{H}$

$$A|H \preceq B|H \Rightarrow (A \vee C)|H \preceq (B \vee C)|H,$$

it follows that $\preceq_H$ is representable by a possibility $\Pi(\cdot|H)$.

Note that (CPO) implies

$$A|H \preceq B|K \text{ and } C|H \preceq D|K \Rightarrow (A \vee C)|H \preceq (B \vee D)|K.$$

In fact, suppose without loss of generality $A|H \preceq C|H$, then from relation (5) and transitivity $(A \vee C)|H \sim C|H \preceq D|K \preceq (B \vee D)|K$ follows.

As concern axiom (CPO) we recall for completeness other results, which essentially goes in the same lines of those given in [12] for unconditional case.

Proposition 1. *Let $\preceq$ be a binary relation on a finite set of conditional events $\mathcal{E} = \mathcal{B} \times \mathcal{H}$. If $\preceq$ is a total preorder satisfying conditions (A1) and (CPO), then it satisfies the following conditions:*

1. *if $A|H \preceq B|H \preceq C|H$, then $A \vee C|H \sim B \vee C|H$,*
2. *if $A|H \prec B|H$ and $C|H \prec B|H$, then $A \vee C|H \prec B|H \sim B \vee C|H \sim A \vee B \vee C|H$.*

Proof: If $A|H \preceq B|H \preceq C|H$, then (see implication (5)) $A \vee C|H \sim C|H \sim B \vee C|H$, so condition 1 follows.

Let, $A|H \prec B|H$ and $C|H \prec B|H$, if $C|H \preceq A|H$, then $A \vee C|H \sim A|H \prec B|H$, while if $A|H \preceq C|H$, then $A \vee C|H \sim C|H \prec B|H$, hence condition 2 holds.

Corollary 1. *Let $\preceq$ be a binary relation on a finite set of conditional events $\mathcal{E} = \mathcal{B} \times \mathcal{H}$, denote by $\mathcal{C}$ the set of atoms contained in the algebra $\mathcal{B}$. If $\preceq$ is a total preorder satisfying conditions (A1) and (CPO), then for any $H \in \mathcal{H}$ there is at least an atom $C \in \mathcal{C}$ with $C \subseteq H$ such that $C|H \sim H|H$.*

Proof: Suppose there is an $H \in \mathcal{H}$ such that for any $C \in \mathcal{C}$ one has $C|H \prec H|H$, then from Proposition 1 (condition 2) it follows that $(\bigvee_{C \in \mathcal{C}} C)|H \prec H|H$, but this is absurd since $(\bigvee_{C \in \mathcal{C}} C)|H = H|H$.

Proposition 2. *Let Π be a coherent T -conditional possibility on a finite set of conditional events $\mathcal{E}$. If $\preceq$ is a binary relation on $\mathcal{E}$ induced by Π , then $\preceq$ admits an extension on $\mathcal{B} \times \mathcal{H} \supseteq \mathcal{E}$ (with $\mathcal{B}$ algebra and $\mathcal{H}$ additive set) satisfying (A0), (A1), (A2), (A3) and (CPO).*

Proof: Let us denote by Π^* any T -conditional possibility on $\mathcal{B} \times \mathcal{H}$ extending Π . Since for any $H \in \mathcal{H}$ the function $\Pi^*(\cdot|H)$ is a possibility measure, then the binary relation $\preceq_H$ induced by $\Pi^*(\cdot|H)$ satisfies (CPO) and (A1). Being $\Pi^*(\emptyset|H) = 0 = \Pi^*(\emptyset|K)$, for any $H, K \in \mathcal{H}$, (A2) holds.

Moreover, condition (A0) is satisfied since $\preceq$ is induced from a real function, while (A3) follows directly from condition 3 of Definition 1.

3.1 Comparative Coherent Conditional Possibility

The above result shows that the axioms (A0) – (A3) and (CPO) are necessary for representing a relation $\preceq$ by a T -conditional possibility, for any t-norm T . Obviously, the choice of the particular t-norm requires an additional axiom; in the case of conditional possibilities (i.e. when the t-norm is the minimum), the relation must satisfy also

$$A \wedge B|H \prec B|H \Rightarrow A \wedge B|H \sim A|B \wedge H \quad (9)$$

for any $A, B \in \mathcal{B}$, $H \in \mathcal{H}$.

The following result shows that axiom (9) links the different relations $\preceq_{H_i}$ (restrictions of $\preceq$ to the events conditioned on the same H_i), for suitable conditioning events H_i .

Proposition 3. *Let $\preceq$ be a relation on $\mathcal{B} \times \mathcal{H}$ satisfying condition (9), then for any $A \subseteq C$ and $B \subseteq C$ the following statements hold:*

- if $A|H \prec B|H \prec C|H \Rightarrow A|C \wedge H \prec B|C \wedge H \prec C|C \wedge H$
- if $A|H \sim B|H \prec C|H \Rightarrow A|C \wedge H \sim B|C \wedge H \prec C|C \wedge H$.

Proof: If $A|H \prec B|H \prec C|H$, then by condition (9)

$$A|C \wedge H \sim A|H \prec B|H \sim B|C \wedge H \prec C|H \preceq C|C \wedge H.$$

If $A|H \sim B|H \prec C|H$, then by condition (9) $A|C \wedge H \sim A|H \sim B|H \sim B|C \wedge H$. So the thesis follows.

Definition 4. Let $\mathcal{E} = \mathcal{B} \times \mathcal{H}$ be a finite set of conditional events with $\mathcal{B}$ a boolean algebra and $\mathcal{H}$ an additive set such that $\mathcal{H} \subset \mathcal{B}$ and $\emptyset \notin \mathcal{H}$. A binary relation $\preceq$ on $\mathcal{E}$ is said comparative conditional possibility iff the following conditions hold:

1. $\preceq$ is a total preorder;
2. for any $A \in \mathcal{B}$ and $H, K \in \mathcal{H}$, $\emptyset|H \sim \emptyset|K \prec H|H \sim K|K$;
3. for any $A, B \in \mathcal{B}$ and $H, B \wedge H \in \mathcal{H}$,

$$A \wedge B|H \preceq A|B \wedge H$$

and moreover if $A \wedge B|H \prec B|H$ or $B|H \sim H|H$, then $A \wedge B|H \sim A|B \wedge H$;

4. $\preceq$ satisfies (CPO).

Actually, condition 3. of Definition 4 is a reinforcement of (A3):

Proposition 4. Let $\preceq$ be a relation on $\mathcal{E} = \mathcal{B} \times \mathcal{H}$ satisfying (CPO), the condition $A \wedge B|H \preceq A|B \wedge H$, for any $A, B \in \mathcal{B}$ and $H, B \wedge H \in \mathcal{H}$, and moreover if $A \wedge B|H \prec B|H$ or $B|H \sim H|H$, then $A \wedge B|H \sim A|B \wedge H$. Then $\preceq$ satisfies (A3).

Proof: Let $A_1|B_1 \wedge C_1, B_1|C_1, A_2|B_2 \wedge C_2, B_2|C_2 \in \mathcal{B} \times \mathcal{H}$ and suppose that $A_1|B_1 \wedge C_1 \preceq A_2|B_2 \wedge C_2$ and $B_1|C_1 \preceq B_2|C_2$.

If $A_1 \wedge B_1|C_1 \prec A_1|B_1 \wedge C_1$, then both the relations $A_1 \wedge B_1|C_1 \prec B_1|C_1$ and $B_1|C_1 \sim C_1|C_1$ cannot hold by hypotheses, so $A_1 \wedge B_1|C_1 \sim B_1|C_1$ and $B_1|C_1 \prec C_1|C_1$.

Then, one has the following situations:

- if $A_1 \wedge B_1|C_1 \prec A_1|B_1 \wedge C_1$ and $A_2 \wedge B_2|C_2 \prec A_2|B_2 \wedge C_2$, then

$$A_1 \wedge B_1|C_1 \sim B_1|C_1 \preceq B_2|C_2 \sim A_2 \wedge B_2|C_2;$$

- if $A_1 \wedge B_1|C_1 \prec A_1|B_1 \wedge C_1$ and $A_2 \wedge B_2|C_2 \sim A_2|B_2 \wedge C_2$, then

$$B_1|C_1 \sim A_1 \wedge B_1|C_1 \prec A_1|B_1 \wedge C_1 \preceq A_2|B_2 \wedge C_2 \sim A_2 \wedge B_2|C_2;$$

- if $A_1 \wedge B_1|C_1 \sim A_1|B_1 \wedge C_1$ and $A_2 \wedge B_2|C_2 \prec A_2|B_2 \wedge C_2$, then

$$A_1 \wedge B_1|C_1 \preceq B_1|C_1 \preceq B_2|C_2 \sim A_2 \wedge B_2|C_2;$$

- if $A_1 \wedge B_1|C_1 \sim A_1|B_1 \wedge C_1$ and $A_2 \wedge B_2|C_2 \sim A_2|B_2 \wedge C_2$, then

$$A_1 \wedge B_1|C_1 \sim A_1|B_1 \wedge C_1 \preceq A_2|B_2 \wedge C_2 \sim A_2 \wedge B_2|C_2.$$

The proof of the second implication of (A3) goes in the same line.

Definition 5. Let $\preceq$ be a binary relation on a set of conditional events $\mathcal{F}$. Then, $\preceq$ is a coherent comparative conditional possibility if and only if there exists a comparative conditional possibility on $\mathcal{B} \times \mathcal{H} \supset \mathcal{E}$, which extends $\preceq$.

Theorem 2. Let $\mathcal{F}$ be a finite set of conditional events. For a binary relation $\preceq$ on $\mathcal{F}$, the following statements are equivalent:

- i. $\preceq$ is a coherent comparative conditional possibility;
- ii. there exists a coherent conditional possibility Π on $\mathcal{F}$ representing $\preceq$.

Proof: We prove that ii implies i. Denote by the same symbol Π any conditional possibility extending Π on $\mathcal{B} \times \mathcal{H} \supset \mathcal{F}$, and let $\preceq$ be the binary relation induced by Π . Then, the conditions 1 and 2 of Definition 4 are trivially satisfied.

Since $\Pi(A \wedge B|H) = \min\{\Pi(A|B \wedge H), \Pi(B|H)\} \leq \Pi(A|B \wedge H)$, one has that if $\Pi(A \wedge B|H) < \Pi(B|H)$ (i.e. $A \wedge B|H \prec B|H$), then $\Pi(A \wedge B|H) = \Pi(A|B \wedge H)$ (i.e. $A \wedge B|H \sim A|B \wedge H$); on the other hand if $\Pi(B|H) = 1$ (i.e. $B|H \sim H|H$), then again $\Pi(A \wedge B|H) = \Pi(A|B \wedge H)$, hence condition 3 is satisfied.

Moreover, since for any $H \in \mathcal{B}^0$ the function $\Pi(\cdot|H)$ on $\mathcal{B}$ is a possibility, i.e.

$$\Pi(A|H) \leq \Pi(B|H) \Rightarrow \Pi(A \vee C|H) \leq \Pi(B \vee C|H)$$

for any $C \in \mathcal{B}$, then $\preceq$ satisfies (CPO), so $\preceq$ on $\mathcal{F}$ is a coherent comparative conditional possibility.

We prove now that i implies ii. Denote by the same symbol $\preceq$ any comparative conditional possibility extending $\preceq$ on $\mathcal{B} \times \mathcal{B}^0 \supseteq \mathcal{F}$.

Let $\mathcal{E}_0, \dots, \mathcal{E}_l$ be a class of subsets of conditional events such that for all $E_i|H_i \in \mathcal{E}_s$ and $E_j|H_j \in \mathcal{E}_k$ we have $E_i|H_i \prec E_j|H_j$, if $s < k$. By condition 2 of Definition 4 follows that $l \geq 1$.

In particular, $E_i|H_i \sim \emptyset|H_i$ if $E_i|H_i \in \mathcal{E}_0$, and $E_j|H_j \sim H_i|H_i$ if $E_j|H_j \in \mathcal{E}_l$.

Now, consider the following function defined by putting

$$f(E|H) = \frac{r}{l} \text{ if } E|H \in \mathcal{E}_r \text{ (with } r = 0, \dots, l \text{),}$$

which obviously represents $\preceq$. We need to prove that $f(\cdot|\cdot)$ is a conditional possibility on $\mathcal{E} = \mathcal{B} \times \mathcal{B}^0$.

First of all, $\emptyset|H \in \mathcal{E}_0$ and $H|H \in \mathcal{E}_l$ for any $H \in \mathcal{B}^0$ from condition 2 of Definition 4, and by construction $f(\emptyset|H) = 0$ and $f(H|H) = 1$. Notice that (CPO) implies $\emptyset|H \preceq E|H \preceq H|H$ and by construction $f(E|H) \in [0, 1]$.

Define the function Π_0 on $\mathcal{B}$ such that $\Pi_0(E) = f(E|\Omega)$ (since $\Omega \in \mathcal{B}^0$). The function $\Pi_0(\cdot)$ on $\mathcal{B}$ is a possibility, in fact (CPO) implies that if $E \preceq F$, then $F \sim (E \vee F)$, i.e. $\Pi_0(E \vee F) = \max\{\Pi_0(E), \Pi_0(F)\}$, for any $A, B \in \mathcal{B}$.

Then, $\Pi_0(\cdot)$ represents the ordering on $\{E|\Omega : E \in \mathcal{B}\}$ by construction.

Let $\mathcal{H}^0 = \{E \in \mathcal{B}^0 : \Pi_0(E) = 1\}$, note that $\mathcal{H}^0$ is an additive class. The possibility Π_0 on $\mathcal{B}$ defines uniquely the conditional possibility Π on $\mathcal{B} \times \mathcal{H}^0$; in fact, since for any event $H \in \mathcal{H}^0$ $\Pi_0(H) = 1$, $\Pi(A \wedge H|\Omega) = \Pi_0(A \wedge H) = \Pi(A|H)$ (by condition 3 of Definition 1).

By condition 3 of Definition 4 since for any event $H \in \mathcal{H}^0$ $H|\Omega \sim \Omega|\Omega$ one has that $A|H \sim A \wedge H|\Omega$, so Π represents $\preceq$ restricted to $\mathcal{B} \times \mathcal{H}^0$, and Π coincides with f on $\mathcal{B} \times \mathcal{H}^0$.

Let $\mathcal{C}^1 = \{C \in \mathcal{B}^o : C|\Omega \prec \Omega|\Omega\}$ and define $H^1 = \bigvee_{C \in \mathcal{C}^1} C$, by definition of $\mathcal{C}^1$ it follows that $H^1|\Omega \prec \Omega|\Omega$ (i.e. $\Pi_0(H^1) < 1$).

Take $\mathcal{H}_1 = \{C \in \mathcal{C}^1 : \exists D \in \mathcal{C}^1 \text{ s.t. } C|\Omega \prec D|\Omega\}$.

Consider the set of conditional events $\{C|H^1 : C \subseteq H^1 \text{ and } C \in \mathcal{C}^1\}$, from Proposition 1 it follows that there is an atom $C \in \mathcal{C}^1$ such that $C|H^1 \sim H^1|H^1$, and $C \in \mathcal{H}^1$ from Proposition 3.

For any $C \in \mathcal{C}^1 \setminus \mathcal{H}^1$ (i.e. $C|\Omega \prec H^1|\Omega$) Proposition 3 implies that $C|\Omega \sim C|H^1$, moreover note that $f(C|H^1)$ is the unique solution of the equation $\Pi_0(C) = \min\{x, \Pi_0(H^1)\}$.

On the other hand, for any $C \in \mathcal{H}^1$ the equation $\Pi_0(C) = \min\{x, \Pi(H^1)\}$ has no unique solution $x \geq \Pi(H^1)$.

Hence, define

$$\Pi_1(A) = \begin{cases} \Pi_0(A) & \text{if } A|\Omega \prec H^1|\Omega \\ \frac{r}{t} & \text{if } A|\Omega \sim H^1|\Omega \text{ and } A|H^1 \in \mathcal{E}_r \end{cases}$$

The function Π_1 on $\mathcal{B}$ is a possibility, moreover $f(A|H) = \Pi_1(A \wedge H)$ for any H such that $H|H^1 \sim H^1|H^1$.

In the same way we build $\mathcal{C}^2, \mathcal{H}^2$ and Π_2 and so on till $\mathcal{C}^k$ contains only the impossible event.

The class of possibilities $\{\Pi_1, \dots, \Pi_k\}$ is a nested class and it is agreeing with the function f , so f is a conditional possibility.

Example 1. Consider an area A , which is not earthquake zone, and let H = “there is an earthquake in A during this year”, E = “in the area A there is no earthquake victim”, F = “in the area A the earthquake victims are more than 10% of population”. We can consider the following relation

$$H \sim \emptyset, \quad E|H \prec F|H$$

the first relation implies that we believe very little in the possibility that an earthquake will be in the zone A , in fact we judge the event H indifferent to the impossible event. While the second relation implies that if we suppose that an earthquake will be in the zone A then we believe more on the fact that the earthquake victims are more than 10% than there is no earthquake victim, in fact a not earthquake zone is usually not equipped to face an earthquake.

The above relation is a coherent comparative conditional possibility and a conditional possibility representing $\preceq$ is, for example,

$$\begin{aligned} \Pi(E \wedge H) &= \Pi(F \wedge H) = \Pi(H) = 0, \\ \Pi(E|H) &= \frac{1}{5}, \quad \Pi(F|H) = \frac{3}{5}, \quad \Pi(E^c \wedge F^c|H) = 1. \end{aligned}$$

Note that the given relation cannot be represented by a conditional possibility in the sense of [13]. In fact, being $\Pi(E \wedge H) = \Pi(F \wedge H) = \Pi(H)$, both the

conditional possibilities on $E|H$ and $F|H$ must be, according to their definition, equal to 1, so the only agreeing relation $\preceq$ is $E|H \sim F|H$.

Our characterization is more general than that representable by conditional possibilities in the sense of [13]. To obtain a characterization for the relations representable by conditional possibilities (in the sense of [13]) is necessary to reinforce condition 3 of Definition 4 by requiring that $A \wedge B|H \sim B|H$ implies $A|B \wedge H \sim H|H$ (from this condition follows that the non-triviality condition can be valid only for conditioning events $H \in \mathcal{H}$ such that $\emptyset \prec H$).

References

1. N.Ben Amor, K.Mellouli, S.Benferhat, D.Dubois, H. Prade (2002) A theoretical framework for possibilistic independence in a weakly ordered setting. *Int. J. of Uncert., Fuzziness and Knowledge-Based Systems*, 10(2):117-155.
2. B. Bouchon-Meunier, G. Coletti, C. Marsala (2002) Independence and Possibilistic Conditioning. *Annals of Mathematics and Artificial Intelligence*, 35, pp. 107-124.
3. B. Bouchon-Meunier, G. Coletti, C. Marsala (2002). Conditional possibility and necessity. *Technologies for Constructing Intelligent Systems*, Bouchon-Meunier, Gutierrez-Rions, Magdalena, Yager (Eds.), Springer-Verlag, 59-71 (Selected papers from IPMU 2000).
4. A.Chateauneuf, R.Kast, A. Lapied (2001) Conditioning capacities and choquet integrals: the role of comonotony, *Theory and Decision* 51: 367-386.
5. G.Coletti, R.Scozzafava (2001) From conditional events to conditional measures: a new axiomatic approach. *Annals of Mathematics and Artificial Intelligence* 32, pp.373-392.
6. G. Coletti, R. Scozzafava (2002) *Probabilistic logic in a coherent setting*, Trends in logic n.15, Kluwer, Dordrecht/Boston/London.
7. G.Coletti, R.Scozzafava (2003) Toward a general theory of conditional beliefs, *Proc. of the 6th Workshop on Uncertainty Processing*, Hejnice, pp. 65-76.
8. G.Coletti, B. Vantaggi (2004) Independence in conditional possibility theory. *Proc. IPMU 2004*, Perugia, pp. 849-856.
9. G.Coletti, B. Vantaggi (2004) Representability of ordinal relations on a set of conditional events. Extended abstract of Conference FUR XI, Paris. An extended version has been submitted to Theory and Decision.
10. G. de Cooman (1997). Possibility theory II: conditional possibility. *Int. J. General Systems*, 25, pp.325-351.
11. B. de Finetti (1931) Sul significato soggettivo della probabilità. *Fundamenta Mathematicae* 17: 293-329.
12. D. Dubois (1986) Belief structure, possibility theory and decomposable confidence measures on finite sets. *Comput. Artificial Intelligence* 5, pp. 403-416.
13. D. Dubois, H. Prade (1988) *Possibility Theory* New York, Plenum Press.
14. E. Hisdal (1978) Conditional possibilities independence and noninteraction, *Fuzzy Sets and Systems*, 1, pp. 283-297.
15. E.P. Klement, R. Mesiar, E. Pap (2000) *Triangular Norms*. Dordrecht: Kluwer.
16. B. Rumbos (2001) Representing subjective orderings of random variables: an extension. *Journal of Mathematical Economics*, 36, pp. 31-43.
17. L.J.Savage (1954) *The Foundations of Statistics*. Wiley, New York.
18. L.A. Zadeh (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3-28.

Second-Level Possibilistic Measures Induced by Random Variables

Ivan Kramosil

Institute of Computer Science, Academy of Sciences of the Czech Republic,
Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic
`kramosil@cs.cas.cz`

Abstract. Given a real-valued random variable defined on a probability space $\langle \Omega, \mathcal{A}, P \rangle$ and given a subset A of the space Ω of all elementary random events, an $\omega_0 \in \Omega$ is called possibly favourable to A with respect to X , if it belongs to the subset A^X of Ω with this property: for every $\omega \in A^X$, $X(\omega) \leq \sup_{\omega_1 \in A} X(\omega_1)$ holds. The mapping Π ascribing to each $A \subset \Omega$ the value $P(A^X)$, i.e., the probability of the set of all elementary random events possibly favorable to A w.r.to X , defines a possibilistic measure on the power-set of all subsets of Ω . Having at hand two random variables X and Y defined on $\langle \Omega, \mathcal{A}, P \rangle$ and repeating our reasoning with A replaced by A^X and with X replaced by Y , we arrive at the idea of second-level possibilistic measures induced by random variables.

1 Introduction and Problem Formulation

Axiomatic probability theory defines random events as (certain, in general) subsets of an a priori introduced universe Ω of elementary random events and ascribes to these random events, as their probabilities, real numbers from the unit interval $[0, 1]$. The system $\mathcal{A}$ of random events is supposed to define a σ -field of subsets of Ω so that, for every $A, A_1, A_2, \dots$ from $\mathcal{A}$, also the sets $\Omega - A$ and $\bigcup_{i=1}^{\infty} A_i$ are in $\mathcal{A}$, i.e., are taken as random events. The mapping P which ascribes to each $A \in \mathcal{A}$ a value $P(A) \in [0, 1]$ is supposed to obey the demands imposed on σ -additive set functions (measures) so that it is consistent with the common intuition behind the idea of size of geometric entities (in particular, sets) known and accepted since the antic times. The triple $\langle \Omega, \mathcal{A}, P \rangle$ is called *probability space* and it is the basic stone on which the construction of the axiomatic probability theory is founded.

What is important when applying probability measure and probability theory as appropriate mathematical tools for uncertainty quantification and processing is the fact that the model re-called above admits an interpretation which is compatible with the intuition behind the most elementary combinatoric probability theory, but also with the idea to reduce uncertainty to the lack of complete knowledge of the actual value of a hidden parameter in a completely deterministic model. Given a probability space $\langle \Omega, \mathcal{A}, P \rangle$, a random event (in the informal sense) can be defined and processed within this framework if and only if each elementary random events $\omega \in \Omega$ is either in favor of the random event in question

or it is not in favor of this random event. Hence, at the level of mathematical formalization accepted in axiomatic probability theory, random event can be (and it is, as a matter of fact) identified with the (classical crisp) set of all elementary random events which are in favor of (or favorable to) the random event under consideration. Moreover, only subsets of Ω which are in $\mathcal{A}$ are taken as random events in order to be able to define their probabilities within the framework of the probability space $\langle \Omega, \mathcal{A}, P \rangle$.

Hence, the occurrence of a particular elementary random event $\omega_0 \in \Omega$ completely determines, beyond any doubts or uncertainty and for any random event $A \in \mathcal{A}$, whether this random event occurs (it is the case when $\omega_0 \in A$) or does not occur (if $\omega_0 \in \Omega - A$). Considering the trivial example of dice tossing: elementary random events can be identified with positive integers from 1 to 6 ($\Omega = \{1, 2, \dots, 6\}$, in symbols). So, e.g., the elementary random event “2” is in favor of the random event “even number”, but it is not in favor of the random event “number greater than 4”.

What is processed by probability measures, and probability theory in general, are just sets of elementary random events and classes of such sets, so that the problem how to decide, given a particular elementary random event $\omega_0 \in \Omega$ and a set $A \subset \Omega$, whether the relation $\omega_0 \in A$ does or does not hold, need not be and is not, as a matter of fact, considered and reflected within the framework of the axiomatic probability theory. However, in what follows, we focus our attention just to this decision problem supposing that it is not only far from being trivial, but that it is solvable only in some cases and only in a partial and negative sense. Namely, given a proper subset A of Ω , for no $\omega \in A$ we are able to decide that this is the case, the only we are able to do is that for *some* (but not for all, in general) elements $\omega \in \Omega - A$ we can decide that this ω indeed is not in A , i.e., that it is not an elementary random event in favor to A . In symbols, for each $A \subset \Omega$ there exists a subset $B_A \subset \Omega - A$ (obviously empty for $\Omega = A$ and perhaps also for some other $A \subset \Omega$) such that, for every $\omega \in B_A$, we are able to decide (check, verify) that $\omega \in A$ does not hold. All $\omega \in \Omega - B_A$ are called *elementary random events possibly favorable to A with respect to B_A* . The inclusion $A \subset \Omega - B_A$ is obvious, hence, each elementary random event favorable to A is also possibly favorable to A w.r.to B_A , so that our application of the adjective “possible” is consistent with the common intuition behind this word. In what follows, we will propose a mathematical model to describe this situation and to process the sketched problems. In particular, we will investigate the mapping $\Pi : \mathcal{A} \rightarrow [0, 1]$ which ascribes to each $A \in \mathcal{A}$ the value $P(\Omega - B_A)$ supposing that $B_A \in \mathcal{A}$ holds, and its possible extensions from $\mathcal{A}$ to $\mathcal{P}(\Omega)$ (the power-set of all subset of Ω).

2 Classification Systems Induced by Random Variables

In [9], the following simple model is introduced and analyzed. Let $\langle \Omega, \mathcal{A}, P \rangle$ be a probability space, let $\mathcal{S} = \langle S_0, S_1, S_2, \dots, S_n \rangle$ be a finite sequence of subsets of Ω such that each S_i is in $\mathcal{A}$ and the relation $\emptyset = S_0 \subset S_1 \subset S_2 \subset \dots \subset S_n = \Omega$

holds. Let us suppose that for each $0 \leq i \leq n$ and each $\omega \in \Omega - S_i$ we are able to verify this fact, i.e., to decide that this is indeed the case. Given $A \subset \Omega$, there exists just one $i(A)$, $0 \leq i(A) \leq n$, such that $A \subset S_{i(A)}$ holds, but $A \subset S_{i(A)-1}$ is not valid, hence, $i(A) = \min\{j : 0 \leq j \leq n, A \subset S_j\}$. So, if $\omega \in \Omega$ is not in S_j , it cannot be in A for each $j \geq i(A)$, so that the set $\Omega - S_j$ can be taken as the set B_A occurring in our informal considerations in the end of the foregoing chapter. Consequently, the set $\Omega - (\Omega - S_i) = S_i$ for $i = i(A)$, i.e., the set $S_{i(A)}$, is the set of all elementary random events possibly favorable to A with respect to $B_A = \Omega - S_{i(A)}$, in other terms, w.r.to the system $\mathcal{S}$ of “negatively decidable” or “refutable” sets from $\mathcal{A}$ (i.e., random events). This system $\mathcal{S}$ is called, in [9], *classification system*. Consequently, the value $\Pi(A) = P(S_{i(A)})$ is defined for every $A \subset \Omega$.

Referring the reader to [9] as far as a more detailed analysis of finite classification systems is concerned, let us focus our attention to infinite classification systems induced by real-valued random variables defined on a fixed probability space $\langle \Omega, \mathcal{A}, P \rangle$. To recall this notion, let $R = (-\infty, \infty)$ be the real line, let $\mathcal{B}$ be the σ -field of all Borel subsets of R , so that the measurable space $\langle R, \mathcal{B} \rangle$ is the well-known Borel line (let us note that $\mathcal{B}$ is the minimal σ -field of subsets of R containing all semi-open intervals $(-\infty, a)$, $-\infty < a < \infty$, where $(-\infty, a) = \{y \in R : -\infty < y < a\}$). A mapping $X : \Omega \rightarrow R$ is called a (*real-valued*) *random variable*, if it is $\mathcal{A} - \mathcal{B}$ -measurable, i.e., if the inverse image of each Borel set is in $\mathcal{A}$, in symbols, if the inclusion

$$\{\{\omega \in \Omega : X(\omega) \in B\} : B \in \mathcal{B}\} \subset \mathcal{A} \quad (2.1)$$

holds, as a matter of fact, the inclusion

$$\{\{\omega \in \Omega : X(\omega) < a\} : -\infty < a < \infty\} \subset \mathcal{A} \quad (2.2)$$

is sufficient. In the definition of $\mathcal{B}$ and in (2.2) $X(\omega) < a$ can be equivalently replaced by $X(\omega) \leq a$.

Let X be a real-valued random variable defined on the probability space $\langle \Omega, \mathcal{A}, P \rangle$. The system $\mathcal{S}$ (or $\mathcal{S}_X$, to explicitate the role of X) defined by $\mathcal{S} = \mathcal{S}_X = \{S_x : -\infty \leq x \leq \infty\}$, where $S_x = \{\omega \in \Omega : X(\omega) \leq x\}$ for $-\infty < x < \infty$, $S_{-\infty} = \emptyset$ (the empty subset of Ω) and $S_{\infty} = \Omega$, is called the *classification system induced by random variable X* . Obviously, for each x_1, x_2 , $-\infty \leq x_1 \leq x_2 \leq \infty$, the inclusion $S_{x_1} \subset S_{x_2}$ holds, so that, denoting by R^* the set $R \cup \{-\infty, \infty\}$, the subset $\{x \in R^*, A \subset S_x\}$ of R^* is an interval for each $A \subset \Omega$ and we can define

$$i(A) = \bigwedge \{x \in R^* : A \subset S_x\}; \quad (2.3)$$

here and below $\bigwedge$ denotes the infimum and $\bigvee$ the supremum in R^* w.r.to the standard linear ordering of real numbers. As can be easily proved, the relation

$$S_{i(A)} = \bigcap_{A \subset S_x} S_x \quad (2.4)$$

is valid for each $A \subset \Omega$, so that $S_{i(A)}$ is the minimum (w.r.to set inclusion) set of elementary random events from Ω possibly favorable to A w.r.to the classification system $\mathcal{S}$.

The value $i(A)$, also denoted by $i(A, \mathcal{S})$, $i(A, \mathcal{S}_X)$ or $i(A, X)$, can be easily explicitly defined. Indeed,

$$\begin{aligned} i(A, \mathcal{S}) &= \bigwedge \{-\infty \leq x \leq \infty : A \subset S_x\} = \\ &= \bigwedge \{-\infty \leq x \leq \infty : A \subset \{\omega \in \Omega : X(\omega) \leq x\}\} = \\ &= \bigwedge \{-\infty \leq x \leq \infty : (\forall \omega \in A)(X(\omega) \leq x)\} = \\ &= \bigwedge \{-\infty \leq x \leq \infty : \bigvee(A, X) \leq x\} = \bigvee(A, X), \end{aligned} \quad (2.5)$$

where $\bigvee(A, X) = \bigvee_{\omega \in A} X(\omega)$ denotes the supremum of values taken by X on A . Consequently, for any $A \subset \Omega$,

$$\begin{aligned} \Pi(A, \mathcal{S}) &= P(S_{\bigvee(A, X)}) = \\ &= P(\{\omega \in \Omega : X(\omega) \leq \bigvee(A, X)\}) = \\ &= F_X(\bigvee(A, X)), \end{aligned} \quad (2.6)$$

where $F_X : \mathbb{R} \rightarrow [0, 1]$ is the distribution function of the random variable X . So, if the supremum value of X is reached on A , i.e., if $\bigvee(A, X) = \bigvee(\Omega, X)$ under our notation, then $\Pi(A, \mathcal{S}) = 1$. As will be shown and analyzed in more detail below, the mapping $\Pi : \mathcal{P}(\Omega) \rightarrow [0, 1]$, defined by (2.6), meets the demands imposed on the so called real-valued possibilistic measures.

Combining the finite, but abstract classification systems from [9], briefly sketched in the introductory paragraph of this chapter, with infinite, but in a special way defined classification systems induced by random variables, we arrive at the following idea.

Let $\langle \Omega, \mathcal{A}, P \rangle$ be a probability space. A system $\mathcal{S} = \{S_t : -\infty \leq t \leq \infty\}$ is called a *general classification system* over $\langle \Omega, \mathcal{A}, P \rangle$, if $\mathcal{S} \subset \mathcal{A}$, $S_{-\infty} = \emptyset$, $S_{\infty} = \Omega$, and if the inclusion $S_{t_1} \subset S_{t_2}$ holds for each $-\infty < t_1 \leq t_2 < \infty$, i.e., if the system $\mathcal{S}$ is nested w.r.to the standard linear ordering in $(-\infty, \infty)$. The mapping Π (or $\Pi_{\mathcal{S}}$, to explicitate the role of $\mathcal{S}$) is defined, for each $A \subset \Omega$, by

$$\Pi(A) = P(S_{i(A)}), \quad (2.7)$$

where,

$$i(A) = \bigwedge \{t \in (-\infty, \infty) : A \subset S_t\}. \quad (2.8)$$

Obviously, $i(A)$ and $\Pi(A)$ are defined for each $A \subset \Omega$, so that Π takes $\mathcal{P}(\Omega)$ into $[0, 1]$. The general classification system $\mathcal{S}$ is called *continuous from above*, if the relation.

$$S_t = \bigcap \{S_s : -\infty < s \leq \infty, s > t\} \quad (2.9)$$

holds for each $-\infty \leq t \leq \infty$.

A nonempty set $\mathcal{H}$ of real random variables defined on $\langle \Omega, \mathcal{A}, P \rangle$ is called *consistent for classification w.r.to intersection (w.r.to union, resp.)*, if the system $\mathcal{S} = \{S_t : -\infty \leq t \leq \infty\}$ of subsets of Ω such that $S_{-\infty} = \emptyset, S_{\infty} = \Omega$, and

$$S_t = \bigcap_{X \in \mathcal{H}} \{\omega \in \Omega : X(\omega) \leq t\} \quad (2.10)$$

$$(S_t = \bigcup_{X \in \mathcal{H}} \{\omega \in \Omega : X(\omega) \leq t\}, \text{ resp.}) \quad (2.11)$$

defines a general classification system over $\langle \Omega, \mathcal{A}, P \rangle$.

3 Classification Systems and Possibilistic Measures

A nonempty system $\mathcal{R}$ of subsets of a nonempty set Ω is called *ample field*, if for each $A \in \mathcal{R}$ and each nonempty subsystem $\mathcal{R}_0 \subset \mathcal{R}$ the sets $\Omega - A$ and $\bigcup \mathcal{R}_0 (= \bigcup_{A \in \mathcal{R}_0} A)$ are also in $\mathcal{R}$. Consequently, due to de Morgan rules, also the set $\bigcap \mathcal{R}_0 (= \bigcap_{A \in \mathcal{R}_0} A)$ is in $\mathcal{R}$. Each ample field $\mathcal{R} \subset \mathcal{P}(\Omega)$ contains the empty set $\emptyset$ and Ω . The systems $\{\emptyset, \Omega\}$ and $\mathcal{P}(\Omega)$ are the most simple, and in the sense of inclusion extremal, examples of ample fields over Ω .

Definition 1. Let Ω be a nonempty set, let $\mathcal{R}$ be an ample field of subsets of Ω . A mapping $\Pi : \mathcal{R} \rightarrow [0, 1]$ is called (real-valued) *possibilistic measure* on $\mathcal{R}$, if $\Pi(\emptyset) = 0$, $\Pi(\Omega) = 1$, and if the equality $\Pi(A \cup B) = \Pi(A) \vee \Pi(B)$ holds for each $A, B \in \mathcal{R}$. The possibilistic measure Π on $\mathcal{R}$ is *complete*, if the equality $\Pi(\bigcup \mathcal{R}_0) = \bigvee \{\Pi(A) : A \in \mathcal{R}_0\}$ holds for each nonempty subsystem $\mathcal{R}_0 \subset \mathcal{R}$. The triple $\langle \Omega, \mathcal{A}, \Pi \rangle$ is then called *possibilistic space*.

Theorem 1. Let $\langle \Omega, \mathcal{A}, P \rangle$ be a probability space, let X be a real-valued random variable defined on $\langle \Omega, \mathcal{A}, P \rangle$, let $\Pi : \mathcal{P}(\Omega) \rightarrow [0, 1]$ be the mapping defined by (2.6). Then Π is a complete possibilistic measure on $\mathcal{P}(\Omega)$.

Proof. The constraints for $\emptyset$ and Ω can be easily verified. Indeed,

$$\begin{aligned} \Pi(\emptyset) &= P(\{\omega \in \Omega : X(\omega) \leq \bigvee(\emptyset, X)\}) = \\ &= P(\{\omega \in \Omega : X(\omega) \leq -\infty\}) = P(\emptyset) = 0, \end{aligned} \quad (3.1)$$

applying the convention according to which $\bigvee\{X(\omega) : \omega \in \emptyset\} = -\infty$, and

$$\Pi(\Omega) = P(\{\omega \in \Omega : X(\omega) \leq \bigvee(\Omega, X)\}) = P(\Omega) = 1. \quad (3.2)$$

Let $\mathcal{A}$ be a nonempty system of subsets of Ω . An easy calculation yields that, for $\bigcup \mathcal{A} = \bigcup_{A \in \mathcal{A}} A$,

$$\begin{aligned}
 \Pi\left(\bigcup \mathcal{A}\right) &= P(\{\omega \in \Omega : X(\omega) \leq \bigvee (\bigcup \mathcal{A}, X)\}) = \\
 &= P(\{\omega \in \Omega : X(\omega) \leq \bigvee_{A \in \mathcal{A}} (\bigvee(A, X))\}) = \\
 &= P(\bigcup_{A \in \mathcal{A}} \{\omega \in \Omega : X(\omega) \leq \bigvee(A, X)\}) = \\
 &= \bigvee_{A \in \mathcal{A}} P(\{\omega \in \Omega : X(\omega) \leq \bigvee(A, X)\}) = \bigvee_{A \in \mathcal{A}} \Pi(A), \quad (3.3)
 \end{aligned}$$

as the sets $\{\omega \in \Omega : X(\omega) \leq \bigvee(A, X)\}$ are nested w.r.to the standard linear ordering of the values $\bigvee(A, X), A \in \mathcal{A}$, as real numbers. The assertion is proved.

A weakened version of Theorem 1 can be proved also in the case of continuous from above generalized classification systems, even if the proof becomes slightly more complicated, as the values of Π cannot be defined directly by the distribution functions of the random variables in question.

Theorem 2. *Let $\langle \Omega, \mathcal{A}, P \rangle$ be a probability space, let $R^* = [-\infty, \infty]$, let $\mathcal{S} = \{S_t : t \in R^*\}$ be a continuous from above general classification system. Then Π is a possibilistic measure on $\mathcal{P}(\Omega)$.*

Proof. If $A = \emptyset$, the inclusion $A \subset S_t$ holds for each $t \in R^*$, so that $i(A) = -\infty, S_{i(A)} = \emptyset$ and $\Pi(\emptyset) = P(\emptyset) = 0$. If $A = \Omega$, then $A \subset S_t$ holds iff $S_t = \Omega$, so that for $i(A)$ we obtain, due to the continuity from above, that

$$S_{i(A)} = \bigcap \{S_t : t > i(A)\} = \Omega, \quad (3.4)$$

as $A_t = \Omega$ for all $t > i(A)$. Hence, $\Pi(\Omega) = P(\Omega) = 1$.

Let us prove that, for each $A, B \subset \Omega$, $i(A \cup B) = i(A) \vee i(B)$. Indeed, for both $C = A, B$ the inequality

$$i(A \cup B) = \bigwedge \{t \in R^* : A \cup B \subset S_t\} \geq \bigwedge \{t \in R^* : C \subset S_t\} = i(C), \quad (3.5)$$

consequently, also the inequality $i(A \cup B) \geq i(A) \vee i(B)$ obviously hold. Suppose that the strict inequality is the case, i.e., that $i(A \cup B) > t_0 \geq i(A) \vee i(B)$ holds for some t_0 . Then $A \subset S_{t_0}$ and $B \subset S_{t_0}$ follows, hence, $A \cup B \subset S_{t_0}$ and $i(A \cup B) \leq t_0$ result – a contradiction. So, $i(A \cup B) = i(A) \vee i(B)$ and we may suppose, without any loss of generality, that $i(A \cup B) = i(A)$. Consequently, $i(A) \geq i(B), S_{i(A)} \supset S_{i(B)}$, and

$$\Pi(A) = P(S_{i(A)}) \geq P(S_{i(B)}) = \Pi(B) \quad (3.6)$$

follows. Hence,

$$\begin{aligned}
 \Pi(A \cup B) &= P(S_{i(A \cup B)}) = P(S_{i(A) \vee i(B)}) = \\
 &= P(S_{i(A)}) = \Pi(A) = \Pi(A) \vee \Pi(B). \quad (3.7)
 \end{aligned}$$

The assertion is proved.

4 Second-Level Possibilistic Measures Induced by Two Random Variables

In [9], Chapter 6, we analyzed the situation when we have at our disposal two classification systems $\mathcal{S}_X$ and $\mathcal{S}_Y$, induced by real-valued random variables X and Y , both defined on the same probability space $\langle \Omega, \mathcal{A}, P \rangle$. Our aim is to take profit of both these classification systems in order to specify the set of elementary random events possibly favorable to a given subset $A \subset \Omega$, using as mathematical tools only the possibilistic measures Π_X and Π_Y induced by the random variables under consideration.

Here we will try to approach the same problem from a different point of view applying, at a higher level, the idea that not only the actual elementary random events are recognizable only partially and in the negative sense, but that the same is the case with possibly favorable elementary random events. Hence, given a subset $A \subset \Omega$ and a random variable X defined on the probability space $\langle \Omega, \mathcal{A}, P \rangle$, we are not able to decide that some $\omega \in \Omega$ is possibly favorable to $A \subset \Omega$ with respect to the classification system $\mathcal{S}_X$, only for *some* (but not for all, in general) elementary random events which are not possibly favorable to A we are able to decide (check, verify) that this is the case, i.e., that they are not possibly favorable to A .

So, let $\langle \Omega, \mathcal{A}, P \rangle$ be a probability space, let X and Y be real-valued random variables defined on $\langle \Omega, \mathcal{A}, P \rangle$, let $\mathcal{S}_X$ and $\mathcal{S}_Y$ be the classification systems induced by X and Y , let $A \subset \Omega$. Recalling that $\bigvee(A, X)$ denotes the supremum of the values taken by X on A , denote by A^X the set

$$A^X = \{\omega \in \Omega : X(\omega) \leq \bigvee(A, X)\}, \quad (4.1)$$

so that the inclusion $A \subset A^X$ trivially holds. Hence, due to the notations used and results obtained above,

$$\Pi_X(A) (= \Pi_{\mathcal{S}_X}(A)) = P(A^X) \quad (4.2)$$

holds for every $A \subset \Omega$. Repeating the same consideration with A replaced by A^X and with X replaced by Y , we obtain that

$$(A^X)^Y = \{\omega \in \Omega : Y(\omega) \leq \bigvee(A^X, Y)\} \quad (4.3)$$

and

$$\Pi_Y(A^X) = P((A^X)^Y) = P(\{\omega \in \Omega : Y(\omega) \leq \bigvee(A^X, Y)\}). \quad (4.4)$$

The value $\Pi_Y(A^X)$ will be denoted by $\Pi_{XY}(A)$ and will be called the *second-level possibilistic measure induced on $\mathcal{P}(\Omega)$ by the ordered pair $\langle X, Y \rangle$ of random variables*. Obviously, Π_{XY} is defined for each $A \subset \Omega$, but the justification of the adjective “possibilistic” remains to be proved.

Theorem 3. Let X, Y be random variables defined on a probability space $\langle \Omega, \mathcal{A}, P \rangle$, let $\Pi_{XY} : \mathcal{P}(\Omega) \rightarrow [0, 1]$ be defined by (4.4). Then (i) Π_{XY} is a possibilistic measure on $\mathcal{P}(\Omega)$, (ii) $\Pi_{XX}(A) = \Pi_X(A)$ for any $A \subset \Omega$, and (iii) the inequalities $\Pi_{XY}(A) \geq \Pi_X(A)$, $\Pi_{XY}(A) \geq \Pi_Y(A)$ hold for each $A \subset \Omega$.

Proof. For each $A, B \subset \Omega$ we obtain that

$$\begin{aligned} (A \cup B)^X &= \{\omega \in \Omega : X(\omega) \leq \bigvee((A \cup B), X)\} = \\ &= \{\omega \in \Omega : X(\omega) \leq \bigvee(A, X) \vee \bigvee(B, X)\} = \\ &= \{\omega \in \Omega : X(\omega) \leq \bigvee(A, X)\} \cup \{\omega \in \Omega : X(\omega) \leq \bigvee(B, X)\} = \\ &= A^X \cup B^X. \end{aligned} \quad (4.5)$$

As Π_Y is a possibilistic measure on $\mathcal{P}(\Omega)$ (cf. Theorem 1), we obtain that

$$\begin{aligned} \Pi_{XY}(A \cup B) &= \Pi_Y((A \cup B)^X) = \Pi_Y(A^X \cup B^X) = \\ &= \Pi_Y(A^X) \vee \Pi_Y(B^X) = \Pi_{XY}(A) \vee \Pi_{XY}(B) \end{aligned} \quad (4.6)$$

and (i) is proved. As can be easily seen, $\bigvee(A^X, X) = \bigvee(A, X)$ holds for each $A \subset \Omega$, so that

$$\begin{aligned} \Pi_{XX}(A) &= P(\{\omega \in \Omega : X(\omega) \leq \bigvee(A^X, X)\}) = \\ &= P(\{\omega \in \Omega : X(\omega) \leq \bigvee(A, X)\}) = \Pi_X(A) \end{aligned} \quad (4.7)$$

and (ii) is also proved.

For each $A \subset \Omega$ the inclusions $A \subset A^X, A \subset A^Y$, hence, also the inclusion $A^X \subset (A^X)^Y$ obviously hold, so that the inequality $\bigvee(A^X, Y) \geq \bigvee(A, Y)$ and the inclusion

$$\{\omega \in \Omega : Y(\omega) \leq \bigvee(A, Y)\} \subset \{\omega \in \Omega : Y(\omega) \leq \bigvee(A^X, Y)\} \quad (4.8)$$

easily follow. So, we obtain that

$$\begin{aligned} \Pi_{XY}(A) &= P(\{\omega \in \Omega : Y(\omega) \leq \bigvee(A^X, Y)\}) \geq \\ &\geq P(\{\omega \in \Omega : Y(\omega) \leq \bigvee(A, Y)\}) = \Pi_Y(A) \end{aligned} \quad (4.9)$$

holds. The inclusion $A^X \subset (A^X)^Y$ yields that the inequality

$$\Pi_{XY}(A) = P((A^X)^Y) \geq P(A^X) = \Pi_X(A) \quad (4.10)$$

is also valid and (iii) holds. The proof is completed.

Let us note that the proof of the relation (4.5) above cannot be extended to infinite systems $\mathcal{A}_0$ of subsets of Ω . In this case, only the inclusion

$$\bigcup_{A \in \mathcal{A}_0} \{\omega \in \Omega : X(\omega) \leq \bigvee(A, X)\} \subset \{\omega \in \Omega : X(\omega) \leq \bigvee_{A \in \mathcal{A}_0} (\bigvee(A, X))\}, \quad (4.11)$$

hence, the inclusion

$$\bigcup_{A \in \mathcal{A}_0} A^X \subset (\bigcup \mathcal{A}_0)^X, \quad (4.12)$$

and the inequality

$$\bigvee_{A \in \mathcal{A}_0} \Pi_{XY}(A) \leq \Pi_{XY}(\bigcup \mathcal{A}_0) \quad (4.13)$$

can be proved. Indeed, if $\mathcal{A}_0$ is infinite, it is possible that for some $\omega_0 \in \Omega$ the equality $X(\omega_0) = \bigvee_{A \in \mathcal{A}_0} (\bigvee(A, X))$, but also the inequalities $X(\omega_0) > \bigvee(A, X)$ for each $A \in \mathcal{A}_0$ hold together. Modifying our definition of $\Pi_X(A)$ by setting $\Pi_X(A) = P(\{\omega \in \Omega : X(\omega) < \bigvee(A, X)\})$ (instead of $\leq \bigvee(A, X)$, as defined above) we would solve the problem of completeness for Π_{XY} , but the classification system $\mathcal{S}_X$ would not be continuous from above, so that our former constructions and assertions would be menaced.

Recalling the semantics behind the idea of possibly favorable elementary random events we can easily see that the best approximation of a set $A \subset \Omega$ by possibly favorable elementary random events induced by two random variables X, Y defined on the probability space $\langle \Omega, \mathcal{A}, P \rangle$ would be the intersection $A^X \cap A^Y$, and the most reasonable and intuitive numerical quantification of this set would be the probability value $P(A^X \cap A^Y)$. A problem is that this reasoning and formalization cannot be embedded into the framework of possibilistic measures and nested classification systems induced by the random variables in question. The next example illustrates the case when the discrepancy between the values $P(A^X \cap A^Y)$ and $\Pi_{XY}(A)$ (or $\Pi_{YX}(A)$) is the most remarkable.

Let $\Omega = R = (-\infty, \infty)$, let $\mathcal{B} = \mathcal{A}$ be the system of all Borel subsets of R , so that $\langle \Omega, \mathcal{A} \rangle$ is the Borel line $\langle R, \mathcal{B} \rangle$, let P be a probability measure on $\mathcal{A}$. Let X be the identity on R , so that $X(\omega) = \omega$ for each $\omega \in R$, let $Y(\omega) = -X(\omega) = -\omega$ for each $\omega \in R$, let $A = [a, b]$ be a closed interval of real numbers such that $a < b$ holds. As can be easily observed, $\bigvee(A, X) = b$ and $A^X = \{\omega \in \Omega : X(\omega) \leq b\} = (-\infty, b]$, so that $\Pi_X(A) = P(A^X) = P((-\infty, b])$. For Y we obtain that $\bigvee(A, Y) = \bigvee\{-\omega : a \leq \omega \leq b\} = -a$, so that $A^Y = \{\omega \in \Omega : -\omega \leq -a\} = \{\omega \in \Omega : \omega \geq a\} = [a, \infty)$ and $\Pi_Y(A) = P([a, \infty))$. In this case, $A^X \cap A^Y = (-\infty, b] \cap [a, \infty) = [a, b] = A$, so that the set A is completely defined by the intersection of the sets of elementary random events possibly favorable to A w.r.to X and w.r.to Y . On the other side, when using only the second-level possibly favourable elementary random events, i.e., the sets $(A^X)^Y$ and/or $(A^Y)^X$, all the information concerning the event A disappears. Indeed,

$$\begin{aligned} \bigvee(A^X, Y) &= \bigvee\{-\omega : \omega \in (-\infty, b]\} = \infty, \\ \bigvee(A^Y, X) &= \bigvee\{\omega : \omega \in [a, \infty)\} = \infty, \end{aligned} \quad (4.14)$$

so that

$$\begin{aligned}\bigvee (A^X)^Y &= \{\omega \in \Omega : Y(\omega) \leq \infty\} = \{\omega \in \Omega : -\omega \leq \infty\} = \Omega = \\ &= \{\omega : \omega \leq \infty\} = \{\omega : X(\omega) \leq \infty\} = (A^Y)^X,\end{aligned}\quad (4.15)$$

hence, $\Pi_{XY}(A) = \Pi_{YX}(A) = 1$.

The mapping $\Pi_{XY} : \mathcal{P}(\Omega) \rightarrow [0, 1]$ is not, in general, commutative in X and Y , i.e., the sets $(A^X)^Y$ and $(A^Y)^X$, as well as the values $\Pi_{XY}(A)$ and $\Pi_{YX}(A)$, may differ for some $A \subset \Omega$. Indeed, let us consider the following example.

Let $\Omega = [0, 1]$, let $\mathcal{A}$ be the system of all Borel subsets of $[0, 1]$, let P be the uniform probability measure on $\mathcal{A}$, so that $P([a, b]) = b - a$ for each interval in $[0, 1]$. Let $X(\omega) = \omega$ for each $\omega \in \Omega$, let $Y(\omega) = 2\omega$, if $\omega \leq 1/2$ holds and $Y(\omega) = 2 - 2\omega$, if $1/2 \leq \omega \leq 1$ holds, hence, the graph of Y defines a triangle shape in $[0, 1] \times [0, 1]$ with the tops in $\langle 0, 0 \rangle$, $\langle 0, 1 \rangle$ and $\langle 1/2, 1 \rangle$. Let $A = [0, b]$ with $b < 1/2$. Then

$$\bigvee (A, X) = \bigvee \{\omega : \omega \in [0, b]\} = b, \quad (4.16)$$

so that $A^X = [0, b] = A$, and

$$\bigvee (A, Y) = \bigvee (A^X, Y) = \bigvee \{2\omega : \omega \leq b\} = 2b, \quad (4.17)$$

as $b < 1/2$ holds. Consequently,

$$(A^X)^Y = A^Y = [0, b] \cup [1 - b, 1], \quad (4.18)$$

as $A^Y = \{\omega \in \Omega : Y(\omega) \leq 2b\} = [0, b] \cup [1 - b, 1]$, so that $\Pi_{XY}(A) = P((A^X)^Y) = P([0, b] \cup [1 - b, 1]) = 2b < 1$.

On the other side,

$$\bigvee (A^Y, X) = \bigvee \{\omega : \omega \in [0, b] \cup [1 - b, 1]\} = 1, \quad (4.19)$$

so that

$$(A^Y)^X = \{\omega \in \Omega : \omega \leq 1\} = \Omega \quad (4.20)$$

and

$$\Pi_{YX}(A) = P(\Omega) = 1 > \Pi_{XY}(A) \quad (4.21)$$

immediately follow.

The construction of the sets $(A^X)^Y$ and the values $P_{XY}(A)$ can be easily generalized to a finite sequence $X_1, X_2, \dots, X_n$ of random variables defined on the probability space $\langle \Omega, \mathcal{A}, P \rangle$ under consideration. Indeed, given $A \subset \Omega$ we define A^{X_1} as above and we denote by $A^{X_1 X_2}$ the set $(A^{X_1})^{X_2}$, also defined above. By induction, having already defined $A^{A_1 \dots X_{n-1}}$, we define $A^{X_1 \dots X_n}$ by $(A^{X_1 \dots X_{n-1}})^{X_n}$ and we set $\Pi_{X_1 \dots X_n}(A) = P(A^{X_1 \dots X_n})$ for each $A \subset \Omega$. As can

be easily seen, the mapping $\Pi_{X_1 \dots X_n}$ defines a possibilistic measure on $\mathcal{P}(\Omega)$ (not necessarily complete) and the inequality $\Pi_{X_1 \dots X_{n-1}}(A) \leq \Pi_{X_1 \dots X_n}(A)$ holds for each $A \subset \Omega$ and each $n \geq 2$. However, let us postpone a more detailed investigation of these possibilistic measures till another occasion.

This contribution has been purposely conceived at an almost self-explanatory level, so that the references listed-below may be of use rather for the reader asking for more detail in the fields of applied mathematics used above in the role of formal tools. For probability theory, the already classical monographs [2] and [10] deal with the abstract mathematical features of the (Kolmogorov) axiomatic probability theory, while [5, 6] and [8] analyze rather the philosophical (ontological and gnoseological) aspects of probabilities and probability theories under various settings. Because of the fact that axiomatic probability theory is settled as a particular case of measure theory, the well-known and also already classical monograph [7] can be worth being introduced explicitly.

Real-valued normalized possibilistic (or possibility) measures were introduced in [11] and analyzed in more detail in [3]. Their generalizations to lattice-valued possibilistic measures were introduced and excellently investigated in [1]. In [4], some relations among the three well-known mathematical tools for uncertainty quantification and processing (probability, possibility and fuzziness) are analyzed and discussed at a surveyal level.

The notions, ideas and results borrowed from [9] are introduced also here in the extent enabling the reader to understand the text above without having been familiar with [9] before.

The author is indebted to an anonymous reviewer for valuable and deeply going remarks and comments which will serve as a useful motivation and inspiration for author's further research work. However, the very limited extent of this contribution does not allow to present, analyze and develop these ideas at a sufficiently detailed level. Some of the reviewer's remarks, comments and suggestions are also touched in [9].

Acknowledgement. This work has been sponsored by the grant Cost Action (TARSKI) No. OC274.001.

References

1. De Cooman, G.: Possibility theory I-III. *International Journal of General Systems*, 25 (1997), pp. 291-323, pp. 325-351, pp. 353-371.
2. Doob, J. L.: *Stochastic Processes*. John Wiley and Sons, New York, 1953.
3. Dubois, D., Prade, H.: *Théorie des Possibilités – Applications à la Représentation des Connaissances en Informatique*. Mason, Paris, 1985.
4. Dubois, D., Nguyen, H., Prade, H.: Possibilist theory, probability theory and fuzzy sets: misunderstandings, bridges and gaps. In: Dubois, D. and Prade, H. (Eds.): *The Handbook of Fuzzy Sets Series*, Kluwer Academic Publishers, Boston, 2000, pp. 343-438.
5. Fine, T. L.: *Theories of Probability – An Examination of Foundations*. Academic Press, New York, 1973.

6. Gillies, D.: *Philosophical Theories of Probability*. Routledge, London and New York, 2000.
7. Halmos, P. R.: *Measure Theory*. D. van Nostrand, New York–Toronto–London, 1950.
8. Khrennikov, A.: *Interpretations of Probability*. VSP, Utrecht-Tokyo, 1999.
9. Kramosil, I.: Possibilistic measures and possibly favorable elementary random events. *Neural Network World* 15, 2 (2005), pp. 85-109.
10. Lo  ve, M.: *Probability Theory*. D. van Nostrand, New York–Toronto–London, 1960.
11. Zadeh, L. A.: Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* 1 (1978), pp. 3-28.

Hybrid Bayesian Estimation Trees Based on Label Semantics

Zengchang Qin and Jonathan Lawry

A.I. Group, Department of Engineering Mathematics,
University of Bristol, Bristol BS8 1TR, U.K
{z.qin, j.lawry}@bristol.ac.uk

Abstract. Linguistic decision tree (LDT) [7] is a classification model based on a random set based semantics which is referred to as label semantics [4]. Each branch of a trained LDT is associated with a probability distribution over classes. In this paper, two hybrid learning models by combining linguistic decision tree and fuzzy Naive Bayes classifier are proposed. In the first model, an unlabelled instance is classified according to the Bayesian estimation given a single LDT. In the second model, a set of disjoint LDTs are used as Bayesian estimators. Experimental studies show that the first new hybrid models has both better accuracy and transparency comparing to fuzzy Naive Bayes and LDTs at shallow tree depths. The second model has the equivalent performance to the LDT model.

1 Introduction

Tree induction algorithms have received a great deal of attention because of their simplicity and effectiveness. There are many decision tree models and tree induction algorithms that been proposed. These range from early discrete decision trees such as ID3 [9] and C4.5 [10] to a variety of fuzzy decision trees. *Linguistic decision tree* (LDT) is a tree-structured model based on a high level knowledge representation framework which is referred to as *Label semantics* [4]. Linguistic expressions (or fuzzy labels) such as *small*, *medium* and *large* are used to build a tree guided by information based heuristics. For each branch, the probability of this branch belonging to a particular class is evaluated based on the proportion of data in this class relative to all the data covered by the linguistic expressions of the branch. Therefore, LDT model can be regarded as a probability estimation tree model based on fuzzy labels. The LDT model has been shown to be an effective model for both classification and prediction. Also a LDT can be represented as a set of linguistic rules and hence provides a high level transparency. However, for complex problems, good probability estimations can only be obtained by deep LDTs, which are not good in terms of transparency. In such cases, how can we build a model which has a good probability estimation with compact LDTs (i.e. LDTs with shallow depths or with less number of branches)? This question motivates the research presented in this paper.

Naive Bayes is a well known and much studied algorithm in machine learning. It is a simple, effective and efficient learning method. Although Naive Bayes classification makes the unrealistic assumption that the values of the attributes of an instance are conditionally independent given the class of the instance, this model is remarkably successful in practice. In this paper, an extended version of Naive Bayes based on label semantics is introduced. The new hybrid models using Naive Bayes classification given a single LDT and a set of disjoint LDTs are proposed and tested on a number of UCI datasets [2].

2 Label Semantics

Label semantics [4] is a framework to represent the use of linguistic expressions to label a value. The underlying question posed by label semantics is how to use linguistic expressions to label numerical values. For a variable x into a domain of discourse Ω we identify a finite set of linguistic labels $\mathcal{L} = \{L_1, \dots, L_n\}$ with which to label the values of x . Then for a specific value $\alpha \in \Omega$ an individual I identifies a subset of $\mathcal{L}$, denoted D_α^I to stand for the description of α given by I , as the set of words with which it is appropriate to label α . If we allow I to vary across a population V , then D_α^I will also vary and generate a random set denoted D_α into the power set of $\mathcal{L}$. The frequency of occurrence of a particular label, say S , for D_α across the population then gives a distribution on D_α referred to as a mass assignment on labels, more formally:

Definition 1 (*Mass Assignment on Labels*)

$$\forall S \subseteq \mathcal{L}, \quad m_x(S) = \frac{|\{I \in V | D_x^I = S\}|}{|V|}$$

For example, given a set of labels defined on the temperature outside: $\mathcal{L}_{Temp} = \{low, medium, high\}$. Suppose 3 of 10 people agree that ‘medium is the only appropriate label for the temperature of 15° and 7 agree ‘both low and medium are appropriate labels’. According to def. 1, $m_{15}(medium) = 0.3$ and $m_{15}(low, medium) = 0.7$ so that the mass assignment for 15° is $m_{15} = \{medium\} : 0.3, \{low, medium\} : 0.7$. More details about the theory of mass assignment can be found in [1].

Consider the previous example, can we know how appropriate for a single label, say low, to describe 15°? In this framework, *appropriateness degrees* are used to evaluate how appropriate a label is for describing a particular value of variable x . Simply, given a particular value α of variable x , the appropriateness degree for labeling this value with the label L , which is defined by fuzzy set F , is the membership value of α in F . The reason we use the new term ‘appropriateness degrees’ is partly because it more accurately reflects the underlying semantics and partly to highlight the quite distinct calculus based on this framework [4]. This definition provides a relationship between mass assignments and appropriateness degrees.

Definition 2 (*Appropriateness Degrees*)

$$\forall x \in \Omega, \forall L \in \mathcal{L} \quad \mu_L(x) = \sum_{S \subseteq \mathcal{L}: L \in S} m_x(S)$$

Consider the previous example, we then can obtain $\mu_{medium}(15) = 0.7 + 0.3 = 1$, $\mu_{low}(15) = 0.7$. Based on the underlying semantics, we can translate a set of numerical data into a set of mass assignments on appropriate labels based on the reverse of definition 2 under the following assumptions: consonance mapping, full fuzzy covering and 50% overlapping [7]. These assumptions are fully described in [7] and justified in [4]. These assumptions guarantee that there is unique mapping from appropriate degrees to mass assignments on labels. Based on these assumptions, we can isolate a set of subsets of $\mathcal{L}$ with non-zero mass assignments. These are referred to as *focal sets*:

Definition 3 (*Focal Set*) *Given a universe Ω for variable x , the focal set of $\mathcal{L}$ is a set of focal elements defined as:*

$$\mathcal{F} = \{S \subseteq \mathcal{L} | \exists x \in \Omega, m_x(S) > 0\}$$

The right-hand side figure of fig. 1 shows the universes of two variables x_1 and x_2 which are fully covered by 3 fuzzy sets with 50% overlap, respectively. For x_1 , the following focal elements occur: $\{small_1\}$, $\{small_1, medium_1\}$, $\{medium_1\}$, $\{medium_1, large_1\}$ and $\{large_1\}$. Since $small_1$ and $large_1$ do not overlap, the set $\{small_1, large_1\}$ cannot occur as a focal element according to def. 3. We can always find a unique translation from a given data point to a mass assignment on focal elements, as specified by the function μ_L . This is referred to as *linguistic translation (LT)* and is defined as follows: *For a particular attribute with an associated focal set, linguistic translation is a process of replacing data elements with masses of focal elements of these data.* For example in fig. 1, $\mu_{small_1}(x_1(1) = 0.27) = 1$, $\mu_{medium_1}(0.27) = 0.6$ and $\mu_{large_1}(0.27) = 0$. They are simply the memberships read from the fuzzy sets. We then can obtain the mass assignment of this data element according to def. 2 under consonance assumption [7]: $m_{0.27}(small_1) = 0.4$, $m_{0.27}(small_1, medium_1) = 0.6$. Similarly, the linguistic translations for $\mathbf{x}_1 = \langle x_1(1) = 0.27, x_2(1) = 158 \rangle$ and $\mathbf{x}_2 = \langle x_1(2) = 0.7, x_2(2) = 80 \rangle$ are illustrated on each attribute independently as follows:

$$\left[\begin{array}{c} x_1 \\ x_1(1) = 0.27 \\ x_1(2) = 0.7 \end{array} \right] \xrightarrow{LT} \left[\begin{array}{ccccc} m_x(\{s_1\}) & m_x(\{s_1, m_1\}) & m_x(\{m_1\}) & m_x(\{m_1, l_1\}) & m_x(\{l_1\}) \\ 0.4 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.8 & 0 \end{array} \right]$$
$$\left[\begin{array}{c} x_2 \\ x_1(2) = 158 \\ x_2(2) = 80 \end{array} \right] \xrightarrow{LT} \left[\begin{array}{ccccc} m_x(\{s_2\}) & m_x(\{s_2, m_2\}) & m_x(\{m_2\}) & m_x(\{m_2, l_2\}) & m_x(\{l_2\}) \\ 0 & 0 & 0 & 0.4 & 0.6 \\ 0.4 & 0.6 & 0 & 0 & 0 \end{array} \right]$$

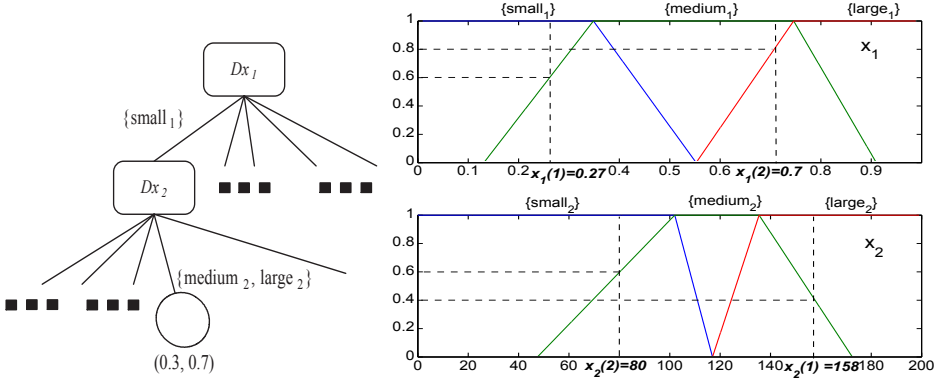


Fig. 1. Left-hand side: A schematic illustration of a linguistic decision tree. Right-hand side: A full fuzzy covering (discretization) with three fuzzy sets with 50% overlap on two attributes x_1 and x_2 , respectively

3 Linguistic Decision Tree

Linguistic decision tree (LDT) [7] is a tree-structured classification model based on label semantics. The information heuristics used for building the tree are modified from Quinlan's ID3 [9] in accordance with label semantics. Given a database of which each instance is labeled by one of the classes: $\{C_1, \dots, C_M\}$. A linguistic decision tree with S consisting branches built from this database can be defined as follows:

$$T = \{\langle B_1, P(C_1|B_1), \dots, P(C_M|B_1) \rangle, \dots, \langle B_S, P(C_1|B_S), \dots, P(C_M|B_S) \rangle\}$$

where $P(C_k|B)$ is the probability of class C_k given a branch B . A branch B with d nodes (i.e., the length of B is d) is defined as:

$$B = \langle F_1, \dots, F_d \rangle$$

where, $d \leq n$ and $F_j \in \mathcal{F}_j$ is one of the focal elements of attribute j . The left-hand side figure of fig 1 gives an schematic illustration of a LDT for a binary classification problem. For example, consider the branch: $\langle \langle \{small_1\}, \{medium_2, large_2\} \rangle, 0.3, 0.7 \rangle$. This means the probability of class C_1 is 0.3 and C_2 is 0.7 given attribute 1 can only be described as *small* and attribute 2 can be described as both *medium* and *large*. We may notice that different fuzzy discretization methods may result in different translations between numerical data and their linguistic models. In this paper, we will use a very intuitive method for generating fuzzy sets referred to as percentile-based (or equal-point) discretization [7, 11]. In this approach, each attribute universe is partitioned into intervals each containing approximately the same number of data elements.

Consider a training set $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ where each instance $\mathbf{x}$ has n attributes: $\langle x_1, \dots, x_n \rangle$. We now describe how the relevant branch probabili-

ties for a LDT can be evaluated from a database. The probability of class C_k ($k = 1, \dots, M$) given B can then be evaluated as follows. First, we consider the probability of a branch B given $\mathbf{x}$:

$$P(B|\mathbf{x}) = \prod_{j=1}^d m_{x_j}(F_j) \quad (1)$$

where $m_{x_j}(F_j)$ for $j = 1, \dots, d$ are mass assignments of single data element x_j . Consider the previous example, where we are given a branch $B = \langle \{small_1\}, \{medium_2, large_2\} \rangle$ in fig. 1 and data element $\mathbf{x}_1 = \langle 0.27, 158 \rangle$ (the linguistic translation of $\mathbf{x}_1$ was given in last section). According to eq. 1:

$$P(B|\mathbf{x}_1) = m_{x_1}(\{small_1\}) \times m_{x_2}(\{medium_2, large_2\}) = 0.4 \times 0.4 = 0.16$$

The probability of class C_k given B can then be evaluated by:

$$P(C_k|B) = \frac{\sum_{i \in \mathcal{D}_k} P(B|\mathbf{x}_i)}{\sum_{i \in \mathcal{D}} P(B|\mathbf{x}_i)} \quad (2)$$

where $\mathcal{D}_k$ is the subset consisting of instances which belong to class k . In the case where the denominator is equals to 0, which may occur when the training database for the LDT is small, then there is no non-zero linguistic data covered by the branch. In this case, we obtain no information from the database so that equal probabilities are assigned to each class.

$$P(C_k|B) = \frac{1}{M} \quad \text{for } k = 1, \dots, M \quad (3)$$

Now consider classifying an unlabeled instance in the form of $\mathbf{x} = \langle x_1, \dots, x_n \rangle$ which may not be contained in the training data set. First we apply linguistic translation to $\mathbf{x}$ based on the fuzzy covering of the training data¹. According to the Jeffrey's rule [3] the probabilities of class C_k given a LDT with S branches are evaluated as follows:

$$P(C_k|\mathbf{x}) = \sum_{s=1}^S P(C_k|B_s)P(B_s|\mathbf{x}) \quad (4)$$

where $P(C_k|B_s)$ and $P(B_s|\mathbf{x})$ are evaluated based on equations 1 and 2 (or 3), respectively.

The goal of tree-structured learning models is to generate subregions partitioned by branches that are less "impure", in terms of the mixture of class labels, than the unpartitioned dataset. For a particular branch, the most suitable free attribute for further expanding (or partitioning), is the one by which the "pureness" is maximumly increased with expanding. That corresponds to selecting the

¹ In the case that a data element appears beyond the range of training data set, we then assign the appropriateness degrees of the minimum or maximum values of the universe to the data element depending on which side of the range it appears.

attribute with maximum information gain. The algorithm for developing linguistic decision trees is fully described in [7] and will not be reproduced here due to the page limitation. Similar to ID3, in developing the tree, the most informative attribute will form the root of a linguistic decision tree, and the tree will expand into branches associated with all possible focal elements of this attribute. For each branch, the attribute that has not appeared in this branch and that has the maximum information gain will be selected as the next node. This is will be repeated from level to level until the tree reaches the maximum specified depth or some other termination criteria are met.

4 Bayesian Estimation Trees with Fuzzy Labels

4.1 Naive Bayes Classification Based on Label Semantics

Bayesian reasoning provides a probabilistic approach to inference based on the Bayesian theorem. Given a test instance, the learner is asked to predict its class according to the evidence provided by the training data. The classification of unknown example $\mathbf{x}$ by Bayesian estimation is on the basis of the following probability,

$$P(C_k|\mathbf{x}) = \frac{P(\mathbf{x}|C_k)P(C_k)}{P(\mathbf{x})} \quad (5)$$

Since the denominator in eq. 5 is invariant across classes, we can consider it as a normalization parameter. So, we obtain:

$$P(C_k|\mathbf{x}) \propto P(\mathbf{x}|C_k)P(C_k) \quad (6)$$

Now suppose we assume for each variable x_j that its outcome is independent of the outcome of all other variables given class C_k . In this case we can obtain the so-called naive Bayes classifier as follows:

$$P(C_k|\mathbf{x}) \propto \prod_{j=1}^n P(x_j|C_k)P(C_k) \quad (7)$$

where $P(x_j|C_k)$ is often called the likelihood of the data x_j given C_k . For a qualitative attribute, it can be estimated from corresponding frequencies. For a quantitative attribute, either probability density estimation or discretization can be employed to estimate its probabilities. In label semantics framework, suppose we are given focal set $\mathcal{F}_j$ for each attribute j . Assuming that attribute x_j is numeric with universe Ω_j , then the likelihood of x_j given C_k can be represented by a density function $p(x_j|C_k)$ determine from the database $\mathcal{D}_k$ and prior density according to Jeffrey's rule [3].

$$p(x_j|C_k) = \sum_{F \in \mathcal{F}_j} p(x_j|F)P(F|C_k) \quad (8)$$

From Bayes theorem:

$$p(x_j|F) = \frac{P(F|x_j)p(x_j)}{P(F)} = \frac{m_{x_j}(F)p(x_j)}{pm(F)} \quad (9)$$

where,

$$pm(F) = \int_{\Omega_j} P(F|x_j)p(x_j)dx_j = \frac{\sum_{\mathbf{x} \in \mathcal{D}} m_{x_j}(F)}{|\mathcal{D}|} \quad (10)$$

Substituting equation 9 in equation 8 and re-arranging gives

$$p(x_j|C_k) = p(x_j) \sum_{F \in \mathcal{F}_j} m_{x_j}(F) \frac{P(F|C_k)}{pm(F)} \quad (11)$$

Also $P(F|C_k)$ can be derived from $\mathcal{D}_k$ according to

$$P(F|C_k) = \frac{\sum_{\mathbf{x} \in \mathcal{D}_k} m_{x_j}(F)}{|\mathcal{D}_k|} \quad (12)$$

Here in this paper, this model is called fuzzy Naive Bayes (FNB) and more details of FNB can be found in [11].

4.2 Bayesian Estimation Given a LDT

Given a decision tree T is learnt from a training database $\mathcal{D}$. According to the Bayesian theorem: A data element $\mathbf{x} = \langle x_1, \dots, x_n \rangle$ can be classified by:

$$P(C_k|\mathbf{x}, T) \propto P(\mathbf{x}|C_k, T)P(C_k|T) \quad (13)$$

We can then divide the attributes into 2 disjoint groups denoted by $\mathbf{x}_T = \{x_1, \dots, x_m\}$ and $\mathbf{x}_B = \{x_{m+1}, \dots, x_n\}$, respectively. $\mathbf{x}_T$ is the vector of the variables that are contained in the given tree T and the remaining variables are contained in $\mathbf{x}_B$. Assuming conditional independence between $\mathbf{x}_T$ and $\mathbf{x}_B$ we obtain:

$$P(\mathbf{x}|C_k, T) = P(\mathbf{x}_T|C_k, T)P(\mathbf{x}_B|C_k, T) \quad (14)$$

Because $\mathbf{x}_B$ is independent of the given decision tree T and if we assume the variables in $\mathbf{x}_B$ are independent of each other given a particular class, we can obtain:

$$P(\mathbf{x}_B|C_k, T) = P(\mathbf{x}_B|C_k) = \prod_{j \in \mathbf{x}_B} P(x_j|C_k) \quad (15)$$

Now consider $\mathbf{x}_T$. According to Bayes theorem,

$$P(\mathbf{x}_T|C_k, T) = \frac{P(C_k|\mathbf{x}_T, T)P(\mathbf{x}_T|T)}{P(C_k|T)} \quad (16)$$

Combining equation 14, 15 and 16:

$$P(\mathbf{x}|C_k, T) = \frac{P(C_k|\mathbf{x}_T, T)P(\mathbf{x}_T|T)}{P(C_k|T)} \prod_{j \in \mathbf{x}_B} P(x_j|C_k) \quad (17)$$

Combining equation 13 and 17

$$P(C_k|\mathbf{x}, T) \propto P(C_k|\mathbf{x}_T, T)P(\mathbf{x}_T|T) \prod_{j \in \mathbf{x}_B} P(x_j|C_k) \quad (18)$$

Further, since $P(\mathbf{x}_T|T)$ is independent from C_k , we have that:

$$P(C_k|\mathbf{x}, T) \propto P(C_k|\mathbf{x}_T, T) \prod_{j \in \mathbf{x}_B} P(x_j|C_k) \quad (19)$$

where $P(x_j|C_k)$ is evaluated according to eq. 11 and $P(C_k|\mathbf{x}_T, T)$ is just the class probabilities evaluated from the decision tree T according to equation 4.

The basic idea of using Bayesian estimation given a LDT is to use the LDT as one estimator and the rest of the attributes as other independent estimators. If we extend this idea, we use a set of small-sized LDTs as estimators, we then have the second hybrid model which is described in the next section.

4.3 Bayesian Estimation from a Set of Trees

Given a training dataset, a small-sized tree (usually the depth is less than 3) can be learnt based on the method we discussed in section 3. We then learn another tree with the same size based on the rest of the attributes, i.e., the attributes which have not been used in previous trees. Successively, a set of trees can be built from training set. If we denote the trees by $\mathcal{T} = \langle T_1, \dots, T_W \rangle$, for each tree T_w , the set of attributes $\mathbf{x}_{T_w}$ are exclusive each other for $w = 1, \dots, W$. For a given unclassified data element $\mathbf{x}$, we can partition it into W group of disjoint set of attributes $\langle \mathbf{x}_{T_1}, \dots, \mathbf{x}_{T_W} \rangle$. If we assume:

$$P(C_k|\mathbf{x}) = P(C_k|\mathbf{x}_{T_1}, \dots, \mathbf{x}_{T_W}) \approx P(C_k|T_1, \dots, T_W) \quad (20)$$

Then, according to the Bayesian theorem:

$$P(C_k|\mathcal{T}) = P(C_k|T_1, \dots, T_W) = \frac{P(T_1, \dots, T_W|C_k)P(C_k)}{P(T_1, \dots, T_W)} \quad (21)$$

Given the assumption the trees are generated independently then it is reasonable to assume that the groups of attributes are conditional independent to each other. Hence,

$$P(T_1, \dots, T_W|C_k) = \prod_{w=1}^W P(T_w|C_k) \quad (22)$$

For a particular tree T_w for $w = 1, \dots, W$, we have

$$P(T_w|C_k) = \frac{P(C_k|T_w)P(T_w)}{P(C_k)} \quad (23)$$

So that,

$$\prod_{w=1}^W P(T_w|C_k) = \frac{\prod_{w=1}^W P(C_k|T_w) \prod_{w=1}^W P(T_w)}{P(C_k)^W} \quad (24)$$

Combine eq. 21, 22 and 24, we obtain

$$P(C_k|\mathcal{T}) \propto \frac{\prod_{w=1}^W P(C_k|T_w) \prod_{w=1}^W P(T_w)}{P(C_k)^{W-1}} \quad (25)$$

Since $\prod_{w=1}^W P(T_w)$ is independent from C_k , we finally obtain:

$$P(C_k|\mathcal{T}) \propto \frac{\prod_{w=1}^W P(C_k|T_w)}{P(C_k)^{W-1}} \tag{26}$$

where $P(C_k|T_w)$ is evaluated according to eq. 4.

5 Experimental Studies

We evaluated the LDT model, single LDT with Bayesian estimation (denoted by BLDT) and Bayesian estimation with a set of trees (denoted by FLDT - a forest of LDTs) on 10 datasets taken from the UCI Machine Learning repository [2]. The descriptions are shown in table 1. Unless otherwise stated, attributes are discretized by 2 trapezoidal fuzzy sets with 50% overlap based on equal-point discretization (see section 3), and classes are evenly split into two sub-datasets randomly, one half for training and the other half for testing, this is referred to as a 50-50 split experiment. For each dataset, we ran 50-50 experiment with random split for 10 times and the average test accuracies with standard deviations are shown against depths of the trees are shown in figures 2. The results of C4.5 ² Fuzzy Naive Bayes (FNB), FLDT and the best results of LDT and BLDT are shown in table 2, where d for LDT and BLDT represents the depth at which the best results are obtained.

From all the figures, we can see that the BLDT model generally performs better at shallow depths than LDT model. However, with the increasing of the tree depth, the performance of the BLDT model remains constant or decreases, while the accuracy curves for LDT increase. For datasets Balance, Ecoli, Wisconsin-Cancer (Wcancer) and Wine, BLDT model performs better at most of depths. For Iris and Hepatitis, the differences are insignificant at all depths. For Pima, LDT model performs better than BLDT model in most the depths and the differences are significant. For the rest of the datasets, the accuracy curves cross somewhere in the middle and the differences are not significant.

Table 1. Descriptions of the datasets for experiments selected from the UCI machine learning repository [2]

Dataset	Classes	Size	Attributes	Dataset	Classes	Size	Attributes
Balance	3	625	4	Ecoli	8	336	8
Glass	6	214	9	Hepatitis	2	155	19
Iris	3	150	4	Liver	2	345	6
Pima	2	768	8	Sonar	2	208	60
Wine	3	178	14	Wcancer	2	699	9

² The results are obtained by WEKA [12] machine toolkit with default settings.

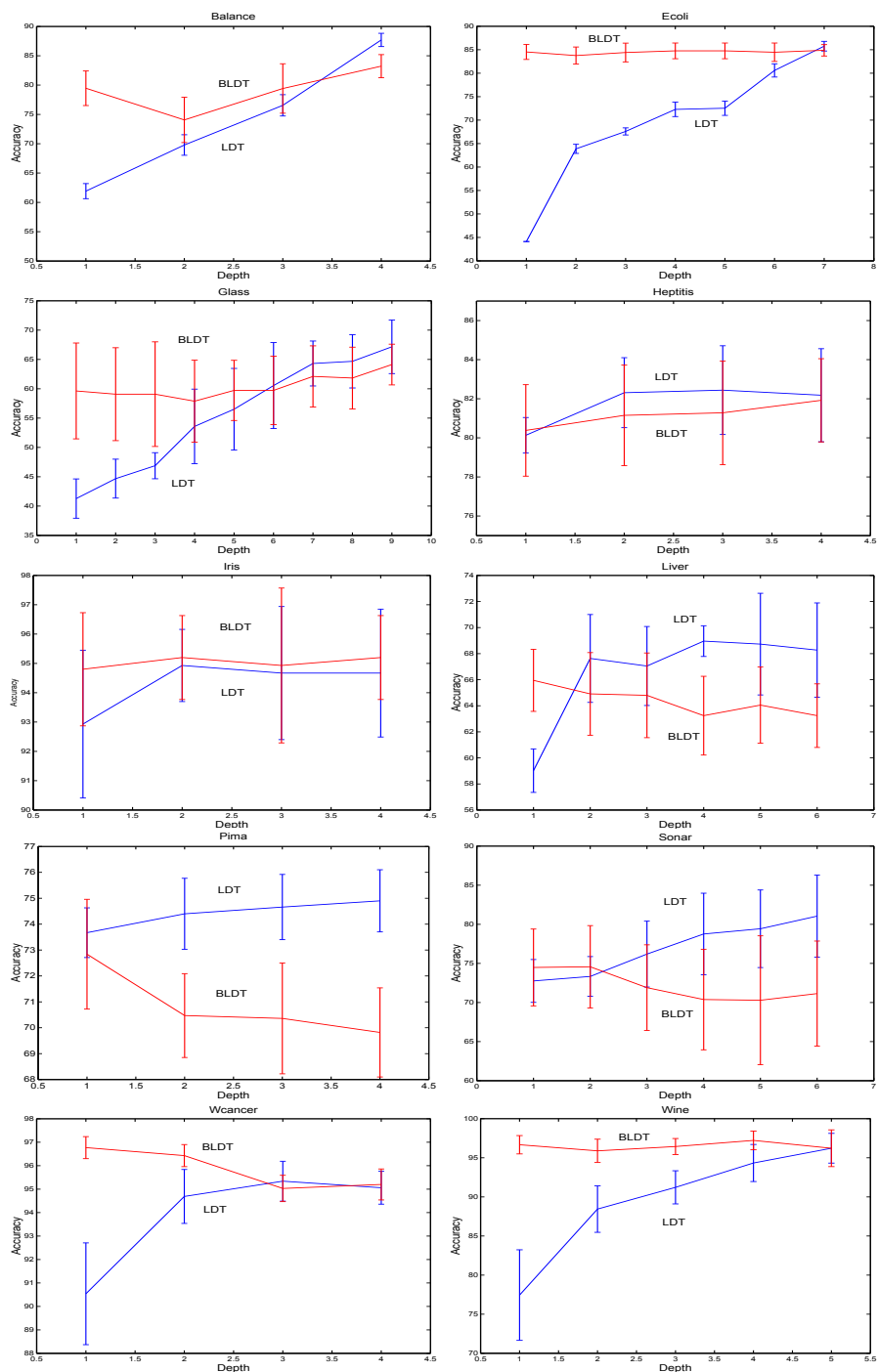


Fig. 2. Results for single LDT with Bayesian estimation: average accuracy with standard deviation on each dataset against the depth of the tree

Table 2. Experimental results on 10 UCI datasets: average accuracy with standard deviation from 10 runs of random 50-50 split experiments

Database	C4.5	FNB	LDT	<i>d</i>	BLDT	<i>d</i>	FLDT	
	Acc	Acc	Acc		Acc		Acc(<i>d</i> =1)	Acc(<i>d</i> =2)
Balance	79.20±1.53	73.77±2.43	87.70±1.13	4	83.23±1.97	4	66.26±2.81	79.42±1.99
Ecoli	78.99±2.23	76.53±4.19	85.76±1.03	7	84.53±1.60	1	80.18±3.45	78.76±1.60
Glass	64.77±5.10	48.35±6.80	59.17±3.70	9	64.13±3.47	9	52.94±8.74	58.53±5.28
Heptitis	76.75±4.68	80.13±2.28	82.44±2.27	3	81.92±2.13	4	80.26±3.15	79.26±0.41
Iris	93.47±3.23	93.73±2.60	94.93±1.23	2	95.20±1.43	2	93.73±1.89	92.00±3.38
Liver	65.23±3.86	63.35±2.38	68.96±3.18	4	65.95±2.38	1	62.43±4.62	59.65±2.09
Pima	72.16±2.80	72.29±2.25	74.90±1.20	4	72.84±2.12	1	72.40±1.48	66.07±1.04
Sonar	70.38±5.23	74.76±4.96	81.05±5.24	6	74.57±5.26	2	76.48±4.82	75.62±2.21
Wcancer	94.38±1.42	96.74±0.54	95.34±0.85	3	96.77±0.47	1	97.17±0.93	98.77±0.85
Wine	88.09±4.14	96.22±1.67	96.22±1.90	5	97.22±1.20	4	96.11±0.79	98.56±1.66

Table 3. Result comparisons (with LDT, BLDT and FLDT are at depth 2) based on t-test with 90% confidence, where ‘√’ represents significant better, ‘-’ represents equivalence and ‘×’ represents significant worse

Database	BLDT vs	BLDT vs	BLDT vs	FLDT vs	FLDT vs	FLDT vs
	C4.5	FNB	LDT	C4.5	FNB	LDT
Balance	√	√	√	-	√	√
Ecoli	√	√	√	-	-	√
Glass	-	√	√	-	√	√
Heptitis	√	-	-	-	-	×
Iris	-	-	-	-	-	-
Liver	-	-	-	-	-	×
Pima	-	-	×	-	-	×
Sonar	-	-	-	-	-	-
Wcancer	√	-	√	√	√	√
Wine	√	-	√	√	-	√

We performed t-tests with a confidence level of 90%³ to compare the models at depth 2 (except for C4.5 and FNB) and the results are shown in table 3. We can see that BLDT and FLDT models are better than Fuzzy Naive Bayes and C4.5. However, if we compare BLDT and FLDT with LDT, we can find that the BLDT model outperforms LDT at shallow depths and FLDT model has the equivalent performance. From fig. 2, we found that most best results for BLDT are obtained at shallow depths, but for LDTs the best results are always obtained with deep depths. So, we can conclude that BLDT model is

³ We generally believe that the confidence level of 90 % is enough to be significant for comparisons among different learning models given these relatively simple data sets.

more efficient than LDT. Compare to BLDT, the FLDT model performs relative worse and less efficient, the reasons are probably because that small-trees are not good estimators. But this still needs more further investigation.

6 Conclusions

In this paper, we proposed two hybrid models by combining Naive Bayes classifier and linguistic decision trees based on label semantics. Through experimental studies, we found that the BLDT (the Bayesian estimation model given a LDT) model outperforms fuzzy naive Bayes, C4.5 and the linguistic decision tree model at shallow tree depths. However, the FLDT (using a set of small-size LDTs as Bayesian estimators) model outperforms fuzzy Naive Bayes classifier and C4.5 but has equivalent accuracy to LDTs. Further research focus on investigating the reasons that FLDTs are not good Bayesian estimators and testing on more datasets.

References

1. J.F. Baldwin, T.P. Martin and B.W. Pilsworth. *Fril-Fuzzy and Evidential Reasoning in Artificial Intelligence*. John Wiley & Sons Inc, 1995.
2. C. Blake and C.J. Merz. UCI machine learning repository.
<http://www.ics.uci.edu/mlearn/MLRepository.html>
3. R.C.Jeffrey. *The Logic of Decision*, Gordon & Breach Inc., New York, 1965.
4. J. Lawry. A framework for linguistic modelling, *Artificial Intelligence*, 155: pp. 1-39, 2004.
5. C. X. Ling. Decision tree with better ranking. *Proceedings of International Conference on Machine Learning (ICML2003)*. Washington DC, 2003.
6. F. Provost and P. Domingos. Tree induction for probability-based ranking. *Machine Learning*. 52, pp.199-215, 2003.
7. Z. Qin and J. Lawry. Decision Tree Learning with Fuzzy Labels. To appear in *Information Sciences*, 2005.
8. Z. Qin and J. Lawry. ROC analysis of a linguistic decision tree merging algorithm. *The Pro. of UK Workshop on Computational Intelligence*, Loughborough, UK, 2004.
9. J.R. Quinlan. Induction of decision trees. *Machine Learning* 1: 81-106. 1986
10. J.R. Quinlan. *C4.5: Programs for Machine Learning*. San Mateo: Morgan Kaufmann, 1993.
11. N.J. Randon and J. Lawry. Classification and query evaluation using modelling with words. *Information Sciences, Special Issue - Computing with Words: Models and Applications*, To appear.
12. I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, 1999.
<http://www.cs.waikato.ac.nz/~ml/weka/>

Selective Gaussian Naïve Bayes Model for Diffuse Large-B-Cell Lymphoma Classification: Some Improvements in Preprocessing and Variable Elimination

Andrés Cano, Javier G. Castellano, Andrés R. Masegosa,
and Serafín Moral

Dept. Computer Science and Artificial Intelligence,
University of Granada, Granada, 18071, Spain
{acu, fjgc, andrew, smc}@decsai.ugr.es

Abstract. In this work, we present some significant improvements for feature selection in wrapper methods. They are two: the first of them consists in a proper preordering of the feature set; and the second one consists in the application of an irrelevant feature elimination method, where the irrelevance condition is subjected to the partial selected feature subset by the wrapper method. We validate these approaches with the *Diffuse Large B-Cell Lymphoma* subtype classification problem and we show that these two changes are an important improvement in the computation cost and the classification accuracy of these wrapper methods in this domain.

1 Introduction

Supervised classification is a task that assigns predefined class labels to data items described by a set of features or attributes. A *classifier* is a function that maps an instance into a class label. In this paper we address the problem of the classification of different subtypes (GCB and ABC) of Diffuse Large-B-Cell Lymphoma from the measured expressions of a large number of genes. Another important characteristic of this problem is that the sample size is always small.

The problem of the automatic induction of classifiers from data sets of preclassified instances has received considerable attention within the machine learning community for a long time. Traditional approaches to this problem include decision trees, neural networks and classical statistical methods [1]. More recently, Bayesian networks have been successfully applied to analyze interaction between genes [2] or to induce classifiers [3].

One of the most used and simple Bayesian network classifiers is the naïve Bayes (NB). It simplifies the learning task by assuming that the attributes are independent given the variable to classify (no structural learning is required,

since the network structure is fixed). Only the parameters have to be estimated from the data set and the classification process is very efficient. The success of this classifier is mainly due to its simplicity and to its surprisingly competitive predictive accuracy outperforming more sophisticated classifiers (especially when the attributes are not strongly correlated) [4, 5, 6]. The other basic assumption of the NB classifier is that all the attributes have some influence on the class variable.

The *selective naïve Bayesian classifier* (SNBC) [7, 8] searches for a good subset of attributes (*Feature Subset Selection* (FSS)), starting from an empty set and iteratively introducing the attribute that most improve the predictive accuracy, using cross-validation. SNBC tries to avoid overamplified influence in the class from strongly correlated attributes and noise in classification because of irrelevant attributes. Previous works [9, 3, 10] show how a selection of features obtains an accurate classification. FSS in the case of DNA microarrays is called *gene selection*.

One of the problem when we face with gene expression profiling data is how to handle continuous data. Previous work in Bayesian classifiers has solved this difficulty by discretizing them [11] or assuming that the predictive variables are distributed as a Gaussian distribution for each variables [12, 13]. We use the a Gaussian distribution for each variable and for each class. The strong assumption about the normality of the data can be a good approximation against discretization which suffers from the loss of information inherited from the discretization method.

There are two basic approaches for Feature Selection Problem. *Filter selection* uses an undirected measure for the goodness of each gene, usually a rank of the attributes. *Wrapper selection* uses the classifier to measure the accuracy for a subset of genes. In general, the wrapper approach obtains more accurate behavior than filter metrics, but this improvement is coupled with a higher time. The computer load is especially important in DNA microarray experiments where we have thousands of gene expression measurements. In this field databases have a high number of features (genes) far exceeding the number of samples.

This work proposes some modifications to traditional wrapper approaches, in order to apply them to the DNA microarray domain where data sets have a high dimensionality. We meaningfully reduce the search space, improving computer load, and achieving a better accurate behaviour. Basically, the modifications consist on establishing a *hierarchical order* in the search space, and also on using an *elimination of irrelevant variables*.

The rest of the paper is organized as follows. Section 3 analyzes the importance of the hierarchical order of the variables in the accuracy of the classification model, reducing the search space of the algorithm. Section 4 shows the algorithm to eliminate irrelevant variables based on a new heuristic. Section 5 shows the experiment results, comparing with the results in the works [14, 15]. Finally, section 6 gives the conclusions and future research in this field.

2 The Wrapper Gaussian Naïve-Bayes Algorithm

The classifier model we have used is a naïve Bayes model where the attributes have a continuous domain. We use a simple wrapper gene selection approach to select the features, the Sequential Forward Selection (SFS) method [16], using the accuracy of classification as the function to select new features.

We shall use $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ to denote the set of features (genes) describing the possible instances to be classified (X_i is the variable related with the i -th gene), and C is the class variable. The supervised classification problem reduces to find c^* such as:

$$c^* = \arg_c \max P(C = c | X_1 = x_1, \dots, X_n = x_n)$$

In the experimental work, when there is no class with a probability higher than a given threshold δ , then the case is left *unclassified*.

Learning a classifier amounts to estimating $P(C = c | \mathbf{X})$ from a set of K labelled training samples which we denote by $\mathcal{T} = \{C_1, \dots, C_K\}$, where $C_j = (x_{1j}, \dots, x_{nj}, c_j)$ correspond to the features and the class of instance j , ($j = 1, \dots, K$).

2.1 Gaussian Naïve Bayes Classifier

In the naïve Bayes classifier [17] no structure learning is required, we assume that the attributes $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ are independent given the variable to classify C . Only the parameters have to be estimated from the data set. To deal with continuous variables we assume that the attributes are distributed as a Gaussian distribution given each class.

Since naïve Bayes assumes conditional independence given the class, the a posteriori probability of the class c_j given a test case $\mathbf{x} = \{x_1, \dots, x_n\}$ verifies:

$$P(C = c_j | \mathbf{x}) \propto p(c_j) \cdot \prod_{i=1}^n f_{\mathcal{N}}(X_i = x_i : \mu_{ij}, \sigma_{ij})$$

where μ_{ij} is the mean and σ_{ij} is the standard deviation of the data set resulting of the projection of the whole data set over the X_i feature and class c_j . And $f_{\mathcal{N}}$ is the density function of a Gaussian distribution.

The predicted class is the one with the highest a posteriori probability.

2.2 Wrapper Feature Selection Algorithm

Wrapper Feature Selection (WFS) begins with an empty set of selected features, and successively add the feature $X_{max} \in \mathbf{X}$ that maximizes a given *evaluation function*. This is known in the literature as *Sequential Forward Selection* (SFS) [16]. We use the accuracy of the classification as evaluation function. This score is gotten by the application of a Gaussian naïve Bayes classifier using a *leave-one-out cross-validation* (LOO) procedure [18]. We apply the classifier only to

the training data set projected over the current set of selected features in the WFS algorithm.

Let be $\mathbf{F}_l$ the set of features selected in step l of the WFS algorithm. Then in step $l + 1$, a new Gaussian naïve Bayes model is learned for the set of features $\mathbf{F}_{l+1} = \mathbf{F}_l \cup \{X_{\max}\}$ ($X_{\max} \in \mathbf{X}$) being $X_{\max}$ the feature that maximizes the increment in classification accuracy in the training data set $\mathcal{T}$ using LOO methodology. The WFS algorithm continues selecting new features until a given *criterion for halting the search* is verified. Suppose $Acc(\mathbf{F}_l)$ is the accuracy of the classification in step l with the set of features $\mathbf{F}_l$. The algorithm stops if $Max\{Acc(\mathbf{F}_l), Acc(\mathbf{F}_{l-1}), \dots, Acc(\mathbf{F}_{l-q+1})\} \leq Acc(\mathbf{F}_{l-q})$, where q is a given parameter of the algorithm. That is, the algorithm stops when q steps are carried out without an improvement in the classification accuracy.

3 Feature Preorder in the Wrapper Search

This section shows how the use of a given preorder of the features, can be used to improve the accuracy of the classification, and to reduce the search space in the wrapper methods, achieving algorithm of lower complexity.

3.1 Description of the Proposed Preorders of Features

In the wrapper algorithm described in section 2.2, it is possible that in step l several features $X_{\max}$, $X_{\max} = arg_X \max\{Acc(\mathbf{F}_l \cup \{X_{\max}\})\}$, produce the same accuracy in classification in step $l + 1$. In the DNA microarray domain, (data sets with a high number of features and few samples), this possibility is high, because the more features we append the more well classified cases we obtain. For example, in a data set where 200 samples are bad classified, we have seen that it is quickly reduced to 40 samples, by appending a few new features. At a given step l we have hundred of candidate features and it is possible that two different ones classify the same number of samples.

We propose a preorder of the feature set. When we find several candidate features $X_{\max}$ then we select the one with the higher ranking. This will have a high influence in the accuracy of the classification as we will show in the experimental work. The methods used to establish the preorder of features are the following:

Random Preorder: The feature is selected randomly among the ones producing the same accuracy.

Anova Preorder: The set of features are ordered according to a filter measure, from higher to lower value. The considered measure is namely Anova coefficient, that is calculated with a standard one-way analysis of variance with respect to the class variable [15]. The genes with a high Anova coefficient have a significant difference between the means of its values for each class.

Accuracy Preorder: If we build a given classifier using only one feature and we apply a leave-one-out cross validation method over the training data set, we can estimate the accuracy of the classifier respect to a concrete feature in the training data set. With this score, we can order the whole set of features, from higher to lower accuracy.

Section 5.1 shows that the accuracy in the classification varies meaningfully depending on the preorder method we use. Particularly, the accuracy preorder produce the best results.

3.2 Reducing the Search Space in Wrapper Methods Using Preorders of Features

This section shows how preorder of features can be used to reduce the search space in wrapper methods, without loss of accuracy in the classification.

Now, we propose to limit the search of the feature $X_{\max}$ in step l to the set of the first t features in a given preorder, where t is a given integer constant. This modification reduces the complexity of the construction of the classifier from $O(K^2 \cdot \eta^2 \cdot n)$ to $O(K^2 \cdot \eta^2)$, in a database with K samples (cases) and n features. The value η represents the maximum number of variables selected by the wrapper algorithm. This value is normally much smaller than n . Also, we need to sum the cost of the calculation of the preorder of the features. This cost is the following:

Anova Preorder. The cost of calculating this preorder is $O(K \cdot n)$.

Accuracy Preorder. Now we need to carry out a cross validation in order to estimate the accuracy for each variable. The resulting cost is $O(K^2 \cdot n)$.

In this way, now the complexity of the wrapper algorithm does not depend on the number of features n in the data set. The number of features only has influence in the preorder stage. The reduction in the search space for new features $X_{\max}$ in step l of the algorithm, does not cause loss in the accuracy classification as we will show in the experimental work in section 5.2. The resulting FSS wrapper algorithm is as follows:

Algorithm 1. *Limited Forward Sequential Selection (LFSS)*

Make $F_0 = \emptyset$, $l = 0$

While $(\text{Max}\{Acc(\mathbf{F}_l), Acc(\mathbf{F}_{l-1}), \dots, Acc(\mathbf{F}_{l-q+1})\} \geq Acc(\mathbf{F}_{l-q}))$

- Let be $\mathbf{X}_{\max} = \{X_{l_1}, \dots, X_{l_p}\}$
- $X_{\max} = \arg_{X_{l_i}} \max\{\text{Order}(X_{l_i}) : X_{l_i} \in \mathbf{X}_{\max}\}$
- $F_{l+1} = F_l \cup X_{\max}$
- Remove $X_{\max}$ from the global set of features $\mathbf{X}$
- $l = l + 1$;

return $\arg_{F_i} \max\{Acc(F_i) : i \in \{1, \dots, l\}\}$;

In previous algorithm $\mathbf{X}_{\max}$ represents the set of the features that obtains maximum accuracy in step l . Suppose this set contains p elements. The features are only chosen among the first t features in the given preorder. That is, each one of the features in $\mathbf{X}_{\max}$ verifies that $X_i = \arg_{X_i} \max\{Acc(F_l \cup X_i) : X_i \in \mathbf{X}^t\}$. $\mathbf{X}^t$ is the set of the first t features in the preorder given by function $Order(X_i)$, and t is a given constant. The algorithm returns the set of features F_i that produce the best accuracy classification in the l steps carried out by the algorithm.

4 Elimination of Irrelevant Variables

The idea of eliminating irrelevant variables in wrapper methods is included in a more general technique called *Backward Elimination* [19]. This method begins with the complete set of features and successively remove the ones that are found as irrelevant. In [19] no evidence is found if this method is better than Sequential Forward Selection. Posterior works [20, 21] develop new variants of the method. In these cases they obtain better accuracy rates, but the complexity of the algorithms are prohibitive when there are too many irrelevant variables.

4.1 Irrelevant Features

There are several possible definitions for relevant and irrelevant variables, see [8, 22]. All these definitions are based in the correlation factor among the states of the variable to consider and the different values of the class variable. We propose a new heuristic method to define irrelevant variables. Let denote Δ to a classifier with a set of features $\mathbf{Y} \subset \mathbf{X}$, built with the data set $\mathcal{T}$. Suppose $C_Y^\Delta = (s_1, s_2, \dots, s_K)$ is a *classification vector* that defines if the classifier Δ classifies well or not each one of the samples in the data set $\mathcal{T}$ using only the features of $\mathbf{Y}$. In a classification vector C_Y^Δ , $s_i = 1$ if the class of sample i is correctly found, and $s_i = 0$ otherwise. Let define now a relation order between two classification vectors:

Definition 1. If $r \in [0, 1]$ is a given parameter and $C_Y^\Delta = (s_1, s_2, \dots, s_K)$ and $C_W^\Delta = (t_1, t_2, \dots, t_K)$ are two classification vectors obtained using two set of features $\mathbf{Y}$ and $\mathbf{W}$ respectively, then $C_Y^\Delta \leq_r C_W^\Delta$ if $\frac{P}{K} < r$, where P is the number of samples that are correctly classified by the classifier C_Y^Δ and not correctly classified by the classifier C_W^Δ . Obviously, $0 \leq P \leq K$.

Previous definition indicates that $C_Y^\Delta \leq_r C_W^\Delta$ if the number of samples correctly classified in C_Y^Δ and not in C_W^Δ , are bellow a given rate r . Now we can define an irrelevant feature in the following way:

Definition 2. The feature X_i is irrelevant with respect to a set of features $\mathbf{Y}$ if $C_{\{X_i\}}^\Delta \leq_r C_Y^\Delta$.

Previous definition indicates that a feature X_i is irrelevant with respect to the set $\mathbf{Y}$ if the samples correctly classified using a classifier with only the feature

X_i , are included in the set of samples correctly classified with a classifier with the set of features $\mathbf{Y}$. The basic intuition idea is to look for new features classifying the cases that were incorrectly classified current features $\mathbf{Y}$. The inclusion is not strict, and we admit a difference lower than a given rate r in the number of samples correctly classified.

4.2 Wrapper Algorithm Based on Elimination of Irrelevant Variables

We propose a new modification in the wrapper algorithm described in section 2.2. Now at every step l of the wrapper algorithm, we eliminate the irrelevant variables with respect to the features included in the classifier. This process is made before doing the search of a new feature $X_{\max}$. In this way, the elimination of irrelevant features is not made a priori as in [19, 20, 21], but it is driven by the search process of the wrapper algorithm. This process reduces the complexity of the wrapper algorithm, and it obtains better accuracy rates as we will show in section 5.3.

The wrapper algorithm that includes this new improvement and the ones specified in sections 3.1 and 3.2 is the following:

Algorithm 2. *Limited Forward Sequential Selection with Variable Elimination (LFSS-VE)*

```

Make  $F_0 = \emptyset$ ,  $l = 0$ 
While ( $\#(\mathbf{X}) > 0$  and  $\text{Max}\{\text{Acc}(\mathbf{F}_l), \text{Acc}(\mathbf{F}_{l-1}), \dots, \text{Acc}(\mathbf{F}_{l-q+1})\} \geq \text{Acc}(\mathbf{F}_{l-q})$  )
    - Let be  $\mathbf{X}_{\max} = \{X_{l_1}, \dots, X_{l_p}\}$ 
    -  $X_{\max} = \arg_{X_{l_i}} \max\{\text{Order}(X_{l_i}) : X_{l_i} \in \mathbf{X}_{\max}\}$ 
    -  $F_{l+1} = F_l \cup X_{\max}$ ;
    - Remove  $X_{\max}$  from the global set of features  $\mathbf{X}$ 
    - Remove  $X_i \in \mathbf{X} : \mathbf{C}_{\{X_i\}}^{\Delta} \leq_r \mathbf{C}_{F_{l+1}}^{\Delta}$  ( $X_i$  is irrelevant with respect to  $F_{l+1}$ )
    -  $l = l + 1$ 
return  $\arg_{F_i} \max\{\text{Acc}(F_i) : i \in \{1, \dots, l\}\}$ 

```

In previous algorithm, the meaning of $\mathbf{X}_{\max}$ and $\text{Order}(X_i)$ is the same than in algorithm 1. Now, the loop contains an additional stopping condition: the set $\mathbf{X}$ is empty. The computational cost of this algorithm is low: $O(K \cdot n)$ where K is the number of samples in the data set and n the number of variables.

5 Classifying Diffuse Large B-Cell Lymphoma

The validation of our proposed approaches is carried out with two different data sets about *Diffuse Large B-Cell subtype classification*[23]:

BD1: This data set has been taken from [23]. It contains 348 genes with 42 samples. There are two classes: GCB and ABC with 21 samples each one.

BD2: This data set has been taken from [14]. This data set contains 8503 features (clones). Class GCB contains 134 samples and class ABC contains 83 samples.

The validation of the classifier for BD1 is carried out with the Leave-One-Out cross validation method [18], because the low number of samples of this data set. For BD2, we randomly partition the data set into two equals size parts: the training and the testing data sets. The number of features in BD2 is reduced using a previous method developed by us, [15]. This method is a filter method based on one way analysis of variance for each feature. This filter method is used with the aim of make possible evaluating the traditional wrapper methods in this big data set, because this evaluation is impossible over its 8503 features. All this process is repeated ten times, so we obtain 10 training data sets and 10 testing data sets, and the mean of the ten evaluations is the final evaluation result. We use this concrete evaluation method in order to compare with the results of [14, 15]. Using a Gaussian Naïve Bayes Classifier, Section 2.1, where we include all the present feature for BD1 and all the features in the transformed BD2, we obtain the results shown in table 1.

Table 1

BD1		BD2	
N of Genes	348 ± 0.0	N of Genes	78.7 ± 4.4
LOO Accuracy Rate	$97.6 \pm 0.7 \%$	Test accuracy rate	$94.1 \pm 1.3 \%$
LOO -log likelihood	0.61 ± 4.9	Test -log likelihood	0.53 ± 0.15

The parameters we have used to implement the procedures in sections 3.1, 3.2, 4.2 are the following:

- Halt Condition of FSS Algorithm. Parameter $q = 2$ (Section 2.2). That is, the FSS algorithm will stop if there are two iterations without an improvement in the classifier accuracy.
- Wrapper Search Limit. Parameter $t = 10$ (Section 3.2). That is, the FSS algorithm only searches in the first ten preordered variables.
- Irrelevant Condition. Parameter $r = 0.02$. (Section 4.1). That is, a feature is irrelevant if the percentage of cases that are correctly classified whit it and that were wrong classified with the current set of variables is lower than 2%.
- Accuracy preorder. (Section 3.1). This is the chosen preorder in all the cases, except when we specify another preorder.

5.1 Experimental Results: Wrapper Dependence of the Feature Preorder

Now, we use the classic FSS algorithm described in Section 2.2. We carry out three distinct runs of this algorithm using the three preorder methods of 3.1. The results are shown in Table 2.

Table 2

Data Base	Date	Random Preorder	Anova Preorder	Accuracy Preorder
BD1	Accuracy	80.9 ± 4.9	81.0 ± 4.9	92.8 ± 2.1
BD1	-logLike	0.39 ± 0.2	0.74 ± 1.42	0.31 ± 0.30
BD1	N Genes	4.3 ± 0.5	3.2 ± 0.1	3.8 ± 0.5
BD1	N Eval	74900	77790	82300
BD2	Accuracy	88.9 ± 0.6	91.0 ± 0.4	89.1 ± 0.5
BD2	-logLike	0.41 ± 0.15	0.35 ± 0.1	0.40 ± 0.13
BD2	N Genes	8.0 ± 3.2	9.0 ± 5.1	7.6 ± 4.0
BD2	N Eval	8002	8630	7709

Table 3

Data Base	Date	Algorithm 1	Data Base	Date	Algorithm 2
BD1	Accuracy	92.8 ± 2.1	BD1	Accuracy	95.2 ± 1.4
BD1	-logLike	0.36 ± 0.44	BD1	-logLike	0.08 ± 0.03
BD1	N Genes	3.8 ± 0.3	BD1	N Genes	5.4 ± 0.1
BD1	N Eval	2840	BD1	N Eval	1882
BD2	Accuracy	91.8 ± 0.4	BD2	Accuracy	93.0 ± 0.4
BD2	-logLike	0.28 ± 0.07	BD2	-logLike	0.25 ± 0.07
BD2	N Genes	7.8 ± 3.0	BD2	N Genes	8.1 ± 5.6
BD2	N Eval	1080	BD2	N Eval	1018

(a)

(b)

Comparing with table 1 we can see as the accuracy rate grows and the *-log likelihood* decreases with the preorder introduction in several data sets.

5.2 Experimental Results: Introduction of a Preorder Limit in the Wrapper Search Feature Space

Table 3a shows the results when we choose new features only among the t first ones in the given preorder, section 3.2, algorithm 1. Comparing with table 2, we can see that there is a significative improvement in the accuracy and *-log likelihood* respect to *Classical FSS* (Random Preorder column in table 2) in both data sets. Secondly, there is an significant reduction of the number of evaluations between the two algorithms, a 96% in BD1 and a 87% in BD2. In addition, we can see that these improvements are not influenced by the number of selected genes, because they are similar in the three cases.

5.3 Experimental Results: Variable Elimination

Table 3b shows the results of applying algorithm 2 of section 4.2 (elimination of irrelevant variables). Comparing with tables 2 and 3a, we can see as the Algorithm 2 improves the accuracy rate and the *-log likelihood* of both data sets. Also, a reduction of the number of evaluations is carried out.

5.4 Experimental Results: Accuracy Order vs Anova Order

The results of Table 4 show us that Algorithm 2 does not depend on the used preorder type, because it obtains better results whatever the preorder we use, accuracy preorder or Anova preorder, although the first one achieves the best results.

Table 4

Data Base	Date	Algorithm 2 Anova Preorder	Algorithm 2 Accuracy Preorder
BD1	Accuracy	88.1 ± 3.3	95.2 ± 1.4
BD1	-logLike	0.59 ± 1.43	0.08 ± 0.03
BD1	N Genes	3.9 ± 0.1	5.4 ± 0.1
BD1	N Eval	2461	1882
BD2	Accuracy	90.7 ± 0.5	93.0 ± 0.4
BD2	-logLike	0.31 ± 0.08	0.25 ± 0.07
BD2	N Genes	7.6 ± 2.7	8.1 ± 5.6
BD2	N Eval	885	1018

5.5 Results Comparison

There are several proposed classifiers [24, 25, 26] for the data set BD1. However there are not too many proposed classifiers with the data set BD2 introduced in [27]. Perhaps the best classification results can be found in [14] and [15]. In [14] it is shown a statistical model based on a *lineal predictor score* (LPS) which is applied to the clustering proposed by [27]. The resulting classifier contains 27 genes. If there is no class with a probability higher than 0.9, then the case is left *unclassified*. In [15], we use a classification process based on two phases, filter phase and wrapper phase. The wrapper phase is a Sequential Forward Selection Wrapper Algorithm with an *Abduction Phase*. The resulting classifier contains 7 genes and its results are similar, but the main difference is that the evaluation of this classifier is carried out in ten distinct training and testing sets, against Wright classifier [14] that is only evaluated in an unique training and testing set.

As we can see in Table 5, the results of our Algorithm 2 are better than the shown ones in [15] and this classifier achieves a similar number of genes, 8.1 versus 7.0. On the other hand, the results of our Algorithm 2 are similar to the ones of [14], but we obtain a lower number of genes, 8 versus 27 and our validation is carried out in ten distinct partitions of the data set respect to the unique evaluation of the classifier of [14]. In fact, we have better results than [14] in several of the ten evaluations of our classifier.

6 Conclusions and Future Work

The wrapper technique has successfully been used in many supervised classification problems, in particular, in supervised classification of gene expression data. But its main disadvantage is its high computational cost, specially in problems

Table 5. (a) Classifier of [14] (b) Classifier of [15] (c) Classifier of Algorithm 2 with cutoff for unclassified equal to 0.9

Training Dataset				Training Dataset				Training Dataset			
True class		Predicted class		True class		Predicted class		True class		Predicted class	
	ABC	GCB	Unclass.		ABC	GCB	Unclass.		ABC	GCB	Unclass.
ABC	37	1	4	ABC	38.9	0.6	2.5	ABC	37.3	1.0	3.7
GCB	1	58	8	GCB	0.7	63.3	3.0	GCB	0.5	60.0	6.5
Test Dataset				Test Dataset				Test Dataset			
True class		Predicted class		True class		Predicted class		True class		Predicted class	
	ABC	GCB	Unclass.		ABC	GCB	Unclass.		ABC	GCB	Unclass.
ABC	38	1	2	ABC	32.7	3.5	4.8	ABC	32.7	1.3	7.0
GCB	2	57	8	GCB	3.2	58.8	5.0	GCB	1.7	57.4	7.9

(a) (b) (c)

as gene expression data classification due to its high dimensionality. Because of it, the filter methods are used with the wrapper methods.

In this work, we have proposed several improvements for the wrapper methods in order to allow its application to the whole data base, without the necessity of use the fast filter methods.

As we saw in the experimental results, the feature preorder and the wrapper search in the only first t features is an excellent method to reduce the computational cost of the wrapper search without a loss in the classifier accuracy rate. On the other hand, the introduction of a new heuristic for irrelevant feature elimination depending on the wrapper search process has shown a very good behaviour when applied to the *Large B-Cell Lymphoma classification*.

A future line of work is the validation of our model with other data sets as breast cancer, colon cancer, leukemia, etc. In addition, we want to use other classification models with the use of more complex structures for the Bayesian Network with the idea of study the dependencies among the genes.

Acknowledgments

This work has been supported by the Spanish Ministerio de Ciencia y Tecnología under Project TIC2001-2973-CO5-01.

References

1. Hand, D.: Discrimination and Classification. John Wiley (1981)
2. Friedman, N., Linial, M., Nachman, I., Pe’er, D.: Using bayesian networks to analyze expression data. Journal of Computational Biology **7** (2000) 601–620
3. Inza, I., Sierra, B., Blanco, R., Larrañaga, P.: Gene selection by sequential wrapper approaches in microarray cancer class prediction. Journal of Intelligent and Fuzzy Systems **12** (2002) 25–34
4. Langley, P., Iba, W., Thompson, K.: An analysis of bayesian classifiers. In: National Conference on Artificial Intelligence. (1992) 223–228

5. Domingos, P., Pazzani, M.J.: On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning* **29** (1997) 103–130
6. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. *Machine Learning* **29** (1997) 131–163
7. Langley, P., Sage, S.: Induction of selective bayesian classifiers. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence* (1994) 399–406
8. John, G.H., Kohavi, R.: Irrelevant features and the subset selection problem. In: *Proceedings of the Eleventh International Conference on Machine Learning*. (1994) 121–129
9. Golub, T.R. et al.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* **286** (1999) 531–537
10. Inza, I., Larrañaga, P., Blanco, R., Cerrolaza, A.: Filter versus wrapper gene selection approaches in dna microarray domains. *Artificial Intelligence in Medicine, special issue in Data mining in genomics and proteomics* **31(2)** (2004) 91–103
11. Hsu, C.N., Huang, H.J., Wong, T.T.: Why discretization works for naïve bayesian classifiers. In: *Proc. 17th International Conf. on Machine Learning*, Morgan Kaufmann, San Francisco, CA (2000) 399–406
12. John, G.H., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Mateo (1995) 338–345
13. Cowell, R., Dawid, A., Lauritzen, S., Spiegelhalter, D.: *Probabilistic Networks and Expert Systems*. Statistics for Engineering and Information Science. Springer-Verlag, New York (1999)
14. Wright, G., Tan, B., Rosenwald, A., Hurt, E.H., Wiestner, A., Staudt, L.M.: A gene expression-based method to diagnose clinically distinct subgroups of diffuse large b cell lymphoma. *Proceedings of National Academy of Sciences of the United States of America* **100** (2003) 9991–9996
15. Cano, A., Castellano, F.G., Masegosa, A., Moral, S.: Application of a selective gaussian naïve bayes model for diffuse large-b-cell lymphoma classification. In: *Proceedings of the Second European Workshop in Probabilistic Graphical Models*, Leiden, Holland (2004) 33–40
16. Kittler, J.: Feature set search algorithms. In Chen, C.H., ed.: *Pattern Recognition and Signal Processing*. Sijthoff & Noordhoff (1978) 41–60
17. Duda, R.O., Hart, P.E.: *Pattern Classification and Scene Analysis*. John Wiley Sons, New York (1973)
18. Stone, M.: An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. *Journal of the Real Statistical Society* **38** (1997) 48–47
19. Aha, D.W., Bankert, R.L.: Feature selection for case-based classification of cloud types: An empirical comparison. In: *Working Notes of the AAAI-94 Workshop on Case-Based Reasoning*, Seattle, WA, AAAI Press (1994) 106112
20. Langley, P., Sage, S.: Oblivious decision trees and abstract cases. In: *Working Notes of the AAAI-94 Workshop on Case-Based Reasoning*, Seattle, WA, AAAI Press (1994)
21. Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artificial Intelligence* **97** (1997) 273–324
22. Allmuallim, H., Dietterich, T.: Learning with many irrelevant features. In: *Ninth National Conference on Artificial Intelligence*, MIT Press (1991) 547–552
23. Alizadeh, A. et al.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* **403** (2000) 503–511

24. Zhang, H., Yu, C.Y., Singer, B.: Cell and tumor classification using gene expression data: Construction of forests. *Proceedings of the National Academy of Sciences* **100** (2003) 4168–4172
25. 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. *Bioinformatics* **17** (2001) 1131–1142
26. Ando, T., Katayama, M., Seto, M., Kobayashi, T., Honda, H.: Selection of causal gene sets from transcriptional profiling by fnn modeling an prediction of lymphoma outcome. *Gene Informatics* **13** (2002) 278–279
27. Rosenwald, A., Wright, G., Chan, W.C., Connors, J.M., Campo, E., Fisher, R.I., Gascoyne, R.D., Muller-Hermelink, H.K., Smealand, E.B., Staudt, L.M.: The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *New England Journal of Medicine* **346** (2002) 1937–1947

Towards a Definition of Evaluation Criteria for Probabilistic Classifiers

Nahla Ben Amor¹, Salem Benferhat², and Zied Elouedi¹

¹ Institut Supérieur de Gestion Tunis,
41 Avenue de la liberté, 2000 Le Bardo, Tunisie
{nahla.benamor, zied.elouedi}@gm.x.fr

² CRIL - CNRS, Université d'Artois,
Rue Jean Souvraz SP 18 62307 Lens, Cedex, France
benferhat@cril.univ-artois.fr

Abstract. This paper deals with the evaluation of "probabilistic" classifiers, where the results of the classification is not a unique class but a probability distribution over the set of possible classes. Our aim is to propose alternative definitions of the well known percent of correct classification (PCC) for probabilistic classifiers. The evaluation functions are called percent of probabilistic-based correct classification (PPCC). We first propose natural properties that an evaluation function should satisfy. Then, we extend these properties to the case when a semantic distance exists between different classes. An example of an evaluation function based on Euclidean distance is provided.

1 Introduction

Probabilistic classifiers (for instance naive Bayes classifiers) are one of well-known classification techniques used in the machine-learning community. These classifiers produce for each instance O , a probability distribution on the set of possible classes $\mathcal{C}$. In general, these probability distributions are not fully exploited. Indeed, they are used to first determine the most plausible classes, then to select one of the plausible classes to be the result of the classification of a given instance. Clearly, focusing on plausible classes, is not satisfactory even if it can make sense in some applications. In particular, it may happen that there exist more than one plausible class. In this case, probabilistic classifiers often proceed to an arbitrary choice of one class, among the most plausible ones. In many applications, this arbitrary choice is not desired and providing a probability distribution, or at least the set of the most plausible classes, is preferred.

The question addressed in this paper is how to evaluate probabilistic classifiers, when the classification result is no longer a unique class, but a probability distribution on the set of possible classes. Clearly, the well-known percent of correct classification (PCC), used to evaluate classifiers, is not appropriate since it ignores the probability distribution relative to different classes by only considering the most probable class.

There are several evaluation criteria that have been developed to compare two probability distributions. These criteria are generally induced from information theory and descriptive statistics. In particular, we can mention Kullback-Leibler (KL) divergence [6], cross entropy, χ^2 , squared errors etc.

This paper deals also with evaluation functions that take into account the whole probability distribution on different classes characterizing the uncertainty on the object to classify.

We first propose natural properties that any evaluation function should satisfy. These properties induce in general a partial pre-order between different probability distributions with respect to the real class. The top element of this pre-order is the probability distribution where the real class has a maximal degree namely 1 (hence alternative classes are impossible).

The minimal elements are those where the real class has the degree 0, and where there exists one alternative class that has the degree 1 (namely, minimal elements are probability distributions which completely missguess the result). We then extend these properties when a semantic distance exists between alternative classes. Namely alternative classes will no longer play symmetric roles, and some classes are considered more close to the real classes than others.

The rest of this paper is organized as follows: Section 2 fixes the notations used in this paper. Section 3 proposes natural properties for an evaluation function. Section 4 presents an example of an evaluation function based on Euclidean distance. Finally, in section 5, an extension of evaluation functions which take into account a semantic distance is proposed.

2 Notations

Let O be an instance to classify. We denote by $\mathcal{C} = \{C_1, \dots, C_n\}$ the set of different classes. The aim of a classifier is to classify the object O , on the basis of some training set, on one of the classes C_i . By convention, we consider that C_1 is the real class. By p_i we denote a probability distribution on $\mathcal{C}$ where $p_i(C_k)$ corresponds to the probability degree on the class C_k . The probability distribution p_i represents the result of the classification of the object O , using a probabilistic classifier (for instance naive bayes classifiers).

A probability distribution p_i will also be represented by a vector $\vec{p}_i = [p_i(C_1), \dots, p_i(C_n)]$. By $\sigma(\vec{p}_i)$ we denote a permutation of elements of the vector $\vec{p}_i$.

A class C_j is said to be *focal* in a probability distribution p_i if $p_i(C_j) > 0$. $nb\_focal\_elts(p_i)$ denotes the number of focal elements in p_i .

We denote by $p_\top$ the probability distribution such that $p_\top(C_1) = 1$ (and hence $\forall i \neq 1, p_\top(C_i) = 0$) and by $p_\perp$ any probability distribution such that $p_\perp(C_1) = 0$ and $\exists j \in \{2..n\}$ s.t $p_\perp(C_j) = 1$. If the classification result corresponds to $p_\top$, this means that the classifier provides a precise predicted class which is the real one. If the classification result corresponds to $p_\perp$, this means that the result of the classifier is precise and it corresponds to one of alternative classes $C_j \neq C_1$.

Our aim is to provide an evaluation function, called percent of probabilistic-based correct classification (PPCC), which evaluates probabilistic classifiers. This function evaluates the closeness of a probability distribution p_i with respect to $p_\top$.

More precisely let $\mathcal{P}$ be the set of probability distributions on $\mathcal{C}$ and let $D_{C_1}^{PCC}$ be a mapping from $\mathcal{P}$ to $[0, 1]$. By convention, the greater $D_{C_1}^{PCC}(p_i)$ is, the better p_i is. Intuitively, we expect $D_{C_1}^{PCC}(p_\top)$ to be equal to 1 since $p_\top$ guesses the real class C_1 . For the sake of simplicity, we use D instead of $D_{C_1}^{PCC}$. When $D(p_1) > D(p_2)$, p_1 is said to be more correct (or more close to the real class) than p_2 .

The following subsection provides some natural properties that the function D should satisfy.

3 Natural Properties of PPCC

The first property expresses the fact that alternative classes play a symmetric role. In particular, if one starts with a probability distribution p_1 and constructs a new probability distribution p_2 by only permuting probability degrees of alternative classes, then the two probability distributions have the same value of D . More formally:

Property 1. *Permutation of Alternative Classes*

• $p_1 \dots p_2$: p_1 and p_2 are two probability distributions on $\mathcal{C}$ such that $p_1(C_1) = p_2(C_1)$ and $p_1(C_i) = p_2(C_{\sigma(i)})$ for $i \in \{2..n\}$.
• $\vec{p_1} = \sigma(\vec{p_2})$ $\implies D(p_1) = D(p_2)$

The second property expresses the fact that the real class and alternative classes do not play symmetrical roles. The real class is more prioritary than each alternative class. More precisely, if one starts with a probability distribution p_1 , and constructs a new probability distribution p_2 by reinforcing the real class C_1 (namely, transferring some probability degrees from alternative classes to the real class C_1) then p_2 will be preferred to p_1 . More formally:

Property 2. *Reinforcement of Real Class*

• $p_1 \dots p_2$: p_1 and p_2 are two probability distributions on $\mathcal{C}$ such that $p_2(C_1) = p_1(C_1) + \alpha_1$ and $p_2(C_i) = p_1(C_i) - \alpha_i$ for $i \in \{2..n\}$ with $\alpha_1 > 0$ and $\sum_{i=2..n} \alpha_i = \alpha_1$.
• $\forall i \in \{2..n\}, p_2(C_i) = p_1(C_i) - \alpha_i$ and $\sum_{i=2..n} \alpha_i = \alpha_1$ and $\alpha_i > 0$.
• $D(p_2) > D(p_1)$

The third property expresses the fact that if p_1 and p_2 have the same probability degree on C_1 , then we prefer the one which minimizes the difference between probability degrees of alternative classes. In particular, if p_2 is constructed from p_1 , by transferring some probability degrees from an alternative class C_i , to another less probable alternative class C_j (while saving their preference order), then p_2 should be considered more correct than p_1 . More formally:

Property 3. "Balance" of Alternative Classes

- $p_1, \dots, p_n, \dots, p_n$: $i, j \in \{2..n\}, i \neq j$: $p_1(C_i) > p_1(C_j)$
- $\alpha > 0$: $p_1(C_i) - \alpha \geq p_1(C_j) + \alpha$
- $p_2, \dots, p_n, \dots, p_n$: $p_2(C_i) = p_1(C_i) - \alpha$
- $p_2(C_j) = p_1(C_j) + \alpha$: $k \neq i, k \neq j$: $p_2(C_k) = p_1(C_k)$
- $D(p_2) > D(p_1)$

The fourth property is called imprecision property. It concerns the case where the classification result is a set of classes $\mathcal{A}$ having the same probability degree, and where the real class also belongs to $\mathcal{A}$. Then the smaller the cardinality of $\mathcal{A}$ is, the better the result is. In other terms, the classification result is better when it is more specific.

Property 4. Imprecision

- $p_1, \dots, p_2, \dots, p_n$: $p_1(C_i) = p_2(C_i) = \frac{1}{n}$
- $C_1, \dots, C_n$: $p_1, \dots, p_2$
- $nb\_focal\_elts(p_1) = m < nb\_focal\_elts(p_2) = n$
- $\forall i \in \{1..m\}$: $C_i \in p_1, p_1(C_i) = \frac{1}{m}$
- $\forall i \in \{1..n\}$: $C_i \in p_2, p_2(C_i) = \frac{1}{n}$
- $D(p_1) > D(p_2)$

Note that if the result of classification is a set of classes A having the same probability degree, and that the real class does not belong to A , then using Property 3, the larger is the result, the better it is.

The following property says that if a given class C_i has the same probability degree in both p_1 and p_2 and that if p_1 is preferred to p_2 , then this preference should be preserved when the degree of C_i is equitably distributed over remaining classes. The converse is also derived. More formally:

Property 5. Preserving Conditioning

- $p_1, \dots, p_2, \dots, p_n$: $i \in \{1..n\}$
- $p_1(C_i) = p_2(C_i) = x > 0$: $p'_1, \dots, p'_2, \dots, p'_n$
- $p_1, \dots, p_2, \dots, p_n$: $p_1(C_i) = p_2(C_i) = x$
- $p'_1(C_i) = 0$: $p'_2(C_i) = 0$
- $\forall j \in \{1..n\}$: $j \neq i, p'_1(C_j) = p_1(C_j) + \frac{x}{n-1}$: $p'_2(C_j) = p_2(C_j) + \frac{x}{n-1}$
- $D(p_1) > D(p_2)$: $D(p'_1) > D(p'_2)$

In Property 5, it is very important to require that the alternative class C_i used in the previous property has the same degree in both p_1 and p_2 . Otherwise, we get an incompatible result with properties 1-4 as it is illustrated by the following example:

Example 1. $\vec{p}_1 = [0.6, 0.2, 0.2]$: $\vec{p}_2 = [0.6, 0.3, 0.1]$: $D(p_1) > D(p_2)$

$\vec{p}'_1 = [0.7, 0, 0.3]$: $\vec{p}'_2 = [0.75, 0, 0.25]$: $D(p'_1) > D(p'_2)$

Note also that in Property 5 the probability degree $p_i(C_i)$ is distributed over all classes. One may require that the probability degree will only be distributed over focal elements (as it is the case with probabilistic conditioning). This is not satisfactory since it can lead to a too strong result. Indeed, property 5 guarantees that the real class will be reinforced with a same degree in both distributions p_1 and p_2 . This is no longer guaranteed if the repartition concerns only focal elements.

We end this section by giving some propositions regarding evaluation functions that can be derived from properties 1-5. Then:

Corollary 1. *Top - Bottom*

- $D(p_\top) = 1$ and $D(p_\perp) = 0$
- $\forall p_i \neq p_\top, D(p_i) < D(p_\top)$
- $\forall p_i \neq p_\perp, D(p_i) > D(p_\perp)$

Property 6. $D(p_\top) = 1$ and $D(p_\perp) = 0$

The following two propositions concerns properties of probability distributions having the same degree of the real class. Let F_α be a family of probability distributions where the real class has a degree α , namely $\forall p \in F_\alpha, p(C_i) = \alpha$. Proposition 1 states that the preferred distribution in F_α is the one which has a uniform distribution on alternative classes. While, the worst distribution, within F_α , is the one having an alternative class with a probability degree equal to $1 - \alpha$.

Proposition 1. *Relative Top - Relative Bottom*

- $\forall p_\top^\alpha \in F_\alpha, \forall j \in \{2..n\}, p_\top^\alpha(C_j) = \frac{1-\alpha}{n-1}$
- $\forall p_\perp^\alpha \in F_\alpha, \exists j \in \{2..n\}, p_\perp^\alpha(C_j) = 1 - \alpha$

The following proposition relates two families of probabilities F_α and F_β

Proposition 2. *Family Top - Family Bottom*

- $\forall \alpha, \beta, \alpha > \beta$
- $\forall p_i \in F_\alpha, \exists p_j \in F_\beta, D(p_i) > D(p_j), D(p_\top^\alpha) > D(p_\top^\beta)$
- $\forall p_j \in F_\beta, \exists p_i \in F_\alpha, D(p_i) > D(p_j), D(p_\perp^\alpha) > D(p_\perp^\beta)$

4 Examples of Evaluation Functions

Generally classifiers are evaluated using the well-known percent of correct classification (PCC) criterion which corresponds to the proportion of the number of

well classified objects on the whole number of objects. The PCC relative to the whole testing set is computed by making comparison, for each testing instance, between its real class (known by us) and the class having the highest probability degree induced by the Bayesian classifier. Namely,

$$PCC = \frac{\text{number of well classified objects}}{\text{number of testing objects}} \tag{1}$$

where the number of well classified objects is computed as the sum of testing objects for which the class obtained by the classifier (the most probable class) is the same as their real class.

Let us first point out that the evaluation function implicitly used in (PCC) is not compatible with properties 1-5. This evaluation function is defined as:

$$D_{PCC}(p_i) = \begin{cases} 1 & \text{if } Select(p_i) = C_1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $Select(p_i)$ selects one class (often arbitrary) among plausible classes in p_i , namely selects one class from the set $\{C_j : \exists C_k, p(C_k) > p(C_j)\}$. This evaluation function is not compatible with properties 1-5. In particular D_{PCC} does not satisfy Property 2 as shown by the following counter example:

Counter-example 1.

$\vec{p_1} = [0.1, 0.6, 0.3]$, $\vec{p_2} = [0.3, 0.4, 0.3]$, $D(p_2) > D(p_1)$, D_{PCC}

An example of evaluation function which satisfies properties 1-6 is the one based on the well-known Euclidean distance. The idea is to compute (in terms of a distance) the difference between the probability distribution corresponding to the truth, known as $p_{\top}$, and the probability distribution p_i given by the classifier. More formally:

$$d(p_i) = (1 - p_i(C_1))^2 + c \sum_{j=2..n} p_i(C_j)^2 \tag{3}$$

where c is a calibrating constant pertaining to $]0, 1]$ allowing to discount the effect of alternative classes. When $c = 1$ we recover standard Euclidian distance, and also quadratic loss function used when the result is a probability distribution over different outcomes.

It is easy to check that $0 \leq d(p_i) \leq 1 + c$. In order to get a degree which pertains to the interval $[0, 1]$, we define the evaluation function as:

$$D(p_i) = 1 - \frac{d(p_i)}{1 + c} \tag{4}$$

When d is maximal D is minimal i.e. $D(p_i) = 0$.

Proposition 3. . D . D

A similar distance have been proposed for accessing the reliability of experts in the framework of transferable belief model [4], [10].

One of the most common measures for comparing probability distributions is Kullback-Leibler (KL) divergence [6] defined by:

$$d_{KL}(p_1, p_2) = - \sum_{1..n} p_1(C_i) \log \frac{p_1(C_i)}{p_2(C_i)} \tag{5}$$

If we adapt this distance to our context in order to develop an evaluation function we should compare a probability distribution p_i to the p_1 , equation (5) becomes:

$$d_{KL}(p_i) = \log_2 p_i(C_1) \tag{6}$$

This evaluation function is equivalent to the one based on informational loss function and only takes into account the probability assigned to the event that actually occurred. This clearly violates Property 3.

5 Semantic Distance

In many applications, alternative classes do not play symmetric roles. For instance, in intrusion detection problem [1] a connection can be classified as either a normal connection or an attack belonging to a list of reported attacks. These attacks are in general split into well known groups of attacks sharing similar properties (e.g. DOS, Probing, U2R, R2L). Therefore, if a connection is misclassified, the impact of the error is not always the same. For example, it is more risky to declare an R2L (Remote to User) attack as a normal connection than Probing if we refer to the cost matrix presented in [7].

In order to take into account interactions existing between different classes, we propose in this section, an extension of previous properties in order to evaluate probabilistic classifiers when a semantic distance exists between possible classes.

The idea of semantic distance is very close to the one of informational loss. However, here we assume a weak information on relative order between different outcomes, which is given by means of total pre-order between outcomes.

Let $\mathcal{L} = C_1 \geq_S C_2 \geq_S \dots \geq_S C_n$ be a total pre-order between different classes expressing a semantic distance between any alternative class and the real one. Namely, the class C_2 in $\mathcal{L}$ is the closest one to C_1 . More generally, C_i denotes here the i^{th} preferred class with respect to C_1 .

One natural requirement for semantic distance-based PCC is that in case where $C_2 =_S \dots =_S C_n$, one should satisfy properties 1-6 provided in previous section. More formally:

Property 7. $C_1 >_S C_2 =_S \dots =_S C_n \implies D(C_1, C_2) = D(C_1, C_n) = D(C_2, C_n) = 0$

Now if alternative classes have not strictly the same priority, it is obvious that properties 1-6 are not appropriate. Indeed, the permutation property (property 1) is meaningless since alternative classes are not interchangeable.

Property 1 can only be applied if C'_i are equally reliable. More precisely, this property is replaced by the following one:

Property 8. $C_i \dots C_j \dots C_i =_S C_j \dots p_1 \dots p_2 \dots C_i \dots C_j \dots p_2(C_i) = p_1(C_j), p_2(C_j) = p_1(C_i) \dots$
 $\forall j \in \{1..n\}, k \neq i, k \neq j, p_2(C_k) = p_1(C_k) \dots D(p_2) = D(p_1)$

Properties 2 and 3 can be substituted by the following one:

Property 9. *Prioritized reinforcement*

$$0 < \alpha < p_1(C_i) \quad p_2 \quad \dots \quad p_n \quad \dots \quad i, j \in \{2..n\} \quad \dots \quad C_i >_S C_j$$

- $$\begin{aligned} & -p_2(C_j) = p_1(C_j) - \alpha, \\ & -p_2(C_i) = p_1(C_i) + \alpha, \\ & -\forall k \in \{2..n\}, k \neq i \dots k \neq j, p_2(C_k) = p_1(C_k) \end{aligned}$$

$$D(p_2) > D(p_1)$$

Property 9 means that if one starts with a probability distribution p_1 and constructs a new one by transferring some probability degrees from a given class C_j to a more preferred class C_i (i.e. $i < j$), then p_2 is preferred to p_1 .

Example 2. $\vec{p}_1 = [0.3, 0.4, 0.2, 0.1]$, $\vec{p}_2 = [0.3, 0.5, 0.1, 0.1]$, $C_2 >_S C_3$, $D(p_2) > D(p_1)$

The preserving conditioning property (i.e. property 5) remains available in the semantic context. Indeed, let us consider two probability distributions p_1 and p_2 such that p_1 is preferred to p_2 . Suppose that there exists a class C_i having the same probability degree in both p_1 and p_2 . Then, if we dispatch the probability degree relative to C_i in a uniform manner on the remaining classes, then we should preserve the initial preference order.

Finally, properties 5 and 9 lead to the following proposition:

Proposition 4. $C_1 >_S C_2 >_S \dots >_S C_n$
 $D(p_1) > D(p_2)$
 $\forall i \in \{2..n\}, p'_1(C_i) = 0 \quad p'_2(C_i) = 0 \quad D(p'_1) > D(p'_2)$

Example 3. $\vec{p}_1 = [0.3, 0.4, 0.2, 0.1]$
 $\vec{p}_2 = [0.3, 0.5, 0.1, 0.1]$ $D(p_2) > D(p_1)$ ($\vec{p}_1 = [0.3, 0.4, 0.2, 0.1]$)
 $\vec{p}'_1 = [0.9, 0, 0.1, 0]$ $\vec{p}'_2 = [0.9, 0.1, 0, 0]$ $D(p'_1) > D(p'_2)$

Proposition 4 means that for strict orders, it is enough to compare probability distributions p_1 , p_2 such that each C_i is either impossible in p_1 or in p_2 .

We now provide some examples of evaluation functions using semantic distance. The first one uses cost-matrices which has been used for instance in evaluating classifiers in KDD'99 competition [7]. Basically, columns in the cost-matrix correspond to predicted class while rows correspond to real classes. Each element $c[i, j]$ of the cost-matrix indicates the "price to pay" if an instance has C_1 as real class and is declared as pertaining to C_j . $\forall j \in \{2..n\}$, $c[1, j] > 0$ and $c[1, 1] = 0$.

Let p_i be the resulted probability distribution, a possible definition of the evaluation function is:

$$d_{Cost}(p_i) = \sum_{j=2..n} c[1, j] * p_i(C_j). \quad (7)$$

The preferred solution is when $p_i(C_1) = 1$ (i.e. $d_{Cost}(p_i) = 0$) while the worst one is when $\exists j \neq 1$ such that $p_i(C_j) = 1$ and $c[1, j]$ is maximal. Hence in order to have an evaluation function pertaining to the interval $[0, 1]$, we can proceed as follows:

$$D_{Cost}(p_i) = 1 - \frac{d_{Cost}(p_i)}{\max_{j=2..n} c[1, j]} \quad (8)$$

Clearly, D_{Cost} , even if it is more appropriate than D_{PCC} is still not satisfactory. In particular, if the cost matrix is uniform, for instance $c[1, 1] = 0$, and $\forall j \in \{2..n\}$, $c[1, j] = 1$, then D_{Cost} does not satisfy property 7 (namely properties 1-6). For instance property 3 is not satisfied, as it is illustrated by the following counter example:

Counter-example 2. $\vec{p}_1 = [0.4, 0.5, 0.1]$, $\vec{p}_2 = [0.4, 0.4, 0.2]$, $c[1, 2] = c[1, 3] = 5$
 $D(p_2) > D(p_1)$, $D_{Cost}(p_2) = D_{Cost}(p_1) = 0.4$

Contrary to evaluation function 8, the following proposition satisfies properties 1-6 when all the costs are equal (i.e. $\forall j, k \in \{2..n\}$, $j \neq k$, $c[1, j] = c[1, k]$):

$$d_{DS}(p_i) = (1 - p_i(C_1))^2 + \sum_{j=2..n} c_j (\beta (1 - \sum_{k=1..j-1} p_i(C_k)) - p_i(C_j))^2 \quad (9)$$

where $\forall j \in \{2..n\}$, $c_j \in]0, 1]$

and $\beta = \begin{cases} 0 & \text{if } \forall i, j \in \{2..n\}, i \neq j, c_i = c_j \\ 1 & \text{otherwise} \end{cases}$

In this equation the c_j is not seen as a "price to pay", but as a closeness coefficient to the real class. In other terms if $C_i \geq_S C_j$ then $c_i \geq c_j$.

The preferred solution is when $p_i(C_1) = 1$ (i.e. $d_{DS}(p_i) = 0$) while the worst one is when $\exists j \neq 1$ such that $p_i(C_j) = 1$ and c_j is minimal. Hence in order to

have an evaluation function pertaining to the interval $[0, 1]$, we can proceed as follows:

$$D_{DS}(p_i) = 1 - \frac{d_{DS}(p_i)}{1 + \beta(\sum_{2..n} c_j - c_n) + (1 - \beta)c_n} \quad (10)$$

$D_{DS}(p_i) = 0$ if p_i is maximal in d_{DS} , $D_{DS} = 1$ if p_i is minimal in d_{DS} , and for each p_i and p_j , $D_{DS}(p_i) > D_{DS}(p_j)$ iff $d_{DS}(p_i) < d_{DS}(p_j)$.

When all the costs are equal equation (9) recovers equation (3) and equation (10) recovers (4). Thus from proposition 3 we can give the following proposition:

Proposition 5. $D_{DS} \dots, C_1 >_S C_2 =_S \dots =_S C_n \dots D_{DS} \dots$

6 Conclusion

This paper deals with the problem of evaluation "probabilistic" classifiers when a probability distribution is provided on different classes. We propose a preliminary set of natural properties that any evaluation function should satisfy. Then, we extend them when some semantic distances between classes are provided. Clearly, natural properties that any proposed in this paper are minimal. A future work will be to characterize evaluation functions satisfying stronger properties (for instance the counter-part of property 1). We will also study evaluation functions based on well-known distance based on entropy function [9]. Another future work is to investigate evaluation functions when the classification result is a possibility distribution [11] or a mass function [10], and to apply them on real world classification problems such intrusion detection.

Acknowledgments. We would like to thank anonymous referees for numerous comments that helped us to improve this paper. This work was supported by the french national project Action Concerte Incitative (ACI) securit et informatique, DADDi (Dependable Anomaly Detection with Diagnosis).

References

1. Axelsson S. (2000): Intrusion detection systems: a survey and taxonomy, Technical report, 99-15.
2. Chan H., and Darwiche A. (2005): A Distance measure for bounding probabilistic belief change, International Journal of Approximate Reasoning (IJAR), 38, 149-174.
3. Duda R. and Hart P. (1973): Pattern Clasification and Scene Analysis. Wiley-Interscience.
4. Elouedi Z., Mellouli K. and Smets P. (2004):Assessing Sensor Reliability for Multisensor Data Fusion with the Transferable Belief Model", IEEE Transactions on System Man and Cybernatics - Part B, 34, 782-787.

5. Garg A. and Roth D. (2001): Understanding probabilistic classifiers. European Conference on Machine Learning (ECML'2001).
6. Kullback S. and Leibler R.A. (1951): On information and sufficiency, *Annals of Mathematical Statistics*, 22, 79-86.
7. <http://kdd.ccs.uci.edu/databases/kddcup99/task.html>.
8. Langley P., Iba W. and Thompson K. (1992): An analysis of Bayesian classifiers. In *Proceedings, Tenth National Conference on Artificial Intelligence*, Menlo Park, CA: AAAI Press, 223-228.
9. Shannon C. E. (1948): A mathematical theory of communication, *The Bell Systems Technical Journal*, 27, 3, 379-423, 623-656.
10. Smets P., and Kennes R. (1994): The transferable belief model, *Artificial Intelligence*, 66, 191-234, 1994.
11. Zadeh, L. A. (1978): Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems* 1: 3-28.

Methods to Determine the Branching Attribute in Bayesian Multinets Classifiers

A. Cano, J.G. Castellano, A.R. Masegosa, and S. Moral

Departamento de Ciencias de la Computación e Inteligencia Artificial,
Granada University, Granada, Spain
{acu, fjgc, andrew, smc}@decsai.ugr.es

Abstract. Bayesian multinets are a Bayesian networks extension where context-specific conditional independences can be represented. The main aim of this work is to study different methods to choose the distinguished attribute in Bayesian multinets when we use them in supervised classification tasks. We have used different approaches: a wrapper method and several filter methods. This will allow us to determine the most appropriate approach that meets our requirements of accuracy and/or time.

1 Introduction

In the supervised classification problem [1] we have a set of observations or cases, made up of a series of attributes or values that we observed and a variable that we want to predict, this is called variable to classify or, simply, class. The classifier obtains a set of rules or laws to predict a value of the class variables for each new observation using the other attributes or predictive variables. In order to build the classifier, different approaches can be used like statistical methods [2], classification trees [3], artificial neural networks or Bayesian classifiers [4, 5].

In Bayesian multinets [6] we have a distinguished variable and a Bayesian network for each value of this variable. Intuitively, we can see a Bayesian multinet as a depth-one classification tree where we have Bayesian networks at the leaves.

In Bayesian multinets classifiers, we can distinguish two types: those in which the distinguished variable is the same variable to classify and a second type where the distinguished variable is an attribute, in this second type we can recursively choose a variable, obtaining a tree structure [7]. The Bayesian multinets are a natural extension of the Bayesian networks methodology. Multinets allow us to work with *context-specific conditional (in)dependencies* [8], also called *asymmetric conditional independences* [6] which is not possible with usual Bayesian networks that only can encode *context-non-specific conditional (in)dependencies* [8] also known as *symmetric conditional independences* [6].

In multinets where the distinguished variable is an attribute, the main problem is to find the best branching variable to build the multinet that better represents the cases. In this search we have followed various approximations: a wrapper approach and several filter approaches. In the wrapper method [9] the goodness for each variable is computed by the estimation of the accuracy of the

classifier. In the filter approach [10] the goodness estimation for each variable is based only in the characteristics of the data. In this paper we have studied this problem in the particular case where we have a Naive Bayes classifier at the leaves.

In the field of Bayesian Classifiers, the first classifier which can be considered as a Bayesian network classifier, is the *Naive Bayes* (NB) classifier [11, 12]. This simple classifier relies on two assumptions: (1) each attribute is conditionally independent from the other attributes given the class and (2) all the attributes have influence on the class. In the DAG structure representing NB all the arcs are directed from the class to the attributes. The success of this classifier is mainly due to its simplicity and exhibits a surprisingly competitive predictive accuracy. It outperforms many more sophisticated classifiers over a large amount of datasets (specially when the attributes are not strongly correlated) [12, 4]. At the other extreme in Bayesian classifiers, any algorithm for learning Bayesian networks can be used to induce a classifier. The relative success of the NB classifier has motivated the development of many methods, which try to improve it. One way is, starting from the NB basic topology, complete it by adding *augmenting arcs* between attributes, for example, *tree-augmented naive Bayesian network* (TAN) [4], where the attributes are linked together as a directed tree or the *Bayesian network augmented naive Bayesian classifier* (BAN). In this case the NB structure is fixed and the search of the augmenting arcs is carried out using some algorithm for learning Bayesian networks [13]. There are also other variations of the NB model, this is the case of the *semi-naive Bayesian classifier* [14] and the model proposed by [15]. They share the idea of merging several correlated attributes into new compound attributes. Other approximations remove correlated and irrelevant attributes using *feature subset selection* (FSS) to improve the naive Bayes classifier, the *selective naive Bayesian classifier* [16, 17] searches for a good subset of attributes, starting from an empty set and iteratively introducing the attribute that most improves the predictive accuracy, using cross-validation.

The rest of the paper is organized as follows. Section 2 describes Bayesian multinets and we show the different approaches used in our experiments. Later Section 3 contains the experimental results of the evaluation for the different proposed methods on a wide range of databases. Finally, Section 4 contains the concluding remarks.

2 Bayesian Multinets

Bayesian networks allow us to represent the distribution properties. In the numerical model, these independences correspond with assertions of the type $P(X|YZ) = P(X|Z) \forall x \forall y \forall z$, which has to be verified for all the possible cases of the variables. If for a case $Z = z$ this equality is not verified then X and Y can not be considered independent, although they are independent for the rest of the cases. This type of independences (where it is supported only for some instances of its variables) are known as *context-specific conditional independences*

[8]. There are different formalisms that can represent these new forms of independence, that enlarge the power of representation of Bayesian networks and at the same time improve the efficiency of the inference. Some of these formalisms are *Similarity Networks* [18] or *Bayesian Multinets* (BMN) [6].

Heckerman [18] distinguishes between two types of asymmetric independences: the first one that he calls *of subset* (when there is a relation among the variable to classify and the attributes) and the other called *of hypothesis-specific* (when there is a relation among only attributes). Multinets where the distinguished variable is the class (so a Bayesian network will be built for each class value) try to represent *asymmetric independences of subset*. While the multinets with an attribute as distinguished variable try to represent *asymmetric independence of hypothesis-specific*. In this paper we are going to focus in the multinets where the distinguished variable is an attribute and we have Naive Bayes at the leaves. In this type of multinet classifier the main problem is the selection of the branching attribute, once this variable is selected we have to build a classifier for each of its cases. As we have Naive Bayes classifiers at the leaves the multinet construction is very simple. In the following, we describe the methods that we have tested to select the branching attribute.

2.1 Wrapper Bayesian Multinet (BMN_w)

In the wrapper algorithm for each attribute X_i , we compute the classifier expanding this variable and determining its accuracy, which will be computed using 5-fold cross-validation. Afterward the node X_j with the greatest accuracy is chosen to branch.

One must keep in mind that this algorithm basically builds all the possible Bayesian multinets and evaluate every one of them, therefore, is quite expensive in time. This is the reason that we only study a type of wrapper Bayesian multinet that will use Naive Bayes in the leaves (BMN_wNB), since the Naive Bayes classifier is very fast to build.

The *Naive-Bayes Tree* (NBTree), proposed by Kohavi in [19], combines the classification tree ideas and Naive-Bayes, having Naive-Bayes in the leaves of the classification tree, can also be seen as a recursive multinet with Naive Bayes in the leaves. The multinet wrapper that we are going to use in our experiments is equal to NBTree but only explores one level, that is, it is not recursive.

We must have take into account that if would not make sense to use a wrapper Naive Bayes Bayesian multinet to build the classifier and once we have found the best branching variable, try to use afterwards another Bayesian classifier at the leaves. Keep in mind that we are optimizing the multinet for using Naive Bayes and not another classifier. Nevertheless, the filter functions are independent of the classifier that we utilize at the leaves and, these filter methods will permit us use any classifier.

2.2 Filter Bayesian Multinet (BMN_f)

In filter Bayesian multinet we use a function that computes for each attribute how good is to branching the multinet. Some of the functions that we have

chosen already have been used in the literature and other are heuristics that we propose. In this functions $X = (x_1, x_2, \dots, x_{r_x})$ is the attribute to measure, $C = (c_1, \dots, c_{r_c})$ is the class variable and n the number of cases.

Filter KullbackLeibler Distance (BMN_{f1}). The Kullback-Leibler[10] divergence is the best-known method for distances measurement between two probability distributions. Its generic formulation for dicotomic problems (two states for the class) is:

$$D_{kl}(P(X), Q(X)) = \sum_{x_i} P(x_i) \log \frac{P(x_i)}{Q(x_i)}$$

For multiclass problems (more than two states for the class), there was necessary to determine which are the two distributions to be used. In cite [20] two possibilities were taken, to compare the 'a priori' marginal odds, and the 'a priory' conditional odds, that is the one that we will use. The divergence is formulated as:

$$KL_{ij}(X; C)_2 = D_{kl}(P(X|c_i), P(X|c_j)) + D_{kl}(P(X|c_j), P(X|c_i))$$

This value measures the degree of dependence between the attribute X and the class C , as if the variables are independent, then this value is 0. The idea is to select the variable with the highest degree of dependence with the class.

Filter Matusita Distance (BMN_{f2}). The original formulation of this metric [10] measures the distance between two probability distributions. In multi-class problems [20], it tries to measure the average distance among the different marginal distributions for each value of the attribute with the values of the class. Its mathematical expression is formulated as:

$$D_m(X; C) = \sum_{i=1}^{r_c} \sum_{j=1}^{j < i} P(c_i) P(c_j) \left[\sum_{k=1}^{r_x} \left(1 - \sqrt{P(x_k|c_i) P(x_k|c_j)} \right) \right]$$

This function is an alternative procedure to measure the degree of dependence between X and C by considering the averaged distance of the conditional distributions. With respect to Kullback Leibler distance, it has the advantage of considering the 'a priori' probabilities of the class values. On the other hand, it is more related with absolute differences of probabilities, being less sensible than Kullback Leibler distance to relative divergences: $P(x_k|c_i)/P(x_k|c_j)$.

Filter Gain (BMN_{f3}). This function computes the gain for each variable, this method was used by Quinlan [3] to select the best variable to branch in classification trees at the ID3 algorithm:

$$Gain(X; C) = H(C) - H(C|X) = \sum_{i=1}^{r_x} \sum_{j=1}^{r_c} P(x_i, c_j) \log \frac{P(x_i, c_j)}{P(x_i) P(c_j)}$$

This is also called mutual information and denoted as $MI(X, C)$, and it is the most usual way of measuring the degree of dependence between X and C .

Filter Gain Ratio (BMN_{f4}). This function computes the gain ratio for each variable, this method was used by Quinlan [3] to select the best variable to branch in classification trees at the C4.5 algorithm. In addition to the calculus of the gain, in C4.5 it is calculated the *Split Info* and the *Gain Ratio*. For an attribute X and a node N with training set T , the gain ratio is defined as follows:

$$GainRatio(X; C) = \frac{Gain(X; C)}{SplitInfo(X)}$$

where,

$$SplitInfo(X) = \sum_{j=1}^{r_x} p(x_j) \log_2(1/p(x_j))$$

This measure is a modification of the mutual information and was proposed by Quinlan [3] to avoid a bias of $Gain(X; C)$ toward the selection of variables with a higher number of cases.

Filter Heuristic Bayesian Score of the Attribute as Parent of the Rest ($BMN_{f5}, BMN_{f6}, BMN_{f7}$). In this heuristic we build a Bayesian net for each attribute X_i , where this attribute is the parent of the rest of variables, including the class. All the variables (except X_i) form a Naive Bayes structure, as we can see in Figure 1(a). Once the network is built we compute its Bayesian score. In our experiments we have tested three scores K2 [21] (BMN_{f5}), BIC [22] (BMN_{f6}) and BDe [23] (BMN_{f7}). The reason of this heuristic is that the used structure is equivalent to a Naive Bayes multinet. This filter method also suffers from the same defect that we had attributed to the wrapper method, this is, only would be completely appropriate for Bayesian multinets with Naive Bayes at the leaves.

This measure follows a different intuitive idea than former ones. It takes into account not only the relationship of a variable X_i with the class variable, but also how X_i affects the rest of the attributes. It is based on a Bayesian network representation of a multinet with X_i as branching node. We have that for each

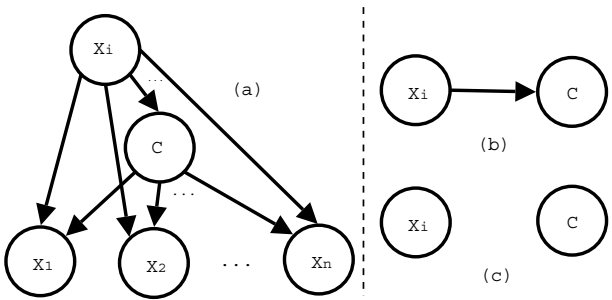


Fig. 1. (a) Structure used in *Bayesian Score of the attribute as parent of the rest*; (b) First structure used in *Bayesian Score of the attribute as parent of the class*; (c) Second structure used in *Bayesian Score of the attribute as parent of the class*

value of X_i , there is a Naive Bayes classifier based on the rest of attributes. This implies that given X_i and C the rest of variables are mutually independent. These are exactly the independence relationships represented in Figure 1(a). So, the score of a variable is obtained by computing the score of the Bayesian network representing the independences associated with the Naive Bayes multinomial with X_i as branching node.

Filter Heuristic Bayesian Score of the Attribute as Parent of the Class ($BMN_{f8}, BMN_{f9}, BMN_{f10}$). Here we compute for each variable X_i the score of the Bayesian net where that attribute is the parent of the class variable 1(b) (the other attributes does not take into account) minus the score of the class variable 1(c). In this method also we use the following Bayesian scores K2 (BMN_{f8}), BIC (BMN_{f9}) and BDe (BMN_{10}).

This is also a measure of the dependence of X_j and C by means of a Bayesian score: the difference between the scores of the Bayesian network expressing the dependence of the two variables and the Bayesian network representing their independence.

Filter Conditional Mutual Information (BMN_{f11}). In this case for each variable X_i we obtain the sum of the conditional mutual information given the class with X_j ($i \neq j$). We select the attribute with the highest value

$$CMuI_{f11}(X_i) = \sum_j MI(X_i, X_j|C).$$

where $MI(X_i, X_j|C) = \sum_{k=1}^{r_c} P(c_k)MI((X_i|C = c_k), (X_j|C = c_k))$ is the conditional mutual information between X_i and X_j given C .

This measure takes also into account the relationships of X_i with the rest of variables. The intuitive idea is that if a Naive Bayes model was the true model, then any couple of variables should be conditional independent given the class, and this measure is always 0.0. Building a multinomial with X_i as branching variable implies to remove X_i from the Naive Bayes model and giving it a most important role (the model is conditioned to the possible values of X_i). So, there is no necessity that this variable is conditionally independent of the rest of variables given the class. In this way, we select the variable which is most dependent with the rest of variables given the class, and this is a measure of how this variable breaks the conditions underlying the Naive Bayes model.

Filter Bhattacharyya (BMN_{f12}). This distance [10] measures the dependence that exists between two probability distributions. In the multiclass case [20] the distributions that will be compared are the 'a priori' distribution for each attribute against the 'a posteriori' distribution given a class value. We try to see what degree of dependence we find among both distributions; as greater be that degree, greater will be the weight of the analyzed variable. Its formulation is:

$$Bh(X; C) = \sum_{i=1}^{r_c} -\log \left[P(c_i) \sum_{j=1}^{r_x} \sqrt{P(x_j|c_i)P(x_j)} \right]$$

This is also a measure of the degree of dependence of X and C , very similar in behavior to Matusita distance.

Filter Gain Dirichlet (BMN_{f13} , BMN_{f14}). In this case we use a smoothed gain to select the variable [24]. We use two version depending of the value for the smoothed parameter s . In this case we try $s = 1$ (BMN_{f13}) and $s = 2$ (BMN_{f14}). In the gain function the probabilities are estimated by maximum likelihood:

$$P(c_j) = freq(c_j)/n$$

here, we makes a Bayesian estimation assuming 'a priori' symmetrical Dirichlet distribution:

$$p^d(c_j) = (s/r_c + freq(c_j))/(s + n)$$

This is modification of the information gain, trying to avoid the effect of maximum likelihood estimation of the probabilities used in its computation. When using maximum likelihood estimation of probabilities, then there is bias with a tendency to overestimate the real information gain between the variables. This measure tries to avoid this bias, but using a Bayesian estimation of probabilities. In this way, we obtain lower estimations of the mutual information than using maximum likelihood probabilities.

3 Experimental Results

We have selected 25 well-known data sets, obtained from the *UCI repository of machine learning databases* [25], except "mofn-3-7-10" and "corral", that were designed in [9]. All these data sets have been widely used in the specialized literature for comparing classifiers.

Table 1 gives is a brief description of the characteristics of the databases: the *Instances* column displays the number of cases, *Attributes* gives the number of attributes used to classify, and *Classes* shows the number of states for the class variable. These data sets have been preprocessed in the following way: the continuous variables have been discretized; the measure used for discretization was the entropy, where the number of intervals was not fixed, it was obtained following the procedure proposed in [26]. The cases with undefined/missing values were eliminated. For this preprocessing stage we have used the MLC++ System [27], available at <http://www.sgi.com/tech/mlc>. For the experiments we have used *10-fold cross validation* to compute the accuracy for each method. We also have utilized the Wilcoxon paired signed Rank Test to determine if the results are statistically significant.

All the algorithms are implemented in the Elvira System [28] and all the experiments have also been carried out with this software. The Elvira System is a Java tool to construct probabilistic decision support systems, which works with Bayesian networks and influence diagrams.

Our objective is to find the most adequate method to build the multinet according to our needs, in this task we will study the wrapper method of section

Table 1. Description of the data sets used in our experiments

#	Data Set	Instances	Attributes	Classes
1	australian	690	14	2
2	breast	682	10	2
3	chess	3196	36	2
4	cleve	296	13	2
5	corral	128	6	2
6	crx	653	15	2
7	diabetes	768	8	2
8	flare	1066	10	2
9	german	1000	20	2
10	glass	214	9	7
11	glass2	163	9	2
12	heart	270	13	2
13	hepatitis	80	19	2
14	iris	150	4	3
15	letter	20000	16	26
16	lymphography	148	18	4
17	mofn-3-7-10	1324	10	2
18	pima	768	8	2
19	satimage	6435	36	6
20	segment	2310	19	7
21	shuttle-small	5800	9	7
22	soybean-large	562	35	19
23	vehicle	846	18	4
24	vote	435	16	2
25	waveform-21	5000	21	3

2.1 and the filters methods seen in 2.2. In Table 2 we can see the results of this comparative study: for each classifier and data set we show the predictive accuracy (the percentage of successful predictions on a test set different from the training set).

We can observe that the wrapper multinet obtains better results in most cases than the filter methods. This differences seem greater in problems with a great number of attributes as *chess*, *waveform-21*, *letter* or *satimage*. This is due to the exhaustive accuracy oriented search of the wrapper method. This is specially significant on these problems with a great number of variables. But, on the negative side we must take into account that this search of the wrapper method consumes too much time because it builds a multinet classifier for each variable and computes its accuracy.

In Table 3 we compare each classifier with the others. The entry in row *i* column *j* represents the number of times that classifier *i* is better/significantly better (using the Wilcoxon test) than classifier *j*. Each row displays the times that the corresponding classifier is better, whereas each column says how many times the classifier is worse. These numbers clearly indicate that the wrapper approach does indeed behave generally better than the others, specially when we

Table 2. Experimental Results: Predictive accuracy and average ratios of running times

#	w	$f1$	$f2$	$f3$	$f4$	$f5$	$f6$	$f7$	$f8$	$f9$	$f10$	$f11$	$f12$	$f13$	$f14$
1	85, 80	86, 38	86, 38	86, 38	86, 38	86, 52	86, 52	86, 52	86, 38	86, 38	86, 38	84, 49	86, 38	86, 38	86, 38
2	95, 90	96, 19	96, 19	96, 19	95, 31	96, 19	96, 19	96, 62	96, 19	96, 19	96, 19	96, 19	96, 19	96, 19	96, 19
3	94, 27	93, 52	93, 52	93, 52	93, 52	84, 58	84, 58	84, 58	93, 52	93, 52	93, 52	84, 58	93, 52	93, 52	93, 52
4	81, 41	81, 78	82, 12	82, 12	82, 09	82, 44	82, 78	82, 78	81, 78	81, 78	81, 78	83, 82	82, 12	81, 78	81, 78
5	89, 10	85, 19	85, 19	85, 19	85, 19	87, 56	88, 40	87, 56	85, 19	85, 19	85, 19	88, 40	85, 19	85, 19	85, 19
6	85, 74	86, 97	86, 97	86, 97	86, 97	86, 97	86, 66	86, 97	86, 97	86, 97	86, 97	85, 75	86, 97	86, 97	86, 97
7	76, 95	77, 07	77, 07	77, 07	77, 07	77, 61	77, 61	77, 61	77, 07	77, 07	77, 07	77, 61	77, 07	77, 07	77, 07
8	82, 08	81, 80	81, 80	81, 80	78, 89	81, 80	81, 80	81, 80	81, 80	81, 80	81, 80	81, 80	81, 80	81, 80	81, 80
9	73, 90	75, 20	75, 20	75, 20	75, 20	73, 00	74, 90	75, 50	75, 20	75, 20	75, 20	73, 50	75, 20	75, 20	75, 20
10	74, 85	71, 06	72, 97	72, 99	72, 47	75, 78	73, 42	73, 42	74, 85	74, 85	74, 85	72, 97	72, 97	73, 46	72, 99
11	84, 71	84, 71	84, 71	84, 71	82, 87	84, 71	84, 12	85, 33	84, 71	84, 71	84, 71	84, 12	84, 71	84, 71	84, 71
12	81, 48	82, 96	82, 96	82, 96	82, 96	82, 22	83, 70	83, 70	82, 96	82, 96	82, 96	83, 33	82, 96	82, 96	82, 96
13	87, 50	87, 50	87, 50	87, 50	87, 50	90, 00	85, 00	85, 00	87, 50	87, 50	87, 50	88, 75	87, 50	87, 50	88, 75
14	93, 33	94, 67	94, 00	94, 00	94, 00	94, 00	94, 67	93, 33	94, 00	94, 00	94, 00	94, 67	94, 00	94, 00	94, 00
15	83, 57	82, 40	82, 40	82, 40	78, 93	83, 57	73, 99	79, 91	82, 40	82, 40	82, 40	83, 57	82, 40	82, 40	82, 40
16	82, 43	82, 33	79, 67	79, 67	81, 05	82, 43	85, 10	84, 48	82, 33	82, 33	82, 33	77, 81	83, 05	82, 33	82, 33
17	93, 81	93, 88	93, 88	94, 03	94, 03	94, 79	94, 79	94, 79	94, 03	94, 03	94, 03	94, 71	93, 88	94, 03	94, 03
18	77, 73	77, 47	77, 47	77, 47	77, 47	77, 73	77, 73	77, 73	77, 47	77, 47	77, 47	77, 73	77, 47	77, 47	77, 47
19	86, 01	84, 29	85, 21	84, 77	84, 29	86, 01	85, 49	86, 01	84, 29	85, 11	84, 29	86, 01	84, 29	84, 77	84, 77
20	94, 98	93, 81	93, 81	93, 81	93, 20	95, 02	92, 08	95, 02	93, 81	93, 81	93, 81	95, 02	93, 81	93, 81	93, 81
21	99, 60	99, 17	99, 17	99, 17	99, 17	99, 53	99, 00	99, 53	99, 17	99, 17	99, 17	99, 17	99, 17	99, 17	99, 17
22	91, 81	91, 64	85, 58	91, 46	91, 46	91, 81	91, 46	92, 00	91, 64	91, 64	91, 64	85, 58	85, 58	91, 64	91, 64
23	69, 76	71, 17	71, 17	71, 17	67, 39	70, 82	69, 88	70, 47	71, 17	70, 70	70, 23	71, 17	71, 17	71, 17	71, 17
24	94, 28	92, 20	92, 20	92, 20	92, 20	93, 12	93, 12	93, 12	92, 20	92, 20	92, 20	93, 82	92, 20	92, 20	92, 20
25	82, 94	81, 72	81, 72	81, 72	81, 52	81, 52	81, 52	81, 52	81, 72	81, 72	81, 72	81, 72	81, 72	81, 72	81, 72
t	1	140	130	80	80	7	21	7	65	99	65	4	142	79	79

focus in differences that are statistical significant. Following the wrapper search, we have the heuristic *Bayesian Score of the attribute as parent of the rest* with K2 metrics (BMN_{f5}). This heuristic is better than the wrapper search in 13 cases, while the wrapper search is only better in 8 cases, although only two of them are statistically significant. With a similar behavior, rather worse than BMN_{f5} , we find the following filter methods: *Bayesian Score of the attribute as parent of the rest* with BDe metrics (BMN_{f7}), *Bayesian Score of the attribute as parent of the class* with BIC metrics (BMN_{f9}) and *Conditional mutual information* (BMN_{f11}). Also we want to remark the good results obtained by *filter Bhattacharyya* (BMN_{f12}), but worse than the mentioned filters and worse than the wrapper search.

In Table 2 (below) we show average ratios of running times spent by each multinet to the time spent by the wrapper approach. As expected, the wrapper multinet is slower than the filter multinets. According to this results we can see three subsets of methods: first, a subset with the wrapper approach; second, another subset with a normal consumption of time (BMN_{f5} , BMN_{f7} , BMN_{f9} , BMN_{f11} and BMN_{f15}) and a third group where we have fast methods (the rest of filter methods). Attending to the previous classification we observe that also those groups represents different accuracy subtypes. In this way, the slow method (wrapper only) obtains better results, the normal methods get very similar results with less time, and faster methods have worse accuracy although the consumed time is quite smaller. In the second group we see that BMN_{f5} obtains a very similar accuracy than the wrapper search although is almost 8

Table 3. Number of times that the multinet in row i is better/significantly better than the multinet in column j

	w	$f1$	$f2$	$f3$	$f4$	$f5$	$f6$	$f7$	$f8$	$f9$	$f10$	$f11$	$f12$	$f13$	$f14$
w	—	13/7	13/6	13/5	16/8	8/2	13/6	10/3	12/6	12/5	12/6	13/4	12/6	13/5	13/7
$f1$	10/0	—	3/1	3/0	10/3	6/1	11/3	6/2	1/0	2/0	2/1	7/2	2/1	1/0	1/0
$f2$	10/0	3/1	—	1/0	10/4	5/1	10/3	6/2	2/1	3/0	3/2	6/1	1/1	2/0	2/0
$f3$	10/0	4/0	3/1	—	10/3	5/1	10/3	6/2	2/0	2/0	3/1	8/2	4/1	1/0	1/0
$f4$	8/0	3/0	3/1	1/0	—	3/1	7/2	3/1	1/0	1/0	1/0	6/2	2/1	1/0	1/0
$f5$	13/0	16/4	16/4	16/3	20/5	—	11/4	5/2	16/4	17/3	17/4	9/3	15/6	16/3	16/3
$f6$	11/0	11/0	13/1	13/0	17/1	6/1	—	3/0	11/0	11/0	11/0	8/2	13/2	11/0	12/0
$f7$	13/0	18/3	18/3	18/2	21/5	7/1	11/3	—	17/3	17/2	18/3	11/4	18/5	17/2	18/2
$f8$	10/0	2/0	4/1	3/0	10/2	5/1	12/3	7/2	—	1/0	1/1	8/2	3/1	1/0	1/0
$f9$	10/0	3/0	4/1	4/0	11/2	4/1	12/3	7/2	1/0	—	2/1	8/2	4/2	2/0	2/0
$f10$	10/0	2/0	4/1	3/0	10/1	4/1	12/3	6/2	0/0	0/0	—	8/2	3/1	1/0	1/0
$f11$	10/0	12/3	12/2	12/2	18/4	8/0	9/2	8/1	12/3	13/2	13/3	—	12/3	12/2	11/2
$f12$	11/0	3/0	1/0	1/0	10/2	6/1	10/3	6/2	2/0	3/0	3/0	6/1	—	2/0	2/0
$f13$	10/0	3/0	4/1	3/0	11/2	5/1	12/3	7/2	1/0	1/0	2/0	8/2	4/1	—	1/0
$f14$	11/1	4/0	5/1	3/0	12/2	5/1	11/3	6/2	2/0	2/0	3/0	8/2	5/1	1/0	—

times faster and in the third the filter BMN_{f12} obtains good results being 142 times faster than the wrapper multinet.

We must keep in mind that the wrapper search is better in databases with a great number of attributes due to a thorough search, but this deeper search has also a high computation cost in time; in our experiments, we can appreciate this in *chess*, *waveform-21*, *satimage* and *soybean-large* databases. Usually a great number of variables implies more complex (symmetric and asymmetric) conditional independence relations between them. For this reason an exhaustive search has a tendency to be better. In data sets with simpler conditional independences, we can use a filter method. In our experiments filter methods work better with *german* or *crx*. Usually, this effect is present in medicine data sets (see *pima*, *breast*, *diabetes* or *heart*) merely because the diagnosis attributes are somewhat independents given the disease.

Summarizing the results, we can choose the wrapper multinet for those cases in which there are no time problems. For those problems in which the computing time is fundamental we can use the BMN_{f12} filter. In the middle point, with good accuracy and good timings, we can use BMN_{f5}

4 Final Remarks

In this work a wide experimental study has been carried out on different methods to obtain the distinguished variable in Bayesian multinets classifiers with Naive Bayes at the leaves. In the studied methods the wrapper approach obtains the best classification predictions, which is reasonable if one keeps in mind that this measure tries to maximize the percentage of successful predictions. However, we have shown filter methods with a very similar accuracy with a smaller running times. The filter methods are always very fast in time which makes them appropriate for problems with a great number of cases or a great number of attributes, where we can not use the wrapper approach. Also we must consider that the filter methods are independent from the classifier used at their leaves, so

this methods can be used with better Bayesian classifiers than the Naive-Bayes. This can not be said of the wrapper multinet, if this algorithm uses Naive-Bayes in the search process, it should use the same classifier when we have chosen a distinguished or branching variable. Finally, we have selected three methods to find the branching variable (BMN_w , BMN_{f5} and BMN_{f12}) to meet our needs.

Now, we are working on the use of hybrid approaches that obtain a good performance in accuracy and running time. These hybrid methods make a variable ranking with a filter method and get a number of the best ranked variables for a wrapper search. For the filter methods we are studying better Bayesian classifiers than the Naive-Bayes classifier. In the near future, we will search for stopping conditions in recursive multinets, using different types of Bayesian classifiers in the same multinet and keep searching better filter methods.

Acknowledgments

This work has been supported by the Spanish Ministerio de Ciencia y Tecnología under Project TIC2001-2973-C05-01. We want thank Rubén Armañanzas for his help in some filter methods.

References

1. Hand, D.J.: Construction and Assessment of Classification Rules. John Wiley and Sons, New York (1997)
2. Hand, D.: Discrimination and Classification. John Wiley (1981)
3. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann Publishers Inc. (1993)
4. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian networks classifiers. Machine Learning, 29 (1997) 131–163
5. Pearl, J.: Probabilistic reasoning in intelligent systems. Morgan Kaufmann, San Mateo, CA (1988)
6. Geiger, D., Heckerman, D.: Knowledge representation and inference in similarity networks and bayesian multinets. Artificial Intelligence, 82 (1996) 45–74
7. Peña, J.M., Lozano, J.A., Larrañaga, P.: Learning recursive bayesian multinets for data clustering by means of constructive induction. Machine Learning, 47 (2002) 63–89
8. Thiesson, B., Meek, C., Chickerin, D.M., Heckerman, D.: Learning mixtures of DAG models. Proceeding of the 14th Conference on Uncertainty in Artificial Intelligence (1998) 504–513
9. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence **97** (1997) 273–324
10. Ben-Bassat, M.: Use of distance measures, information measures and error bounds in feature evaluation. In Krishnaiah, P.R., Kanal, L.N., eds.: Handbook of Statistics. Volume 2., North-Holland Publishing Company, Amsterdam (1982) 773–791
11. Duda, R., Hart, P.: Pattern classification and scene analysis. John Wiley and Sons, New York (1973)
12. Langley, P., Iba, W., Thompson, K.: An analysis of bayesian classifiers. In: National Conference on Artificial Intelligence. (1992) 223–228

13. Cheng, J., Greiner, R.: Comparing bayesian network classifiers, Morgan Kaufmann Publishers (1999) 101–108
14. Kononenko, I.: Semi-naive bayesian classifier. In: European working session on learning on Machine learning. (1991) 206–219
15. Pazzani, M.: Searching for dependencies in bayesian classifiers. Learning form data: Artificial Intelligence and stadistics V (1996) 239–248
16. Langely, P., Sage, S.: Induction of selective Bayesian classifiers. Proceeding of the Tenth Conference on Uncertainty in Artificial Intelligence (1998) 399–406
17. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: International Conference on Machine Learning. (1994) 121–129
18. Heckerman, D.: Probabilistic similarity networks. ACM Doctoral dissertation award series (1991)
19. Kohavi, R.: Scaling up the accuracy o naive-bayes classifier: a decision tree hybrid. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. (1996)
20. Armañanzas, R.: Medidas de filtrado de selección de variables mediante la plataforma Elvira. Master's thesis, Computer Science and Artificial Intelligence department, University of the Basque Country (2004) In Spanish.
21. Cooper, G., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Machine Learning 9 (1992) 309–347
22. Schwarz, G.: Estimating the dimension of a model. Annals of Statistics 6 (1978) 461–464
23. Heckerman, D., Geiger, D., Chickering, D.M.: Learning bayesian networks: The combination of knowledge and statistical data. Machine Learning 20 (1995) 197–243
24. Castellano, J.G., Moral, S., Cano, A.: Árboles de clasificación usando una estimación bayesiana. In: X conferencia de la Asociación Española para la Inteligencia Artificial CAEPIA-TTIA'2003, Donostia-San Sebastián (Spain) (2003) 97–100
25. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)
26. Fayyad, U., Irani, K.: Multi-valued interval discretization of continuous-valued attributes for classification learning. Proceeding of the 13th International joint Conference on Artificial Intelligence, Morgan Kaufmann, San Mateo, 1022-1027 (1993)
27. Kohavi, R., Sommerfield, D., Dougherty, J.: Data mining using MLC++: A machine learning library in C++. International Journal on Artificial Intelligence Tools 6 (1997) 537–566
28. Consortium, E.: Elvira: An environment for probabilistic graphical models. J.A. Gámez and A. Salmerón (Eds.): Procs. of the 1st European Workshop on Probabilistica Graphical Models (PGM 2002). pp. 222-230 (2002)

Qualitative Inference in Possibilistic Option Decision Trees

Ilyes Jenhani, Zied Elouedi, Nahla Ben Amor, and Khaled Mellouli

LARODEC, Institut Supérieur de Gestion de Tunis, 41 Avenue de la liberté,
2000 Le Bardo, Tunisie

ilyes.jenhani@gmx.net, {zied.elouedi, nahla.benamor}@gmx.fr,
khaled.mellouli@ihec.rnu.tn

Abstract. This paper presents a classification technique using possibility theory, namely the possibilistic option decision trees (PODT) which offers a more flexible building procedure by selecting more than one attribute in each decision node. Then, a classification method, using the PODT, to determine the class value of instances characterized by uncertain/missing attributes is proposed.

1 Introduction

Decision trees are considered as one of the most popular classification techniques [12]. They have the specific aim of allowing us to predict the class value of an object given known values of its attributes. Their popularity is due to their ability to solve complex problems by providing knowledge representations that are easy to understand and to interpret by experts and even by ordinary users.

Building a decision tree from a given training set represents a NP-Complete problem [10]. So, the majority of the decision tree algorithms is applying an heuristic of inductive inference known as Ockham's razor [2]. Generally speaking, this principle, which is widely employed in classification techniques, is useful when we face two models that perform identically in the training set: "...

... " [2].

In the decision tree context, "... " refers to the size of the tree and "... " refers to its accuracy. So, according to this principle, it is more plausible, but not necessarily true, that a smaller tree will be more accurate than a complex one. So, when building a decision tree, we collide with a kind of uncertainty which is related to the choice of an attribute at a given decision node. The idea is to develop what we call a possibilistic option decision tree (PODT). A technique that we have included in our approach is options [4] [7]. An option tree is a decision tree in which a decision node can be split according to more than one attribute (using multiple attribute-value tests, or "options"). Different options, in the PODT, are quantified via possibility distributions.

Uncertainty can also pervade the classification phase [1]. In fact, new instances presented to the inference procedure within the PODT may have some or even all attributes with uncertain values. This will be investigated in this paper, by developing a classification method using qualitative possibility theory.

This paper is organized as follows: Section 2 focuses on the basics of decision trees. Section 3 recalls some aspects of possibility theory and details the building procedure of the possibilistic option tree approach. Then, Section 4 describes the classification of uncertain objects within this approach. Finally, Section 5 presents and analyzes experimental results carried out on standard learning data sets from the U.C.I. repository [8].

2 Standard Decision Trees

We first give some notations. Let $\mathcal{I} = \{I_1, I_2, \dots\}$ be the set of all objects (instances) already classified or to classify. $\mathcal{I}$ is denoted the **object universe**. Each instance in $\mathcal{I}$ corresponds to a description of an object in attribute-value form. $\mathcal{A} = \{A_1, A_2, \dots, A_m\}$ denotes the set of **attributes** (properties) of each object (E.g.: color, age, height, etc.). Each attribute A_k in $\mathcal{A}$ has a finite domain denoted by $\mathcal{D}(A_k)$. $\mathcal{C} = \{C_1, C_2, \dots, C_n\}$ denotes the set of n mutually exclusive **classes** in which each object of the universe $\mathcal{I}$ has to belong. Finally, $\mathcal{I}_l$ denotes the **training set** (the set of all labeled instances whose classes are known) from which the decision tree will be induced. Once the problem is defined, the main objective is to build a decision tree that classifies well instances belonging to $\mathcal{I}$ and especially new instances whose class labels have to be predicted.

A decision tree is a flow-chart-like hierarchical tree structure which is composed of three basic elements: Decision nodes corresponding to attributes, edges or branches which correspond to the different possible attribute values. The third component is leaves including objects that typically belong to the same class or that are very similar. Such representation allows us to induce decision rules that will be used to classify new instances. In fact, each path from the root to a leaf corresponds to a conjunction of test attributes and the tree is considered as a disjunction of these conjunctions.

The majority of decision trees is made up of two major procedures: the building and the classification procedures also known, respectively, as the induction and the inference procedures.

- **Building Procedure:** Given a training set, building a decision tree is usually done by starting with an empty tree and selecting for each decision node the 'appropriate' test attribute using an attribute selection measure. The principle is to select the attribute that maximally diminish the mixture of classes between each training subset created by the test, thus, making easier the determination of object's classes. The process continues for each sub decision tree until reaching leaves and fixing their corresponding classes.
- **Classification Procedure:** To classify a new instance having precise and certain values of all its attributes and whose class value is unknown, we start with the root of the constructed tree and follow the path corresponding to the

observed value of the attribute in the interior node of the tree. This process is continued until a leaf is encountered. Finally, we use the associated label to obtain the predicted class value of the instance at hand.

Several algorithms for building decision trees have been developed. The most popular and applied ones are: simple **ID3** [10] and its successor **C4.5** "the state-of-the-art" algorithm developed by Quinlan [11]. We can also mention the **CART** algorithm of Breiman and al. [3].

Decision tree algorithms have many parameters to be defined which are considered as generic parameters of common algorithms. These parameters are described as follows:

a) The Attribute Selection Measure generally based on information theory, serves as a criterion in choosing among a list of candidate attributes at each decision node, the attribute that generates partitions where objects are distributed less randomly, with the aim of constructing the smallest tree among those consistent with the data. The measure used in this paper is the one used in **C4.5**, namely the gain ratio of Quinlan [11].

Given an attribute A_k , the information gain relative to A_k is defined as follows:

$$Gain(T, A_k) = Info(T) - Info_{A_k}(T) \quad (1)$$

$$where Info(T) = - \sum_{i=1}^n \frac{freq(C_i, T)}{|T|} \log_2 \frac{freq(C_i, T)}{|T|} \quad (2)$$

$$and Info_{A_k}(T) = \sum_{v \in D(A_k)} \frac{|T_v^{A_k}|}{|T|} Info(T_v^{A_k}) \quad (3)$$

... $(C_i, \dots)$ denotes the number of objects in the training set belonging to the class C_i and $|T_v^{A_k}|$ denotes the number of objects for which the attribute A_k has the value ... $Info(T)$ corresponds to the entropy of the set T . Then, the gain ratio is given by:

$$Gr(T, A_k) = \frac{Gain(T, A_k)}{Split Info(T, A_k)} \quad (4)$$

$$where Split Info(T, A_k) = - \sum_{v \in D(A_k)} \frac{|T_v^{A_k}|}{|T|} \log_2 \frac{|T_v^{A_k}|}{|T|} \quad (5)$$

$SplitInfo(\dots, A_k)$ represents the potential information generated by dividing into n subsets.

b) The Partitioning Strategy consisting in partitioning the training set according to all possible attribute values (for symbolic attributes) which leads to the generation of one partition for each possible value of the selected attribute. For continuous attributes, a discretization step is needed.

c) The Stopping Criteria stopping the partitioning process. Generally, we stop the partitioning if all the remaining objects belong to only one class, then the node is declared as a leaf labeled with this class value or if there is no further attribute to test.

3 Possibilistic Option Decision Trees

3.1 Possibility Theory

In this section, we will give a brief recalling on possibility theory (for more details see [5]). In possibility theory, uncertainty can be represented quantitatively as well as qualitatively. In this paper, we use both of these representations.

In the quantitative setting, uncertainty is modelled by means of numerical values from the interval $[0, 1]$ called the possibilistic scale. The basic concept of possibility theory in its quantitative aspect is the notion of *possibility*, denoted by π and which corresponds to a function which associates to each element ω of the universe of discourse Ω a value from the interval $[0, 1]$. This value is called a *possibility degree*: it encodes our knowledge, denoted by π , on the real world.

By convention, $\pi(\omega) = 1$ means that it is fully possible that $\omega = \cdot$ is the real world, $\pi(\omega) = 0$ means that $\omega = \cdot$ cannot be the real world (is impossible), and $\pi(\omega) > \pi(\omega')$ means that $\omega = u$ is preferred to $\omega' = u$ (or is more plausible).

A possibility distribution π is said to be **normalized** if there exists at least one state ω which is totally possible (i.e. $\pi(\omega) = 1$). In this paper, we only deal with normalized possibility distributions. We define the *possibility measure* of any event $\varphi \subseteq \Omega$ by:

$$\Pi(\varphi) = \max_{\omega \in \varphi} \pi(\omega). \quad (6)$$

This measure evaluates at which level φ is *possible* with our knowledge represented by π . As we can see from (6), the basic axiom of possibility theory is the **maximum** operator. Hence, the possibility of the disjunction of two events φ_1 and φ_2 is the maximum of the respective possibility of the individual events.

$$\Pi(\varphi_1 \vee \varphi_2) = \max(\Pi(\varphi_1), \Pi(\varphi_2)) \quad (7)$$

Suppose that a possibility distribution π is provided by a given source (e.g. expert, sensor) and suppose that the degree of certainty that this source is reliable is given by β , then π can be updated into [5]:

$$\pi' = \max(\pi, 1 - \beta) \quad (8)$$

Note that when $\beta=1$ (fully reliable source), $\pi' = \pi$ and in the case of absolutely unreliable source ($\beta=0$), $\forall \omega$, $\pi'(\omega) = 1$ (total ignorance). Equation (8) represents a form of **discounting** of a given possibility distribution.

In the qualitative framework, instead of giving exact numerical values from the unit interval $[0, 1]$ to model the uncertainty, we give an order between all the possible values using a finite and totally ordered scale denoted by $\mathcal{L} = \{\alpha_0 = 1, \alpha_1, \dots, \alpha_n, \alpha_{n+1} = 0\}$ such that $\alpha_0 = 1 > \alpha_1 > \dots > \alpha_n > \alpha_{n+1} = 0$. If δ is a set of uncertainty degrees, we define $\min(\delta) = \alpha_j$ (resp. $\max(\delta) = \alpha_j$) such that $\alpha_j \in \delta$ and $\nexists \alpha_k \in \delta$ such that $\alpha_k < \alpha_j$ (resp. $\alpha_k > \alpha_j$).

A qualitative possibility distribution (**QPD**) is a function which associates to each element ω of the universe of discourse Ω an element from $\mathcal{L}$, thus, enabling us to express that some states are more plausible than others without referring to any numerical value. The **QPD** covers all the properties of the quantitative possibility distributions mentioned in this section.

3.2 Building Possibilistic Option Decision Trees

Recall that the heart of any decision tree algorithm is the attribute selection measure parameter which is used to build a decision tree. As it is described, the standard building procedure [11] chooses at each decision node the attribute having the maximum or the minimum value (according to the context) of this measure, assuming that it leads to the smallest tree, and the remaining attributes are rejected: at this point, Ockham's razor is applied.

For instance, suppose that at a node n , we find that $\text{Gr}(T, A_1) = 0.87$ and $\text{Gr}(T, A_2) = 0.86$. In standard decision tree building procedure, the node n will be split according to the values of A_1 whereas A_2 is rejected in spite of the fact that the two values are almost equal. When looking into the second part of the assumption underlying Ockham's Razor: "the smaller the tree, the better", and after computing the gain ratios of the different attributes, one should establish priorities between these candidate attributes according to the obtained values and select attributes that appears possible to a certain extent as well instead of choosing only the one with the highest gain ratio and rejecting all the remaining attributes. Thus, the idea is to assign to each decision node n , a normalized possibility distribution π_{A_n} over the set of remaining attributes at this node, based on the set of gain ratios of the different attributes $GR = \{\text{Gr}(T_n, A_k) \text{ s. t. } A_k \in A_n\}$. T_n denotes the training subset relative to the node n .

Let A_n be the set of remaining attributes at a decision node n and GR the set corresponding to their gain ratios. We define a quantitative possibility distribution π_{A_n} by the following equation:

$$\pi_{A_n}(A_k) = \begin{cases} 0 & \text{if } \text{Gr}(A_k) \leq 0 \\ 1 & \text{if } \text{Gr}(A_k) = \max(GR) \\ \frac{\text{Gr}(A_k)}{\text{Gr}(A_k^*)} & \text{otherwise.} \end{cases} \quad (9)$$

We interpret $\pi_{A_n}(A_k)$ as the possibility degree that a given attribute A_k is reliable for the node n . An alternative manner to quantify the attributes was proposed by Hllermeier in [6], but the characteristics of our possibility distribution is that it proportionally preserves the gap between the different attributes according to their gain ratios and it does not use any additional parameter. Once possibility degrees are generated for each attribute, we use the possibilistic approach [4], i.e., a decision node n will not be only split according to the best attribute A_k^* but rather for all attributes in the set A_n^* which we define by:

$$A_n^* = \{A_k \in A_n \text{ s. t. } \text{distance}(A_k^*, A_k) \leq \Delta\}. \quad (10)$$

where $\text{distance}(A_k^*, A_k) = \pi_{A_n}(A_k^*) - \pi_{A_n}(A_k)$, A_n denotes the set of candidate attributes at the node n and Δ represents an arbitrary threshold varying in the interval $[0, 1]$. The fixed value of Δ has a direct effect on the size of the tree. In fact, for a large (resp. small) value of Δ , the number of the selected attributes, at each node, will increase (resp. decrease) and hence, the tree will have a larger (resp. smaller) size. The extreme cases occur when:

- $\Delta = 0$, we recover a standard decision tree as C4.5 of Quinlan.
- $\Delta = 1$, we obtain a huge decision tree composed of all the combinations of the different attribute values. This case is not interesting because it increases the time and space complexity. In addition, selecting attributes with low possibility degrees of being reliable in a given option node is nonsensical.

Since we can have more than one attribute at a given decision node n (an option-node), the partitioning is realized as follows: For each attribute $A_k \in A_n^*$ and each value $v \in D(A_k)$, one outgoing edge is added to n . This edge is labeled with the value v and the possibility degree $\pi_{A_n}(A_k)$ which is interpreted as the reliability degree of that edge. Obviously, we keep the same stopping criteria as in the standard decision trees.

Let us use the golf data set [8] to illustrate the induction of a possibilistic option decision tree (PODT). Let T be the training set composed of fourteen objects which are characterized by four attributes:

- **Outlook:** sunny or overcast or rain.
- **Temp:** hot or mild or cool.
- **Humidity:** high or normal.
- **Wind:** weak or strong.

Two classes are possible either, C_1 (play) or C_2 (don't play). The training set T is given by Table 1:

Assume $\Delta = 0.4$ in Equation (10).

Let us compute the gain ratios of the different attributes at the root node $n = 0$:

$$\text{Gr}(T_0, \text{Outlook}) = \frac{\text{Gain}(T_0, \text{Outlook})}{\text{Split Info}(T_0, \text{Outlook})} = \frac{0.246}{1.577} = 0.156;$$

$$\text{Gr}(T_0, \text{Temp}) = \frac{\text{Gain}(T_0, \text{Temp})}{\text{Split Info}(T_0, \text{Temp})} = \frac{0.029}{1.556} = 0.018;$$

$$\text{Gr}(T_0, \text{Humidity}) = \frac{\text{Gain}(T_0, \text{Humidity})}{\text{Split Info}(T_0, \text{Humidity})} = \frac{0.151}{1} = 0.151;$$

$$\text{Gr}(T_0, \text{Wind}) = \frac{\text{Gain}(T_0, \text{Wind})}{\text{Split Info}(T_0, \text{Wind})} = \frac{0.048}{0.985} = 0.048;$$

We remark that the attribute "Outlook" has the highest gain ratio. Let's now, compute the possibility degrees of the different attributes, using Equation (9), in order to define the set A_0^* :

$$\pi_{A_0}(\text{Outlook}) = 1$$

$$\pi_{A_0}(\text{Temp}) = \frac{\text{Gr}(T_0, \text{Temp})}{\text{Gr}(T_0, \text{Outlook})} = \frac{0.018}{0.156} = 0.12;$$

$$\pi_{A_0}(\text{Humidity}) = \frac{\text{Gr}(T_0, \text{Humidity})}{\text{Gr}(T_0, \text{Outlook})} = \frac{0.151}{0.156} = 0.97;$$

$$\pi_{A_0}(\text{Wind}) = \frac{\text{Gr}(T_0, \text{Wind})}{\text{Gr}(T_0, \text{Outlook})} = \frac{0.048}{0.156} = 0.31;$$

Given $\Delta = 0.4$, the set of attributes which will be assigned to the root n_0 of the possibilistic option tree is given by: $A_0^* = \{Outlook, Humidity\}$.

The possibilistic option tree induced from the training set T ($\Delta = 0.4$ in Equation (10)), which we denote by $PODT_{0.4}$, is given by Fig. 1. For clarity reasons, abbreviations of the attribute values are used instead of complete words.(e.g. "ho" for the value "hot", "hi" for "high", "we" for "weak", etc.).

Table 1. Training set

Outlook	Temp	Humidity	Wind	Class
sunny	hot	high	weak	C_2
sunny	hot	high	strong	C_2
overcast	hot	high	weak	C_1
rain	mild	high	weak	C_1
rain	cool	normal	weak	C_1
rain	cool	normal	strong	C_2
overcast	cool	normal	strong	C_1
sunny	mild	high	weak	C_2
sunny	cool	normal	weak	C_1
rain	mild	normal	weak	C_1
sunny	mild	normal	strong	C_1
overcast	mild	high	strong	C_1
overcast	hot	normal	weak	C_1
rain	mild	high	strong	C_2

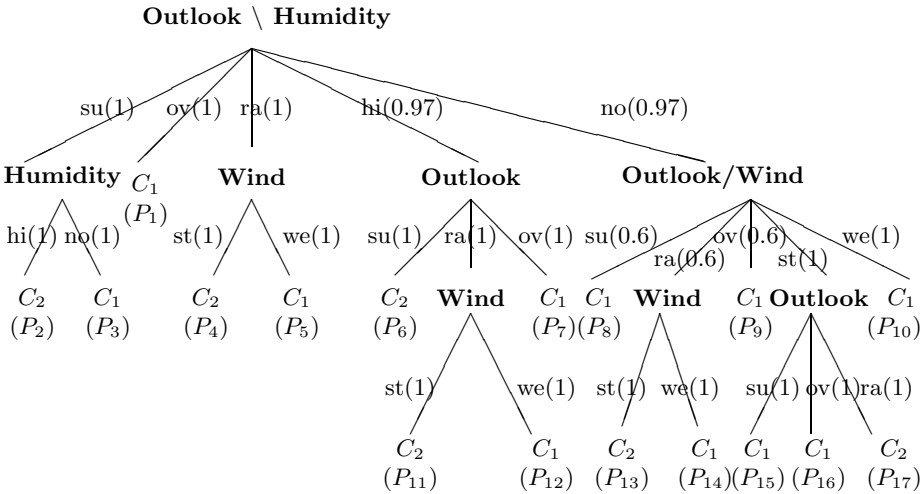


Fig. 1. Final possibilistic option tree

4 Qualitative Inference with Possibilistic Option Trees

In this section, we are interested on how to classify objects characterized by uncertain or missing attribute values within possibilistic option trees. Uncertainty here is handled in a qualitative possibilistic framework.

For each attribute, we assign a qualitative possibility distribution (QPD) to express the uncertainty on the real value of that attribute. Given the set of attributes A , the instance to classify is described by a vector of possibility distributions $\vec{i} = (\pi'_{A_1}, \dots, \pi'_{A_n})$. An attribute A_k whose value is known with certainty has exactly one value $v \in D(A_k)$, such that $\pi'_{A_k}(v) = 1$, and for all other values $v' \in D(A_k) - \{v\}$, $\pi'_{A_k}(v') = 0$. An attribute A_k whose value is uncertain is represented by a uniform possibility distribution, i.e., $\forall v \in D(A_k)$, $\pi'_{A_k}(v) = 1$. Table 2 gives an example of an uncertain instance $\vec{i}_1$ to classify. Note that $1 > \alpha_1 > \alpha_2 > \alpha_3 > \alpha_4 > \alpha_5$. In order to classify an uncertain instance (e.g. $\vec{i}_1$) within a possibilistic option tree *PODT*, we need to carry out the following steps:

Table 2. Instance $\vec{i}_1$

$\pi'_{outlook}$		π'_{temp}		$\pi'_{humidity}$		π'_{wind}	
sunny	α_4	hot	1	high	1	strong	1
overcast	α_1	mild	1	normal	α_2	weak	α_5
rain	1	cool	α_3				

Step One: The Instance Propagation

At each option node of a possibilistic option tree, the instance to classify can branch in different directions depending on the chosen attribute to test on. To each one of these attributes, we have assigned a possibility degree $\pi_{A_n}(A_k)$ (Equation (9)) indicating the possibility that a given attribute is reliable for a given option node n .

Thus, throughout a given *PODT*, whenever an instance follows an attribute A_k , the related QPD in the instance to classify (π'_{A_k}) should be discounted according to the possibility degree of the followed attribute ($\pi_{A_n}(A_k)$) using Equation (8). The discounted possibility degrees will replace the degrees labeling the *PODT*.

Step Two: Exploring the Paths

Once the propagation is made within the *PODT* (step 1), we should explore all its paths in order to determine their corresponding possibility degrees based on the 'new' discounted possibility degrees labeling the tree. Since we deal with qualitative possibility distributions, we have chosen the $\min$ operator to define the possibility degree of a path $p = (n_0, \dots, n_k)$ as

$$\pi_{path}(p) = \min_{0 \leq i < l(p)-1} \pi_{edge}((n_i, n_{i+1})) \quad (11)$$

where $\pi_{edge}((n_i, n_{i+1}))$ denotes the possibility degree labeling the edge (n_i, n_{i+1}) and $l(p)$ denotes the length (number of nodes) of the path p .

Step Three: Exploring the Classes

The results obtained from the computation of the possibilities of the different paths should be refined. For each possible class $c_k \in C$, we define a vector $\vec{c}_k$ ranking the possibility degrees of the different paths pertaining to c_k in decreasing order. The class of the instance to classify will be the one relative to the preferred vector which will be determined using the following definition:

Definition 1. $\vec{c_k} = \{\pi(p_{c_k,1}), \dots, \pi(p_{c_k,m})\}$ and $\vec{c_l} = \{\pi(p_{c_l,1}), \dots, \pi(p_{c_k,n})\}$ are the k -th and l -th column vectors of $\pi(C)$ respectively.

- $$\begin{aligned}
& - \vec{c_k} \leq \vec{c_l}, \dots, \vec{c_k} >_p \vec{c_l} \\
& \bullet \dots i \in \{1, \dots, \min(n, m)\} \dots \pi(p_{c_k, i}) > \pi(p_{c_l, i}) \\
& \quad \forall j < i, \pi(p_{c_k, j}) = \pi(p_{c_l, j}) \\
& \bullet \dots, \forall i \in \{1, \dots, \min(n, m)\}, \pi(p_{c_k, i}) = \pi(p_{c_l, i}) \text{ and } m > n \\
& - \vec{c_k} \leq \vec{c_l}, \dots, \vec{c_k} =_p \vec{c_l}, \dots \forall i, \pi(p_{c_k, i}) = \pi(p_{c_l, i})
\end{aligned}$$

In the case of equally preferred vectors, we choose a class at random.

Suppose we have to classify the instance $\vec{t}_1$ given in Table 2 within the induced $PODT_{0.4}$ of Example 1. Assume $\alpha_1 = 0.8$, $\alpha_2 = 0.5$, $\alpha_3 = 0.4$, $\alpha_4 = 0.2$ and $\alpha_5 = 0.1$. The assigned values only preserve the ranking between α_i and hence they have no sense. So, we get the following instance:

Table 3. Instance $\overrightarrow{i_1}$

$\pi'_{outlook}$		π'_{temp}		$\pi'_{humidity}$		π'_{wind}	
sunny	0.2	hot	1	high	1	strong	1
overcast	0.8	mild	1	normal	0.5	weak	0.1
rain	1	cool	0.4				

STEP 1: Instance Propagation

Starting from the root node of the $PODT_{0.4}$ (see Fig. 1), the instance $\vec{i}_1$ can follow both the 'Outlook' attribute and the 'Humidity' attribute whose reliability degrees are respectively 1 and 0.97. According to the reliability of each followed attribute A_k , we will discount the corresponding possibility distribution π'_{A_k} as mentioned above.

The different edges of the $PODT_{0.4}$ will be labeled by the discounted QPD's of the instance to classify. We do not show the figure here for reasons of space.

STEP 2: Exploring the Paths

Let us compute the possibility degree relative to each path using Equation (11):

$P_1 : 0.8 \Rightarrow (C_1, 0.8)$, $P_2 : \min(0.2, 1) = 0.2 \Rightarrow (C_2, 0.2)$, $P_3 : \min(0.2, 0.5) = 0.2 \Rightarrow (C_1, 0.2)$, $P_4 : \min(1, 1) = 1 \Rightarrow (C_2, 1)$, $P_5 : \min(1, 0.1) = 0.1 \Rightarrow (C_1, 0.1)$, $P_6 : \min(1, 0.2) = 0.2 \Rightarrow (C_2, 0.2)$, $P_7 : \min(1, 0.8) = 0.8 \Rightarrow (C_1, 0.8)$, $P_8 : \min(0.5, 0.4) = 0.4 \Rightarrow (C_1, 0.4)$, $P_9 : \min(0.5, 0.8) = 0.5 \Rightarrow (C_1, 0.5)$, $P_{10} : \min(0.5, 0.1) = 0.1 \Rightarrow (C_1, 0.1)$, $P_{11} : \min(1, 1, 1) = 1 \Rightarrow (C_2, 1)$, $P_{12} : \min(1, 1, 0.1) = 0.1 \Rightarrow (C_1, 0.1)$, $P_{13} : \min(0.5, 1, 1) = 0.5 \Rightarrow (C_2, 0.5)$, $P_{14} : \min(0.5, 1, 0.1) = 0.1 \Rightarrow (C_1, 0.1)$, $P_{15} : \min(0.5, 1, 0.2) = 0.2 \Rightarrow (C_1, 0.2)$, $P_{16} : \min(0.5, 1, 0.8) = 0.5 \Rightarrow (C_1, 0.5)$, $P_{17} : \min(0.5, 0.8, 1) = 0.5 \Rightarrow (C_2, 0.5)$.

STEP 3: Exploring the Classes

Refining the results found, using Definition 1, we get:

$$\begin{aligned}\vec{C}_1 &= \{0.8, 0.8, 0.5, 0.5, 0.4, 0.2, 0.2, 0.1, 0.1, 0.1, 0.1\}. \\ \vec{C}_2 &= \{1, 1, 0.5, 0.5, 0.2, 0.2\}.\end{aligned}$$

Then, the class corresponding to the instance $\vec{i}_1$ is C_2 since $\vec{C}_2 >_p \vec{C}_1$.

Note that the classification method described in this Section collapses to the standard classification procedure when testing instances are certain and $\Delta = 0$.

5 Experimental Results

For the evaluation of the possibilistic option tree approach, we have developed programs in Matlab V6.5, implementing both of the building and the classification procedures relative to the PODT. Then, we have applied our approach to two real databases obtained from the U.C.I repository of Machine Learning databases [8]. A brief description of these nominal-valued databases is presented in Table 3. #Tr, #Ts, #attributes, #classes denote respectively the number of training instances, the number of testing instances, the number of attributes and the number of classes. For the testing sets, we have generated uncertainty relative to attribute's values of the different testing instances in an artificial manner. In this experimentation, we were interested by the impact of varying Δ in Equation (10) on the *PCC* ($= \frac{\text{number of well classified instances}}{\text{total number of classified instances}}$) by considering parameters relative to the tree size (#nodes, #leaves) and temporal parameters (time relative to the building phase (T. build.) and to the classification phase (T. classif.)). Table 4 and Table 5 summarize different results relative to the Wisconsin breast cancer and Nursery databases, respectively. Note that the experimentations were performed using a Centrino 1.4 GHz PC with 512 MB of RAM running Windows XP.

It is important to mention that, during the experimentations, we have varied Δ from 0 to 0.5. We stopped at 0.5 since it becomes not interesting to consider attributes whose reliability is less than 0.5, i.e., attributes that seem to become

Table 4. Description of databases

Database	#Tr	#Ts	#attributes	#classes
Wisconsin Breast Cancer	629	70	8	2
Nursery	750	75	8	5

Table 5. The experimental measures (W. breast cancer)

Δ	#nodes	#leaves	T. build. (s)	T. classif. (s)	PCC (%)
0	101	168	15.27	55.42	81.42
0.1	154	259	17.5	96.54	88.57
0.2	320	550	27.27	204.38	80.00
0.3	529	933	38.89	366.15	80.00
0.4	879	1602	59.41	673.62	78.57
0.5	1802	3263	110.0	1635.98	75.71

Table 6. The experimental measures (Nursery)

Δ	#nodes	#leaves	T. build. (s)	T. classif. (s)	PCC (%)
0	60	108	12.34	17.84	88.00
0.1	107	197	13.55	32.61	90.66
0.2	176	333	16.25	57.81	92.00
0.3	224	424	18.86	72.88	86.66
0.4	294	554	21.05	98.34	86.66
0.5	401	781	26.26	134.87	84.00

far from the fully reliable one. As it is shown in Table 4 and Table 5, the *PCC* increases progressively and becomes to decrease when reaching a specific value of Δ . For instance, in the W. breast cancer database, the *PCC* increases from 81.42 % to 88.57 % when varying Δ from 0 to 0.1 and becomes to decrease from 88.57 % to 75.71 % for $\Delta \in [0.1, 0.5]$. The value of Δ for which we obtain the most accurate *PODT* (0.1 for the W. breast cancer database and 0.2 for the Nursery database) is determined experimentally and depends on the used training set. These results confirm the results obtained in [9]: smaller tree(s) is (are) not necessarily more accurate than the slightly larger one(s). It is important to note that the *PODT* approach has the advantage of classifying instances having uncertain or missing attribute values.

6 Conclusion

In this paper, we have developed a new approach so-called possibilistic option decision tree. This approach has two advantages. The first is that it considers more than one attribute at a given decision node by breaking Ockham’s razor principle. The second advantage is the ability of classifying instances characterized by uncertain/missing attribute values. The experimental results presented

in this paper are encouraging. In fact, the classification accuracy of the PODT increases when varying Δ until reaching a specific value which is purely experimental. This value is relatively small and hence the time and space complexity are reasonable. We believe that the pruning issue should be investigated and aim to extend our approach to handle continuous attributes in the future.

References

1. Ben Amor, N., Benferhat, S., Elouedi, Z.: Qualitative classification and evaluation in possibilistic decision trees, FUZZ-IEEE'2004.
2. Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.: Occam's razor, Information Processing Letters, 24, 377-380, 1987.
3. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: Classification and regression trees, Monterey, CA : Wadsworth & Brooks, 1984.
4. Buntine, W.: Learning classification trees, Statistics and Computing, 63-73, 1990.
5. Dubois, D., Prade, H.: Possibility theory: An approach to computerized processing of uncertainty, Plenum Press, New York, 1988.
6. Hllermeier, E.: Possibilistic Induction in decision tree learning, ECML'2002.
7. Kohavi, R., Kunz, C.: Option decision trees with majority votes, ICML'97.
8. Murphy, P. M., Aha, D. W., UCI repository of machine learning databases, 1996.
9. Murphy, P. M., Pazzani, M. J.: Exploring the decision forest: An empirical investigation of Occam's Razor in decision tree induction, JAIR, 257-275, 1994.
10. Quinlan, J. R.: Induction of decision trees, Machine Learning, 1, 81-106, 1986.
11. Quinlan, J. R.: C4.5: Programs for machine learning, Morgan Kaufmann, 1993.
12. Weiss, S. M., Kulikovski, C. A.: Computer systems that learn, Morgan Kaufmann, San Mateo, California, 1991.

Partially Supervised Learning by a Credal EM Approach

Patrick Vannoorenberghe¹ and Philippe Smets²

¹ PSI, FRE 2645 CNRS, Université de Rouen,
Place Emile Blondel, 76821 Mont Saint Aignan cedex, France

Patrick.Vannoorenberghe@univ-rouen.fr

² IRIDIA, Université Libre de Bruxelles,
50, av. Roosevelt, 1050 Bruxelles, Belgique
psmets@ulb.ac.be

Abstract. In this paper, we propose a Credal EM (CrEM) approach for partially supervised learning. The uncertainty is represented by belief functions as understood in the transferable belief model (TBM). This model relies on a non probabilistic formalism for representing and manipulating imprecise and uncertain information. We show how the EM algorithm can be applied within the TBM framework when applied for the classification of objects and when the learning set is imprecise (the actual class of each object is only known as belonging to a subset of classes), and/or uncertain (the knowledge about the actual class is represented by a probability function or by a belief function).

Keywords: Learning, belief functions, EM, transferable belief model.

1 Introduction

Supervised learning consists in assigning an input pattern $\mathbf{x}$ to a class, given a learning set $\mathcal{L}$ composed of N patterns $\mathbf{x}_i$ with known classification. Let $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ be the set of K possible classes. Each pattern in $\mathcal{L}$ is represented by a p -dimensional feature vector $\mathbf{x}_i$ and its corresponding class label y_i . When the model generating the data is known, the classical methods of discriminant analysis (DA) permits the estimation of the parameters of the model.

Still these methods assumed in practice that the actual class y_i of each case in the learning set is well known. Instead suppose the data of the learning set are only partially observed, i.e., the actual class of a given object is only known to be one of those in a given subset C of Ω . Classical methods for parametric learning encounter then serious problems. One of the solution was based on the EM algorithm (Dempster, Laird, & Rubin, 1977; McLacian & Krishnan, 1997).

Parametric learning requires a model of the generation of the data and an algorithm for estimating the parameters of this model using the available information contained in the learning set. A major drawback of many parametric methods is their lack of flexibility when compared with nonparametric methods. However, this problem can be circumvented using mixture models which

combine much of the flexibility of nonparametric methods with certain of the analytic advantages of parametric methods. In this approach, we assume that the data $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ are generated independently from a mixture density model which probability density function (pdf) is given by:

$$f(\mathbf{x}_i; y_i = \omega_k, \theta) = \sum_{g=1}^{G_k} \pi_{kg} f_{kg}(\mathbf{x}_i; \alpha_{kg}) \quad (1)$$

where G_k is the number of components in the mixture for the cases in class ω_k , π_{kg} are the mixing proportions, f_{kg} denotes a component, i.e. a probability distribution function parametrized by α_{kg} , and $\theta = \{(\pi_{kg}, \alpha_{kg}) : g = 1, \dots, G_k; k = 1, \dots, K\}$ are the model parameters to be estimated. For mixture of Gaussian pdfs, the function $f_{kg}(\mathbf{x}_i; \alpha_{kg})$ is a Gaussian pdf and α_{kg} is a set of parameters $\alpha_{kg} = (\boldsymbol{\mu}_{kg}, \boldsymbol{\Sigma}_{kg})$ where $\boldsymbol{\mu}_{kg}$ is the mean and $\boldsymbol{\Sigma}_{kg}$ the variance-covariance matrix of the Gaussian pdf f_{kg} .

Generally, the maximum likelihood estimation of the parameters of this model cannot be obtained analytically, but learning θ could be easily achieved if the particular component f_{kg} responsible for the existence of each observation $\mathbf{x}_i$ was known. In reality, this ideal situation is hardly encountered.

Several real world contexts can be described.

1. **The precise teacher case.** For each learning case, we know the actual class to which it belongs. The missing information is the g value for each case. The classical approach to solve this problem is the EM algorithm.
2. **The imprecise teacher case.** For each learning case, we only know that the actual class belongs to a subset of Ω . The missing information is the k and the g values for each case, where k is constrained to a subset of $1, \dots, K$. The EM algorithm can be extended to such a case (Hastie & Tibshirani, 1996; Ambroise & Govaert, 2000).
3. **The precise and uncertain teacher case.** For each learning case, we only have some beliefs about what is the actual class to which the case belongs. The uncertainty is represented by a probability function on Ω . The uncertainty concerns the k value, and the g values are still completely unknown.
4. **The imprecise and uncertain teacher case.** For each learning case, we only have some beliefs about what is the actual class to which the case belongs. The uncertainty is represented by a belief function on Ω . The uncertainty and imprecision concern the k value, and the g values are still completely unknown. The EM algorithm can be further extended to such a case as done here.

In this paper, we consider the imprecise teacher case and the imprecise and uncertain teacher case, the first case being covered by the second one. Uncertainty is represented by belief functions as understood in the TBM (Smets & Kennes, 1994; Smets, 1998). We propose to use the advantages of both the EM algorithm and the belief functions to learn the parameter of a TBM classifier. This algorithm is called the ‘Credal EM’ (CrEM) and its related classifier is called the ‘CrEM classifier’.

Previous work on comparing a TBM classifier with an EM based classifier was performed in (Ambroise, Denœux, Govaert, & Smets, 2001). Performance were analogous, but the TBM classifier was much simpler to use. The TBM classifier used in that comparison was based on non parametric methods as developed by (Denœux, 1995; Zouhal & Denœux, 1998). Here the TBM is used for parameter estimation and the final TBM classifier is based on a parametric method. This paper is organized as follows. The basic concepts of belief functions theory are briefly introduced in Section 2. The notion of likelihood is extended into the TBM in Section 3. The principle of parameters estimation via the EM algorithm is recalled in Section 4. The proposed algorithm is presented in Section 5. Finally, Section 6 gives some experimental results using synthetic data.

2 Background Materials on Belief Functions

Let Ω be a finite space, and let 2^Ω be its power set. A belief function defined on Ω can be mathematically defined by introducing a set function, called the basic belief assignment (bba) $m^\Omega : 2^\Omega \rightarrow [0, 1]$ which satisfies:

$$\sum_{A \subseteq \Omega} m^\Omega(A) = 1. \quad (2)$$

Each subset $A \subseteq \Omega$ such as $m^\Omega(A) > 0$ is called a focal element of m^Ω . Given this bba, a belief function bel^Ω and a plausibility function pl^Ω can be defined, respectively, as:

$$bel^\Omega(A) = \sum_{\emptyset \neq B \subseteq A} m^\Omega(B), \quad \forall A \subseteq \Omega. \quad (3)$$

$$pl^\Omega(A) = \sum_{A \cap B \neq \emptyset} m^\Omega(B), \quad \forall A \subseteq \Omega. \quad (4)$$

The three functions bel^Ω , pl^Ω and m^Ω are in one-to-one correspondence and represent three facets of the same piece of information. We can retrieve each function from the others using the fast Möbius transform (Kennes, 1992). Let m_1^Ω and m_2^Ω be two bbas defined on the same frame Ω . Suppose that the two bbas are induced by two distinct pieces of evidence. Then the joint impact of the two pieces of evidence can be expressed by the conjunctive rule of combination which results in the bba:

$$m_{12}^\Omega(A) = (m_1^\Omega \odot m_2^\Omega)(A) = \sum_{B \cap C = A} m_1^\Omega(B) \cdot m_2^\Omega(C). \quad (5)$$

In the TBM, we distinguish the credal level where beliefs are entertained (formalized, revised and combined) and the pignistic level used for decision making. Based on rationality arguments developed in the TBM, Smets proposes to transform m^Ω into a probability function $BetP$ on Ω (called the *pignistic* probability function) defined for all $\omega_k \in \Omega$ as:

$$BetP(\omega_k) = \sum_{A \ni \omega_k} \frac{m^\Omega(A)}{|A|} \frac{1}{1 - m^\Omega(\emptyset)} \quad (6)$$

where $|A|$ denotes the cardinality of $A \subseteq \Omega$ and $BetP(A) = \sum_{\omega \in A} BetP(\omega)$, $\forall A \subseteq \Omega$. In this transformation, the mass of belief $m(A)$ is distributed equally among the elements of A (Smets & Kennes, 1994; Smets, 2005).

Let us suppose the two finite spaces X , the observation space, and Θ , the unordered parameter space. The Generalized Bayesian Theorem (GBT), an extension of Bayes theorem within the TBM (Smets, 1993), consists in defining a belief function on Θ given an observation $x \subseteq X$, the set of conditional bbas $m^X[\theta_i]$ over X , one for each $\theta_i \in \Theta^1$ and a vacuous *a priori* on Θ . Given this set of bbas (which can be associated to their related belief or plausibility functions), then for $x \subseteq X$ and $\forall A \subseteq \Theta$, we have:

$$pl^\Theta[x](A) = 1 - \prod_{\theta_i \in A} (1 - pl^X[\theta_i](x)). \quad (7)$$

3 Explaining the Likelihood Maximization Within the TBM

Suppose a random sample of a distribution with parameters $\theta \in \Theta$ and let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N : \mathbf{x}_i \in \mathbb{R}^p\}$ be the set of observations. In probability theory many estimation procedures for θ are based on the maximization of the likelihood, i.e. $P^{\mathbb{R}^p}(\mathbf{X}|\theta)$ considered as a function of θ . How do we generalize this procedure within the TBM? We reconsider the issue.

For each $\theta \in \Theta$, we have a conditional bba on $\mathbb{R}$, denoted $m^{\mathbb{R}}[\theta]$. We observe $\mathbf{x} \subseteq \mathbb{R}$. This induce a bba on Θ by the application of the GBT. So we get the bba $m^\Theta[\mathbf{x}]$. How to estimate θ_0 , the actual value of Θ ? We could select the θ that maximizes $BetP^\Theta[\mathbf{x}]$, thus the most ‘probable’ value of Θ . This last solution means finding the modal value of $BetP^\Theta[\mathbf{x}]$. We feel this principle fits with the idea underlying the maximum likelihood estimators.

So we must find the $\theta \in \Theta$ such that $BetP^\Theta[\mathbf{x}](\theta) \geq BetP^\Theta[\mathbf{x}](\theta_i)$, $\forall \theta_i \in \Theta$. This maximization seems hard to solve, but we can use theorem III.1. in (Delmotte & Smets, 2004) which states that the θ that maximizes $BetP^\Theta[\mathbf{x}]$ is the same as the one that maximizes the plausibility function $pl^\Theta[\mathbf{x}](\theta)$, provided the *a priori* belief on Θ is vacuous, as it is the case here.

Theorem 1. *Given $\mathbf{x} \subseteq X$ and $pl^X[\theta](\mathbf{x})$ for all $\theta \in \Theta$, let $pl^\Theta[\mathbf{x}]$ be the plausibility function defined on Θ and computed by the GBT, and $BetP^\Theta[\mathbf{x}]$ be the pignistic probability function constructed on Θ from $pl^\Theta[\mathbf{x}]$, then:*

$$BetP^\Theta[\mathbf{x}](\theta_i) > BetP^\Theta[\mathbf{x}](\theta_j) \quad \text{iff} \quad pl^\Theta[\theta_i](\mathbf{x}) > pl^\Theta[\theta_j](\mathbf{x}). \quad (8)$$

In the TBM, $pl^\Theta[\mathbf{x}](\theta)$ is equal to $pl^X[\theta](\mathbf{x})$. Furthermore when N i.i.d. data $\mathbf{x}_i, i = 1, \dots, N$, are observed, we get $pl^{X^N}[\theta](\mathbf{x}_1, \dots, \mathbf{x}_N) = \prod_{i=1}^N pl^X[\theta](\mathbf{x}_i)$.

¹ We use the next notational convention for the indices and $[\]$: $m^D[u](A)$ denotes the mass given to the subset A of the domain D by the conditional bba $m^D[u]$ defined on D given u is accepted as true.

This last term is easy to compute and leads thus to applicable algorithms. Maximizing the likelihood over θ turns out to mean maximizing over θ the conditional plausibilities of the data given θ .

4 Parameter Estimation by EM Algorithm

We introduce the classical EM approach to find the parameters of a mixture models from a data set $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ made of cases which belong to a same class. The aim is to estimate the posterior distribution of the variable $\mathbf{y}$ which indicates the component of the mixture that generated $\mathbf{x}_i$ taking into account the available information $\mathcal{L}$. For simplicity sake, we do not indicate the class index k . For that estimation, we need to know π_g , f_g and α_g for $g = 1, \dots, G$. For their estimation, we use the EM algorithm to maximize according to θ the log likelihood:

$$L(\theta; \mathbf{X}) = \log\left(\prod_{i=1}^N f(\mathbf{x}_i; \theta)\right) = \sum_{i=1}^N \log\left(\sum_{g=1}^G \pi_g f_g(\mathbf{x}_i; \alpha_g)\right). \quad (9)$$

In order to solve this problem, the idea is that if one had access to a hidden random variable $\mathbf{z}$ that indicates which data point was generated by which component, then the maximization problem would decouple into a set of simple maximizations. Using this indicator variable $\mathbf{z}$, relation (9) can be written as the next complete-data log likelihood function:

$$L_c(\theta; \mathbf{X}, \mathbf{z}) = \sum_{i=1}^N \sum_{g=1}^G z_{ig} \log(\pi_g f_g(\mathbf{x}_i; \alpha_g)) \quad (10)$$

where $z_{ig} = 1$ if the Gaussian pdf having generated the observation $\mathbf{x}_i$ is f_g , and 0 otherwise. Since $\mathbf{z}$ is unknown, L_c cannot be used directly, so we usually work with its expectation denoted $Q(\theta|\theta^l)$ where l is used as the iteration index. As shown in (Dempster et al., 1977), $L(\theta; \mathbf{X})$ can be maximized by iterating the following two steps:

- E step: $Q(\theta|\theta^l) = E[L_c(\theta; \mathbf{X}, \mathbf{z})|\mathbf{X}, \theta^l]$
- M step: $\theta^{l+1} = \arg \max_{\theta} Q(\theta|\theta^l)$

The E (Expectation) step computes the expected complete data log likelihood and the M (Maximization) step finds the parameters that maximize that likelihood. $Q(\theta|\theta^l)$ can be rewritten as

$$Q(\theta|\theta^l) = \sum_{i=1}^N \sum_{g=1}^G E[z_{ig}|\mathbf{X}, \theta^l] \log(\pi_g f_g(\mathbf{x}_i; \alpha_g)) \quad (11)$$

In a probabilistic framework, $E[z_{ig}|\mathbf{X}, \theta^l]$ is nothing else than $P(z_{ig} = 1|\mathbf{X}, \theta^l)$, the posterior distribution easily computed from the observed data.

5 CrEM: The Credal Solution

In this section, we introduce a credal EM approach for partially supervised learning. The imprecision or/and uncertainty on the observed labels are represented by belief functions (cf. section 5.1). We consider the imprecise and uncertain teacher case (section 5.2).

5.1 Partially Observed Labels

Thanks to its flexibility, a belief function can represent different forms of labels including hard labels (HL), imprecise labels (IL), probabilistic labels (PrL), possibilistic (PoL) labels and credal labels (CrL). Table 1 illustrates an example of the bbas that characterize the knowledge about the labels on a three-class frame. Note that a possibility measure is known to be formally equivalent to a consonant belief function, i.e., a belief function with nested focal elements (Denœux & Zouhal, 2001). Unlabeled samples (UL) can be encoded using the vacuous belief function m_v defined as $m_v(\Omega) = 1$. This show that handling the general case based on belief functions covers all cases of imperfect teacher (imprecise and/or uncertain). Of course, the TBM covers the HL, IL, PrL and CrL cases. For the PoL, the CrEM algorithm presented here has to be adapted as we use the GBT and other combination rules that differ from their possibilistic counterparts.

Table 1. Example of imprecise and uncertain labeling with belief functions

$A \subseteq \Omega$	HL	IL	PrL	PoL	CrL	UL
$\{\omega_1\}$	0	0	0.2	0	.1	0
$\{\omega_2\}$	1	0	0.6	0	0	0
$\{\omega_1, \omega_2\}$	0	1	0	0	.2	0
$\{\omega_3\}$	0	0	0.2	0.7	.3	0
$\{\omega_1, \omega_3\}$	0	0	0	0.2	.3	0
$\{\omega_2, \omega_3\}$	0	0	0	0	0	0
Ω	0	0	0	0.1	.1	1

5.2 The Imprecise and Uncertain Teacher Case

Let $\Omega = \{\omega_1, \dots, \omega_K\}$ be a set of K mutually exclusive classes². Let $\mathcal{L}$ be a set of N observed cases and called the learning set. For $i = 1, \dots, N$, let c_i denotes the i -th case. For case c_i , we collect a feature vector $\mathbf{x}_i$ taking values in $\mathbb{R}^p$, and a bba m_i^Ω that represents all we know about the actual class $y_i \in \Omega$ to which case c_i belongs. We then assume that the probability density function (pdf) of $\mathbf{x}_i$ is given by the next mixture of pdfs :

² In the TBM, we do not require Ω to be exhaustive, but one could add this requirement innocuously.

$$f(\mathbf{x}_i; y_i = \omega_k, \theta_k) = \sum_{g=1}^{G_k} \pi_{kg} f_{kg}(\mathbf{x}_i; \alpha_{kg}) \quad (12)$$

where f_{kg} is the p -dimensional Gaussian pdf with parameters $\alpha_{kg} = (\boldsymbol{\mu}_{kg}, \boldsymbol{\Sigma}_{kg})$.

Let the available data be $\{(\mathbf{x}_1, m_1^\Omega), \dots, (\mathbf{x}_N, m_N^\Omega)\}$ where $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ is an i.i.d sample. Let $\mathbf{Y} = (y_1, \dots, y_N)$ be the unobserved labels and $\mathbf{m}^\Omega = (m_1^\Omega, \dots, m_N^\Omega)$ are the bbas representing our beliefs about the actual values of the y_i 's. For the estimation of the parameters $\theta = (\{\alpha_{kg} : j = 1, \dots, G_k, k = 1, \dots, K\}, \mathbf{Y})$, we use the EM algorithm to maximize the log likelihood given by:

$$L(\theta; \mathcal{L}) = \log\left(\prod_{i=1}^N f(\mathbf{x}_i; y_i = \omega_k, \theta_k)\right) = \sum_{i=1}^N \log\left(\sum_{g=1}^{G_k} \pi_{kg} f_{kg}(\mathbf{x}_i; \alpha_{kg})\right). \quad (13)$$

We can rephrase the relation by considering all the Gaussian pdfs. There are $G = \sum_{k=1}^K G_k$ Gaussian pdfs. Let J_k be the indexes in the new ordering of the components of the class ω_k . So $J_k = \{j : \sum_{\nu=1}^{k-1} G_\nu < j \leq \sum_{\nu=1}^k G_\nu\}$ where $\sum_{\nu=1}^0 G_\nu = 0$. This reindexing is analogous to a refinement $\mathcal{R}$ of the classes in $\Omega = \{\omega_k : k = 1, \dots, K\}$ into a set of new 'classes' $\Omega^* = \{\omega_j^* : j = 1, \dots, G\}$ where ω_k is mapped onto $\{\omega_j^* : j \in J_k\}$. The bba m_i^Ω can be refined on Ω^* as $m_i^{\Omega^*}$ where

$$\begin{aligned} m_i^{\Omega^*}(\mathcal{R}(A)) &= m_i^\Omega(A) & \forall A \subseteq \Omega \\ &= 0 & \text{otherwise} \end{aligned} \quad (14)$$

For each case c_i , we must find out which of the G pdfs generated their $\mathbf{x}_i$ data. So, equation (13) can be written as:

$$L(\theta; \mathcal{L}) = \sum_{i=1}^N \log\left(\sum_{j=1}^G \pi_j f_j(\mathbf{x}_i; \alpha_j)\right) \quad (15)$$

where the sum of the π_j taken on the j indexes corresponding to the possible classes of c_i must add to 1, all others being 0.

We reconsider the EM algorithm when the teacher is imperfect. We need for each case c_i the plausibility of $\mathbf{x}_i$ given the bba $m_i^{\Omega^*}$ about its class in Ω^* . If the actual class is ω_j^* , then $pl^{IR^p}[\omega_j^*](\mathbf{x}_i)$ is given by $f_j(\mathbf{x}_i, \alpha_j)$. If $\mathbf{x}_i$ is a singleton (as usual and assumed hereafter) then $pl^{IR^p}[\omega_j^*](\mathbf{x}_i) = f_j(\mathbf{x}_i, \alpha_j)d\mathbf{x}$ where we put $d\mathbf{x}$ to mention that a plausibility is a set function whereas f itself is a density. This $d\mathbf{x}$ term will cancel when normalizing. Let $A \subseteq \Omega^*$, then from the disjunctive rule of combination associated to the GBT we get:

$$pl^{IR^p}[A](\mathbf{x}_i) = 1 - \prod_{j: \omega_j^* \notin A} (1 - pl^{IR^p}[\omega_j^*](\mathbf{x}_i)). \quad (16)$$

We then assess the bba on Ω^* given θ^l and $\mathbf{x}_i$. From the GBT, we get $m^{\Omega^*}[\mathbf{x}_i, \theta^l]$. We combine this bba with the prior bba given by $m_i^{\Omega^*}$ by the conjunctive combination rule. The term to maximize is then:

$$Q(\theta|\theta^l) = \sum_{i=1}^N \sum_{A \subseteq \Omega^*} (m^{\Omega^*}[x_i, \theta^l] \odot m_i^{\Omega^*})(A) \log(pl^{\mathbb{R}^p}[A](\mathbf{x}_i)) \quad (17)$$

where $pl^{\mathbb{R}^p}[A](\mathbf{x}_i)$ is given by relation (16).

6 Simulations Results

In this section, we propose to illustrate the performance of the CrEM algorithm described in the previous sections using two learning tasks.

6.1 Learning Task 1: Isosceles Triangles

In this task, we have three classes: $\Omega = \{\omega_1, \omega_2, \omega_3\}$ and two-dimensional data. In each class, there are 2 components ($G_k = 2, k = 1, 2, 3$). For a given subset, each vector $\mathbf{x}$ is generated from a Gaussian $f(\mathbf{x}|\omega_g) \sim \mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ where $\boldsymbol{\Sigma}_g = \sigma \mathbf{I}$. The parameters for the 6 pdfs are presented in table 2. The pdf corresponds to 3 largely spread data ($\sigma = 2$) located at the 3 corners of an isosceles triangle, and to 3 clustered data ($\sigma = 0.5$) located at the 3 corners of another isosceles triangle. The pair of pdf corresponding to one class are thus located at one corner and half way on the line between the other 2 corners. In figure 1, we illustrate an example of such a learning set with its respective isosceles triangles (fine lines).

We generate a sample of 50 cases from each of the 6 pdfs. Labels for each case can be of two types, either imprecise (IL) or credal (CrL). In the IL case, the labels for the 50 cases from the largely spread data (those at the corners) are precise. The other 50 cases are randomly split into two groups of 25 cases. Their labels are imprecise and made of 2 classes, the actual class being one of them. So for the 50 cases in subset 2 of class ω_1 , 25 are labeled $\{\omega_1, \omega_2\}$ and 25 are labeled $\{\omega_1, \omega_3\}$. In the CrL case, the labels are subsets of Ω randomly

Table 2. Parameters of the learning set for task 1 with imprecise labels (IL) and the estimations obtained with the CrEM for one run

	$\omega_1(+)$ subset1	$\omega_1(+)$ subset2	$\omega_2(\times)$ subset1	$\omega_2(\times)$ subset2	$\omega_3(\cdot)$ subset1	$\omega_3(\cdot)$ subset2
μ_a	10	17.5	15	15	20	12.5
μ_b	10	14.3	18.6	10	10	14.3
σ	2	0.5	2	0.5	2	0.5
IL cases	50 ω_1	25 ω_1, ω_2 25 ω_1, ω_3	50 ω_2	25 ω_1, ω_2 25 ω_2, ω_3	50 ω_3	25 ω_1, ω_3 25 ω_2, ω_3
m_a	9.13	17.54	15.60	14.92	20.36	12.42
m_b	10.35	14.32	18.95	10.12	9.86	14.35
s	2.57	0.38	1.85	0.37	3.24	0.35
r	0.185	0.152	0.178	0.148	0.179	0.154

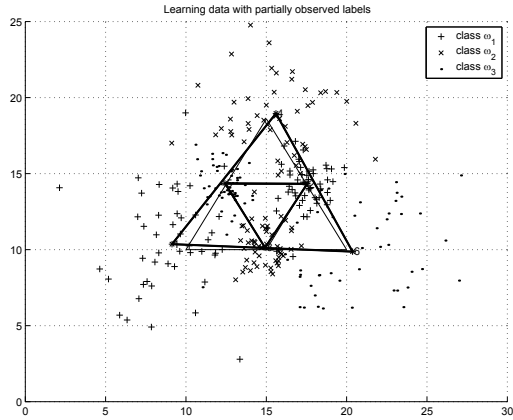


Fig. 1. Learning set in the feature space

Table 3. Percentage of correct classification for classical EM and CrEM algorithms

Triangles	1	2	3	4	5	6	7	8	9	10	mean	std
EM	85.3	84.3	86.3	88.0	86.7	87.0	83.3	85.7	90.7	88.0	86.5	2.1
CrEM IL	86.3	85.3	88.0	90.3	88.0	87.3	84.0	88.0	91.0	88.0	87.6	2.0
CrEM CrL	87.0	86.6	87.6	90.0	87.6	88.0	85.3	88.3	91.3	86.7	87.8	1.7

generated and each one receives a random mass. We thus generate imprecise and uncertain learning sets as they can be encountered in real world applications.

We run 10 simulations. For each of them, we generate the labels for the IL and CrL cases. In figure 1, we present the data for one simulation. The bold line triangle illustrates the result of the application of the CrEM for the IL case. As can be seen, the means (the corners of the triangles) are well located. The estimated parameters are listed at the bottom of table 2. On the IL data, we apply both a classical EM algorithm and the CrEM. On the CrL, we apply only the CrEM algorithm as the classical does not seem fitted for such type of data. In table 3, we present the Percentage of Correct Classification (PCC) obtained for each of the 10 independent training sets. Each method produces very similar results but only the CrEM algorithm is able to use credal labels, a much more flexible information than the one encountered in the IL case.

6.2 Learning Task 2: Qualitative Example

This learning set is drawn using three bi-dimensional Gaussian classes of standard deviation 1.5 respectively centered on (3,0), (0,5) and (0,0). Figure 2 illustrates this learning task associated to the decision regions computed using parameters of the CrEM algorithm learnt from credal labels (CrL). A very important, but classical feature using EM and mixture models algorithms, is the

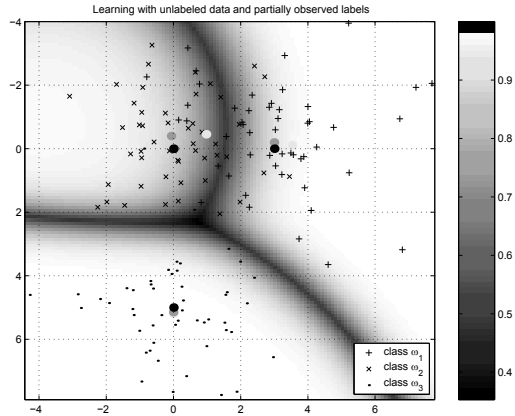


Fig. 2. Maximum pignistic probabilities as grey level values

Table 4. Estimated parameters of the learning task 2

	$\omega_1(+)$	$\omega_1(+)$	$\omega_2(\times)$	$\omega_2(\times)$	$\omega_3(\cdot)$	$\omega_3(\cdot)$
	μ_a	μ_b	μ_a	μ_b	μ_a	μ_b
Real values	3.00	0.00	0.00	0.00	0.00	5.00
Training set 1	3.52	-0.10	0.96	-0.45	-0.00	5.18
Training set 2	2.99	-0.19	-0.07	-0.40	-0.00	5.14

ability to cope with unlabeled samples. The first intuition is that these unlabeled data don't bring any information for learning the parameters of the generated data. Contrary to this idea, we can show on this illustrative example that unlabeled data give clearly a more precise idea of the real distributions. To highlight this issue, two training sets were considered: a training set (set 1) which contains all the data except that we randomly remove 40 cases (80%) of class ω_2 , and a training set (set 2) with all the data (150 cases). In this second learning set, we replace the credal labels generated for the 40 previous cases with vacuous belief functions (UL) before applying the CrEM classifier. Table 4 shows the estimated parameters for these two learning tasks. Additionally, estimated means are illustrated with gray levels disks in figure 2. This last capacity makes CrEM a very suitable algorithm for cluster analysis which is under study. In all these simulations, the estimation of the number of components G_k is a difficult model choice problem for which there is a number of possible solutions (Figueiredo & Jain, 2002). This problem is left for future works.

7 Conclusion

In this paper, a credal approach for partially supervised learning has been presented. The proposed methodology uses a variant of EM algorithm to estimate

parameters of mixture models and can cope with learning set where the knowledge about the actual class is represented by a belief function. Several simulations have proved the good performance of this CrEM algorithm compared to classical EM estimation in learning mixture of Gaussians.

Numerous applications of this approach can be mentioned. As example, let us consider Bayesian networks which use EM algorithms to estimate parameters of unknown distributions. Using CrEM algorithm can be a good alternative for belief networks. Future work is concerned with model selection issue which includes the choice of the number of components, shape of each component. . . Another important issue is the detection of outliers which can be solved by adding an extra component (uniform for example) in the mixture.

References

- Ambroise, C., Denoeux, T., Govaert, G., & Smets, P. (2001). Learning from an imprecise teacher: probabilistic and evidential approaches. In *Proceedings of asmda'2001* (Vol. 1, pp. 100–105). Compiègne, France.
- Ambroise, C., & Govaert, G. (2000). EM algorithm for partially known labels. In *Proceeding of IFCS'2000* (Vol. 1). Namur, Belgium.
- Delmotte, F., & Smets, P. (2004). Target identification based on the transferable belief model interpretation of Dempster-Shafer model. *IEEE Transactions on Systems, Man and Cybernetics, A* 34, 457–471.
- Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood estimation from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, series B*, 39, 1–38.
- Denceux, T. (1995). A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics*, 25(5), 804–813.
- Denceux, T., & Zouhal, L. M. (2001). Handling possibilistic labels in pattern classification using evidential reasoning. *Fuzzy Sets and Systems*, 122, 47–62.
- Figueiredo, M. A. T., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3), 381–396.
- Hastie, T., & Tibshirani, R. J. (1996). Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society B*, 58, 155–176.
- Kennes, R. (1992). Computational aspects of the Möbius transform of a graph. *IEEE-SMC*, 22, 201–223.
- McLacelan, G. J., & Krishnan, T. (1997). *The EM algorithm and extensions*. New York: John Wiley.
- Smets, P. (1993). Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem. *Int. J. Approximate Reasoning*, 9, 1–35.
- Smets, P. (1998). The transferable belief model for quantified belief representation. In D. M. Gabbay & P. Smets (Eds.), *Handbook of defeasible reasoning and uncertainty management systems* (Vol. 1, pp. 267–301). Kluwer, Dordrecht, The Netherlands.
- Smets, P. (2005). Decision making in the TBM: the necessity of the pignistic transformation. *Int. J. Approximate Reasoning*, 38, 133–147.

- Smets, P., & Kennes, R. (1994). The transferable belief model. *Artificial Intelligence*, 66, 191–234.
- Zouhal, L. M., & Dencœux, T. (1998). An evidence theoretic k-nn rule with parameter optimisation. *IEEE Transactions on Systems, Man and Cybernetics - Part C*, 28, 263–271.

Default Clustering from Sparse Data Sets

J. Velcin and J.-G. Ganascia

LIP6, Université Paris VI, 8 rue du Capitaine Scott, 75015 Paris, France
{julien.velcin, jean-gabriel.ganascia}@lip6.fr

Abstract. Categorization with a very high missing data rate is seldom studied, especially from a non-probabilistic point of view. This paper proposes a new algorithm called *default clustering* that relies on default reasoning and uses the local search paradigm. Two kinds of experiments are considered: the first one presents the results obtained on artificial data sets, the second uses an original and real case where political stereotypes are extracted from newspaper articles at the end of the 19th century.

Introduction

Missing values are of great interest in a world in which information flows play a key role. Most data analysis today has to deal with a lack of data due to voluntary omissions, human error, broken equipment, etc. [1]. Three kinds of strategies are generally used to handle such data: ignoring the incomplete observations (the so-called “list-wise deletion”), estimating the unknown values with other variables (single or multiple imputation, k-nearest-neighbors [2], maximum likelihood approaches [3]) or using the background knowledge to complete the “gaps” automatically with default values (arbitrary values, default rules).

The present work proposes a strategy which is not based on information completion but on default reasoning. The goal is to extract a set of some very complete descriptions that summarize as well as possible the whole data set. For this purpose, a clustering algorithm is proposed that is based on local search techniques and constraints specific to the context of sparse data.

Section 1 presents a new approach to conceptual clustering when missing information exists. Section 2 proposes a general framework, applied to the attribute-value formalism. The new notion of default subsumption is introduced, before seeing how the concept of stereotype makes it possible to name clusters. A stereotype set extraction algorithm is then presented. Section 3 concerns experiments, first on artificial data sets and secondly with a real data case generated from newspaper articles.

1 Dealing with Missing Values

1.1 Missing Values and Clustering

Generally, in Data Analysis, missing values are primarily solved just before starting the “hard” analysis itself (e.g. Multiple Correspondence Analysis [4]). But

this sort of pre-processing method is not really flexible for classification purposes, especially with a high rate of missing values.

This paper presents the problem performed in a non-supervised way, as with the well-known clustering algorithms k-means (its categorical version: k-modes) and EM (Expectation-Maximization). But contrary to these algorithms that can easily lead to local optima, we have chosen to achieve the clustering using a combinatorial optimization approach, like in [5] or [6]. Our goal here is not only to cluster examples but also and mainly to describe the cluster easily and in an understandable way. The problem can thus be stated as finding readable, understandable, consistent and rich descriptions.

Each of these descriptions covers part of the data set. The examples belonging to a part can be considered as *equivalent* according to the covering description. Note that our interest is focused on the similarity between the examples and the cluster descriptions, and not between the examples themselves.

1.2 Default Clustering

E. Rosch saw the categorization itself as one of the most important issues in cognitive science [7]. She introduced the concept of prototype as the ideal member of a category. Whereas categorization makes similar observations fit together and dissimilar observations be well separated, clustering is the induction process in data mining that actually build such categories. More specifically, conceptual clustering is a machine learning task defined by R. Michalski [8] which does not require a teacher and uses an evaluation function to discover classes named with appropriate conceptual descriptions. Conceptual clustering was principally studied in a probabilistic context (see, for instance, D. Fisher's Cobweb algorithm [9]) and seldom used really sparse data sets. For instance, the experiments done by P.H. Gennari do not exceed 30% of missing values [10].

This paper proposes a new technique called *default clustering* which is inspired by the default logic of R. Reiter [11]. We use a similar principle but for induction, when missing information exists. The main assumption is the following: if an observation is grouped with other similar observations, you can use these observations to complete unknown information in the original fact if it remains consistent with the current context. Whereas default logic needs implicit knowledge expressed by default rules, default clustering only uses information available in the data set. The next section presents this new framework. It shows how to extract stereotype sets from very sparse data sets: first it extends the classical subsumption, next it discusses stereotype choice, and finally it proposes a local search strategy to find the best solution.

2 Logical Framework

This section presents the logical framework of default clustering in the attribute-value formalism (an adaptation to conceptual graphs can be found in [12]). The description space is noted $\mathcal{D}$, the attribute space $\mathcal{A}$, the descriptor space (i.e. the values the attributes can take) $\mathcal{V}$ and the example set E . The function δ maps

each example $e \in E$ to its description $\delta(e) \in \mathcal{D}$. Note that this logical framework only presents categorical attributes, but it has been easily extended to ordinal attributes.

2.1 Default Subsumption

Contrary to default logic, the problem here is not to deduce, but to induce knowledge from data sets in which most of the information is unknown. Therefore, we put forward the notion of *default subsumption*, which is the equivalent for subsumption of the default rule for deduction. Saying that a description $d \in \mathcal{D}$ subsumes $d' \in \mathcal{D}$ by default means that there exists an implicit description d'' such that d' completed with d'' , i.e. $d' \wedge d''$, is more specific than d in the classical sense, which signifies that $d' \wedge d''$ entails d . The exact definition follows:

Definition 1. d subsumes d' by default (noted $d \leq_D d'$) iff $\exists d_c$ such that $d_c \neq \perp$ and $d \leq d_c$ and $d' \leq d_c$ where $t \leq t'$ stands for t subsumes t' in the classical sense. d_c is a minorant of d and d' in the subsumption lattice.

To illustrate our definition, here are some descriptions based on binary attributes that can be compared with respect to the default subsumption:

$$\begin{aligned} d_1 &= \{(\text{Traitor}=\text{yes}), (\text{Internationalist}=\text{yes})\} \\ d_2 &= \{(\text{Traitor}=\text{yes}), (\text{Connection\_with\_jews}=\text{yes})\} \\ d_3 &= (\text{Patriot}=\text{yes}) \end{aligned}$$

$d_1 \leq_D d_2$ and $d_2 \leq_D d_1$ because $\exists d_c$ such that $d_1 \leq d_c$ and $d_2 \leq d_c$:

$$d_c = \{(\text{Traitor}=\text{yes}), (\text{Internationalist}=\text{yes}), (\text{Connection\_with\_jews}=\text{yes})\}$$

However, considering that a patriot cannot be an internationalist and vice-versa, i.e. $\neg((\text{Patriot}=\text{yes}) \wedge (\text{Internationalist}=\text{yes}))$, which was an implicit statement for many people living in France at the end of the 19th century, d_1 does not subsume d_3 by default, i.e. $\neg(d_1 \leq_D d_3)$.

Property 1. The notion of default subsumption is more general than classical subsumption since, if d subsumes d' , i.e. $d \leq d'$, then d subsumes d' by default, i.e. $d \leq_D d'$. The converse is not true: if $d \leq_D d'$, we do not know if $d \leq d'$.

Property 2. The default subsumption relationship is symmetrical, i.e. $\forall d \forall d'$ if $d \leq_D d'$ then $d' \leq_D d$.

Note that the notion of default subsumption may appear strange for people accustomed to classical subsumption because of the symmetrical relationship. As a consequence, it does not define an ordering relationship on the description space $\mathcal{D}$. The notation $\leq_D$ may be confusing with respect to this symmetry, but it is relative to the underlying idea of generality.

2.2 Concept of Stereotype

In the literature of categorization, Rosch introduced the concept of prototype [7, 13] inspired by the family resemblance notion of Wittgenstein [14] (see [15])

for an electronic version and [16] for an analysis focused on family resemblance). Even if our approach and the original idea behind the concept of prototype have several features in common, we prefer to refer to the older concept of stereotype that was introduced by the publicist W. Lippman in 1922 [17]. For him, stereotypes are perceptive schemas (a structured association of characteristic features) shared by a group about other person or object categories. These simplifying and generalizing images about reality affect human behavior and are very subjective. Below are three main reasons to make such a choice.

First of all, the concept of prototype is often misused in data mining techniques. It is reduced to either an average observation of the examples or an artificial description built on the most frequent shared features. Nevertheless, both of them are far from the underlying idea in family resemblance. Especially in the context of sparse data, it seems more correct to speak about a combination of features found in different example descriptions than about average or mode selection. The second argument is that the notion of stereotype is often defined as an imaginary picture that distorts the reality. Our goal is precisely to generate such pictures even if they are caricatural of the observations. Finally, these specific descriptions are better adapted for fast classification (we can even say discrimination) and prediction than prototypes, which is closely linked to Lippman's definition.

In order to avoid ambiguities, we restrict the notion to a specific description $d \in \mathcal{D}$ associated to (we can say "covering") a set of descriptions $D \subset \mathcal{D}$. However, the following subsection does not deal just with stereotypes but with stereotype sets to cover a whole description set. The objective is therefore to automatically construct stereotype sets, whereas most of the studies are focused on already fixed stereotype usage [18, 19]. Keeping this in mind, the space of all the possible stereotype sets is browsed in order to discover the best one, i.e. the set that best covers the examples of E with respect to some similarity measure. But just before addressing the search itself, we should consider both the relation of relative cover and the similarity measure used to build the categorization from stereotype sets.

2.3 Stereotype Sets and Relative Cover

Given an example e characterized by its description $d = \delta(e) \in \mathcal{D}$, consider the following statement: the stereotype $s \in \mathcal{D}$ is allowed to cover e if and only if s subsumes d by default. It means that in the context of missing data each piece of information is so crucial that even a single contradiction prevents the stereotype from being a correct generalization. Furthermore, since there is no contradiction between this example and its related stereotype, the stereotype may be used to complete the example description.

In order to perform the clustering, a very general similarity measure M_{sim} has been defined, which counts the number of common descriptors of $\mathcal{V}$ belonging to two descriptions, ignores the unknown values and takes into account the default subsumption relationship:

$$\begin{aligned}
 M_{sim} : \mathcal{D} \times \mathcal{D} &\longrightarrow \mathbb{N}^+ \\
 (d_i, d_j) &\longmapsto M_{sim}(d_i, d_j) = |\{v \in d/d = d_i \wedge d_j\}| \text{ if } d_i \leq_D d_j, \\
 &M_{sim}(d_i, d_j) = 0 \text{ if } \neg(d_i \leq_D d_j).
 \end{aligned}$$

where $d_i \wedge d_j$ is the least minorant of d_i and d_j in the subsumption lattice.

Let us now consider a set $S = \{s_\emptyset, s_1, s_2 \dots s_n\} \subset \mathcal{D}$ of stereotypes. $s_\emptyset$ is the absurd-stereotype linked to the set $E_\emptyset$. Then, a categorization of E can be calculated using S with an affectation function which we called *relative cover*:

Definition 2. *The relative cover of an example $e \in E$, with respect to a set of stereotypes $S = \{s_\emptyset, s_1, s_2 \dots s_n\}$, noted $C_S(e)$, is the stereotype s_i if and only if:*

1. $s_i \in S$,
2. $M_{sim}(\delta(e), s_i) > 0$,
3. $\forall k \in [1, n], k \neq i, M_{sim}(\delta(e), s_i) > M_{sim}(\delta(e), s_k)$.

It means that an example $e \in E$ is associated to the most similar and “covering-able” stereotype *relative* to the set S . If there are two competitive stereotypes with an equal higher score or if there is no covering stereotype, then the example is associated to the absurd-stereotype $s_\emptyset$. In this case, no completion can be calculated for e .

Note that C_S defines an equivalence relation on E . Given an example e , consider now the projection of its description $\delta(e)$ on the descriptors belonging to $C_S(e)$. This projection, noted $\delta(e)|_{C_S}$, naturally subsumes the original description $\delta(e)$. If e_i and e_j are covered by the same stereotype, i.e. $C_S(e_i) = C_S(e_j)$, then the projection of e_i can be subsumed by default by the projection of e_j . More formally:

Property 3. $\forall e_i, e_j \in E^2, C_S(e_i) = C_S(e_j) \Rightarrow \delta(e_i)|_{C_S} \leq_D \delta(e_j)|_{C_S}$.

This means that the examples covered by the same stereotype are considered equivalent *if we consider as negligible* the descriptors that do not belong to this stereotype. This shows that, beyond the use of stereotypes, it is the examples themselves that are used to complete the sparse descriptions.

2.4 Stereotype Extraction

In this paper, default reasoning is formalized using the notions of both default subsumption and stereotype set. Up to now, these stereotype sets were supposed to be given. This section shows how the classification can be organized into such sets in a non-supervised learning task. It can be summarized as follows. Given:

1. An example set E .
2. A description space $\mathcal{D}$.
3. A description function $\delta: E \longrightarrow \mathcal{D}$ which associates a description $\delta(e) \in \mathcal{D}$ to each example belonging to the training set E .

The function of a non-supervised learning algorithm is to organize the initial set of individuals E into a structure (for instance a hierarchy, a lattice or a pyramid).

In the present case, the structure is limited to partitions of the training set, which corresponds to searching for stereotype sets as discussed above. These partitions may be generated by $(n + 1)$ stereotypes $S = \{s_\emptyset, s_1, s_2 \dots s_n\}$: it is sufficient to associate to each s_i the set E_i of examples e belonging to E and covered by s_i relative to S . The examples that cannot be covered by any stereotype are put into the $E_\emptyset$ cluster and associated to $s_\emptyset$.

To choose from among the numerous possible partitions, which is a combinatorial problem, a non-supervised algorithm requires a function for evaluating stereotype set relevance. Because of the categorical nature of data and the previous definition of relative cover, it appears natural to make use of the similarity measure M_{sim} . This is exactly what we do by introducing the following cost function h_E :

Definition 3. *E being an example set, $S = \{s_\emptyset, s_1, s_2 \dots s_n\}$ a stereotype set and C_S the function that associates to each example e its relative cover, i.e. its closest stereotype with respect to M_{sim} and S , the cost function h_E is defined as follows:*

$$h_E(S) = \sum_{e \in E} M_{sim}(\delta(e), C_S(e))$$

While k-modes and EM algorithms are straightforward, i.e. each step leads to the next one until convergence, we reduce here the non-supervised learning task to an optimization problem. This approach offers several interesting features: avoiding local optima (especially with categorical and sparse data), providing “good” solutions even if not the best ones, better control of the search. In addition, it is not necessary to specify the number of expected stereotypes that is also discovered during the search process.

There are several methods for exploring such a search space (hill-climbing, simulated annealing, etc.), but we have chosen the meta-heuristic called *tabu search* which improves the local search algorithm. Remember that the local search process can be schematized as follows: 1. An initial solution S_{ini} is given (for instance at random). 2. A neighborhood is calculated from the current solution S_i with the assistance of permitted movements. These movements can be of low influence (enrich one stereotype with a descriptor, remove a descriptor from another) or of high influence (add or retract one stereotype to or from the current stereotype set). 3. The best movement, relative to the evaluation function h_E , is chosen and the new current solution S_{i+1} is computed. 4. The process is iterated a specific number of times and the best up-to-now discovered solution is recorded. Then, the solution is the stereotype set S_{max} that best maximizes h_E in comparison to all the crossed sets.

As in almost all local search techniques, there is a trade-off between exploitation, i.e. choosing the best movement, and exploration, i.e. choosing a non optimal state to reach completely different areas. The tabu search extends the basic local search by manipulating short and long-term memories which are used to avoid loops and to intelligently explore the search space. We shall not detail here this meta-heuristic but suggest you read the book by Glover and Laguna [20].

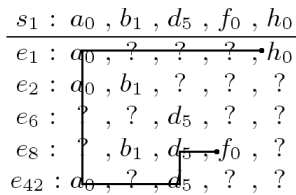
2.5 Constraints on Stereotypes

A “no-redundancy” constraint has been added in order to obtain a perfect separation between the stereotypes. In the context of sparseness, it seems really important to extract contrasted descriptions, which are used to quickly classify the examples, as does the concept of stereotype introduced by Lippman.

A new constraint called *cognitive cohesion* is now defined. It verifies cohesion within a cluster, i.e. an example set $E_j \subset E$, relative to the corresponding stereotype $s_j \in S$. Cognitive cohesion is verified if and only if, given two descriptors v_1 and $v_2 \in \mathcal{V}$ of s_j , it is always possible to find a series of examples that make it possible to pass by correlation from v_1 to v_2 . Below are two example sets with their covering stereotype. The example on the left verifies the constraint, the one on the right does not.

$s_1 : a_0, b_1, d_5, f_0, h_0$	$s_2 : a_0, b_1, d_5, f_0, h_0$
$e_1 : a_0, ?, ?, ?, h_0$	$e_0 : a_0, b_1, ?, ?, ?$
$e_2 : a_0, b_1, ?, ?, ?$	$e_8 : ?, ?, ?, f_0, ?$
$e_6 : ?, ?, d_5, ?, ?$	$e_9 : a_0, b_1, ?, ?, ?$
$e_8 : ?, b_1, d_5, f_0, ?$	$e_{51} : ?, ?, d_5, ?, h_0$
$e_{42} : a_0, ?, d_5, ?, ?$	$e_{101} : ?, ?, d_5, ?, h_0$

Hence, with s_2 it is never possible to pass from a_0 to d_5 , whereas it is allowed by s_1 (for instance with e_2 and then e_8). In the case of s_1 , you are always able to find a “correlation path” from one descriptor of the description to another, i.e. examples explaining the relationship between the descriptors in the stereotype. The graph below gives an example of a path between the descriptor h_0 and the descriptor f_0 , using e_1 , e_{42} and e_8 :



3 Experiments

This section presents experiments performed on artificial data sets. This is followed by an original comparison in a real data case using three well-known clusterers. Default clustering was implemented in a Java program called PRESS (*Programme de Reconstruction d’Ensembles de Stéréotypes Structurés*). All the experiments for k-modes, EM and Cobweb were performed using the Weka platform [21]. Note that the data sets used in the following correspond to the default clustering assumptions.

3.1 Validation on Artificial Data Sets

These experiments use artificial data sets to validate the robustness of our algorithm. The first step is to give some contrasted descriptions of $\mathcal{D}$. Let us note n_s the number of these descriptions. Next, these initial descriptions are duplicated n_d times. Finally, missing data are artificially simulated by removing a percentage p of descriptors at random from these $n_s \times n_d$ artificial examples. The evaluation is carried out by testing different clusterers on these data and comparing the discovered cluster representatives with the initial descriptions. We verify what we call *recovered descriptors*, i.e. the proportion of initial descriptors that are found. This paper presents the results obtained with $n_s = 5$ and $n_d = 50$ over 50 runs. The number of examples is 250 and the descriptions are built using a language of 30 binary attributes. Note that these experiments are placed in the Missing Completely At Random (MCAR) framework.

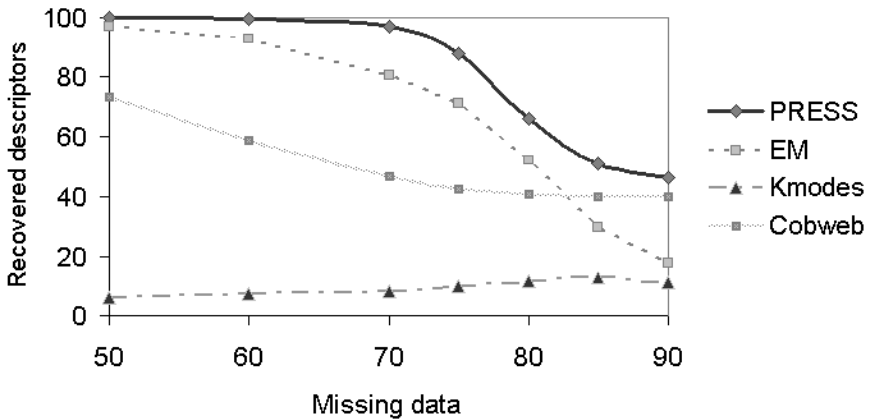


Fig. 1. Proportion of recovered descriptors

Fig. 1 shows firstly that the results of PRESS are very good using a robust learning process. The stereotypes discovered correspond very well to the original descriptions up to 75% of missing data. In addition, this score remains good (nearly 50%) up to 90%. Whereas Cobweb seems stable relative to the increase in the number of missing values, the results of EM rapidly get worse above 80%. Those obtained using k-modes are the worst, although the number of expected medoids has to be specified.

3.2 Studying Social Misrepresentation

The second part of the experiments deals with real data extracted from a newspaper called “Le Matin” from the end of the 19th century in France. The purpose is to automatically discover stereotype sets from events related to the political disorder in the first ten days of September 1893. The results of PRESS are

compared to those of the three clusterers k-modes, EM and Cobweb. It should be pointed out that our interest focuses on the cluster descriptions, which we call *representatives* to avoid any ambiguity, rather than on the clusters themselves.

The articles linked to the chosen theme were gathered and represented using a language with 33 attributes. The terms of this language, i.e. attributes and associated values, were extracted manually. Most of the attributes are binary, some accept more than two values and some are ordinals. The number of extracted examples is 63 and the rate of missing descriptors is nearly 87%, which is most unusual.

3.3 Evaluation of Default Clustering

In order to evaluate PRESS, a comparison was made with three classical clusterers: k-modes, EM and Cobweb. Hence, a non-probabilistic description of the clusters built by these algorithms was extracted using four techniques: (1) using the more frequent descriptors (mode approach); (2) the same as (1) but forbidding contradictory features between the examples and their representative; (3) dividing the descriptors between the different representatives; (4) the same as (3) but forbidding contradictory features. Two remarks need to be made. Firstly, the cluster descriptions resulting from k-modes correspond to technique (1). Nevertheless, we tried the other three techniques exhaustively. Secondly, representatives resulting from extraction techniques (3) and (4) validate *by construction* the no-redundancy constraint. The comparison was made according to the following three points:

The first approach considers the contradictions between an example and its representative. The *example contradiction* is the percentage of examples containing at least one descriptor in contradiction with its covering representative. In addition, if you consider one of these contradictory examples, *average contradiction* is the percentage of descriptors in contradiction with its representative. This facet of conceptual clustering is very important, especially in the sparse data context.

Secondly, we check if the constraints described in section 2.5 (i.e. cognitive cohesion and no-redundancy) are verified. They are linked to the concept of stereotype and to the sparse data context.

Finally, we consider the degree of similarity between the examples and their covering representatives. This corresponds to the notion of compactness within clusters, but without penalizing the stereotypes with many descriptors. The function h_E seems really adapted to render an account of representative relevance. In fact, we used a version of h_E normalized between 0 and 1 by dividing, by the total number of descriptors.

3.4 Results

Fig. 2 gives the results obtained from the articles published in Le Matin. Experiments for the k-modes algorithm were carried out with $N = 2 \dots 8$ clusters, but only $N = 6$ results are presented in this comparison. The rows of the table show the number n of extracted representatives, the two scores concerning contradic-

	k-Modes				EM				Cobweb				PRESS
	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)	(1)	(2)	(3)	(4)	
n	6	6	6	6	2	2	2	2	2	2	2	2	6
ex. contradiction	27	0	27	0	48	0	48	0	56	0	57	0	0
av. contradiction	42	0	44	0	56	0	56	0	52	0	51	0	0
h_E	.89	.60	.74	.50	.85	.66	.83	.65	.82	.56	.68	.46	.79
redundancy	70	63	0	0	17	7	0	0	72	55	0	0	0
cog. cohesion	×	×	×	×	×	×	×	×	×	×	×	×	✓

Fig. 2. Comparative results on *Le Matin*

tion, the result of h_E , the redundancy score and whether or not the cognitive cohesion constraint is verified. The columns represent each type of experiment (k-modes associated with techniques (1) to (4), EM and Cobweb as well, and finally our algorithm PRESS).

Let us begin by considering the contradiction scores. They highlight a principal result of default clustering: using PRESS, the percentage of examples having contradictory features with their representative is always equal to 0%. In contrast, the descriptions built using techniques (1) and (3) (whatever the clusterer used) possess at least one contradictory descriptor with 27% to 57% of the examples belonging to the cluster. Furthermore, around 50% of the descriptors of these examples are in contradiction with the covering description, and that can in no way be considered as a negligible noise. This is the reason why processes (1) and (3) must be avoided, especially in the sparse data context, when building such representatives from k-modes, EM or Cobweb clustering. Hence, we only consider techniques (2) and (4) in the following experiments.

Let us now study the results concerning clustering quality. This quality can be expressed thanks to the compactness function h_E , the redundancy rate and cognitive cohesion.

PRESS marked the best score (0.79) for cluster compactness with six stereotypes. That means a very good homogeneity between the stereotypes and the examples covered. It is perfectly consistent since our algorithm tries to maximize this function. The redundant descriptors rate is equal to 0%, according to the no-redundancy constraint. Furthermore, PRESS is the only algorithm that is able to verify cognitive cohesion. EM obtains the second best score and redundant descriptor rate remains acceptable. However, the number of expected classes must be given or guessed using a cross-validation technique, for instance. K-modes and Cobweb come third and fourth and also have to use an external mechanism to discover the final number of clusters.

Note that the stereotypes extracted using PRESS correspond to the political leanings of the newspaper. For instance, the main stereotype produces a radical, socialist politician, corrupted by foreign money and Freemasonry, etc. It corresponds partly to the difficulty in accepting the major changes proposed by the radical party and to the fear caused in France since 1880 by the theories of Karl Marx. We cannot explain here in more detail the semantics of discovered stereotypes, but these first results are really promising.

4 Conclusion

Sparse data clustering is seldom studied in a non-probabilistic way and with such a high number of missing values. However, it is really important to be able to extract readable, understandable descriptions from such type of data in order to complete information, to classify new observations quickly and to make predictions. In this way, the default clustering presented in this paper tries to provide an alternative to the usual clusterers.

This algorithm relies on local optimization techniques that implement a very basic version of the tabu search meta-heuristic. Part of our future work will be to extend these techniques for stereotype set discovering. Hence, an efficient tabu search has to develop a long-term memory and to use more appropriate intensification and diversification strategies (e.g. *path-relinking* strategy).

The results obtained, on both artificial data sets and a real case extracted from newspaper articles, are really promising and should lead to other historical studies concerning social stereotypes. Another possible extension is to apply these techniques to the study of *social representations*, a branch of social psychology introduced by S. Moscovici in 1961 [23]. More precisely, this approach is really useful for *press content study* which up to now is done manually by experts. Here it would be a question of choosing key dates of the Dreyfus affair and automatically extracting stereotypical characters from different newspapers. These results will then be compared and contrasted with the work of sociologists and historians of this period.

Acknowledgments

The authors would particularly like to thank Rosalind Greenstein for reading and correcting the manuscript.

References

1. Newgard, C.D., Lewis, R.J.: The Imputation of Missing Values in Complex Sampling Databases: An Innovative Approach. In: Academic Emergency Medicine, Volume 9, Number 5484. Society for Academic Emergency Medicine (2002).
2. Huang, C.-C., Lee, H.-M.: A Grey-Based Nearest Neighbor Approach for Missing Attribute-Value Prediction. In: Applied Intelligence, Volume 20. Kluwer Academic Publishers (2004) pp.239–252.
3. Ghahramani, Z., Jordan, M.-I.: Supervised learning from incomplete data via an EM approach. In: Advances in Neural Information Processing Systems, Volume 6. Morgan Kaufmann Publishers (1994), San Francisco.
4. Benzecri, J.P.: Correspondence Analysis Handbook, New York: Marcel Dekker (1992).
5. Figueroa, A., Borneman, J., Jiang, T.: Clustering binary fingerprint vectors with missing values for DNA array data analysis (2003).
6. Sarkar, M., Leong, T.Y.: Fuzzy K-means clustering with missing values. In: Proc AMIA Symp. PubMed (2001) pp.588–92.

7. Rosch, E.: Cognitive representations of semantic categories, In: *Journal of Experimental Psychology: General*, number 104 (1975) pp.192–232.
8. Michalski, R.S.: Knowledge acquisition through conceptual clustering: A theoretical framework and algorithm for partitioning data into conjunctive concepts. In: *International Journal of Policy Analysis and Information Systems*, 4 (1980) pp.219–243.
9. Fisher, D.H.: Knowledge Acquisition Via Incremental Conceptual Clustering. In: *Machine Learning*, number 2 (1987) pp.139–172.
10. Gennari, J.H.: An experimental study of concept formation. Doctoral dissertation (1990), Department of Information & Computer Science, University of California, Irvine.
11. Reiter, R.: A logic for default reasoning. In: *Artificial Intelligence*, number 13 (1980) pp.81–132.
12. Velcin, J., Ganascia, J.-G.: Modeling default induction with conceptual structures, In: *ER 2004 Conference Proceedings*. Lu, Atzeni, Chu, Zhou, and Ling editors. Springer-Verlag (2004), Shangai, China.
13. Rosch, E.: Principles of categorization, In: *Cognition and Categorization*. NJ: Lawrence Erlbaum, Hillsdale (1978) pp.27–48.
14. Wittgenstein, L.: *Philosophical Investigations*. Blackwell (1953), Oxford, UK.
15. Shawver, L.: Commentary on Wittgenstein's *Philosophical Investigations*. In: <http://users.rcn.com/rathbone/lw65-69c.htm>.
16. Narboux, J.-P.: Ressemblances de famille, caractères, critères, In: *Wittgenstein : mtaphysique et jeux de langage*. PUF (2001) pp.69–95.
17. Lippman, W.: *Public Opinion*, Ed. MacMillan (1922), NYC.
18. Rich, E.: User Modeling via Stereotypes. In: *International Journal of Cognitive Science*, 3 (1979) pp.329–354.
19. Amosy, R., Herschberg Pierrot, A.: *Stéréotypes et clichés : langues, discours, société*. Nathan Université (1997).
20. Glover, F., Laguna, M.: *Tabu Search*, Kluwer Academic Publishers (1997).
21. Garner, S.R.: WEKA: The waikato environment for knowledge analysis, In: *Proc. of the New Zealand Computer Science Research Students Conference* (1995) pp.57–64.
22. Moscovici, S.: *La psychanalyse : son image et son public*. PUF (1961), Paris.

New Technique for Initialization of Centres in TSK Clustering-Based Fuzzy Systems

Luis Javier Herrera, Héctor Pomares, Ignacio Rojas,
Alberto Guillén, and Jesús González

University of Granada, Department of Computer Architecture and Technology,
E.T.S. Computer Engineering, 18071 Granada, Spain
<http://atc.ugr.es>

Abstract. Several methodologies for function approximation using TSK systems make use of clustering techniques to place the rules in the input space. Nevertheless classical clustering algorithms are more related to unsupervised learning and thus the output of the training data is not taken into account or, simply the characteristics of the function approximation problem are not considered. In this paper we propose a new approach for the initialization of centres in clustering-based TSK systems for function approximation that takes into account the expected output error distribution in the input space to place the fuzzy system rule centres. The convenience of proposed the algorithm comparing to other input clustering and input/output clustering techniques is shown through a significant example.

1 Introduction

The problem of function approximation deals with estimating an unknown function f from samples of the form $\{(\mathbf{x}^m; z^m) \ ; m = 1, 2, \dots, M; \text{ with } z^m = f(\mathbf{x}^m) \in \mathbb{R}, \text{ and } \mathbf{x}^m \in \mathbb{R}^m\}$ and is a crucial problem for a number of scientific and engineering areas. The main goal is thus to learn an unknown functional mapping between the input vectors and their corresponding output values, using a set of known training samples. Later, this generated mapping will be used to obtain the expected output given any new input data. Regression or function approximation problems deal with continuous input/output data in contrast to classification problems that deal with discrete, categorical output data.

Fuzzy Systems are widely applied for both classification and Function Approximation problems. Specifically, for function approximation problems, two main techniques appear in the literature, Grid-Based Fuzzy Systems (GBFSs) [5] and Clustering-Based Fuzzy Systems (CBFSs) [6], whose main difference is the type of partitioning of the input space. GBFS have the advantage that they perform a thorough coverage of the input space, but at the expense of suffering from the curse of dimensionality that makes them inapplicable for problems with moderate complexity. In contrast, Clustering-Based Fuzzy System (CBFSs) techniques place the rules in the zones of the input space in which they are needed,

being more suitable thus for example for Time Series Prediction problems in which the input data is more centralized in some regions of the input space, or for problems with moderated complexity and a higher number of input variables.

CBFS techniques usually utilize a clustering approach [3] for the initialization of the rule centres and afterwards perform an optimization process in order to obtain the pseudo-optimal rule parameters (centres and weights) using gradient descent, constraint optimization [4], etc. The use of clustering approaches for the initialization of the rule centres is mainly based on the idea of performing clustering in the input space and associating a weight or functional value to this region of the input space. Nevertheless, this idea might be more appropriate in classification problems; in function approximation problems, input space cluster interrelation does not necessarily carry out such output cluster interrelation. Input-output clustering techniques [1, 2] solve partially this problem since they consider the output variable/s in the clustering process. The input/output CFA clustering algorithm [1] for example performs a output-variance weighted input-space clustering according to a modified distortion measure.

In this paper we present a new approach for rule centres initialization that does not minimize a classical clustering distortion function, but that uses the final function-approximation-error function

$$J = \sum_{m \in D} (f(\mathbf{x}^m) - z^m)^2 \quad (1)$$

to place the centres pseudo-optimally. The idea of our approach is to place the centres so that the estimated error along each corresponding input space region is similar. Or, similarly, by forcing each centre to have a similar error, according to $\dots - 1$, in each side of every input dimension.

The rest of the paper is organized as follows. Section 2 presents and discuss our Error Equidistribution Method (EEM) for the initialization of centres in CBFS. Section 3 presents an example and compares our EEM approach with other previous clustering methodologies. Finally in Section 4 we present the conclusions obtained from this work.

2 Error Equidistribution Method for Initialization of Rule Centres in CBFS for Function Approximation

In this section we present the new methodology proposed for the initialization of rule centres in CBFS for function approximation in the context of a general learning methodology.

Typically, the structure of a multiple-input single-output (MISO) Takagi-Sugeno-Kang (TSK) fuzzy system and its associated fuzzy inference method comprises a set of K IF-THEN rules in the form

$$Rule^k : \text{IF } x_1 \text{ is } \mu_1^k \text{ AND } \dots \text{ AND } x_n \text{ is } \mu_n^k \text{ THEN } y = R^k \quad (2)$$

where the μ_i^k are fuzzy sets characterized by membership functions $\mu_i^k(x_i)$ in universes of discourse U_i (in which variables x_i take their values), and where R_k are the consequents of the rules.

The output of a fuzzy system with rules in the form shown in . . . 2 can be expressed (using weighted average aggregation) as

$$F(\mathbf{x}) = \frac{\sum_{k=1}^K \mu_k(\mathbf{x}) y_k}{\sum_{k=1}^K \mu_k(\mathbf{x})} \quad (3)$$

provided that $\mu_k(\mathbf{x})$ is the activation value for the antecedent of the rule k , which can be expressed as

$$\mu_k(\mathbf{x}) = \mu_1^k(x_1) \mu_2^k(x_2) \dots \mu_n^k(x_n) \quad (4)$$

Given this formulation the learning process in a CBFS with a fixed number of fuzzy rules, can be subdivided in two main steps: optimization of rule consequents and optimization of rule antecedents, i.e. optimization of the membership function (MF) parameters.

Optimization of Fuzzy Rule Consequents. Given a fixed membership functions configuration, we can obtain optimally the rule consequents (no matter the degree of the polynomial rule consequent). The Least Squares approach (LSE) by obtaining the partial derivatives of J (see . . . 1) with respect to each of the consequents coefficients obtains a linear equation system that can be solved using any of the well-known mathematical methods for this purpose. In particular we will use Singular Value Decomposition (SVD) since it allows to detect redundancies (that make the problem ill-conditioned) in the equation system and easily remove them.

Optimization of Fuzzy Rule Antecedents. Given a fixed number of rules, according to the function approximation problem formulation, we wish to minimize the error function J (see . . . 1), but in this case the rule antecedent parameters (membership function parameters) can not be expressed as a linear function with respect to J . Thus a gradient descent or a constrained optimization could be applied, that would make use of the optimal rule consequents coefficients calculation. But these techniques have the drawback that they can easily fall in local minima.

Therefore, several approaches have been proposed for CBFS in order to find a good starting point for the rule centres, being most of them based in clustering techniques. Traditional clustering algorithms used in CBFS attempt to place the rule centres according to the set of vectors selected by a clustering technique, typically a fuzzy clustering algorithm [3]. These clustering algorithms can be divided into two conceptually different families [1]: input clustering and input/output clustering. Here we present a novel approach, more intuitive from the point of view of the function approximation problem formulation that is based on a previous work for GBFS [5].

2.1 Initialization of the Rule Centres Using the Error Equidistribution Method

For the general model we present in this paper, we will make use of gaussian membership functions. Thus, the parameters to be optimized for each MF would be the centre (composed by one centre value for input dimension) and the width, but for the sake of simplicity of our initialization approach we will use one width per centre for every dimension, that will be automatically calculated using the nearest centre criteria [8]. Therefore the only parameters that our initialization process will obtain will be the rule (cluster) centres.

The main purpose of our approach, instead of trying to minimize a classical distortion measure based on the distance of the training data points to the rule centres, it will try to place the rule centres so that the errors (according to 5.1) are homogeneously distributed over the whole output range.

The methodology to obtain such distribution of rule centres stays as follows. Starting from a random initialized (or using any simple clustering approach like k-means [10]) rule centres distribution, we will consider that a rule centre k is responsible for the error corresponding to each training point $\mathbf{x}^m$ using the next formula

$$J^k(\mathbf{x}^m) = \frac{\mu_k(\mathbf{x})}{\sum_{j=1}^K \mu_j(\mathbf{x})} (f(\mathbf{x}^m) - z^m)^2 \quad (5)$$

being thus

$$J = \sum_{m \in D} \sum_{j=1}^K J^k(\mathbf{x}^m) \quad (6)$$

Every rule centre k will have associated parameters S_{i-}^k and S_{i+}^k , that will reflect the error according to 5.5 on the “left” (minus sign) and on the “right” (plus sign) of the centre c_i^k (i.e. centre of the MF in rule k in dimension i).

$$S_{i+}^k = \sum_{\substack{m \in D \\ x_i^m \geq c_i^k}} J^k(\mathbf{x}^m) \quad (7)$$

$$S_{i-}^k = \sum_{\substack{m \in D \\ x_i^m < c_i^k}} J^k(\mathbf{x}^m) \quad (8)$$

Using this two parameters S_{i-}^k and S_{i+}^k , we will define a slope parameter

$$S_i^k = \frac{S_{i+}^k - S_{i-}^k}{\sigma_y^2} \quad (9)$$

(where σ_y^2 is the output variance) that will reflect the need of that centre to move “left” or “right” in order to equidistribute the error J along the input space.

An iterative process is performed now that will calculate this slope parameter for each dimension in each rule centre and that afterwards will move the centres according to the next formula

$$\Delta c_i^k = \begin{cases} \frac{d_{min}}{b} \frac{S_i^k}{S_i^k + T_i^k}, & \text{if } S_i^k \geq 0 \\ \frac{d_{min}}{b} \frac{|S_i^k|}{|S_i^k| + T_i^k}, & \text{if } S_i^k < 0 \end{cases} \quad (10)$$

where d_{min} is the Euclidean distance of the rule centre k to its nearest centre (or to the boundary of the input space), b is the active radius which is the maximum variation distance and is used to guarantee that the clusters don't substitute their relative positions in the input space (typical value is $b=4$, meaning that, at most, a centre can be moved as far as half the midpoint between it and its neighbour), and T_i^k is an inverse-temperature parameter that will endow our methodology with a "Simulated Annealing" behavior, in the sense that it will begin with a small value (allowing large rule centres movements) and as the sign of S_i^k changes it will get larger (thus allowing smaller rule centres movements).

In this iterative process we have introduced a migration step, in order to place rule centres for which (for any i) the total distortion

$$D^k = S_{i+}^k + S_{i-}^k \quad (11)$$

is lower than the distortion mean value, near to those centres for which the distortion is higher than the mean. This migration process will assure that we will not leave any region of the input space insufficiently covered and will speed up the convergence.

The process finalizes when we obtain the configuration of centres that achieves an equidistant distribution of errors, i.e., the sum of slopes

$$E = \sum_{\substack{k=1 \dots K \\ i=1 \dots n}} |S_i^k| \quad (12)$$

does not decrease with respect to the previous iterations. This procedure requires few iterations to obtain a clustering-based fuzzy system which, although itself does not represent a good approximation, it is assumed to be a good starting point to search for an approximation close to the optimum of the target function, using local-minimum search techniques. $\therefore$ 1 shows the flow diagram of the whole EEM process.

Migration Process. As we have mentioned, in the iterative process we will perform a migration step, similar to that presented for the ELBG algorithm [7] or for the CFA algorithm. Using the distortion D^k we will move those centres with a lower relative distortion ($D^k - \text{mean}(D^k)$) near to those with a higher relative distortion. The selection of which centre we will move near to which one will be based on a probability proportional to the relative distortion.

When a rule centre v is to be moved near to another one u , the n -dimensional box compounded by that input space region for which the rule centre u has a higher influence (see $\S$ 5) is considered. Then this input region is subdivided by considering one of the n -dimensional diagonals of this n -dimensional box and placing one of the centres v in the 1/4 of it and u in the 3/4 of it. If the sum of

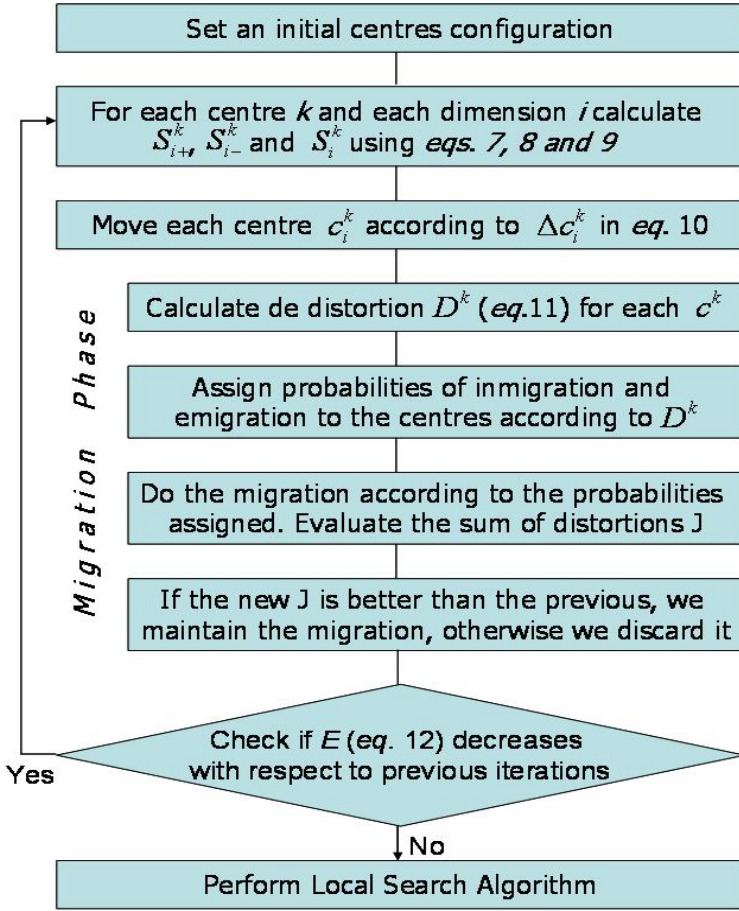


Fig. 1. Error Equidistribution Method for the initialization of rule centres in CBFSs

slopes (function . 12) according to this new distribution of the rule centres is better than the previous distribution, then this migration is confirmed, otherwise it is discarded.

2.2 Local Search Procedure

Finally, once we have a good starting rule centres configuration we can launch any local search procedure to find a local optimum according to this initial configuration provided. Though in our Error Equidistribution approach we have obtained the width of the gaussian functions automatically, in this local search we can try to find the best rule centres but also the optimal rule widths, thus obtaining a more accurate result according to the error function . 1.

In this paper we have used the Levenberg-Marquardt algorithm [9] due to its efficiency; nevertheless any local search procedure could be executed according to the nature of the CBFS taken into account.

3 Simulations Section

In this section we will analyze the application of the Error Equidistribution method in Clustering-Based Fuzzy System to a significative example first presented in [1]. The function f_1 is expressed as

$$f_1 = \frac{\sin(2\pi x)}{\exp(x)} \tag{13}$$

In the simulations performed in this section, 1000 randomly distributed data points are extracted from this expression. The form of the function is shown in Fig. 2.

3.1 Detailed Execution of the Error Equidistribution Method

In this subsection we will show a detailed execution of the error equidistribution method using the example function f_1 . Given a fixed number of rules, for example 5, and using a k-means [10] algorithm to get the starting rule centres.

Since the training datapoints are randomly distributed in the input domain, the MF centres obtained by the k-means algorithm are equally distributed along that domain. Fig. 3 shows the output obtained using the rule centres obtained using the k-means, and the distribution of the error obtained in the input domain (the NRMSE [5] obtained is 0.93).

Starting from this initial configuration (though we could use any randomly distributed centres configuration) we show now the evolution of the Error Equidistribution Method that will try to find a better starting configuration for the local search process. In the first iterations of the EEM, the centres trend to move to the left side of the figure as expected. Fig. 4 shows a migration step that takes

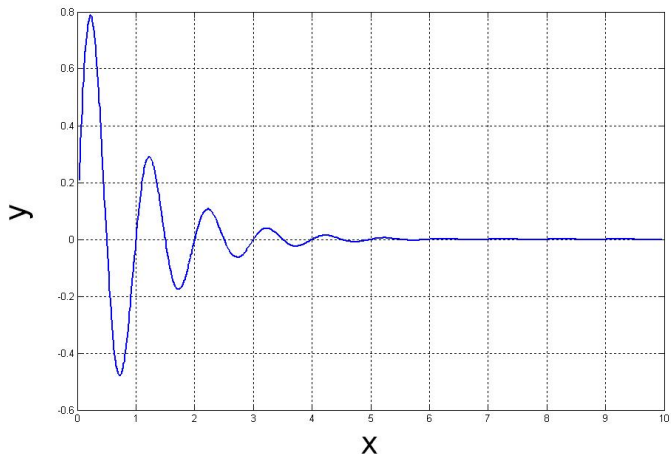


Fig. 2. Function f_1

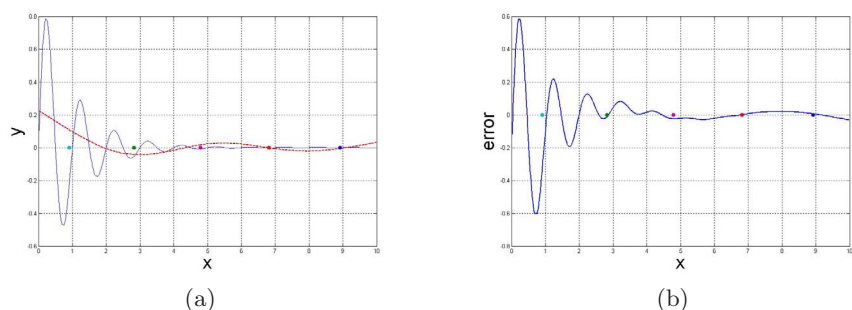


Fig. 3. a) CBFS output using the initial centres assignment using k-means b) Distribution of the error in the input domain

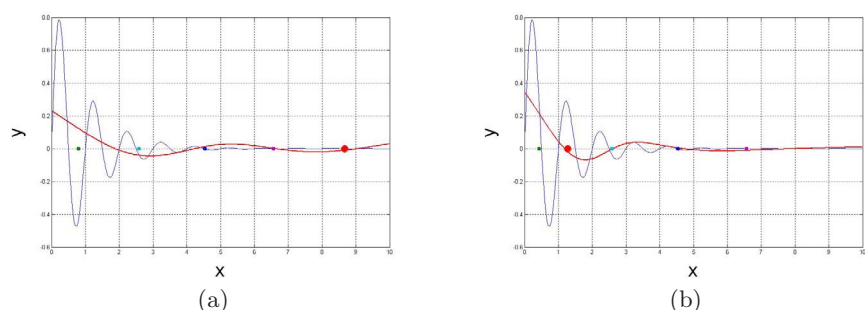


Fig. 4. Example of migration of a cluster. The thicker line indicates the model output and the thicker dot points to the migrating rule centre: a) Situation before the migration b) Situation after the migration

place after the first iteration. According to the probabilities of migration, the top left centre has more chances of being selected to be moved, and the top right centre has more chances of receiving a centre. Since the error obtained using the new centres position is better than the previous one, the migration is accepted; Fig. 4b shows the cluster distribution and the model output after the migration.

In Fig. 5 we show the final centres distribution obtained by the EEM and the distribution of the error. It can be noted from the error distribution that each centre receives approximately the same amount of error and that it is approximately similar on both sides of the rule centres.

Fig. 6 shows the evolution of the parameter E (see Eq. 12) in the iterative process. We can see how there are several peaks in the evolution, but thanks to the “simulated annealing” effect of the EEM, E trends to stabilize itself and in this case reaches almost 0 (ideal case).

Using this starting centres configuration, in Fig. 7 we show the final approximation of the function f_1 using five rules in a Clustering-Based Fuzzy System with gaussian-type MFs.

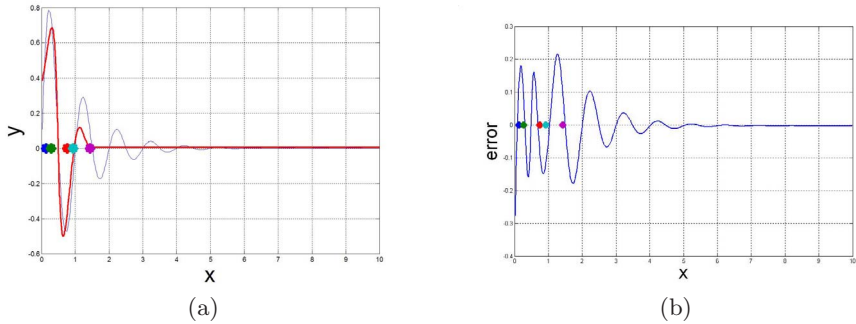


Fig. 5. a) CBFS output using the centres provided by the Error Equidistribution Method b) Distribution of the error in the input domain using the centres provided by the EEM

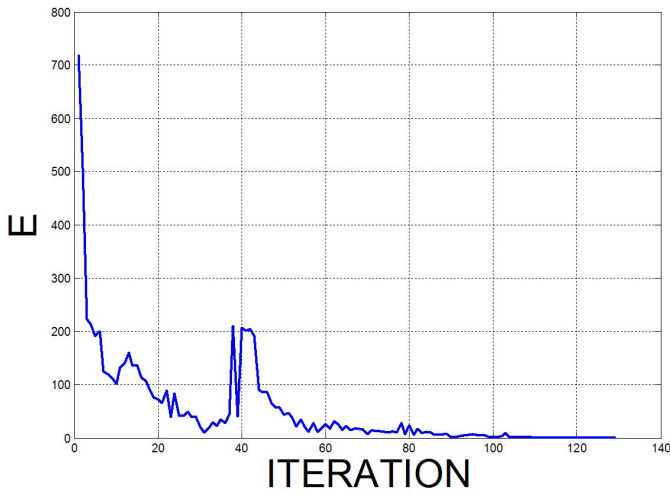


Fig. 6. Evolution of the parameter E (see Eq. 12) as the EEM algorithm evolves

3.2 Comparison with Other Approaches

Now we will show a comparison of different methodologies for the centres initialization in CBFSs. Using the same function f_1 , Fig. 1 shows the mean and standard deviation of the approximation error after the initialization procedure using a CBFS from five to ten rule centres to approximate f_1 for the k-means approach, the CFA algorithm and the EEM approach presented in this paper. From these results we can see that the EEM approach obtains the best error of all the centres initialization approaches.

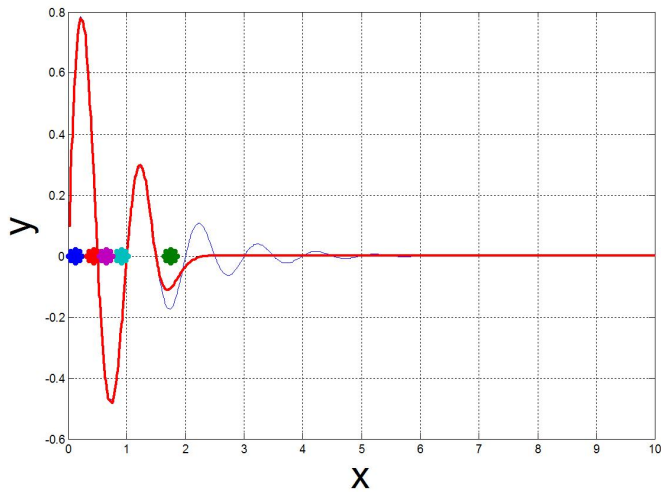


Fig. 7. Final approximation of function f_1 using 5 rules after the local search procedure taking the centres provided by the EEM algorithm

Table 1. Mean and standard deviation of the approximation NRMSE after the initialization procedure using a CBFS for different methodologies considering from five to ten rule centres to approximate

Num.Clusters	K-means	CFA	Error Equidistribution
5	0.936(0.017)	0.910(0.029)	0.585(0.041)
6	0.910(0.027)	0.826(0.035)	0.393(0.057)
7	0.899(0.031)	0.794(0.035)	0.263(0.043)
8	0.871(0.036)	0.735(0.058)	0.291(0.127)
9	0.843(0.038)	0.733(0.065)	0.188(0.035)
10	0.826(0.043)	0.662(0.174)	0.122(0.029)

After obtaining the initial models for each centres initialization approach, the LevenbergMarquardt algorithm was applied to find the optimum of rule centres and widths configurations. Table 2 shows the NRMSE mean and standard deviation for the previous simulations after applying the local minimization algorithm. The CFA algorithm produces a better performance than the k-means algorithm since it takes into account the variability of the output. In [1], the results presented showed how the CFA algorithm outperformed the previous approaches for initialization of RBF centres in RBF networks. But as we can see, the EEM methodology presented in this paper obtains the best results, outperforming the CFA algorithm and the k-means algorithm, thus showing its convenience for CBFS for function approximation since it suits better the function approximation problem formulation.

Table 2. Mean and standard deviation of the approximation NRMSE after the execution of the Levenberg-Marquardt algorithm (local search) using a CBFS for different methodologies considering from five to ten rule centres to approximate

Num.Clusters	K-means	CFA	Error Equidistribution
5	0.701(0.160)	0.307(0.046)	0.143(0.010)
6	0.538(0.207)	0.286(0.064)	0.112(0.033)
7	0.699(0.150)	0.287(0.060)	0.083(0.031)
8	0.500(0.218)	0.164(0.121)	0.072(0.021)
9	0.541(0.208)	0.112(0.124)	0.041(0.013)
10	0.310(0.118)	0.127(0.135)	0.031(0.004)

4 Conclusions

In this paper we have presented a new methodology for the initialization of the rule centres in a Clustering-Based Fuzzy System. The Error Equidistribution Method tries to place each rule centre so that the error influenced by each rule is similar at both sides at each dimension of the rule centre. The results shown in the simulation section shows that this new approach can outperform classical input and input/output clustering algorithms for the initialization of the rule centres at the initialization phase, and also after a local search process is executed. This results show the convenience of the EEM algorithm to avoid bad local optimums.

Acknowledgements

This work has been partially supported by the Spanish CICYT Project TIN2004-01419.

References

1. González, J., Rojas, I., Pomares, H., Ortega, J., Prieto, A.: A new Clustering Technique for Function Aproximation. *IEEE Transactions on Neural Networks*, **13**(1)(2002) 132-142
2. Uykan, Z., Gzelis, C., Celebei, M. E., Koivo, H. N.: Analysis of Input-Output Clustering for Determining Centers of RBFN. *IEEE Transactions on Neural Networks*, **11**(4) (2000) 851-858
3. Bezdek, J. C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, Nueva York, 1981
4. Kasabov, N., Song, Q.: DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction. *IEEE Transactions on Fuzzy Systems*, **10** (2) (2002) 144-154.
5. Pomares, H., Rojas, I., Ortega, J., Prieto, A.: A systematic approach to a self-generating fuzzy rule-table for function approximation. *IEEE Trans. Syst., Man, Cybern.* **30** (2000) 431-447

6. Babuska, R.: Fuzzy modelling for control. Kluwer Academic, 1998.
7. Russo, M., Patane, G.: Improving the LBG Algorithm. Lecture Notes in Computer Science. New York: Springer-Verlag, **1606** (1999) 621-630
8. Moody, J., Darken, C.J.: Fast learning in networks of locally-tuned processing units. *Neural Computation* **1(2)** (1989) 281-294
9. More, J.J.: The Levenberg-Marquardt algorithm implementation and theory. *Lecture Notes in Mathematics* **630** (1978) 105-116
10. Duda, R. O., Hart, P. E.: Pattern Classification and Scene Analysis. New York: Wiley (1973)

Learning Methods for Air Traffic Management

Frank Rehm¹ and Frank Klawonn²

¹ German Aerospace Center, Braunschweig, Germany
`frank.rehm@dlr.de`

² University of Applied Sciences Braunschweig/Wolfenbüttel, Germany
`f.klawonn@fh-wolfenbuettel.de`

Abstract. Weather is an important source of delay for aircraft. Recent studies have shown that certain weather factors have significant influence on air traffic. More than 50% of all delay accounts to weather and causes among others high costs to airlines and passengers. In this work we will show to what extent weather factors in the closer region of Frankfurt Airport have an impact on the delay of flights. Besides the results of a linear regression model we will also present the results of some modern data mining approaches, such as regression trees and fuzzy clustering techniques. With the clustering approach we will show that several weather conditions have a similar influence on the delay of flights. Our analyses focus on the delay that will be explicitly caused by weather factors in the vicinity of the airport, the so-called terminal management area (TMA). Thus, delay caused by weather at the departure airport or by other circumstances during the flight will not bias our results. With our methods it becomes possible to predict the delay of flights if certain weather factors are known. We will specify these factors and quantify their effects on delay.

1 Introduction

Traffic at Frankfurt Airport is increasing every year. As for car traffic, obstruction or delay in air traffic is mostly caused by high traffic volume. Apart from this, weather is an important source of delaying, particularly in the vicinity of an airport. In this paper we discuss which weather factors at Frankfurt Airport may have an effect on the travel time on approaching aircraft. Delays according to a certain schedule will not be concern of this work. To be independent for the most part from any delay that might be caused at the departure airport or during the flight (. . .), we only consider in this study the travel time that aircraft need from the entrance in the terminal management area (TMA) until landing. In average, this travel time is about 30 minutes.

Our analyses focus on a weather dataset that consists of more than ten weather factors which are captured at the airport at least half-hourly by different sensors. Another dataset that contains all flights for the same time period is used to bring it in correlation with the weather data. The objective of these analyses is the prediction of the travel time of an aircraft given a comparable weather description. With this information it is possible to manage landside procedures, such as the disposition system, taxiing, arrangement on the apron

and the aircraft's stand. Optimising these procedures could afford saving money since these resources are always scarce.

We will present in this paper the results of three statistical methods, which we apply on this data. Beside linear regression, we use two modern learning methods namely regression trees and fuzzy clustering. The clustering of the weather data shows that there are some significant weather conditions at Frankfurt Airport which have a certain effect on the travel time.

2 Related Work

There are several publications on this topic analysing only a short time period or considering only a few different weather factors.

In [4] a model is described that predicts a daily flight time index for Los Angeles Airport. This index represents the daily flight time by weighting several factors such as weather but also origin airport departure delay. The problem of weather normalisation is addressed in [3] to improve a performance measure which aims at comparing different airspace systems.

The effect of thunderstorms on delay at Frankfurt Airport is discussed in [7]. This study shows that one thunderstorm affects the air traffic in the TMA at Frankfurt Airport for three hours and causes approximately one thousand minutes delay for one hundred aircraft.

[8] investigates the effects of weather at Frankfurt Airport to compute the daily punctuality at Frankfurt Airport. These analyses base on daily indices of punctuality and local weather data of the airport. By means of multiple linear regression, more than twenty weather parameters are determined with effect on punctuality. With this model it can be estimated the punctuality of one day, given the expected weather conditions.

3 The Weather Data

This study is based on the ATIS* weather dataset of the years 1998 and 1999. This dataset consists of more than ten attributes, describing certain weather factors at Frankfurt Airport which were captured by different sensors over a timeframe of some minutes. The following table shows these weather factors.

We will call a datum that is composed of these attributes a weather report. Regularly, weather reports will be recorded in an ATIS dataset every 30 minutes. Only when the weather changes quickly, weather reports will be recorded in a shorter interval.

Apart from the precipitation attributes, these weather factors are described by means of numer-

ATIS
atmospheric pressure
temperature
visibility
cloud coverage
cloud layers
precipitation
precipitation intensity
wind speed [†]
wind direction [†]

* Automatic Terminal Information Service.

[†] From two different sensors.

ical values. For certain methods, precipitation, which is actually nominal, must be converted into numerical values, too.

4 The Flight Data

For the observed time period, we use a flight dataset, which contains the arrival times of all aircraft. Since we are mainly interested in the delay that is caused by certain weather factors in the vicinity of the airport, we consider the point in time of the aircraft's entrance in the TMA and the time when the corresponding aircraft is landing. As mentioned above, this travel time is about 30 minutes in average.

Additionally, the dataset contains the identification of the runway that was used for the respective landing. The runway configuration, the direction that is actually used for approaches and departures, is of importance when analysing the effect of certain wind components, such as headwind and tailwind.

5 Data Preprocessing

The weather dataset consists of attributes which have different ranges of values. For the linear regression method and the clustering, which we intend to use, it is essential to normalise the data. Thus, coefficients resulting from linear regression can be interpreted directly as degree of importance. Clustering is sensitive to different scaled variables, too. Without normalising the data large distances that may occur regarding wide range attributes such as visibility and atmospheric pressure would result in misleading clusters.

A first insight into the weather data we obtain by applying a principle component analysis (PCA). The first three components have eigenvalues which are greater than 1. These three hypothetical components cover more than 70% of the variation in the data.

Figure 1 shows two components of the weather data, which result from applying PCA. Obviously, there are two linearly separable clusters. The clear separation of the data into two clusters is due to two different weather conditions: cloudy weather and cloudless weather.

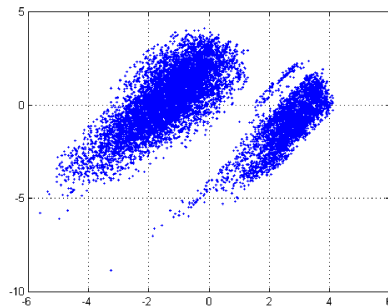


Fig. 1. Two Components of the Weather Data after PCA

Table 1. Data Preprocessing

Atmospheric Pressure	Tem- perature	Visi- bility	Precipi- tation		Atmospheric Pressure	Tem- perature	Visi- bility	Precipitation		
1015	30	30000	-	→	-0.06	2.5	0.53	0	0	0
1020	22	25000	SHRA		0.5	1.4	0.16	1	0	0
1020	8	20000	BCFG		0.5	-0.5	-0.21	0	1	0
1018	6	2000	BR		0.27	-0.77	-1.55	0	0	1

When considering the precipitation attribute, another important conversion has to be made. Some statistical methods can not deal with nominal attributes directly. Usually, a numerical representation for such variables has to be found. In most cases, values of such nominal variables will be converted into dichotomous (0/1-coded) variables. That means, that for every value that occurs in column precipitation a new variable will be created, that is either 1 if this kind of precipitation has been recorded or 0 otherwise. Thus, normalising and converting variables for a small extract of the data (see left part of table 1) leads to the right part of table 1.

From the flight data two additional variables can be extracted. It is obvious, that the travel time depends to a high degree on the capacity of the airport. Thus, shortages mainly result from the demand, that is a factor determined by means of the amount of aircraft that wish to land.

Another essential attribute is the travel time. We intend to assign a specific travel time value to each weather report. Since weather reports occur usually half-hourly and arrivals may occur with inter-arrival times of a few minutes, it is necessary to determine a specific value for the travel time. To obtain these values for a certain weather report, we propose to consider all approaching flights, that have entered the TMA already and those flights that have been landed after that point in time where the preceding weather report has been recorded. Thus, we count the number of flights according to the above definition and call this attribute $\dots f_{fi}$. Further, we examine all travel times of those flights and choose the median travel time. In some cases, the mean estimator would be not appropriate, particularly when some normal flights will be pooled with one flight that has an extreme long travel time.

For some data mining methods, such as for clustering and regression trees, it is advisable to eliminate the dependence of the travel time on the traffic. Since traffic is very important to predict the travel time, regression trees would mostly use the traffic attribute for the prediction and weather factors would be rarely represented in the tree. Further on, clusters of weather conditions would not be meaningful when comparing them with average travel times.

For each flight in the flight dataset, the corresponding landing runway can be read. Because the wind sensors measure wind speed and wind direction one can compute the related headwind components and crosswind components. This might be of interest, since runway configuration changes, when tailwind exceeds five knots.

6 Statistical Analyses

In this section we describe the results of three statistical methods. For a more detailed description of the principles of the algorithms we refer to the literature.

6.1 Multiple Linear Regression

Linear regression aims at estimating the conditional expected value of one variable y given the values of some other variable or variables x . The variable of interest, y , is called the *dependent variable*. The other variables x are called the *independent variables*. A multiple linear regression model is typically stated in the form

$$y = a_0 + ax_1 + a_2x_2 + \dots + a_px_p.$$

Usually, the parameters $a_1, a_2, \dots$ will be estimated by the method of least squares. In our context, the dependent variable is the travel time. The independent variables are the weather factors and the traffic. Including the new variables when converting the precipitation attribute, the coefficients for more than 40 independent variables have to be determined.

Usually, one starts applying linear regression using all given variables and eliminating variables stepwise that do not contribute significantly to the prediction. From air traffic experts we know that some variables might have a non-linear, for instance logarithmic or quadratic influence. Therefore, we also extended the regression function correspondingly.

Table 2 shows the final results applying this procedure. Since most of the variables are normalised, the values of the estimated parameters can be directly interpreted as degree of impact. For the precipitation variables the 0/1-coded values are used, traffic is used as described in section 5 and for visibility we recommend to use the logarithm of visibility. With these variables we obtain a model with a coefficient of determination $R^2 = 0.63$.

Obviously, a certain kind of precipitation affects the travel time significantly. Thus, snow prolongs the travel time by some minutes. Snowfall decreases visibility and occurs often in conjunction with iced runways. In such cases, runways but also aircraft must be cleared and deiced. This may cause delay if traffic volume is high, because these procedures take some time and must be repeated as the case may be.

Mist near the ground might be problematic, since the airport is often still visible from a larger height, but when approaching visibility decreases dramatically. Also fog, thunderstorms and rain affect the air traffic in terms of visibility. When visibility decreases, then separation of aircraft must be enlarged. Rain leads to wet runways and causes therewith a risk of skidding and elongated braking distances. Thunderstorms imply very high risks for air traffic, since they are associated with a number of weather phenomena. Generally, their avoidance leads to significant impairment.

Increasing values for temperature, cloud layer₁ and visibility favour short travel times. Increased temperatures and high cloud layers stand often for good

Table 2. Results for Linear Regression

Variable	Parameter Estimate	P
traffic	0.05	≤ 0.0001
traffic ²	-1.3	≤ 0.0001
traffic ³	22.4	≤ 0.0001
log(visibility)	-22.3	≤ 0.0001
temperature	-25.5	≤ 0.0001
cloud coverage ₂	-34.1	≤ 0.0001
cloud layer ₁	-41.3	≤ 0.0001
headwind	12.8	0.0029
south wind	9.1	≤ 0.0001
wind speed ₁	41.9	≤ 0.0001
wind speed ₂	12.5	≤ 0.0001
fog	385.9	≤ 0.0001
fog patches	86.9	≤ 0.0001
mist	48.4	≤ 0.0001
rain	26.3	≤ 0.0001
rain, mist	111.1	≤ 0.0001
snow grains	658.5	≤ 0.0001
snow grains, mist	287.1	≤ 0.0001
snow, mist	-183.7	0.0095
snow shower	286.8	≤ 0.0001
thunderstorm, rain	422.6	≤ 0.0001

weather conditions at all. Wind speed is an important source of delay at Frankfurt Airport. The high probabilities for the two measured wind speed values and the computed wind components ascertain that these variables contribute significantly to the regression model. Depending on the wind speed and the wind direction the runway configuration changes. Since high wind speed correlates mostly with other bad weather conditions separation will be enlarged and therefore delay increases.

6.2 Regression Trees

Regression tree learning is a method for approximating continuous-valued target functions, in which the learned function is represented by a tree. Learned trees can also be represented as sets of $\text{if} \dots \text{then} \dots$ rules. Such rules can be easily interpreted by humans.

With the well known CART (Classification and Regression Tree) [2] algorithm we aim at inducing such trees, with the objective to determine important weather factors and significant values for these weather factors, which enable us to predict the travel time.

Figure 2 shows a simple tree that consists of fifteen nodes and eight leaves. Each node represents a decision based on a certain weather factor and an associated value. The leaves of the tree stand for the predicted travel time that will be measured in average when applying the respective rule. CART determines

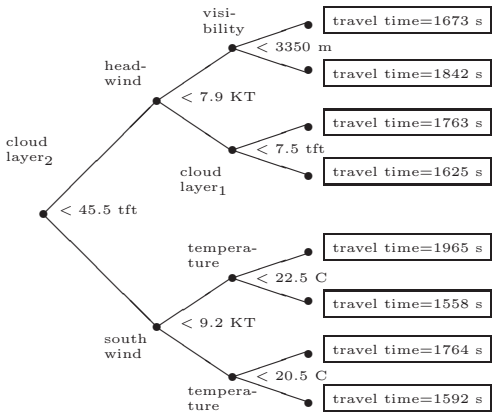


Fig. 2. A simple Regression Tree for Travel Time Prediction

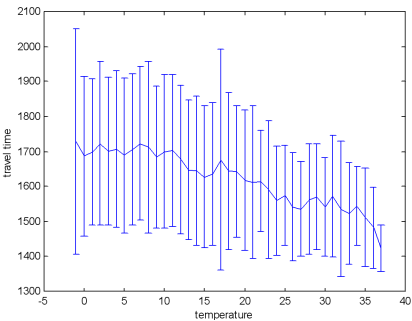


Fig. 3. Temperature affects the Travel Time

the optimal split in a greedy way considering every single attribute and dividing the dataset at this point that minimises the variance for the child nodes and maximises the difference between the sum of quadratic deviations in the parent node and those in the child nodes.

When inducing different regression trees with different datasets by means of cross validation one can observe that these trees have not always the same structure. But in general, attributes that are represented in top levels of one tree will be also represented in the top level of most other trees. We consider these attributes as important ones, which can be found very often in the tree and whose values are quite similar.

The following table shows some results. Analysing a set of regression trees, for some weather factors, such as visibility and certain wind components, a specific value can be obtained. Thus, wind speed about a value of 7-10 kt in average, which includes very high peaks of about 35 kt, leads to long travel times. Also low visibility complicates the air traffic. Although, temperature is represented very often in regression trees, no specific value can be given. This is due to a main disadvantage of regression trees, which can only produce orthogonal data splits. A diagonal curve progression as shown in figure 3 will be stepwise approximated by several nodes with different values.

frequently represented attributes	rarely represented attributes
head wind (7-10 kt)	cross wind
south wind (9 kt)	cloud coverage ₁
visibility (2000-3500m)	cloud coverage ₂
temperature	atmospheric pressure
cloud layer ₁	
cloud layer ₂	

Nevertheless, some attributes that the experts suspect to be important, such as crosswind and cloud coverage, occur quite rarely in the regression trees. Obviously, the other wind components are mostly higher and therefore represented in the top levels of the trees. As well, cloud coverage, that affects actually visibility, is according to our results less important to predict the travel time than the height of cloud layers or visibility.

Finally, we produce regression trees of different complexity to acquire the appropriate size for these trees. Generally, trees becoming more and more complex can predict the travel time for the training dataset better and better. With hundreds of branches, a training dataset can be adapted nearly perfectly. Since outliers and natural deviations may occur in such datasets, those complex trees cannot predict the travel time very well for new data. Quite the contrary, rather small trees are capable to predict well even for new data.

As a result we can make good predictions of similar quality as with linear regression, with regression trees of depth eight. Bigger trees tend to overfit the training data and yield poor results on test data.

6.3 Clustering

Even though, it is valuable to know which single weather factors affect the air traffic, due the complex interaction of different weather factors it is essential to inspect real weather conditions as a whole and their effect on the air traffic.

With clustering it becomes possible to partition the weather data into groups, the so called clusters. Such a weather cluster might describe a certain weather condition by means of a prototype, that is the centre of the respective cluster. As it can be clearly seen in figure 1 the dataset contains two separable clusters. These two clusters mainly describe the difference between weather reports referring to cloudy weather and those referring to cloudless weather. Beside this, other weather conditions seem not to be separated linearly that clearly. If these clusters differ also relating to the travel time, then we can predict this value for future flights, too.

Nevertheless, even if weather is subject of smooth transitions, one may distinguish some weather conditions anyway. We apply the fuzzy clustering algorithm *Fuzzy C-Means* (FCM), which allows us – as our tests have shown – to find stable weather clusters. As always when clustering with a prototype-based clustering technique, the number of prototypes is of concern. Common validity measures [6, 9, 1, 5] give no definite answer toward the question, how many prototypes should be used for the clustering on this dataset. Therefore, it is recommended to experiment with different numbers of clusters in order to minimise the prediction error as well as maintain the interpretability of the clusters in terms of weather situations.

Table 3 shows the most interesting prototypes which result from partitioning both the cloudy weather reports and the cloudless weather reports separately into eight clusters. The prototypes appear in ascending order regarding the average travel time for flights that correspond to the described weather conditions.

Table 3. Prototypes of FCM Clustering

clus- ter	tempera- ture	cloud coverage ₁	cloud coverage ₂	atmospheric pressure	visi- bility	cloud layer ₁	cloud layer ₂	head- wind	cross- wind	south- wind
1	22.9	-	-	1017.4	30166.4	-	-	1.6	2.0	-0.1
2	14	-	-	1013.5	29109.9	-	-	10.2	7.9	10.7
3	16.5	2.1	3.5	1014.8	29127.9	3.32	6.91	4.9	3.4	1.7
4	13.4	2.2	3.7	1014.1	25703.5	4.06	24.57	4.9	3.3	2.5
5	6.7	2.1	3.6	1022.0	21926.3	2.02	3.63	5.8	6.7	-6.2
6	4.7	2.1	3.8	1021.1	9705.5	1.24	3.15	3.5	3.0	2.3
7	12.2	2.1	3.3	1009.1	35328.9	2.63	5.1	12.9	4.4	6.7
8	10.6	2.1	3.7	1008.2	20356.5	1.76	3.6	9.2	8.8	11.2

Cluster 1 and 2 describe weather conditions of cloudless weather. The estimated travel time for flights corresponding to the weather conditions of cluster 1 is 1566s in average with a standard deviation about 157s. Therefore, this cluster represents the best weather conditions regarding the travel time. Obviously, the reason for this are increased temperatures, good visibility and almost no wind. The second cluster represents weather conditions which lead to the longest travel times when considering only cloudless weather. Although most weather factors indicate quite good flight conditions, the strong wind leads to these long travel times (1687s/254s)*.

Cluster 3 (1606s/183s) and cluster 4 (1639s/187s) represent good weather conditions when cloudy weather was recorded. Both clusters indicate increased temperatures, good horizontal and vertical visibility and moderate wind speed. Increased temperatures mostly represent physical conditions, which attend higher approach speed and consequently shorter travel time. The connection between temperature and the travel time, that can also be observed in figure 3, is revealed in cluster 5 (1675s/235s) and cluster 6 (1676s/257s), too. Additionally, cluster 6 is featured by lowest visibility be it horizontal or vertical. The longest travel times due to the weather conditions are represented by cluster 7 (1715s/252) and cluster 8 (1737s/231s). Wind has the most important effect on the travel time in these two clusters. As other studies have shown, reduced approach speed (i.e. due to headwind) affect the capacity of the airport. Particularly when approaching, due to strong wind flight paths are deformed, which forces pilots to enlarge separation to preceding aircraft. If in addition to this visibility is reduced due to a low cloud layer and a visual approach is not possible, this might lead to a considerable enlargement of separation and to significant delay consequently. Cluster 7 mainly contains weather reports where rain and also snow was recorded, which additionally affects visibility and thus the air traffic.

7 Conclusions

In this paper we have discussed some learning methods to predict the travel time of approaching aircraft at Frankfurt Airport with the emphasis to gain knowledge

* Notation: (average travel time/standard error).

about the influence of the weather conditions. By means of linear regression we have investigated which weather factors affect the air traffic to which extent. With regression tree induction, we have applied a method whose results are easy to interpret. Such trees give hints toward the question, what the critical values are for certain variables to predict the travel time. Clustering enables us to examine weather conditions by partitioning the weather dataset into weather clusters. As the results indicate, weather clusters have characteristic travel times. The results of the different learning methods have one thing in common, there is a considerable variance, which is not explained yet by these models. Hence, future work will be to improve accuracy of the prediction. To achieve this, we plan to estimate missing values so that we can use such weather reports that have been left out so far. Furthermore, an additional treatment of outliers could lower the prediction error.

References

1. 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.
2. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: *Classification and Regression Trees*, Chapman & Hall, New York, 1984.
3. Callaham, M.B., DeArmon, J.S., Cooper, A., Goodfriend, J.H., Moch-Mooney, D., Solomos, G.H.: *Assessing NAS Performance: Normalizing for the Effects of Weather*. 4th USA/Europe Air Traffic Management R&D Symposium, Santa Fe, 2001.
4. Hanson, M., Bolic, T.: *Delay and Flight Time Normalization Procedures for Major Airports: LAX Case Study*. Research Report UCB-ITS-RR-2001-5, University of California at Berkeley, 2001.
5. Höppner, F., Klawonn, F., Kruse, R., Runkler, T.: *Fuzzy Cluster Analysis*, John Wiley & Sons, Chichester, 1999.
6. Rubens, M.: Fuzzy clustering algorithms and their cluster validity. *European Journal of Operational Research*, 10, pp. 294–301, 1992.
7. Sasse, M., Hauf, T.: A study of thunderstorm-induced delays at Frankfurt Airport, Germany. *Meteorological Applications*, 10, pp. 21–30, 2003.
8. Spehr, U.: *Analyse des Wettereinflusses auf die Pünktlichkeit im Flugverkehr*. Diplomarbeit, Universität Hannover, 2003.
9. Windham, M.P.: Cluster validity for fuzzy clustering algorithms. *Fuzzy Sets and Systems*, 5, pp. 177–185, 1981.

Molecular Fragment Mining for Drug Discovery

Christian Borgelt¹, Michael R. Berthold², and David E. Patterson³

¹ School of Computer Science, Otto-von-Guericke-University of Magdeburg,
Universitätsplatz 2, 39106 Magdeburg, Germany
borgelt@iws.cs.uni-magdeburg.de

² Department of Computer Science, University of Konstanz,
Box M712, 78457 Konstanz, Germany
berthold@inf.uni-konstanz.de

³ Tripos Inc., 1699 South Hanley Road, St Louis, MO 63144, USA
patterson@tripos.com

Abstract. The main task of drug discovery is to find novel bioactive molecules, i.e., chemical compounds that, for example, protect human cells against a virus. One way to support solving this task is to analyze a database of known and tested molecules in order to find structural properties of molecules that determine whether a molecule will be active or inactive, so that future chemical tests can be focused on the most promising candidates. A promising approach to this task was presented in [2]: an algorithm for finding molecular fragments that discriminate between active and inactive molecules. In this paper we review this approach as well as two extensions: a special treatment of rings and a method to find fragments with wildcards based on chemical expert knowledge.

1 Introduction

The computer-aided analysis of molecular databases plays an increasingly important role in drug discovery. One of its major goals is to find structural properties of molecules that determine whether a molecule is active or inactive, for example, w.r.t. the protection of human cells against a virus. Knowledge about such properties can then be used to focus the expensive and time-consuming real chemical tests on the most promising candidate compounds.

The approach to this task we are considering here is based on finding a set of *discriminative fragments*. Such fragments are parts of molecules that are frequent in the set of active molecules and rare in the set of inactive ones and thus discriminate between the two classes. The rationale is that discriminative fragments may be the key substructures that determine the activity of compounds and therefore can be used to predict the activity of molecules.

Several algorithms have been suggested for finding discriminative fragments efficiently. [4] presented an approach that finds linear fragments using a method that is based on the well-known Apriori algorithm for association rule mining [1]. However, the restriction to chains of atoms is limiting in many real-world applications, since substructures of interest often contain rings or branching points.

The approach presented in [2] does not suffer from this restriction. It is based on a transfer of the Eclat algorithm for association rule mining [8] and thus avoids the expensive reembedding of fragments. Another approach that can find arbitrary fragments was proposed in [5]. The main differences between these approaches are the ways in which fragments of interest are generated, an issue we consider in more detail when we study the algorithm of [2] below.

In this paper we also review two extensions of the algorithm suggested in [2]. Due to a symmetry problem that makes it impossible to suppress all redundant search, this algorithm runs into problems if fragments contain a lot of rings. The first extension solves this problem by finding rings in a preprocessing step and then treating them as units in the mining algorithm, thus making the search much more efficient [3]. The second extension is inspired by the fact that certain atom types can sometimes be considered as equivalent from a chemical point of view. To handle such cases, we introduce wildcard atoms into the search [3].

As a test case for our extended algorithm we use a publicly available database of the National Cancer Institute, namely the NCI-HIV database (DTP AIDS Antiviral Screen) [7], which was also used in [2] and [4]. This database is the result of an effort to discover new compounds capable of inhibiting the HI virus (HIV). The screen utilized a soluble formazan assay to measure protection of human CEM cells from HIV-1. Compounds able to provide at least 50% protection to CEM cells were retested. Compounds providing at least 50% protection on retest were listed as moderately active (*CM*). Compounds that reproducibly provided 100% protection were listed as confirmed active (*CA*). Compounds not meeting these criteria were listed as confirmed inactive (*CI*). The dataset consists of 41,316 compounds, of which we have used 37,171. For the remaining 4,145 no structural information was available at the time the experiments reported here were carried out. Each compound has a unique identification number (NSC number), by which we refer to it. Out of the 37,171 compounds we used, 325 belong to class *CA*, 896 to class *CM*, and the remaining 35,950 to class *CI*. We neglected the 896 moderately active compounds in our experiments. The size of this database stresses the need for a truly efficient and scalable algorithm. The method described in [2], for example, could only be applied to this data set by using small initial fragments as “seeds” for the underlying search algorithm.

For some other experiments we used the NCI-H23 database (DTP Human Tumor Cell Line Screen) [9] of the National Cancer Institute’s Development Therapeutics Program, which has a similar purpose as the NCI-HIV database and which is also publicly available. This database lists 35,985 compounds, which have been tested for evidence of the ability to inhibit the growth of human tumor cell lines. The compounds were tested at five different concentrations and w.r.t. sixty different human tumor cell lines. We used the results for a specific type of lung cancer. In these results 5,996 compounds are classified as active based on the concentration parameter GI50 being greater than 5 (see [9] for more details). With our algorithm we tried to find discriminative fragments that separate these active compounds from the rest.

2 Molecular Fragment Mining

As stated above, the goal of molecular fragment mining is to find discriminative fragments in a database of molecules, which are classified as either active or inactive. To achieve this goal, the algorithm presented in [2] represents molecules as attributed graphs and carries out a depth first search on a tree of fragments. Going down one level in this search tree means extending a fragment by adding a bond and maybe an atom to it. For each fragment a list of embeddings into the available molecules is maintained, from which the lists of embeddings of its extensions can easily be constructed. As a consequence, expensive reembeddings of fragments are not necessary. The support of a fragment, that is, the number of molecules it is contained in, is then determined by simply counting the number of different molecules the embeddings refer to. (Note that there may be more than one embedding of a fragment into the same molecule.) If the support of a fragment is high in the set of active molecules and low in the set of inactive molecules it is reported as a discriminative fragment.

The most important ingredients of the algorithm are different search tree pruning methods, which can be grouped into three categories: *size based pruning* (which restricts the size of fragments), *support based pruning* (which discards too infrequent fragments), and *structural pruning*. Among these the latter is the most important and most complicated. It is based on a set of rules that defines a local order of the extensions of a fragment, which is used to avoid redundant searches, that is, multiple construction and test of the same fragment.

The main idea is that two extensions, if both are applied (one after the other) to the same fragment, lead to the same extended fragment, regardless of which of them is applied first. Thus, without restrictions, the same fragments would be considered in several branches of the search tree, because there are several ways of building the same fragment by adding bonds and atoms to a core fragment (which may be a single atom). To avoid this redundancy, the extensions are ordered, and once an extension is carried out, any extension preceding it in this order is prohibited. The order is defined locally, because it depends on the structure of the fragment to be extended. Ideally, ordering the extensions should lead to a situation in which there is exactly one way of constructing each possible fragment, so that there is a unique path in the search tree leading to it.

To be more specific, the algorithm works as follows: The atoms of the fragments that are constructed are numbered in the order in which they have been added to the fragment. Whenever an embedding is extended, the number of the atom, from which the added bond started, is recorded in the resulting extension. When the extended embedding is to be extended itself, only bonds are considered that start from atoms having numbers no less than this recorded number. That is, only the atom extended in the preceding step and atoms added later than this atom can be the starting point of a new bond. With this simple scheme it is immediately avoided that two bonds, call them A and B , which start from different atoms, are added in the order A, B in one branch of the search tree and in the order B, A in another. Since either the atom A starts from or the atom B starts from must have a smaller number, one of the orders is ruled out.

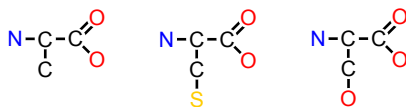


Fig. 1. The amino acids glycine, cysteine and serine

However, two or more bonds may start from the same atom. Therefore it is also necessary to define an order on bonds, to avoid that two different bonds A and B that start from the same atom are added in the order A, B in one branch of the search tree and in the order B, A in another. This order on bonds is, of course, arbitrary. Most naturally, however, single bonds precede aromatic bonds, which precede double bonds, which precede triple bonds. Finally, within extensions by bonds of the same type starting from the same atom, the order is determined by (1) whether the atom the bond leads to is already in the fragment or not and (2) the type of this atom. To take care of the bond type etc., it is recorded in each embedding which bond was added last.

Ideally, extensions would be totally ordered by these rules. However, this is not the case. There are situations in which two bonds of the same type start from the same atom and lead to atoms of the same type. In such cases the locally available information is not sufficient to define an order of the extensions. However, it is also not possible to simply process these extensions in an arbitrary order, as an example in [2] demonstrates. As a consequence, they have to be considered in every possible order, which can lead to considerable redundant search. This is the main reason for the first extension of this algorithm (see below).

Since the search tree we traverse in the modified algorithm differs slightly from the one used in [2], we illustrate it with a simple example. Figure 1 shows the amino acids glycine, cysteine and serine with hydrogens and charges neglected. The upper part of the tree (or forest if the empty fragment at the root is removed) that is traversed by the algorithm for these molecules is shown in Figure 2. The first level contains individual atoms, the second connected pairs and so on. The dots indicate subtrees that are not depicted in order to keep the figure understandable. The numbers next to these dots state the number of fragments in these subtrees, giving an indication of the total size of the tree.

The order, in which the atoms on the first level of the tree are processed, is determined by their frequency of occurrence in the molecules. The least frequent atom type is considered first. Therefore the algorithm starts on the left by embedding a sulfur atom into the example molecules. That is, the molecules are searched for sulfur atoms and their locations are recorded. In our example there is only one sulfur atom in cysteine, which leads to one embedding of this (one atom) fragment. This fragment is then extended (depth first search) by a single bond and a carbon atom ($-C$), which produces the fragment $S-C$ on the next level. All other extensions of fragments that are generated by going down one level in the tree are created in an analogous way.

If a fragment allows for more than one extension (as is the case, for instance, for the fragments $O-C$ and $S-C-C-C$), we sort them according to the local ordering

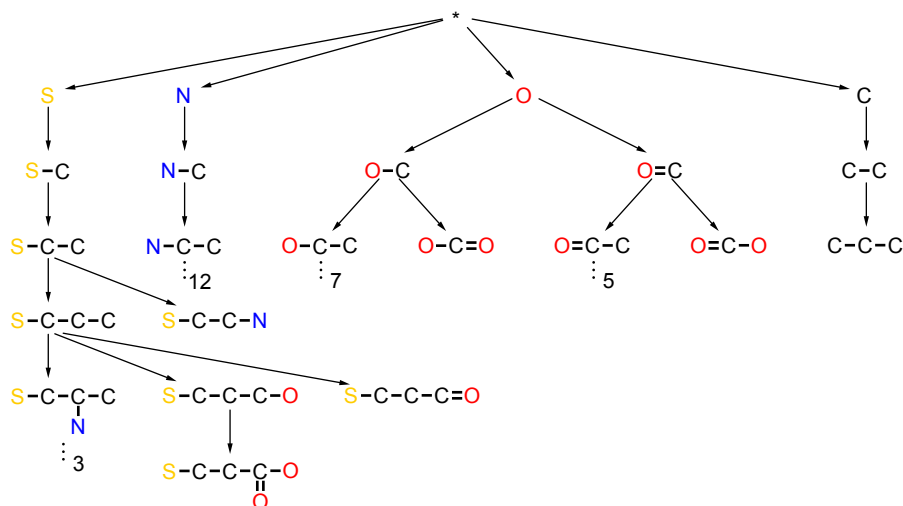


Fig. 2. The tree of fragments for the amino acids example

rules listed above. As explained a few paragraphs up, the main purpose of this local order is to prevent certain extensions to be generated in certain branches of the search tree, in order to avoid redundant searches. For instance, the fragment S-C-C-C-O is not extended by adding a single bond to a nitrogen atom at the second carbon atom, because this extension has already been considered in the subtree rooted at the left sibling of this fragment.

Furthermore, in the subtree rooted at the nitrogen atom, extensions by a bond to a sulfur atom are ruled out, since all fragments containing a sulfur atom have already been considered in the tree rooted at the sulfur atom. Similarly, neither sulfur nor nitrogen are considered in the tree rooted at the oxygen atom, and the rightmost tree contains fragments that consist of carbon atoms only.

Although the example molecules considered here do not contain any rings, it is clear that we have to handle them in the general case. While the original algorithm treated extensions to new atoms and extensions closing rings basically alike, only handling them through an ordering rule to avoid redundant search (see above), the new algorithm processes them in two separate phases: Even though extensions closing rings are allowed in any step, once a ring has been closed in a fragment, the possible future extensions are restricted to closing rings as well. While this does not lead to performance improvements, it has certain advantages w.r.t. the order in which fragments are generated.

Up to now we described how the search tree is organized, i.e., the manner in which the candidates for discriminative fragments are generated and the order in which they are considered. However, in an application this search tree is not traversed completely—that would be much too expensive for a real-world database. Since a discriminative fragment must be frequent in the active molecules and extending a fragment can only reduce the support (because only fewer molecules

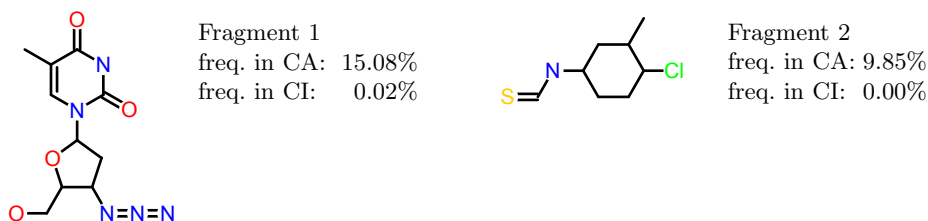


Fig. 3. Discriminative fragments

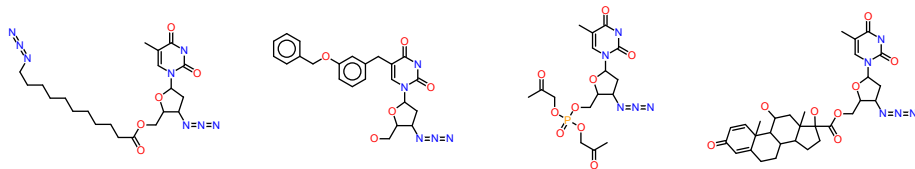


Fig. 4. Example molecules containing Fragment 1

can contain it), subtrees can be pruned as soon as the support falls below a user-defined threshold (support based pruning). Furthermore, the depth of the tree may be restricted, thus limiting the size of the fragments (size based pruning).

Discriminative fragments should also be rare in the inactive molecules, defined formally by an user-specified upper support threshold. However, this upper threshold cannot be used to prune the search tree: Even if a fragment does not satisfy this threshold, its extension may (again extending a fragment can only reduce the support), and thus it has to be generated. Therefore this threshold is only used to filter the fragments that are frequent in the active molecules. Only those fragments that satisfy this threshold (in addition to the minimum support threshold in the active molecules) are reported as discriminative fragments.

In [2] the original form of this algorithm was applied to the NCI-HIV database with considerable success. Several discriminative fragments were found, some of which could be related to known classes of HIV inhibitors. Among the results are, for example, the two fragments shown in Figure 3. Fragment 1 is known as AZT, a well-known nucleoside reverse transcriptase inhibitor, which was the first drug approved for anti-HIV treatment. AZT is covered by 49 of the 325 (15.08%) compounds in *CA* and by only 8 of the 35,950 (0.02%) compounds in *CI*. Four examples of compounds in the NCI-HIV dataset that are covered by Fragment 1 are shown in Figure 4. Fragment 2 is remarkable, because it is not covered by any compound of the 35,950 inactive ones, but is covered by 31 molecules in *CA* (9.85%). (Fragment 2 was not reported in [2], but in [3]).

3 Rings

An unpleasant problem of the search algorithm as we presented it up to now is that the local ordering rules for suppressing redundant searches are not complete.

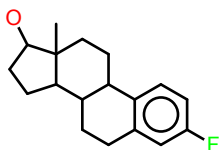


Fig. 5. Example of a steroid, which can lead to problems due to the large number of rings

Two extensions by identical bonds leading to identical atoms cannot be ordered based on locally available information alone. As mentioned, a simple example demonstrating this problem and the reasons underlying it can be found in [2]. As a consequence, a certain amount of redundant search has to be accepted in this case, because otherwise incorrect support values would be computed and it could no longer be guaranteed that all discriminative fragments are found.

Unfortunately, the situation leading to these redundant searches occurs almost always in molecules with rings, whenever the fragment extension process enters a ring. Consequently, molecules with a lot of rings, like the steroid shown in Figure 5—despite all clever search tree pruning—still lead to a prohibitively high search complexity: The original algorithm considers more than 300,000 fragments for the steroid, a large part of which are redundant.

To cope with this problem, a preprocessing step is added to the algorithm, in which rings of the molecules are found and marked. This is done by a simple depth first search that is restricted in depth to the maximum size of rings that are to be found. Usually it suffices to find rings with five and six atoms, because these are most interesting to chemists. In the search process itself, whenever an extension adds a ring bond to a fragment, not only this bond, but the whole ring is added (all bonds and atoms of the ring are added in one step). As a consequence, the depth of the search tree is reduced (from five or six levels to construct a ring to one) and much of redundant search of the basic algorithm is avoided, which leads to enormous speed-ups. For example, the number of fragments generated for the steroid shown in Figure 5 reduces to 93.

On the NCI-HIV database our implementation of the basic algorithm generates 407,364 fragments in *CA* that have a support higher than 15%, for which it needs 45 minutes.¹ Using ring extensions reduces the number of fragments to 456 which are generated in 13 seconds. Another illustration of the gains resulting from one step ring extensions is provided by the diagrams shown in Figure 6. They display the number of nodes per level in the search tree for the two versions of the algorithm. As can be seen in the left diagram, the search tree of the basic algorithm has a width of approximately 80,000 fragments and a depth of 22 levels. In contrast to this, the width of the search tree using one step ring extensions is approximately 80 fragments, almost a factor of 1000 less.

¹ Our experiments were run using a Java implementation on a 1GHz Xeon Dual-Processor machine with 1GB of main memory using jre1.4.0.

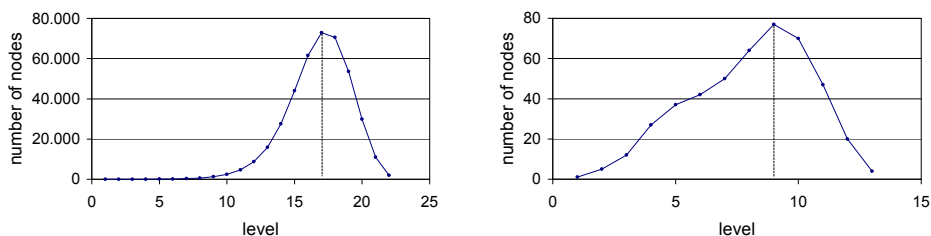


Fig. 6. Number of nodes per level for the algorithms with and without ring extensions

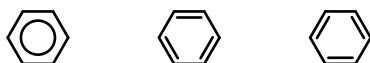


Fig. 7. Aromatic and Kekulé representations of benzene



Fig. 8. Frequencies in CA with and without ring extensions

Furthermore, the shapes of the search trees are remarkable. While with the basic algorithm the tree width grows roughly exponentially with the depth of the tree, until a maximum is reached, with the ring extension algorithm the tree width grows only roughly linearly with the depth. As a side note, these diagrams also justify the use of a depth first search in our algorithm: The typical and extremely high width to depth ratio of the tree makes a breadth first search disadvantageous or, for the basic algorithm, infeasible.

Besides a considerable reduction of the running time, treating rings as units when extending fragments has other advantages as well. In the first place, it makes a special treatment of atoms and bonds in rings possible, which is especially important for aromatic rings. For example, benzene can be represented in different ways as shown in Figure 7. On the left, all six bonds are represented as aromatic bonds, while the other two representations—called *Kekulé structures*—use alternating single and double bonds. A chemist, however, considers all three representations as equivalent. With the basic algorithm this is not possible, because it would require treating single, double, and aromatic bonds alike, which obviously leads to undesired side effects. By treating rings as units, however, these special types of rings can be identified in the preprocessing step and then be transformed into a standard aromatic ring.

Secondly, one step ring extensions make the support values associated with a fragment more easily interpretable. To see this, consider the two fragments shown

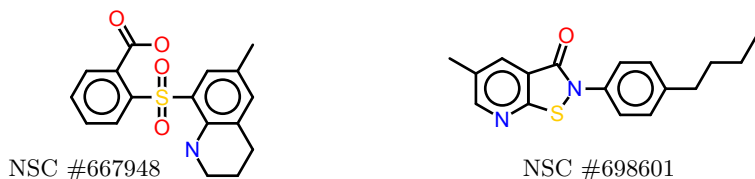


Fig. 9. Compounds that contain Fragment 3 but not Fragment 4

in Figure 8. The left was found with the basic algorithm, the right with the ring extension algorithm. At first glance it may be surprising that the two fragments look alike, but are associated with different support values (74 compounds, or 22.77%, for the basic algorithm and 65 compounds, or 20.00%, for the ring extension algorithm). However, the reason for this is very simple. While with the basic algorithm the N-C-C chain extending from the aromatic ring may be a real chain or may be a part of another ring, it must be a real chain with the ring extension algorithm.

The reason is that with the latter it is not possible to walk into a ring bond by bond, because rings are always added in one step. Therefore the ring extension algorithm does not find this fragment, for instance, in the two compounds shown in Figure 9. Although it may appear as a disadvantage at first sight, this behavior is actually highly desirable by chemists and biologists, because atoms in rings and atoms in chains are considered to be different and thus should not be matched by the same fragment. A fragment should either contain a ring or not, but not parts of it, especially if that part can be confused with a chain.

4 Fragments with Wildcards

Chemists and biologists often regard fragments as equivalent, even though they are not identical. Acceptable differences are, for instance, that an aromatic ring consists only of carbon atoms in one molecule, while in another ring one carbon atom is replaced by a nitrogen atom. An example of such a situation from the NCI-HIV database is shown in Figure 10. A chemist would say that the two molecules shown on the left have basically the same structure and that the difference consists only in the oxygen atom attached to the selenium atom.

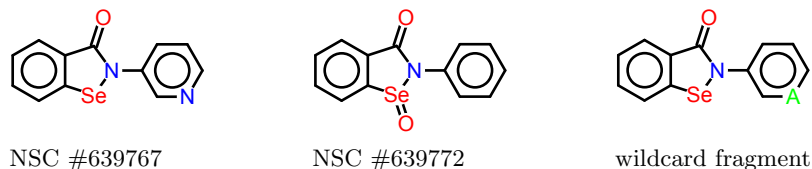


Fig. 10. Matching molecules with a fragment with a wildcard atom

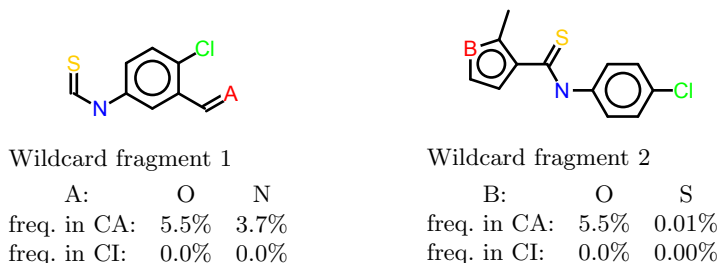


Fig. 11. Fragments with wildcards in the NCI-HIV database

Therefore they can be treated alike. However, the original algorithm will not find an appropriate fragment because of this mismatch in atom-type [2].

More generally, under certain conditions certain types of atoms can be considered as equivalent from a chemical point of view. To model this equivalence we introduce wildcard atoms into the search, which can match atoms from a user-specified range of atom types. As input we need chemical expert knowledge to define equivalence classes of atoms, possibly conditioned on whether the atom is part of a ring or not (that is, certain atom types may be considered as equivalent in a ring, but as different in a chain). In the example shown in Figure 10 we may introduce a wildcard atom into the search that can match carbon as well as nitrogen atoms in rings. This enables us to find the fragment on the right, where the green A denotes the wildcard atom.

Formally, wildcard atoms can be introduced into the search process by adding to each node a branch for each group of equivalent atoms, provided that there are at least two molecules that contain different atoms from the group at the position of the wildcard atom. In the later search process, this branch has to be treated in a special way, because it can be pruned if—due to further extensions—the molecules matching the fragment all contain the same atom for the wildcard atom. In our implementation we restrict the number of wildcard atoms in a fragment to a user-defined maximum number.

We applied the wildcard atom approach to the NCI-HIV database, restricting the number of wildcard atoms to one. With this approach the two wildcard fragments shown in Figure 11 are found (among others). The left fragment has a wildcard atom indicated by a red A on the right, which may be replaced by either an oxygen or a nitrogen atom. The (non-wildcard) fragment that results if A is replaced with an oxygen atom has a support of 5.5% in the active compounds, whereas it does not occur in the inactive ones. The fragment with A replaced with nitrogen has a support of 3.7% in the active compounds, and also does not occur in the inactive ones. Consequently, the wildcard fragment has a support of 9.2% in the active compounds. This demonstrates one of the advantages of wildcard fragments, that is, they can be found with much higher minimum support thresholds, which can cut down the output of irrelevant fragments.

However, wildcard fragments can also make it possible to find certain interesting fragments in the first place. This is demonstrated with the wildcard

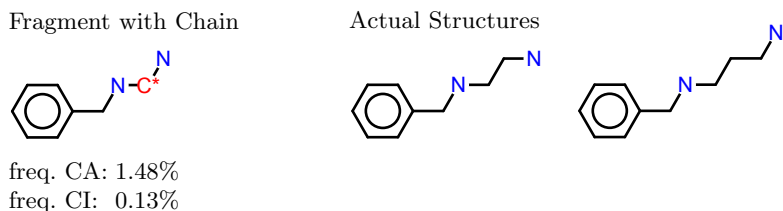


Fig. 12. Example of a carbon chain of varying length from the NCI cancer dataset

fragment shown in the right in Figure 11. It contains a wildcard atom denoted by a red B in the five-membered aromatic ring on the left, which may be replaced by either an oxygen or a sulfur atom. The interesting point is that the (non-wildcard) fragment that results from replacing B with sulfur has a very low support of 0.01% in the active molecules. This (non-wildcard) fragment would not be found without wildcard atoms, because it is not possible to run the search with a sufficiently low minimum support threshold.

An alternative type of “fuzzy” matches of a fragment to a molecule is concerned with chains of carbon atoms [6]. Often the exact length of a carbon chain in a molecule is not important, as long as it is within a certain range. In this case, the fragment contains an indication of a chain, which may match a certain number of chained carbon atoms in a molecule, where the number may differ for different molecules. This type of fuzzy match can be handled in a similar way as the ring extension we suggested in the previous section: Instead of bond by bond, a carbon chain is added in one step, allowing for different lengths (different number of carbon atoms). An example of a result that can be obtained in this way on the NCI cancer dataset is shown in Figure 12.

5 Conclusions

This paper reviewed some limitations of the molecular fragment mining algorithm introduced in [2] as well as approaches to overcome them. The first extension of the original algorithm deals with the frequent occurrence of rings and their effect on the width of the generated search tree [3]. By preprocessing the molecular database we can identify rings before the actual search starts and then treat them as one entity during the search. This not only speeds up the computation tremendously, but also finds chemically more meaningful fragments since chemists tend to think of rings as one molecular unit. The second extension addresses the need to find fragments that behave chemically similar but are different from a graph point of view. We have proposed a mechanism that allows the user to specify equivalence classes of atoms. Fragments can then contain a certain maximum number of such *wildcard* atoms [3]. Using this method we not only find chemically meaningful fragments, but we can also detect fragments that would otherwise fall below the support threshold. Finally, the search method can be extended to find carbon chains of varying length [6].

The extensions presented here make the original algorithm described in [2] directly applicable to much larger databases and allow the chemist to introduce expert knowledge about similarities on an atomic level, as demonstrated on the NCI-HIV dataset. As a result, the method presented here has been used very successfully, among others, at Tripos Receptor Research (Bude, UK) for the prediction of the success of compound synthesis.

References

1. R. Agrawal, T. Imieliński, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. *Proc. Conf. on Management of Data*, 207–216. ACM Press, New York, NY, USA 1993
2. C. Borgelt and M.R. Berthold. Mining Molecular Fragments: Finding Relevant Substructures of Molecules. *Proc. IEEE Int. Conf. on Data Mining (ICDM 2002, Maebashi, Japan)*, 51–58. IEEE Press, Piscataway, NJ, USA 2002
3. H. Hofer, C. Borgelt, and M.R. Berthold. Large Scale Mining of Molecular Fragments with Wildcards, *Proc. 5th Int. Symposium on Intelligent Data Analysis (IDA2003, Berlin, Germany)*, 376–385. Springer-Verlag, Heidelberg, Germany 2003
4. S. Kramer, L. de Raedt, and C. Helma. Molecular Feature Mining in HIV Data. *Proc. 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2001, San Francisco, CA)*, 136–143. ACM Press, New York, NY, USA 2001
5. M. Kuramochi and G. Karypis. An Efficient Algorithm for Discovering Frequent Subgraphs. Technical Report TR 02-026, Dept. of Computer Science/Army HPC Research Center, University of Minnesota, Minneapolis, USA 2002
6. T. Meinl, C. Borgelt, and M.R. Berthold. Mining Fragments with Fuzzy Chains in Molecular Databases. *Proc. 2nd Int. Workshop on Mining Graphs, Trees and Sequences (MGTS 2004, Pisa, Italy)*, 49–60. University of Pisa, Pisa, Italy 2004
7. O. Weislow, R. Kiser, D. Fine, J. Bader, R. Shoemaker, and M. Boyd. New Soluble Formazan Assay for HIV-1 Cytopathic Effects: Application to High Flux Screening of Synthetic and Natural Products for AIDS Antiviral Activity. *Journal of the National Cancer Institute*, 81:577–586. Oxford University Press, Oxford, United Kingdom 1989
8. M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New Algorithms for Fast Discovery of Association Rules. *Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD'97)*, 283–296. AAAI Press, Menlo Park, CA, USA 1997
9. http://dtp.nci.nih.gov/docs/cancer/cancer_data.html

Automatic Selection of Data Analysis Methods

Detlef D. Nauck, Martin Spott, and Ben Azvine

BT, Research and Venturing, Intelligent Systems Research Centre,
Aadal Park, Orion pp 1/12, Ipswich IP5 3RE, UK
{detlef.nauck, martin.spott, ben.azvine}@bt.com

Abstract. Modern business face the challenge of to make use of one of there most valuable assets – data about their customers and processes – in real time in order to stay ahead of global competition. In order to achieve real time business intelligence it is necessary to automate data analysis to address the lack of available experts, empower business users and produce analysis results where and when they are required. In this paper we describe how our intelligent data analysis platform SPIDA automatically selects suitable data analysis methods for application to data analysis problems.

1 Introduction

Modern business are facing ever increasing competition, globalisation and rapidly changing customer needs and technologies. In order to address new challenges quickly and effectively enterprise decision makers must rely on accurate and timely results from data analysis. Business Intelligence (BI) is a term used to identify methods and tools used by businesses to gather information about their enterprise and support decision making.

Traditional ways of conducting analysis and presenting results, for example in form of monthly or annual reports by marketing or finance department are no longer sufficient to react to a global dynamic competitive environment. Decision makers are no longer satisfied with scheduled analytics reports, pre-configured key performance indicators or fixed dashboards. They demand ad hoc queries to be answered quickly, they demand actionable information from analytic applications using real-time business performance data, and they demand these insights accessible to the right people exactly when and where they need them.

Real Time Business Intelligence (RTBI) is an approach to address the problems that are created by such demands:

- Real-time data access in order to produce intelligence on the current state of the business.
- Automatic real-time analytics to produce predictions, models, explanations and decision support on demand without the need for data anlysis experts.
- Real-time data fusion in order to combine data for heterogenous enterprise-wide data sources and legacy systems.

- Real-time data quality management in order to prevent low-quality data from entering systems or being used for analysis.
- Real-time decision making in order to act upon analytical results immediately and adapt operational parameters as required.

The meaning of RTBI mainly depends on understanding what *real-time* means for a business. It can mean

- the requirement to obtain zero latency within a process,
- that a process has access to information whenever it is required,
- that a process provides information whenever it is required by management, and
- the ability to derive key performance measures that relate to the situation to the current point in time and not just to some historic situation.

Based on these descriptions RTBI provides the same functionality as the traditional business intelligence, but operates on data that is extracted from operational data sources with zero latency and provides means to propagate actions back into business processes instantly.

In this paper we concentrate on our research programme to automate data analysis and provide a platform that not only empowers business user but also can function as an automatic data analysis engine for RTBI systems. Nowadays, available data analysis tools are still very much a collection of data analysis methods that require analytics experts as users. Business users are typically overwhelmed by the sheer knowledge that is required to use currently available tools. We consider this one of the reasons for the fact that modern machine learning techniques like decision trees, neural networks, fuzzy techniques, support vector machines, Bayesian networks etc. are still not industry standard.

Business users require a much more user-centric and problem-oriented approach to data analysis. Rather than knowing analysis methods, they are experts in the data domain and they typically know what they want to achieve with data analysis. Ideally, such users would simply feed high-level objectives and the available data into a tool that would then automatically find the best algorithm in terms of requirements, configure it, run it and create a software module that can be plugged into a business application.

Based on these ideas we developed SPIDA (Soft computing Platform for Intelligent Data Analysis) [1] and equipped it with a Wizard that, to certain extent, automates the data analysis process. In this paper, we focus on a way to select the most appropriate data analysis algorithm given a problem definition, a set of requirements and a data file.

2 Ranking Data Analysis Models

The selection of an appropriate data analysis method depends on a number of factors. First of all, the type of analysis problem (classification, approximation, segmentation, finding dependencies etc.) restricts the list of applicable methods.

The second category of requirements is concerned with preferences regarding the solution. These comprise properties like accuracy and simplicity of the solution, whether the method is adaptable to new data (rather than learning from scratch whenever new data is available), whether it offers an explanation facility like rule-based systems or functional models like linear regression and how simple the explanation should be. The user might have requirements for the execution time of a method, for prior knowledge to be integrated etc. Finally, the data might constrain the applicability of methods. The number of data records, for example, might be too small for some statistical methods, or generally, some methods might cope better with certain types of data than others.

Depending on the user group, the level at which the requirements are defined will vary considerably. Whereas some users will be able to understand the difference between function approximation and classification, for instance, others will certainly not. One could therefore think of hierarchical approaches where requirements are iteratively mapped onto lower level requirements until the lowest level is reached.

In order to evaluate the suitabilities of analysis methods, the requirements have to be mapped onto properties of the methods. In some cases, requirements are directly expressed as desired method properties. Adaptability is an example. A user might request a highly adaptable method, which can be directly matched to adaptability properties of given methods. Especially for higher level requirements this is not the case. For instance, 'I would like to detect fraudulent cases every night and I would like to understand the solution' could be mapped onto 'classification problem requiring a highly adaptable method with explanation facility providing a very simple explanation'.

Another observation is that non-expert users are often only able to specify requirements vaguely like 'I would like a simple explanation in terms of rules'. In this case the term 'simple' either indicates that the user cannot specify more accurately, what he means, because he does not know what to expect. Or the user does not want to give a more precise specification because his requirement is inherently vague, i.e. the exact degree of simplicity is not important.

In order to provide a mapping from requirements onto desired analysis method properties that can deal with vague requirements and properties we propose to use a fuzzy rule base. We map user preferences into target properties which are then matched to method properties and model properties. Method properties are static features of an analysis method like adaptability, explanation facility etc., whereas model properties can only be observed after an actual model has been created. Examples are accuracy and simplicity of an explanation. For instance, we may decide to use decision trees. Before we have actually learned a tree we cannot tell the accuracy nor do we know its size. On the other hand we know that decision trees in general are not adaptable.

The actual ranking procedure works as follows. First the user defines all preferences, which will be mapped onto desired properties. The method properties of all available analysis methods will then be matched with the respective desired properties and an initial ranking of methods will be produced. The user can

then select the methods he wants SPIDA to create models of. After the creation of actual models, model properties are available and then all desired properties can be matched with method and model properties. Depending on the degree of match of some of the model properties like accuracy or simplicity, SPIDA might decide to adapt learning parameters in order to improve the match. If a decision tree for example is very big and the user required a simple explanation, we might want to reduce the maximum tree height and prune or relearn the tree. If no more improvement can be achieved, the final ranking of created models can be presented. If several models have been created for the same method, the best one will be picked according to the user's preferences. However, the properties of all models can be shown and the user can pick a different one.

The rest of this section is organised as follows. Section 2.1 introduces the concept of combining fuzzy words which allow us to fuse symbolic and numeric requirements and properties in a sound way. Finally, we propose a method to measure the match between desired and actual analysis method properties in Section 2.2.

2.1 Combination of Fuzzy Words

As already laid out in Section 2, we use fuzzy techniques to describe user requirements as well as method, model and desired properties. Also, mappings between requirements and properties will be modelled as fuzzy rules. Looking closer at the requirements and properties reveals that some of them carry symbolic (categorical) values, whereas others are of numeric nature. Examples for the first category are the type of analysis problem (classification, function approximation, clustering etc.), while accuracy and simplicity represent the second category. Obviously, fuzzy sets can be defined on numeric as well as symbolic (categorical) domains. However, a more homogeneous approach has been presented in [2]. The main idea is to represent information at two levels of abstraction. The lower level is the level of details like the potentially infinite number of possible values in a continuous numeric domain of discourse. At the higher level of abstraction we only deal with a finite number of symbols by abstracting from details. In other words, the granularity of information can be either fine or coarse (potentially, entire hierarchies of granularity are conceivable). If we measure accuracy of a prediction model we can do so at the level of details as a value in $[0, 1]$, for example, or at the symbolic level in terms of symbols like 'high accuracy', 'medium accuracy', 'low accuracy'. What this approach allows us to do is to represent inherently symbolic information like the type of analysis problem mentioned above at the same level as symbols like 'high accuracy' which are in fact abstractions from a level of finer granularity.

The advantage of this approach compared to simply defining fuzzy sets on symbolic or numeric domains becomes more obvious when we think of expressing information using a combination of symbols like 'I fully accept a model of high accuracy, but to a lower degree I would accept a model of medium accuracy, as well.' This expression could be quantified in a requirement as 'high accuracy (1.0) + medium accuracy (0.6)'. We call such an expression a *weighted combination*

of symbols or in short a *combination*. In [3] it has been shown how to process such expressions at the symbolic level. Thereby, it does not matter, if the terms used are inherently symbolic or if they are actually fuzzy sets themselves. In this way, all symbols can be treated in a coherent way.

In [2, 3] we proposed a probabilistic model for the weights in combinations. That means in particular that the sum of the weights is 1. In Section 2.2 this model will be extended by allowing a sum of weights larger than 1. Obviously, such weights can not be interpreted as probabilities, they rather represent the existence of alternatives.

2.2 Measuring the Match of Requirements and Properties

We assume that the original user requirements have already been mapped onto desired properties and that information about the respective method and model properties is available. Each property is represented by a fuzzy variable that takes combinations of fuzzy words as their values. The desired accuracy, for example, could be 'medium (1.0) + high (1.0)' whereas a created analysis model might be accurate to the degree of 'low (0.3) + medium (0.7)'. In other words, we are looking for a model with medium or high accuracy, and the created model's accuracy is low with degree 0.3 and medium with degree 0.7. In this example, the weights for the desired accuracy sum up to 2, whereas the ones for the actual accuracy add up to 1. We interpret a combination of fuzzy words with sum of weights greater than one as alternative fuzzy words. Rather than modeling the weights as probabilities, we assume a possibility density function [4, 2] on the fuzzy words, which allows for alternative values which, as an extreme case, could all be possible without restriction. Thereby, we define the degree of possibility of a fuzzy word in conjunction with probabilities of the related method or model property. The degree stands for the maximum acceptable probability of a property. In the example above, we were entirely happy with an analysis model of medium or high accuracy. We therefore assigned the possibilistic weight 1 to both of them, i.e. models exhibiting the property 'low (0.0) + medium (a) + high (b)' with $a + b = 1$ are fully acceptable. In case of the requirement 'low (0.0) + medium (0.0) + high (1.0)' and the above property with $a > 0$ the weight a exceeds the possibility for 'medium' and therefore at least partially violates the requirements. Degrees of possibility can be any real number in $[0, 1]$.

Building on these ideas we stipulate that *requirements* are represented as a possibilistic combination of fuzzy words, i.e. at least one of the fuzzy words carries the weight 1, so the sum of weights of the fuzzy words is at least 1. The second property is based on the assumption that at least one of the alternative requirements is fully acceptable. *Properties* on the other hand stand for existing properties of a given model, so we assume that they are represented by a combination of fuzzy words, where the sum of weights of fuzzy words equals 1.

The following part of the section deals with the question of how a match of requirements and properties can be quantified, i.e. we are looking to measure the compatibility of properties with requirements. We will first focus on the

compatibility of a single pair of requirement/property and then consider ways of combining several degrees of compatibility in one value.

In order to find an appropriate compatibility measure we stipulate a number of required properties. For reasons of simplicity, we refer to combinations of fuzzy words simply as fuzzy sets (on fuzzy words) $\tilde{R}$ for requirements and $\tilde{P}$ for properties, defined on a finite universe of discourse $\mathcal{X}$. Subsethood of fuzzy sets is defined in the traditional way [5]

$$\tilde{A} \subseteq \tilde{B} \Leftrightarrow \forall x \in \mathcal{X} \mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x) \quad (1)$$

with $\mu_{\tilde{A}}$ being the membership function of $\tilde{A}$. For a compatibility measure $C(\tilde{P}, \tilde{R})$ we require

C1 $C(\tilde{P}, \tilde{R}) \in [0, 1]$

C2 If the fuzzy sets are disjoint, compatibility is 0:

$$\tilde{P} \cap \tilde{R} = \emptyset \Rightarrow C(\tilde{P}, \tilde{R}) = 0$$

C3 If the requirement fuzzy set is a superset of the property fuzzy set, compatibility is 1:

$$\tilde{R} \supseteq \tilde{P} \Rightarrow C(\tilde{P}, \tilde{R}) = 1$$

C4 Monotony in both arguments:

$$\text{a) } \tilde{P}' \subseteq \tilde{P} \Rightarrow C(\tilde{P}', \tilde{R}) \geq C(\tilde{P}, \tilde{R})$$

$$\text{b) } \tilde{R}' \supseteq \tilde{R} \Rightarrow C(\tilde{P}, \tilde{R}') \geq C(\tilde{P}, \tilde{R})$$

C1 is simply a normalisation, whereby a match value of 1 means full match and a value of 0 stands for no match. If the fuzzy sets are disjoint the property does not meet any requirement and the match is therefore 0 (C2). In C3 we make sure that the degree of match is 1 as long as the requirement covers the property, i.e. the property is fully acceptable. Finally, monotony in C4 means that the more relaxed the spectrum of requirements or the more specific the properties the higher the degree of compatibility.

Properties C1-4 resemble typical properties of measures for fuzzy set inclusion. Normally, such measures are generalisations of the inclusion of crisp sets based on a number of axioms [6, 7]. However, many of these axioms are too strict for our application. Also, the inclusion measure $Inc(\tilde{A}, \tilde{B}) = \min_{x \in \mathcal{X}} I(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$ with fuzzy implication operator I proposed in the above publications suffers from the serious drawback that it draws its value from a minimum operation, i.e. for quite different fuzzy sets $\tilde{A}$, for instance, Inc would return the same degree of inclusion. In mathematical terms, Inc is not strictly monotonous. Furthermore, it does not take into account the relation in size of the subset of $\tilde{A}$ that is included in $\tilde{B}$ and the one that is not included.

We therefore propose a different measure which meets all our requirements. As already mentioned above, we assume $\sum_{x \in \mathcal{X}} \mu_{\tilde{R}}(x) \geq 1$ and $\sum_{x \in \mathcal{X}} \mu_{\tilde{P}}(x) = 1$.

$$C(\tilde{P}, \tilde{R}) := 1 - \frac{1}{2} \left(\sum_{x \in \mathcal{X}} |\mu_{\tilde{R}}(x) - \mu_{\tilde{P}}(x)| - \sum_{x \in \mathcal{X}} \mu_{\tilde{R}}(x) + 1 \right) \quad (2)$$

which can be rewritten as

$$C(\tilde{P}, \tilde{R}) = 1 - \sum_{\substack{x \in \mathcal{X}: \\ \mu_{\tilde{P}}(x) > \mu_{\tilde{R}}(x)}} \mu_{\tilde{P}}(x) - \mu_{\tilde{R}}(x) \quad (3)$$

Before we proof that the measure C fulfils requirements C1-4 let us explain what the measure actually does. Figure 1 shows a fuzzy set $\tilde{R}$ for requirements on the left hand side (we used a continuous domain for better illustration), the right triangular function is the fuzzy set $\tilde{P}$ for properties. The right term in (3) measures the size of the grey area, which can be interpreted as a degree to which the properties violate the requirements. The size of the area is bounded by the area underneath the membership function of $\tilde{P}$ which we stipulated to be 1. That means that the right term measures the proportion of properties $\tilde{P}$ that violate the requirements. From (3) immediately follows

Lemma 1. *For the fuzzy set $\tilde{R}'$ with $\forall x \in \mathcal{X} \mu_{\tilde{R}'}(x) = \min\{\mu_{\tilde{R}}(x), \mu_{\tilde{P}}(x)\}$ holds $C(\tilde{P}, \tilde{R}') = C(\tilde{P}, \tilde{R})$.*

The Lemma shows that the requirements $\tilde{R}$ can be stripped down to the intersection $\tilde{R}'$ of requirements and properties $\tilde{P}$ (using min as t-norm) without changing the compatibility of properties with requirements, see Fig. 1.

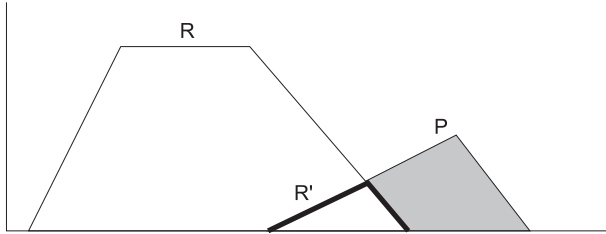


Fig. 1. Fuzzy sets of requirements $\tilde{R}$, properties $\tilde{P}$ and the intersection $\tilde{R}'$. The grey area represents incompatibility of properties with requirements

Lemma 2. *The measure C defined above conforms to properties C1-4.*

Proof. 1. Since the right term in (3) is obviously not negative and the following holds

$$\sum_{\substack{x \in \mathcal{X}: \\ \mu_{\tilde{P}}(x) > \mu_{\tilde{R}}(x)}} \mu_{\tilde{P}}(x) - \mu_{\tilde{R}}(x) \stackrel{(*)}{\leq} \sum_{x \in \mathcal{X}} \mu_{\tilde{P}}(x) = 1 \quad (4)$$

C meets requirement C1.

2. In case of $\tilde{P} \cap \tilde{R} = \emptyset$, we have equality at $(*)$ in (4) and therefore C2 is met.
3. If $\tilde{R} \supseteq \tilde{P}$, then the set $x \in \mathcal{X} : \mu_{\tilde{P}}(x) > \mu_{\tilde{R}}(x)$ is empty, the right term in (3) is 0 and therefore $C(\tilde{P}, \tilde{R}) = 1$.

4. If $\tilde{R}$ grows or $\tilde{P}$ shrinks the value of the right term in (3) cannot decrease (equivalently the size of the grey area in Fig. 1 cannot shrink) therefore the value of C does not decrease.

Furthermore, it turns out that C is a measure of satisfiability as defined in [8] which is not surprising since their notion of satisfiability is very similar to our understanding of compatibility.

Since we deal with a set of requirements and properties, we end up having one match value for each requirement/property pair. Requiring a set of properties can formally be interpreted as a logical conjunction of the individual requirements. Given the assumption that all requirements are equally important, we therefore propose to use a t-norm to aggregate the individual match values. We decided to use multiplication, as it is a strictly monotonous operator. Strict monotony basically means that the overall match value decreases with any of the individual match values. Other operators like minimum do not have this property. In case of minimum, the overall match obviously is the minimum of the individual matches. That means that all the match values apart from the minimum can be increased without changing the overall value. This is not the desired behaviour in our case since many different sets of properties would result in the same overall match value as long as the minimal value is the same. So the proposed measure for a multi-criteria match is

$$\begin{aligned}
 C(\tilde{P}, \tilde{R}) &= \prod_{j=1}^m C(\tilde{P}_j, \tilde{R}_j) \\
 &= \prod_{j=1}^m \left(1 - \sum_{\substack{x \in \mathcal{X}_j: \\ \mu_{\tilde{P}_j}(x) > \mu_{\tilde{R}_j}(x)}} \mu_{\tilde{P}_j}(x) - \mu_{\tilde{R}_j}(x)\right)
 \end{aligned} \tag{5}$$

3 The SPIDA Wizard for Analysis Model Selection

Based on the techniques described in the preceding sections, we implemented a wizard for our data analysis tool SPIDA. In a series of dialogs the user specifies the data analysis problem (prediction, grouping, dependencies), chooses the data source and gives his preferences regarding the solution (explanation facility, type of explanation, simplicity of explanation, facility to take prior knowledge, adaptability, accuracy etc.).

Figure 2 shows the dialog for specifying requirements for an explanation facility. Possible selections are a mixture of fuzzy terms like 'at least medium' or 'simple' for simplicity, and crisp terms like 'Rules' and 'Functions' for type of explanation. The dialogs for other preferences look very similar. A typical ranking of data analysis methods according to user preferences is shown in Fig. 3, where the match or compatibility of method properties with preferences is given as suitability. At this stage, no models have been created, so model properties like accuracy and simplicity are not taken into account for the suitability.

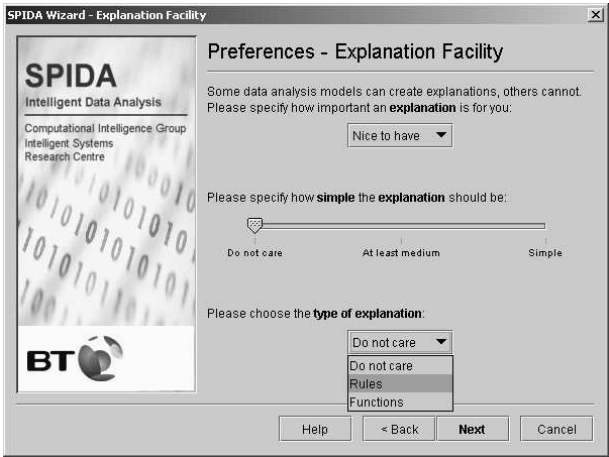


Fig. 2. Specifying preferences, here regarding an explanation facility

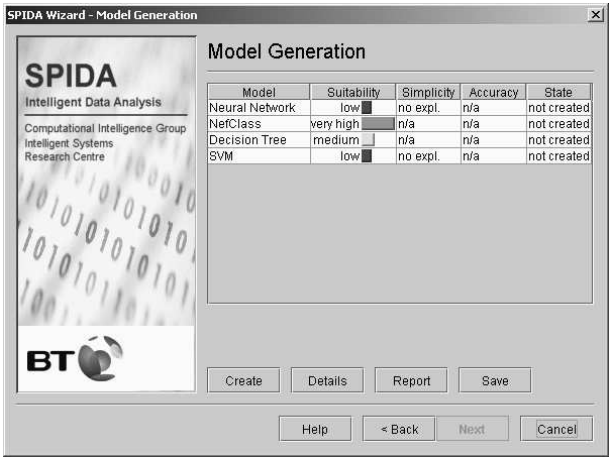


Fig. 3. Ranking of analysis models

The user can preselect the most suitable methods and trigger the creation of models for them. As already mentioned in Section 2, the wizard will then create models for each selected method, evaluate model properties afterwards and try to improve on the match with the respective desired properties. This is achieved by changing learning parameters of the methods, which have been collected from experts in the field. If no improvement can be achieved, anymore, the final overall suitability can be shown. Figure 4 shows five different models of the Neuro-Fuzzy classifier Nefclass [9]. The user has asked for a simple model, so the wizard tried to force Nefclass to produce a simple solution but keeping the accuracy up. As can be seen in the figure SPIDA produced three models with high simplicity,

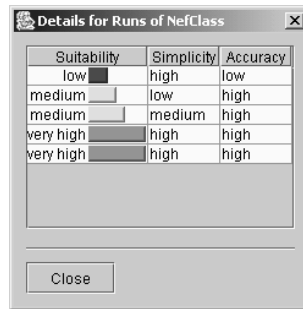


Fig. 4. Accuracy, simplicity and overall suitability of different Nefclass models

but considerably different accuracy – in this case between 44% and 55% (the actual values for accuracy can be revealed in tool tips). The user can balance the importance of simplicity against accuracy as one of the preferences, so the wizard decides on the best model according to this. Nevertheless, the user can pick a different model based on the information in Fig. 4.

3.1 User Preferences and Method Properties

In the current version of the SPIDA wizard, we measure the suitability of a data analysis method according to the following method properties

- type of analysis problem (classification, function approximation, clustering, dependency analysis etc.)
- if an explanation facility exists
- type of explanation (rules or functions)
- adaptability to new data
- if prior knowledge can be integrated

and model properties

- simplicity of an explanation
- accuracy

Another conceivable model property is execution time, which can be crucial for real-time applications. Examples for property profiles are shown in Table 1.

Table 1. Property profiles for decision trees, neural networks and Nefclass

Method	Problem	Explain	Adapt	Prior Knowl.
Decision Tree	classification	rules	no	no
Neural Network	classification, func. approx.	no	medium	no
Nefclass	classification	rules	high	yes

The method properties above are symbolic, whereas the model properties are numeric. In general, of course, this is not necessarily the case. For all numeric properties, fuzzy sets have to be defined as granularisation of the underlying domain. For example, if accuracy was measured as value in $[0, 1]$ fuzzy sets for 'high', 'medium' and 'low' accuracy could be defined on $[0, 1]$ as fuzzy values for accuracy. Since accuracy is heavily dependent on the application, the definition of the fuzzy terms is as well. We ask users to specify a desired accuracy and the lowest acceptable accuracy whenever they use the wizard. These two crisp accuracy values are then used as cross-over points for three trapezoidal membership functions for 'high', 'medium' and 'low'. In case the user cannot specify accuracy due to a lack of knowledge, accuracy will simply not be used to determine the suitability of an analysis model. For other properties, fuzzy sets can be defined accordingly, either by the user or by the expert who designs the wizard. Fuzzy sets can even be adapted by user feedback. If the wizard, for instance, recommends a supposedly simple model that is not simple at all from the user's perspective the underlying fuzzy set can be changed accordingly (user profiling).

In the current version of the wizard, user preferences are specified at a similar level as desired method and model properties. They include

- type of analysis problem (classification, function approximation, clustering, dependency analysis etc.)
- importance of an explanation facility (do not care, nice to have, important)
- type of explanation (do not care, rules, functions)
- adaptability to new data (do not care, nice to have, important)
- integration of prior knowledge (do not care, nice to have, important)
- simplicity of an explanation
- accuracy
- balance importance of accuracy and simplicity

The mapping from user preferences onto desired properties is therefore quite simple, in some cases like accuracy almost a one-to-one relation like 'If accuracy preference is at least medium, then desired accuracy is medium or high'. For others like simplicity it is slightly more complicated with rules like 'If simplicity preference is high and an explanation is important, then desired simplicity is medium (0.6) + high (1.0)'.

The balance for the importance of accuracy and simplicity is not used to compute the suitability of models, since we can assume that the user has specified his preferences regarding these properties. It is only taken into account if several models of the same analysis method get the same suitability score, so the wizard can decide on the better one. The balance is also used when the wizard decides to rerun an analysis method with different learning parameters because accuracy and/or simplicity are not satisfactory. Depending on a combination of accuracy and simplicity score and their balance the wizard changes parameters in order to either improve on accuracy or simplicity.

Some properties like the level of accuracy can easily be measured and compared for all models, whereas others like the level of simplicity are more difficult. In [10] we proposed a way to measure the interpretability of rule sets (crisp or fuzzy), which

can be used as a measure of simplicity for most rule-based models. Measuring the simplicity of models which are based on functions is more difficult, especially since we require such a measure to be comparable with a measure for rule sets (commensurability). Nevertheless, heuristically defined measures that take into account the number of arguments (as in rule sets) and the complexity of a function usually work well enough, in particular, since we finally evaluate simplicity on the basis of a handful of fuzzy values and not on the underlying continuous domain.

4 Conclusion

As a new direction in automating data analysis, we introduced the concept of using soft constraints for the selection of an appropriate data analysis method. These constraints represent the user's requirements regarding the analysis problem in terms of the actual problem (like prediction, clustering or finding dependencies) and preferences regarding the solution. Requirements can potentially be defined at any level of abstraction. Expert knowledge in terms of a fuzzy rule base maps high-level requirements onto required properties of data analysis methods which will then be matched to actual properties of analysis methods. As a result of our work, we introduced a new measure for the compatibility of fuzzy requirements with fuzzy properties that can be applied to other problems in the area of multi-criteria decision making. The methods presented above have been implemented as a wizard for our data analysis tool SPIDA, which has been successfully used to produce solutions to a variety of problems within BT, e.g. fraud detection, travel time prediction and customer satisfaction analysis.

References

1. Nauck, D., Spott, M., Azvine, B.: Spida – a novel data analysis tool. *BT Technology Journal* **21** (2003) 104–112
2. Spott, M.: Combining fuzzy words. In: *Proc. of FUZZ-IEEE 2001*, Melbourne, Australia (2001)
3. Spott, M.: Efficient reasoning with fuzzy words. In Halgamuge, S.K., Wang, L., eds.: *Computational Intelligence for Modelling and Predictions*. Springer Verlag (2004) (to appear).
4. Gebhardt, J., Kruse, R.: The context model—an integrating view of vagueness and uncertainty. *Intern. Journal of Approximate Reasoning* **9** (1993) 283–314
5. Zadeh, L.A.: Fuzzy sets. *Information and Control* **8** (1965) 338–353
6. Sinha, D., Dougherty, E.: Fuzzification of set inclusion: theory and applications. *FSS* **55** (1993) 15–42
7. Cornelis, C., Van der Donck, C., Kerre, E.: Sinha-dougherty approach to the fuzzification of set inclusion revisited. *FSS* **134** (2003) 283–295
8. Bouchon-Meunier, B., Rifqi, M., Bothorel, S.: Towards general measures of comparison of objects. *Fuzzy Sets and Systems* **84** (1996) 143–153
9. Nauck, D., Kruse, R.: A neuro-fuzzy method to learn fuzzy classification rules from data. *FSS* **89** (1997) 277–288
10. Nauck, D.: Measuring interpretability in rule-based classification systems. In: *Proc. IEEE Int. Conf. on Fuzzy Systems 2003*, St. Louis (2003) 196–201

Author Index

- Aguzzoli, Stefano 650, 662
Alsinet, Teresa 353
Amgoud, Leila 269, 527
Arieli, Ofer 563
Avron, Arnon 625
Awad, Mohammed 613
Azvine, Ben 1014
- Baroni, Pietro 329
Barragáns Martínez, A. Belén 638
Bell, David A. 465, 501
Ben Amor, Nahla 921, 944
Benferhat, Salem 452, 921
Benhamou, Belaïd 477
Bennaim, Jonathan 452
Berthold, Michael R. 1002
Besnard, Philippe 427
Biazzo, Veronica 775
Björkegren, Johan 136
Bonnefon, Jean-Francois 269
Borgelt, Christian 100, 1002
Bosc, Patrick 812
Bouckaert, Remco R. 221
- Cano, Andrés R. 908, 932
Capotorti, Andrea 750
Castellano, Javier G. 174, 908, 932
Cayrol, Claudette 366, 378
Chesñevar, Carlos 353
Cholvy, Laurence 390
Cobb, Barry R. 27
Coletti, Giulianella 872
Cornelis, Chris 563
Coste-Marquis, Sylvie 317
- Daniel, Milan 539, 824
D'Antona, Ottavio M. 650
de Campos, Luis M. 123, 174
Dencœux, Thierry 552
Deschrijver, Glad 563
Devred, Caroline 317
Díaz Redondo, Rebeca P. 638
Dubois, Didier 293, 305, 848
- Eklund, Patrik 341
Elouedi, Zied 921, 944
- Fargier, Hélène 305
Farrokh, Arsalan 198
Fernández Vilas, Ana 638
Fernández-Luna, Juan M. 123
Flaminio, Tommaso 714
Flores, M. Julia 63
Fuentetaja, Raquel 88
- Gámez, José A. 63, 161
Gammerman, Alex 111
Ganascia, Jean-Gabriel 968
García Duque, Jorge 638
Garcia, Laurent 402
Garmendia, Luis 576, 587
Garrote, Luis 88
Gauwin, Olivier 514
Gebhardt, Jörg 3
Georgescu, Irina 257
Gerla, Brunella 662
Giacomin, Massimiliano 329
Gil Solla, Alberto 638
Gilio, Angelo 775
Godo, Lluís 353
Gómez, Manuel 123
González, Jesús 980
Guglielmann, Raffaella 600
Guillén, Alberto 613, 980
- Haenni, Rolf 788
Herrera, Luis Javier 613, 980
Huete, Juan F. 123
Hüllermeier, Eyke 848
Hunter, Anthony 415
- Ikodinović, Nebojša 726
Ironi, Liliana 600
- Jeansoulin, Robert 452
Jenhani, Ilyes 944
Jensen, Finn V. 76
Jin, Zhi 440
Jøsang, Audun 824

- Kaci, Souhila 281, 293, 527
 Kerre, Etienne 563
 Khelfallah, Mahat 452, 477
 Klawonn, Frank 992
 Konieczny, Sébastien 514
 Kramosil, Ivan 884
 Krishnamurthy, Vikram 198
 Kruse, Rudolf 3, 100

 Lagasquie-Schiex, Marie Christine 366, 378
 Lagrue, Sylvain 452
 Lang, Jérôme 15
 Larrañaga, Pedro 148
 Lawry, Jonathan 896
 Lee, Jae-Hyuck 186
 Li, Wenhui 836
 Lindgren, Helena 341
 Liu, Weiru 415, 440, 465, 501
 López Nores, Martín 638
 Lozano, Jose A. 148
 Lu, Ruqian 440
 Lucas, Peter 244
 Lukasiewicz, Thomas 737
 Luo, Zhiyuan 111

 Majercik, Stephen M. 209
 Manara, Corrado 662
 Marchioni, Enrico 701
 Marquis, Pierre 317, 514
 Marra, Vincenzo 650
 Martínez, Irene 51
 Masegosa, Andrés R. 908, 932
 Mellouli, Khaled 944
 Mercier, David 552
 Meyer, Thomas 489
 Miranda, Enrique 860
 Molina, Martin 88
 Moral, Serafin 1, 51, 63, 908, 932
 Mu, Kedian 440

 Nauck, Detlef D. 1014
 Neufeld, Eric 233
 Nicolas, Pascal 402
 Nielsen, Thomas D. 76

 Ognjanović, Zoran 726

 Papini, Odile 452
 Patterson, David E. 1002
 Pazos Arias, José J. 638
 Peña, Jose M. 136

 Perrussel, Laurent 489
 Pini, Maria Silvia 800
 Pivert, Olivier 812
 Pomares, Héctor 613, 980
 Poole, David 763
 Pope, Simon 824
 Pozos Parra, Pilar 489
 Prade, Henri 269, 293, 675
 Puerta, J. Miguel 161

 Qi, Guilin 465, 501
 Qin, Zengchang 896
 Quost, Benjamin 552

 Ramos Cabrer, Manuel 638
 Rehm, Frank 992
 Rodríguez, Carmelo 51
 Rojas, Ignacio 613, 980
 Rossi, Francesca 800
 Rumí, Rafael 39

 Salmerón, Antonio 39, 51
 Salvador, Adela 576, 587
 Sanscartier, Manon J. 233
 Santafé, Guzmán 148
 Serrurier, Mathieu 675
 Shenoy, Prakash P. 27
 Simari, Guillermo 353
 Smets, Philippe 956
 Smyth, Clinton 763
 Spott, Martin 1014
 Stéphan, Igor 402
 Straccia, Umberto 687
 Studený, Milan 221
 Sun, Haibin 836

 Tegnér, Jesper 136

 Valenzuela, Olga 613
 van der Torre, Leendert 281
 van der Weide, Theo 244
 van Gerven, Marcel 244
 Vannoorenberghe, Patrick 956
 Vantaggi, Barbara 872
 Velcin, Julien 968
 Venable, Brent 800

 Würbel, Eric 452
 Wilson, Nic 452

 Zagoraiou, Maroussa 750

Lecture Notes in Artificial Intelligence

Subseries of Lecture Notes in Computer Science

The LNAI series reports state-of-the-art results in artificial intelligence research, development, and education, at a high level and in both printed and electronic form. Enjoying tight cooperation with the R&D community, with numerous individuals, as well as with prestigious organizations and societies, LNAI has grown into the most comprehensive artificial intelligence research forum available.

The scope of LNAI spans the whole range of artificial intelligence and intelligent information processing including interdisciplinary topics in a variety of application fields. The type of material published traditionally includes

- proceedings (published in time for the respective conference)
- post-proceedings (consisting of thoroughly revised final full papers)
- research monographs (which may be based on PhD work)

More recently, several color-cover sublines have been added featuring, beyond a collection of papers, various added-value components; these sublines include

- tutorials (textbook-like monographs or collections of lectures given at advanced courses)
- state-of-the-art surveys (offering complete and mediated coverage of a topic)
- hot topics (introducing emergent topics to the broader community)

In parallel to the printed book, each new volume is published electronically in LNCS/LNAI Online.

Detailed information on LNCS/LNAI can be found at
<http://www.springeronline.com>

Proposals for publication should be sent to

LNCS/LNAI Editorial, Tiergartenstr. 17, 69121 Heidelberg, Germany
E-mail: lncs@springer-sbm.com

ISSN 0302-9743

ISBN 3-540-27326-3



springeronline.com

**Lecture Notes in
Artificial Intelligence**

Lecture Notes in Computer Science