Foreword

Scientific progress is driven, in large measure, by questioning the validity of axioms, dogmas and traditions. One of the most deep-seated traditions in science is that of according much more respect to numbers than to words. The essence of this tradition was articulated succinctly by Lord Kelvin in 1883:

> In physical science the first essential step in the direction of learning any subject is to find principles of numerical reckoning and practicable methods for measuring some quality connected with it. I often say that when you can measure what you are speaking about and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind: it may be the beginning of knowledge but you have scarcely in your thoughts, advanced to the state of science, whatever the matter may be.

Adherence to the tradition of primacy of numbers over words has led to brilliant successes that are visible to all. But does it follow that the tradition should be accorded this status of immutable truth? No, certainly not.

The reality is that alongside brilliant successes we see sobering failure. We have sent men to the Moon but we cannot build a robot that can play tennis. We can build computers that can execute billions of instructions per second but cannot create programs that can summarize a non-stereotypical story, much less a book. We cannot automate driving in city traffic and we cannot construct machine translation programs that can perform at the level of a human interpreter, and so on. Does this reality suggest that the tradition of primacy of numbers over words may be a part of the problem? In a sense, 'Modelling with Words' may be viewed as an affirmative answer to this question.

In retrospect, my 1973 paper, 'Outline of a New Approach to the Analysis of Complex Systems and Decision Processes,' may be seen as an initial step in the direction of enhancing the power of words through the use of words as objects of computation. A basic concept that was introduced in that paper is that of a linguistic variable, that is, a variable whose values are words or phrases drawn from a natural language. The concept of a linguistic variable is closely linked to the concept of granulation and forms the basis for the concept of a linguistic if – then rule.

Reflecting the depth of the tradition of the primacy of numbers over words, the initial reaction to the concept of a linguistic variable ranged from mixed to skeptical and hostile. Commenting on my first presentation of the concept of a linguistic variable in 1972, Professor Rudolf Kalman, a brilliant mind, had this to say:

> I would like to comment briefly on Professor Zadeh's presentation. His proposals could be severely, ferociously, even brutally criticized from a

> technical point of view. This would be out of place here. But a blunt question remains: Is Professor Zadeh presenting important ideas or is he indulging in wishful thinking? No doubt Professor Zadeh's enthusiasm for fuzziness has been reinforced by the prevailing climate in the U.S. – one of unprecedented permissiveness. 'Fuzzification' is a kind of scientific permissiveness; it tends to result in socially appealing slogans unaccompanied by the discipline of hard scientific work and patient observation.

Today, almost all applications of fuzzy logic and fuzzy set theory involve the concept of a linguistic variable. Underlying these applications are two principal rationales. The first, and most obvious, is related to problems in which the values of variables and their interrelations are not known with sufficient precision to justify the use of numbers. However, it should be noted that it is common practice to employ numbers, even when their use is not justified, for the simple reason that numbers carry more respect than words. The second rationale relates to the use of words – since words are intrinsically less precise than numbers – to exploit the tolerance for imprecision to achieve tractability, robustness and, above all, low solution cost. It is this rationale that underlies the extensive use of fuzzy logic in consumer products and industrial systems.

Seen against this backdrop, 'Modelling with Words' may be viewed as a presentation of a persuasive ease for the development and adoption of the methodology of linguistic modelling – a methodology that is an extension and generalization of conventional, numerically-based approaches to modelling. In this perspective, the machinery of modelling with words is an important addition to the armamentarium of modelling methodologies.

'Modelling with Words' is an impressive collection of authoritative contributions that relate to a wide range of issues in modelling and systems analysis, extending from conceptual graphs and fuzzy quantifiers to humanist computing and self-organizing maps. There is much that is new in Modelling with Words, and numerous examples add significantly to the value of the information.

A basic issue, which is addressed by some of the contributors, is the relationship between fuzziness and randomness. This issue has been an object of discussion and debate since the early days of fuzzy set theory and fuzzy logic. The crux of the issue is the question: 'Is fuzziness distinct from randomness?' A history of discussions of this issue may be found in my 1995 paper in Technometrics entitled 'Probability Theory and Fuzzy Logic Are Complementary Rather than Competitive.'

It was pointed out by Robbins in 1965, Loginov in 1966, Orlov in 1980, and many others since then that a fuzzy set may be generated by a random set. A physical analogy that is described in my 1995 paper is the following. A fuzzy shadow on the base of an enlarger may be generated by dodging, that is, by a pattern of randomly moving an opaque planar object under the lens of the enlarger. However, the same shadow can also be generated in a non-random way by placing under the lens a stack of partially translucent sheets shaped like the α-cuts of the fuzzy shadow. Thus, what can be said is that a fuzzy set may be generated from a family of crisp sets by various operations,

some random and some not. Furthermore, anything that can be done with crisp random sets can be done with convex combinations of crisp non-random sets. The mathematical linkage between fuzzy sets and random sets may be used to obtain useful results, as was done in the books by Orlov and Goodman, and by the authors in 'Modelling with Words.' However, the mathematical linkage between fuzzy sets and random sets does not suggest that fuzzy set theory is a part of probability theory or vice-versa. It may be argued, as I have done in my 1995 paper, that the two theories are complementary, but my more recent and more radical view is that probability theory should be based on fuzzy logic, rather than on the bivalent logic that it has been based on since its inception.

Another issue that I should like to comment on is the relation between Modelling with Words (MW) and Computing with Words (CW).

As its label suggests, in Computing with Words the accent is on computation rather than on modelling. Thus, in CW the objects of computation are words, phrases or propositions drawn from a natural language. A major focus of attention in CW is computation with perceptions, with the understanding that perceptions are described by propositions drawn from a natural language. A key component of CW is Precisiated Natural Language (PNL).

The relationship between Modelling with Words and Computing with Words is close and is likely to become even closer with the passage of time. In the final analysis, both are aimed at enlarging the role of natural languages in scientific theories and, especially, in knowledge management, decision and control.

'Modelling with Words' is an important contribution to the conception, design and utilization of intelligent systems. It is informative, authoritative and reader-friendly. The editors, J. Lawry, J. Shanahan and A. Ralescu, the contributors and the publisher, Springer-Verlag, have done an excellent job and deserve our thanks and congratulations.

July 2003

Lotfi A. Zadeh
Berkeley, CA

Preface

The development of high-performance computers and the corresponding advances in global communications have lead to an explosion in data collection, transmission and storage. Large-scale multi dimensional databases are being generated to describe a wide variety of systems. These can range from engineering applications such as computer vision, to scientific data such as that from the genome project, to customer and price modelling in business and finance. In all of these cases the data is useless without methods of analysis by which we can discover the important underlying trends and relationships, integrate other background information, and then carry out inference on the learnt models. For a number of reasons we argue that in order to fulfill these requirements we should move towards a modelling paradigm that is as close to natural language as possible.

In recent years the area of machine learning has focused on the development of induction algorithms that is maximize predictive accuracy. However, since there has been little emphasis on knowledge representation the models derived are typically 'black box' and therefore difficult to understand and interpret. For many applications a high level of predictive accuracy is all that is required. However, in a large number of cases, including many critical application, a clear understanding of the prediction mechanisms is vital if there is to be sufficient confidence in the model for it to be used as a decision-making tool. Model transparency of this kind is best achieved within a natural-language-based modelling framework that allows for the representation of both uncertainty and fuzziness. We must be aware, however, of an inherent trade-off between model accuracy and transparency. Simple models, while the most transparent, are often inadequate to capture the complex dependencies that exist in many practical modelling problems. Alternatively, more complex models are much more difficult to represent in a clear and understandable manner. This trade-off is best managed by close collaboration with domain experts who can provide the modeller with an unbiased assessment of the transparency of their models while also establishing what level of accuracy is necessary for the current problem. Another important justification for learning models at a linguistic level is that it facilitates their fusion with background knowledge obtained from domain experts.

In any data modelling problem there is almost certain to be some expert knowledge available, derived from either an in-depth understanding of the underlying physical processes or from years of practical experience. In expert systems the emphasis is placed almost entirely on this expert information, with data being used only to optimize the performance of the model. On the other hand, in machine learning, background knowledge is largely ignored, except perhaps in the limited role of constraining prior distributions in Bayesian methods. As part of modelling with words we propose that there should be a high-level fusion of expert- and data-derived knowledge. By integrating these two types

of information it should be possible to improve on the performance of models that are based solely on one or the other. Furthermore, the effective use of background knowledge can allow for the application of simpler learning algorithms, producing simpler, and hence more transparent, models.

Given a model of a data problem it is highly desirable that practitioners be able to interrogate it in order to evaluate interesting hypotheses. Since these hypotheses are most likely to be in natural-language form, to achieve this a high-level inference mechanism on linguistic terms is required. Such an inference process is, in essence, what Zadeh calls 'computing with words.' The nature of any reasoning mechanism at this level will depend on the nature of the data models. For example, if the models take the form of a fuzzy rule base then methods similar to those proposed by Zadeh may be appropriate. Alternatively, if the model consists of conceptual graphs then graph matching and other similar methods from conceptual graph theory will need to be used. However, no matter what methodology is applied it must be formally well defined and based on a clear underlying semantics. In this respect modelling with words differs from natural language since we require a much more formal representation and reasoning framework for the former than for the latter. In fact this high level of formal rigor is necessary if we are to obtain models that are sufficiently transparent to satisfy practitioners of their validity in critical applications. Certainly, a modelling process cannot be truly transparent if there are significant doubts regarding the meaning of the underlying concepts used or the soundness of the learning and reasoning mechanisms employed. This formal aspect of modelling with words is likely to mean that some of the flexibility and expressiveness of natural language will need to be sacrificed. The goal, however, is to maintain rigor within a representation framework that captures many of the important characteristics of natural language so as to allow relative ease of translation between the two domains. This is very similar to the idea behind Zadeh's 'precisiated natural language.'

Modelling with words can be defined in terms of the trilogy, learning, fusion and reasoning as carried out within a formal linguistic representation framework. As such this new paradigm gives rise to a number of interesting and distinct challenges within each of these three areas. In learning, how can the dual goals of good predictive accuracy and a high level of transparency be reconciled? Also, how can we scale our linguistic algorithms to high-dimensional data problems? In fusion, what are the most effective methods for integrating linguistic expert knowledge with data-derived knowledge, and how does this process constrain the representation of both types of knowledge? In reasoning, what sound and useful rules of inference can be identified and what type of queries can they evaluate? In general, how can we effectively integrate fuzzy and probabilistic uncertainty in data modelling and what type of knowledge representation framework is most appropriate? This volume contains a collection of papers that begin to address some of these issues in depth. Papers by E. Hernandez et al. and A. Laurent et al. investigate the use of fuzzy decision trees to derive linguistic rules from data. H. Ishibuchi et al. and R. Alcala et al. describe how genetic algorithms can be used

to improve the performance of fuzzy models. The area of fuzzy conceptual graphs is the topic of papers by T. Cao and P. Paulson et al. Linguistic modelling and reasoning frameworks based on random sets are discussed in papers by J. Lawry and F. Diaz-Hermida et al., and Q. Shen introduces an algorithm according to which rough sets can be used to identify important attributes. The application of fuzzy sets to text classification is investigated by Y. Chen, and J. Rossiter discusses the paradigm of humanist computing and its relationship to modelling with words.

June 2003

Jonathan Lawry
Jimi Shanahan
Anca Ralescu

Author Index

Alcalá, Rafael 44

Barro, Senén 1
Bouchon-Meunier, Bernadette . 102
Bugarín, Alberto 1

Cao, Tru H. 80
Cariñena, Purificación 1
Chen, Yi-Ping Phoebe 153
Cordón, Oscar 44

Díaz-Hermida, Felix 1

Hernández, Enric 26
Herrera, Francisco 44

Ishibuchi, Hisao 209

Laurent, Anne 102
Lawry, Jonathan 186

Marsala, Christophe 102

Paulson, Patrick 168

Recasens, Jordi 26
Rossiter, Jonathan 124

Shen, Qiang 64

Tzanavari, Aimilia 168

Yamamoto, Takashi 209

Random Set-Based Approaches for Modelling Fuzzy Operators*

Felix Díaz-Hermida, Purificación Cariñena, Alberto Bugarín, and Senén Barro

Dep. Electrónica e Computación Univ. de Santiago de Compostela
15782 Santiago de Compostela, Spain
felixdh@usc.es
{puri,alberto,senen}@dec.usc.es

Abstract. In this work two approaches for extending set functions to fuzzy set functions are presented. The first approach describes an extension mechanism based on a random set interpretation of fuzzy sets. In the second approach fuzzy sets are interpreted on the basis of random trivalued sets. Examples showing that both techniques exhibit behaviours that are well-suited for modelling different tasks such as fuzzy quantification, fuzzy cardinality and fuzzy temporal operators are presented.

Keywords. Fuzzy cardinality, extension of fuzzy operators, extension principle, fuzzy quantification, random sets, trivalued sets.

1 Introduction

In contrast with traditional computation, based on the manipulation of numbers and symbols, humans principally employ uncertain or vague terms for the representation of knowledge and reasoning. In order to emulate this process of reasoning the development of a methodology for computing with words is essential [31]. Computing with words makes it possible to simulate human reasoning, enabling the manipulation and use of imprecise knowledge, and improving the tolerance and robustness of applications.

The development of computing with words poses us two principal problems: on one hand, the definition of a vocabulary which will establish the set of expressions that can be manipulated; on the other hand, the development of semantic models to interpret the expressions and to perform inferences on them. For example, in [4] a grammar is proposed that defines a broad set of expressions of interest for the field of fuzzy temporal reasoning and fuzzy temporal control, enabling the evaluation of expressions with a degree of complexity higher than usual (e.g., "the mean of high temperatures within the last ten minutes", "the minimum of recent high pressure values"). In [6] a classification of quantified sentences is defined that makes it possible to increase the expressive power of the underlying systems (e.g., fuzzy databases [3], data mining [22], information

* Authors wish to acknowledge the support of the Spanish Ministry of Science and Technology and the European Commission through grant TIC2000-0873.

J. Lawry, J. Shanahan, A. Ralescu (Eds.): Modelling with Words, LNAI 2873, pp. 1–25, 2003.
© Springer-Verlag Berlin Heidelberg 2003

retrieval [2]). In [9, 13, 16] semantic models are developed for the evaluation of fuzzy quantified sentences. In general, most of fuzzy logic applications are based to a greater or lesser degree on techniques aimed at making computing with words. Nevertheless, capacity of fuzzy systems is generally under-exploited. For example, in the paradigmatic case of fuzzy control very simple types of expressions and rules are usually stated.

In this work we propose two mechanisms based on the random set theory [1, 17, 24] which make it possible to define the extension of a set function to fuzzy sets. These mechanisms will enable the evaluation of operators more complex than the habitual ones (e.g., *m*ean, *l*ast, *a*ccumulated value, *t*he majority, ...). Thus, we are tackling the second of the objectives of computing with words: i.e., the development of semantic models for the operators of the vocabulary.

Within the literature regarding the fuzzy extension of functions, fuzzy cardinality measures and fuzzy quantifiers have been thoroughly studied [5, 9, 11, 16, 20, 21, 25, 26, 30]. Other works [11, 12, 27] present an approach that is more general, and in principle, valid for any type of operator. This is also our aim in the present work, where we describe general mechanisms that are highly related with the proposals developed in [9, 13] for the modelling of fuzzy quantifiers.

The paper also deals with the interpretation of the two extension mechanisms that are proposed. Although these extension mechanisms are not discussed from a formal perspective, their underlying interpretation is provided, together with a number of examples that make it possible to understand the differences and problems of the various approaches. We claim that the mechanisms that are presented are superior to possibilistic approaches [11, 27]. In this respect some examples of the poor behaviour of possibilistic models are given.

The paper is organized as follows. Firstly, we analyze some possibilistic proposals that are based on the extension principle. We then go on to explain two interpretations of fuzzy sets that are based on probability theory: random sets and what we call "trivalued random sets". In the last section we explain how these concepts can be used in the definition of mechanisms for extending functions.

1.1 Formulation of the Problem

Let $E = \{e_1, \ldots, e_n\}$ be the finite set of objects of the domain of the problem. By $\wp(E)$ we denote the crisp powerset of E and by $\widetilde{\wp}(E)$ the fuzzy powerset. Let us suppose that a list of attributes (or properties) $A_1, \ldots, A_u$ (e.g., height) can be measured on the elements of E. By Ω_j we will denote the range of the attribute A_j. Throughout this explanation we will assume $\Omega_j = \mathbb{R}$.

We use a special element θ in order to indicate the non-applicability of attributes. Abusing of notation, by $A_j : E \to \Omega_j \cup \{\theta\}$ we denote the function that associates to each $e \in E$ the value θ when the corresponding attribute A_j is not applicable to e, and its corresponding value $\omega_r \in \Omega_j$ otherwise.

We suppose that a set of labels (or terms) $LA_{A_j} = \{L_1, \ldots, L_{k_j}\}$ is associated to each attribute (e.g. small, medium, high) [27].

The set of all the labels that are relevant in the domain is denoted as $LA = \{L_1, \ldots, L_\eta\}$. The membership function of a fuzzy set X is denoted as usual by μ_X. If X is a crisp or trivalued set its membership function is denoted by χ_X.

Let $X \in \wp(E)$, $L_i \in LA_{A_j}$. By $L_i(X)$ we denote the fuzzy set on X whose membership funtion is

$$\mu_{L_i(X)}(e_i) = \begin{cases} 0 & A_j(e_i) = \theta \\ \mu_{L_i}(A_j(e_i)) & A_j(e_i) \neq \theta \end{cases}, e_i \in X$$

Our aim is to define models for the extension of a set function $f : \wp(E)^s \to \mathbb{R} \cup \{\theta\}$ to a fuzzy set function $\widetilde{f}$. We represent by θ all those cases where f is non applicable. We will not define at this point the image space of $\widetilde{f}$, since depending on which of the mechanisms is employed this can be a single value, a probability function, or a fuzzy set.

We now go on to give a number of examples of functions to which these mechanisms can be applied:

Example 1 (Arithmetic mean). Let A_j be an attribute of the elements of E. We can calculate the arithmetic mean of A_j for a set $X \in \wp(E)$ using the following function:[1]

$$f(X) = \begin{cases} \theta & |X| = 0 \\ \frac{1}{|X|} \sum_{e \in X} A_j(e) & otherwise \end{cases}$$

Example 2 (Last function). Let us suppose that an order has been defined on E ($e_1 < e_2 < \ldots < e_n$). The function *last* of the elements of a set $X \in \wp(E)$ can be defined as

$$f(X) = \begin{cases} \theta & |X| = 0 \\ \max(X) & otherwise \end{cases}$$

Example 3 (Semi-fuzzy quantifiers). In [13, 16] the concept of semi-fuzzy quantifier for the evaluation of fuzzy quantified sentences is defined. A semi-fuzzy quantifier is a function $Q : \wp(E)^s \to [0,1]$ that associates a degree of truth to a tuple of sets $(X_1, \ldots, X_s), X_i \in \wp(E)$. For example, for the evaluation of the sentence "about 80% or more of the X_1 are X_2" we can define the semi-fuzzy quantifier

$$\text{about80\%}\mathbf{or_more_ot_the}(X_1, X_2) = \begin{cases} S_{0.5,0.8}\left(\frac{|X_1 \cap X_2|}{|X_1|}\right) & X_1 \neq \varnothing \\ 1 & X_1 = \varnothing \end{cases}$$

where

$$S_{\alpha,\gamma}(x) = \begin{cases} 0 & x < \alpha \\ 2\left(\frac{(x-\alpha)}{(\gamma-\alpha)}\right)^2 & \alpha < x \leq \frac{\alpha+\gamma}{2} \\ 1 - 2\left(\frac{(x-\gamma)}{(\gamma-\alpha)}\right)^2 & \frac{\alpha+\gamma}{2} < x \leq \gamma \\ 1 & \gamma < x \end{cases}$$

[1] In these examples, we ignore the situation $\theta \in X$, in which it would be reasonable to define $f(X) = \theta$.

2 Analysis of Some Proposals

In this section a brief overview of some possibilistic proposals for the extension of functions that have been presented in the literature is made. All of them are based on defining a possibility distribution over the crisp representatives of fuzzy sets $X_1, \ldots, X_s$ in order to apply Zadeh's extension principle.

In this section we will often refer to fuzzy cardinality, which has attracted a great attention [11, 12, 20, 25, 26, 30] in the field of extension of functions.

Let $X \in \widetilde{\wp}(E)$, X can be represented on the basis of its α-cuts [27, 30] since:

$$\mu_X(e) = \sup\{\alpha \in [0,1] : e \in X_{\geq\alpha}\}, e \in E$$

where $X_{\geq\alpha} = \{e \in E : \mu_X(e) \geq \alpha\}$

This representation allows X to be interpreted as a possibility distribution over its α-cuts; i.e., the possibility for a given $Y \in \wp(E)$ to be a representative of fuzzy set X can be defined as: [2]; i.e.,

$$\pi_X(Y) = \begin{cases} \sup\{\alpha \in (0,1] : X_{\geq\alpha} = Y\} & \exists \alpha \in (0,1], X_{\geq\alpha} = Y \\ 0 & otherwise \end{cases} \quad Y \in \wp(E) \tag{1}$$

Let $f : \wp(E) \to \mathbb{R} \cup \{\theta\}$. The interpretation of X as a possibility distribution over its α-cuts enables us to apply Zadeh's extension principle [27] to f:

$$\mu_{f(X)}(r) = \begin{cases} \sup\{\alpha \in (0,1] : f(X_{\geq\alpha}) = r\} & \exists \alpha, f(X_{\geq\alpha}) = r \\ 0 & otherwise \end{cases} \tag{2}$$

For the case of functions of arity s ($f : \wp^s(E) \to \mathbb{R} \cup \{\theta\}$) the following possibilities can be considered:

- Using cuts of the same level:

$$\mu_{f(X_1,\ldots,X_s)}(r) = \begin{cases} \sup\left\{\alpha \in (0,1] : f\left((X_1)_{\geq\alpha}, \ldots, (X_s)_{\geq\alpha}\right) = r\right\} & \exists \alpha, f\left((X_1)_{\geq\alpha}, \ldots, (X_s)_{\geq\alpha}\right) = r \\ 0 & otherwise \end{cases} \tag{3}$$

- Using cuts of different levels:

$$\mu_{f(X_1,\ldots,X_s)}(r) = \begin{cases} \sup\left\{ \begin{array}{c} \min(\alpha_1,\ldots,\alpha_s), \alpha_i \in (0,1]: \\ f\left((X_1)_{\geq\alpha_1}, \ldots, (X_s)_{\geq\alpha_s}\right) = r \end{array} \right\} & \begin{array}{c} \exists \alpha_1, \ldots, \alpha_s, \\ f\left((X_1)_{\geq\alpha_1}, \ldots, (X_s)_{\geq\alpha_s}\right) = r \end{array} \\ 0 & otherwise \end{cases} \tag{4}$$

[2] This representation is questionable if the representation of X is compared with that of its fuzzy complement (defined, as usual, by $\mu_{\neg X}(e) = 1 - \mu_X(e)$). For example, for $X = \{1/e_1, 0.8/e_2\}$ the representatives of X are $L_1 = \{e_1\}$ with possibility 1, and $L_2 = \{e_1, e_2\}$ with possibility 0.8. But if the possibility of $\{e_1\}$ being a representative of X is 1, then the possibility of $\{e_2\}$ being a representative of the complement of X will also be 1. Thus, the representatives of the complement of X should be $L_1' = \{e_2\}$ with possibility 1, and $L_2' = \varnothing$ with possibility 0.8. However this is not a representation of a fuzzy set, and it does not coincide with the representation of $\widetilde{\neg} X = \{0/e_1, 0.2/e_2\}$.

Other schemes can be formulated by replacing the supremun by a *tconorm* and the minimum by a *tnorm* in expressions (3) and (4).

This mechanism is used in [30] to formulate the following fuzzy cardinality measure:

$$\mu_{card(X)}(i) = \sup\{\alpha \in (0,1] : f(X_{\geq\alpha}) = i\}, i \in \mathbb{N} \tag{5}$$

The theoretical behaviour of this definition is studied in [11], with the conclusion that it has some deficiencies (i.e., non-satisfaction of the property of additivity).

Another of the problems is the lack of continuity in results. Let us consider the following example, in which this mechanism is used in fuzzy temporal reasoning [4]:

Example 4. Suppose the evaluation of the sentence "the last temporal point at which the temperature was high occurred around half an hour ago". Let $E = \{e_1, e_2, e_3, e_4, e_5\}$ be the temporal referential with $e_1 < e_2 < \ldots < e_5$; and let

$$high(E) = \{1/e_1, 0.5/e_2, 0/e_3, 0.75/e_4, 1/e_5\}$$
$$ar_half_hour_ago(E) = \{1/e_1, 1/e_2, 0.5/e_3, 0/e_4, 0/e_5\}$$

be the fuzzy sets representing "high temperature" and the fuzzy temporal reference "around half an hour ago". Computing function "last" by means of expression (2) we obtain

$$last(high(E)) = \{0/e_1, 0/e_2, 0/e_3, 0/e_4, 1/e_5\}$$

and therefore the degree of possibility of the sentence is

$$\pi = \sup_{e \in E}\{\min(ar_half_hour_ago(E), last(high(E)))\} = 0$$

Nevertheless, when considering a case that is very similar to this one (in which only a small variation has been made in the membership function of the element e_5)

$$high'(E) = \{1/e_1, 0.5/e_2, 0/e_3, 0.75/e_4, 0.99/e_5\}$$

we obtain[3]

$$last(high'(E)) = \{1/e_1, 0/e_2, 0/e_3, 0/e_4, 0.99/e_5\}$$
$$\pi' = \sup_{e \in E}\left\{\min\left(ar_half_hour_ago(E), last\left(high(E)'\right)\right)\right\} = 1$$

[3] Temporal constraint "around half an hour ago" can be interpreted as a function $ar_half_hour_ago : E \rightarrow [0,1]$ such that $ar_half_hour_ago(e_1) = ar_half_hour_ago(e_2) = 1$, $ar_half_hour_ago(e_3) = 0.5$, $ar_half_hour_ago(e_i) = 0, i > 3$. In this way, by using Zadeh's extension principle we would obtain a result on the universe of truth values $[0,1]$: $\mu_{ar_half_hour_ago(high(E))} = \{1/0, 0/0.5, 0/1\}$, $\mu_{ar_half_hour_ago(high'(E))} = \{0.99/0, 0/0.5, 1/1\}$. Note that in the first case the solution is necessarily false. The problems are similar, apart from practical difficulties when dealing with type-II sets.

The lack of continuity shown by the previous example may be of critical importance in some practical applications, where the values of the variables always exhibit a lack of precision due to the tolerance of the sensors.

The behaviour of (5) is aimed to be improved by the FG_{Count} fuzzy cardinality [30]. It can be interpreted that this measure arises from applying Zadeh's extension principle on a new set of representatives that is made up of adding to the set of representatives of X those that are found between two α-cuts. Measure FG_{Count} has also been criticized in [11]. It is not difficult to find examples of non intuitive behaviour when this approach is used for extending other functions.

In [30] the fuzzy cardinality measure FE_{Count} is proposed. This measure is equivalent to the one suggested in [25], and can be formulated on the basis of the definition of equipotencies between fuzzy and crisp sets. As we explain in section (5.2), mechanisms that are based on trivalued cuts applied to the cardinality function also give rise to the FE_{Count} measure.

In [11] the following possibility assignment for representing a fuzzy set X is proposed:

$$\pi_X(Y) = \begin{cases} \inf\{\mu_X(e) : e \in Y\} & ker(X) \subseteq Y \subseteq supp(X) \\ 0 & otherwise \end{cases}, Y \in \wp(E) \qquad (6)$$

It is worthy noting that this representation exhibits the following behaviour: it is possible for a representative to include an element e_2 and not to include another element e_1, even if it is the case that the degree of satisfaction of X by e_2 is lower than the degree of satisfaction of X by e_1.

By applying the extension principle the following cardinality measure can be defined [11]

$$\mu_{card(X)}(i) = \sup\{\pi_X(S) : Ker(X) \subseteq S \subseteq supp(X) \wedge |S| = i\}, i \in \mathbb{N} \qquad (7)$$

The following example, which is similar to example 4, shows some continuity problems of this approach.

Example 5. Let us consider that we wish to evaluate the sentence "the last temporal point of high temperature", for E, $high$ and $high'$ defined as in the previous example.
By calculating the function "last" by means of the extension principle applied to (6) we obtain

$$last(high(E)) = \{0/e_1, 0/e_2, 0/e_3, 0/e_4, 1/e_5\}$$

whilst for the set $high'$ the result is

$$last(high'(E)) = \{1/e_1, 0.5/e_2, 0/e_3, 0.75/e_4, 0.99/e_5\}$$

By using examples, some non-plausible behaviours for some possibilistic extension mechanisms have been presented in this section. The following sections are devoted to presenting the preliminaries, interpretations and formal definitions of two random-set based approaches for the fuzzy extension of functions.

3 Random Set Based Interpretation of Fuzzy Sets

In this section we describe the two interpretations of fuzzy sets that we use for defining function extension mechanisms. Both are based on the notion of random sets [1, 17, 24], the first applied to crisp sets, and the second to trivalued sets.

3.1 Random Bivalued Set Based Interpretation of Fuzzy Sets

This section describes how the membership function of a linguistic term can be interpreted within the theory of random sets and voting models [1, 17, 24].

Once a random set has been associated to each one of the linguistic terms of the domain of our problem, we use these sets to induce a probability function on the referential E. This probability is used to calculate a probability function on the image space of f.

Under the random set interpretation, a fuzzy set (or possibility distribution) is the result of carrying out a random experiment. Let us consider the following example:

Example 6. Four individuals (henceforth voters) $V = \{v_1, v_2, v_3, v_4\}$ are asked what the height values for a male adult to be considered tall are. Let us represent the choice of an individual v_i by $\Gamma(v_i)$, and let us suppose that the results of the experiment are as follows:

$$\Gamma(v_1) = [170, \infty), \Gamma(v_2) = [175, \infty)$$
$$\Gamma(v_3) = [175, \infty), \Gamma(v_4) = [180, \infty)$$

In this manner, each individual is supplying different information on the meaning of the concept "tall".

The following definition [17] enables us to formalize the mentioned experiment:

Definition 1 (Random set). *Let $(V, \sigma(V), P)$ be a finite probability space, Ω an arbitrary non-empty referential set, and $\Gamma : V \to \wp(\Omega)$ a set valued mapping. The pair (P, Γ) is called a random set. The sets $\Gamma(v)$, $v \in V$, are called the focal elements of (P, Γ).*

The probability measure P defines a probability on the focal elements of the random set.

Definition 2. *Let (P, Γ) be a random set. The probability associated to a focal element $\Gamma_i \in \wp(\Omega)$ is*

$$m(\Gamma_i) = \sum_{v \in V : \Gamma(v) = \Gamma_i} P(\{v\})$$

This definition is equivalent to the *mass assignment function* that is used in the mass assignment theory [1], where a *mass assignment function* is any function $m : 2^{\Omega} \rightarrow [0,1]$ such that $\sum_{S \subseteq \Omega} m(S) = 1$. In turn, *mass assignment functions* are a weak form of Dempster Shafer's definition [23], in which the probability of the empty set can be greater than 0.

Example 7. (continuation of example 6). Consider that $P(v_i) = 0.25$ (i.e., we assign an equitative measurement to all individuals). The focal elements of our experiment are denoted by

$$\Gamma_1 = [170, \infty), \Gamma_2 = [175, \infty), \Gamma_3 = [180, \infty)$$

hence

$$P(\Gamma_1) = 0.25, P(\Gamma_2) = 0.5, P(\Gamma_3) = 0.25$$

On the basis of random sets we introduce the definition of contour function, which can also be found in [17].

Definition 3. *Let (P, Γ) with $\Gamma : V \rightarrow \wp(\Omega)$ be a random set, then the function*

$$\begin{aligned} &\mu_{\Gamma} : \Omega \rightarrow [0,1] \\ &\omega \in \Omega \mapsto P(\{v \in V : \omega \in \Gamma(v)\}) \end{aligned}$$

is called the contour function of (P, Γ).

The interpretation of fuzzy sets based on random sets is based on identifying the membership function of a fuzzy set X associated to a property A with the contour function of the random set resulting from the evaluation of A for a set of individuals:

$$\mu_X(\omega) = \mu_{\Gamma}(\omega), \omega \in \Omega$$

Example 8. (continuation of example 6) The contour function of the fuzzy set *high* is

$$\begin{aligned} \mu_{high}(\omega) = \mu_{\Gamma}(\omega) = 0.25, \omega \in [170, 175) \\ \mu_{high}(\omega) = \mu_{\Gamma}(\omega) = 0.5, \omega \in [175, 180) \\ \mu_{high}(\omega) = \mu_{\Gamma}(\omega) = 0.25, \omega \in [180, \infty) \end{aligned}$$

It should be noted that this approach does not assume that sets $\Gamma(v)$ are consonant; i.e., that $\Gamma(v_i) \subseteq \Gamma(v_j)$ or $\Gamma(v_j) \subseteq \Gamma(v_i)$ for any $v_i, v_j \in V$.

The situation of consonance is important as it is fairly reasonable to consider that the terms associated to the linguistic variables arise for consonant votes. In our example, the selection of ω as a representative of tall conditions the choice of all those $\omega' \geq \omega$, due to which the selections of individuals are consonant.

In this case, function μ_{Γ} allows the recovery of the focal elements of (P, Γ) and the probability mass function that is associated to them. Let us suppose that $\mu_{\Gamma}(\omega_1) \leq \mu_{\Gamma}(\omega_2) \leq \ldots \leq \mu_{\Gamma}(\omega_n)$, and let us denote $\Gamma_i = \{\omega_i, \ldots, \omega_n\}, i =$

$1, \ldots, n$. It holds that Γ_i is a focal element iff $\mu_\Gamma(\omega_{i-1}) < \mu_\Gamma(\omega_i)$. Moreover, the probability mass that is associated to Γ_i is [1, 12]

$$m(\Gamma_i) = \mu_\Gamma(\omega_i) - \mu_\Gamma(\omega_{i-1}), m(\varnothing) = 1 - \mu_\Gamma(\omega_n)$$

where we assume $\mu_\Gamma(\omega_0) = 0$.

Example 9. Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_5\}$ be a set of individuals for which the set $X \in \widetilde{\wp}(E)$

$$X = \{0.8/\omega_1, 1/\omega_2, 0.5/\omega_3, 0.5/\omega_4, 0.2/\omega_5\}$$

represents the fulfilment of the label "tall". It is reasonable for X to arise on the basis of a consonant voting. The arrangement of the referential on the basis of their height is $\omega_2 > \omega_1 > \omega_3 = \omega_4 > \omega_5$. The focal elements and their associated probability masses are

$$\begin{aligned}
&\Gamma_1 = \{\omega_2\}, m(\Gamma_1) = \mu_X(\omega_2) - \mu_X(\omega_1) = 0.2\\
&\Gamma_2 = \{\omega_1, \omega_2\}, m(\Gamma_2) = \mu_X(\omega_1) - \mu_X(\omega_3) = 0.3\\
&\Gamma_3 = \{\omega_1, \omega_2, \omega_3, \omega_4\}, m(\Gamma_3) = \mu_X(\omega_3) - \mu_X(\omega_5) = 0.3\\
&\Gamma_4 = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5\}, m(\Gamma_4) = \mu_X(\omega_5) = 0.2
\end{aligned}$$

In the case the number of voters tends towards infinity it can be assumed the contour function (associated to a linguistic term) resulting from the random experiment is given by a continuous function. It is also important to highlight that fuzzy sets associated to linguistic terms are normalized.

In the continuous case we can interpret that for each label $L_i \in LA$, each possible voter v has an associated level of specificity α. Also, a uniform probability function is defined on the interval $[0, 1]$ of levels of specificity:

Definition 4. *Let $L_i \in LA$ be a label associated to a linguistic term defined on a universe Ω, and P be an uniform probability function on $[0, 1]$; i.e., $P(\alpha) = 1, \alpha \in [0, 1]$. On the basis of L_i we define the function*

$$\begin{aligned}
\Gamma^{L_i} &: [0, 1] \to \wp(\Omega)\\
&\alpha \in [0, 1] \to (L_i)_{\geq \alpha}
\end{aligned}$$

The random set associated with label L_i is denoted by $\left(P, \Gamma^{L_i}\right)$

Note that:

$$\begin{aligned}
\mu_{\Gamma^{L_i}}(\omega) &= \int_0^1 \chi_{\Gamma^{L_i}(\alpha)}(\omega)\, d\alpha, \omega \in \Omega\\
&= \int_0^1 \chi_{(L_i)_{\geq \alpha}}(\omega)\, d\alpha, \omega \in \Omega
\end{aligned}$$

We do not impose any restriction on the levels of specificity (or α-cuts) that individuals use for different labels. Let $L_1, \ldots, L_\eta$ be the set of labels in our

domain. We suppose that a tuple $(\alpha_1, \ldots, \alpha_\eta) \in [0,1]^\eta$ of levels is associated to each voter v; such level α_i is associated to the corresponding label $L_i, 1 \leq i \leq \eta$. This allows us to work with $[0,1]^\eta$ instead of with the set of voters V. Later on, we will explain how we can use probability functions for establishing relations between the cut levels that individuals use for each label.

We now define the constraint induced by a label $L_i \in LA$ in the subset of the referential $X \in \wp(E)$.

Definition 5. *Let $L_i \in LA$, $X \in \wp(E)$. $\tilde{e}$ define the constraint of level $\alpha, \alpha \in [0,1]$, of X by L_i as*

$$R_X^{L_i}(\alpha) = \begin{cases} \left\{x \in X : A_j(x) \in \Gamma^{L_i}(\alpha)\right\} & \alpha > 0 \\ X & \alpha = 0 \end{cases}$$

Note that

$$(L_i(X))_{\geq\alpha} = R_X^{L_i}(\alpha)$$

Definition 6. *Let $L_i \in LA_{A_j}$, $X \in \wp(E)$. We define the constraint of X by L_i as*

$$\mu_{R_X^{L_i}}(e) = \int_0^1 \chi_{R_X^{L_i}(\alpha)}(e)\, d\alpha, e \in E$$

Note that

$$\mu_{R_X^{L_i}}(e) = \mu_{L_i}(e), e \in E$$

We have used a continuous probability function on $\wp(\Omega)$ in order to represent linguistic labels. An advantage of this formulation is that it helps to explains why the use of normalization methods [18] in models for extending functions is not very suitable. This point will be dealt with briefly at a later stage.

In definitions (5) and (6) a single label has been considered. Although it will not be considered in the present work, it is natural to extend them to *"formulas over labels"*, i.e., to use logical expressions that combine more than one label. In this way it will be possible to formulate methods for the evaluation of quantified sentences and for the extension of functions in a "purely" probabilistic way. A first approximation in the field of fuzzy quantification has been made in [8].

3.2 Random Trivalued Set Based Interpretation of Fuzzy Sets

In this section we explain how it is possible to interpret the membership function of a linguistic variable as a probability function on trivalued sets. The interpretation is highly related with the approach used in [13] for formulating one of the best models for the evaluation of fuzzy quantified sentences.

Trivalued sets imply a trivalued evaluation of the referential set. We denote a trivalued set associated to a property A by a pair $X = \left(X^{\min}, X^{\max}\right)$,

where $X^{\min}$ represents the elements certainly fulfils A, and $X^{\max} \backslash X^{\min}$ represents the elements for which fulfilment of A is uncertain. The membership function of a trivalued set X is defined as:

$$\chi_X(\omega) = \begin{cases} 0 & \omega \notin X^{\max} \\ \frac{1}{2} & \omega \in X^{\max} \backslash X^{\min} \\ 1 & \omega \in X^{\min} \end{cases}$$

We can represent X using fuzzy notation; e.g., $X = \left\{1/\omega_1, \frac{1}{2}/\omega_2, 1/\omega_3, 0/\omega_4\right\}$.

Following the notation used in [13] we denote the trivalued powerset of Ω by $\overset{\cup}{\wp}(\Omega)$.

Using similar arguments to the ones used in section (3.1), we can interpret the membership function of linguistic labels on the basis of a random experiment in which individuals characterize the elements of the referential by means of trivalued sets.

Example 10. Let $v \in V$. Instead of categorizing the referential crisply, we could consider that v classifies the elements of Ω with regard to the property "height" by way of a trivalued set. Thus $\overset{\cup}{\Gamma}(v)$ could be given by

$$\chi_{\overset{\cup}{\Gamma}(v)} = \begin{cases} 0 & \omega \leq 170 \\ \frac{1}{2} & 170 < \omega < 180 \\ 1 & 180 \leq \omega \end{cases}$$

We now go on to introduce the definition of *trivalued random set.* This definition is obtained on the basis of definition (1), by replacing function $\Gamma : V \to \wp(\Omega)$ with a function $\overset{\cup}{\Gamma} : V \to \overset{\cup}{\wp}(\Omega)$ whose range is on the set of trivalued parts of Ω.

Definition 7. *Let $(V, \sigma(V), P)$ be a finite probability space, Ω an arbitrary non-empty referential set, and $\overset{\cup}{\Gamma} : V \to \overset{\cup}{\wp}(\Omega)$ a trivalued set valued mapping. The pair $\left(P, \overset{\cup}{\Gamma}\right)$ is called trivalued random set. The sets $\overset{\cup}{\Gamma}(v)$, $v \in V$, are the focal elements of (P, Γ).*

Definition 8. *Let $\left(P, \overset{\cup}{\Gamma}\right)$ be a trivalued random set. The probability associated to a focal element $\overset{\cup}{\Gamma}_i \in \overset{\cup}{\wp}(\Omega)$ is*

$$m\left(\overset{\cup}{\Gamma}_i\right) = \sum_{v \in V, \overset{\cup}{\Gamma}(v) = \overset{\cup}{\Gamma}_i} P(v)$$

and is called the mass assignment related to $\overset{\cup}{\Gamma}_i$.

Definition 9. *Let* $\left(P, \overset{\cup}{\Gamma}\right)$ *a trivalued random set. The function*

$$\mu_{\overset{\cup}{\Gamma}} : \Omega \to [0,1]$$
$$\omega \in \Omega \mapsto P\left(\left\{v \in V : \chi_{\overset{\cup}{\Gamma}(v)}(\omega) = 1\right\}\right) + \frac{1}{2}P\left(\left\{v \in V : \chi_{\overset{\cup}{\Gamma}(v)}(\omega) = \frac{1}{2}\right\}\right)$$

is called the contour function of $\left(P, \overset{\cup}{\Gamma}\right)$.

In a similar manner as for random sets we interpret that

$$\mu_X(\omega) = \mu_{\overset{\cup}{\Gamma}}(\omega), \omega \in \Omega$$

Regarding random sets, we can propose the recovery of a trivalued random set on the basis of their contour function. The key concept, in a similar manner as the consonance of bivalued random sets, is that of consonance with respect to specificity.

Definition 10 (Specificity in trivalued sets). *We say that a trivalued set* $X = \left(X^{\min}, X^{\max}\right)$ *is less or equaly specific than another one* $Y = \left(Y^{\min}, Y^{\max}\right)$ *(*$X \preceq Y$*) if and only if* $X^{\min} \subseteq Y^{\min}$ *and* $X^{\max} \supseteq Y^{\max}$*. If* $X \preceq Y$ *and* $X \neq Y$ *we will say that* X *is strcitly less specific than* Y *(*$X \prec Y$*).*

Example 11. For trivalued sets $X = \left\{1/\omega_1, 1/\omega_2, \frac{1}{2}/\omega_3, \frac{1}{2}/\omega_4, \frac{1}{2}/\omega_5, 0/\omega_6\right\}$ and $Y = \left\{1/\omega_1, 1/\omega_2, 1/\omega_3, \frac{1}{2}/\omega_4, 0/\omega_5, 0/\omega_6\right\}$. It holds that $X \prec Y$.

On the basis of the previous example, we introduce the following definition.

Definition 11. *We say that a trivalued random set is consonant with respect to specificity if and only if for each* $v_i, v_j \in V$ *either* $\overset{\cup}{\Gamma}(v_i) \preceq \overset{\cup}{\Gamma}(v_j)$ *or* $\overset{\cup}{\Gamma}(v_j) \preceq \overset{\cup}{\Gamma}(v_i)$

For the case of consonant with respect to specificity trivalued random sets $\mu_{\overset{\cup}{\Gamma}}$ enables the recovery of the focal elements of $\left(P, \overset{\cup}{\Gamma}\right)$ and the probability mass function which is associated to them.

Proposition 1. *Let* X *be a fuzzy set. There exists a single decomposition* $\overset{\cup}{\Gamma}_1 : m_1, \overset{\cup}{\Gamma}_2 : m_2, \ldots, \overset{\cup}{\Gamma}_r : m_r$ *(*$\overset{\cup}{\Gamma}_1 \prec \ldots \prec \overset{\cup}{\Gamma}_r$*) of* X *into consonant with respect to specificity trivalued sets.*

Definition 12 (Cautiousness cut of level γ**).** *[13] Let* μ_X *be the contour function associated to the trivalued random set* $\overset{\cup}{\Gamma}$*. The cautiousness cut of level*

$\gamma \in [0,1]$, *which is denoted by* X^{γ}, *is a trivalued set with membership function*

$$\chi_{X^{\gamma}}(\omega) = \begin{cases} 0 & \mu_X(\omega) \leq \frac{1}{2} - \frac{1}{2}\gamma \\ \frac{1}{2}\ \frac{1}{2} - \frac{1}{2}\gamma < \mu_X(\omega) < \frac{1}{2} + \frac{1}{2}\gamma \\ 1 & \mu_X(\omega) \geq \frac{1}{2} + \frac{1}{2}\gamma \end{cases}, \quad \gamma \in (0,1] \tag{8}$$

$$\chi_{X^{\gamma}}(\omega) = \begin{cases} 0\ \mu_X(\omega) < \frac{1}{2} \\ \frac{1}{2}\ \mu_X(\omega) = \frac{1}{2} \\ 1\ \mu_X(\omega) > \frac{1}{2} \end{cases}, \quad \gamma = 0$$

For the maximum level of cautiousness ($\gamma = 1$) only crisp elements (with a truth value of 0 or 1) are considered to be precise. For a minimum level of cautiousness ($\gamma = 0$) all elements e such that $\mu_X(e) \neq 0.5$ are considered to be precise.

Definition 13. *Let* X *be a fuzzy set constructed on the basis of a trivalued representation. By* $T(X)$ *we define the set*

$$T(X) = \left(\begin{array}{c} \left\{ 2\mu_X(\omega) - 1 : \mu_X(\omega) > \frac{1}{2} \right\} \\ \cup \left\{ 1 - 2\mu_X(\omega) : \mu_X(\omega) < \frac{1}{2} \right\} \end{array} \right) \setminus \{1, 0\}$$

Proposition 2. *Let* X *be a fuzzy set* $t_1 > t_2 > \ldots > t_m$ *the ordered elements of* $T(X)$, $t_0 = 1, t_{m+1} = 0$. *The trivalued representation of* X *is*

$$\overset{\cup}{\Gamma}_i = X^{t_i}, i = 0, \ldots, m$$
$$m_i = t_i - t_{i+1}$$

Example 12. For

$$X = \{0/\omega_1, 0.2/\omega_2, 0.25/\omega_3, 0.6/\omega_4, 0.6/\omega_5, 1/\omega_6, 1/\omega_7\}$$

we have $T(X) = \{0.2, 0.5, 0.6\}$. Thus $t_0 = 1, t_1 = 0.6, t_2 = 0.5, t_3 = 0.2, t_4 = 0$, and then

$$\Gamma_0 = \left\{ 0/\omega_1, \frac{1}{2}/\omega_2, \frac{1}{2}/\omega_3, \frac{1}{2}/\omega_4, \frac{1}{2}/\omega_5, 1/\omega_6, 1/\omega_7 \right\}, m(\Gamma_1) = 0.4$$
$$\Gamma_1 = \left\{ 0/\omega_1, 0/\omega_2, \frac{1}{2}/\omega_3, \frac{1}{2}/\omega_4, \frac{1}{2}/\omega_5, 1/\omega_6, 1/\omega_7 \right\}, m(\Gamma_2) = 0.1$$
$$\Gamma_2 = \left\{ 0/\omega_1, 0/\omega_2, 0/\omega_3, \frac{1}{2}/\omega_4, \frac{1}{2}/\omega_5, 1/\omega_6, 1/\omega_7 \right\}, m(\Gamma_3) = 0.3$$
$$\Gamma_3 = \{0/\omega_1, 0/\omega_2, 0/\omega_3, 1/\omega_4, 1/\omega_5, 1/\omega_6, 1/\omega_7\}, m(\Gamma_4) = 0.2$$

It can be interpreted in the continuous case, similarly as it happens with bivalued random sets, that for each label $L_i \in LA$, each possible voter v has an associated cautiousness level γ. This enables us to suppose that a uniform probability function exists on $[0,1]$, and to associate the cautiousness cut $(L_i)^{\gamma}$ to each γ. We introduce the following definition:

Definition 14. *Let $L_i \in LA$ be a label associated to a linguistic property defined on a universe Ω, and P a probability function over $[0,1]$ ($P(\gamma) = 1, \gamma \in [0,1]$). On the basis of L_i we define*

$$\begin{aligned} \overset{\cup}{\Gamma}^{L_i} &: [0,1] \to \overset{\cup}{\wp}(\Omega) \\ &\gamma \in [0,1] \to (L_i)^{\gamma} \end{aligned}$$

The trivalued random set that is associated to the label L_i is denoted by $\left(P, \overset{\cup}{\Gamma}^{L_i}\right)$.

Definition 15. *Let $L_i \in LA_{A_j}$, $X \in \wp(E)$. We define the cautiouness constraint of level $\gamma, \gamma \in [0,1]$ on X by L_i ($\overset{\cup}{R}{}^{L_i}_X(\gamma)$) as follows:*

$$\chi_{\overset{\cup}{R}{}^{L_i}_X(\gamma)}(e) = \chi_{\overset{\cup}{\Gamma}{}^{L_i}(\gamma)}(A_j(e)), e \in E$$

Note that

$$(L_i(X))^{\gamma} = \overset{\cup}{R}{}^{L_i}_X(\gamma)$$

Definition 16. *Let $L_i \in LA_{A_j}$, $X \in \wp(E)$. We define the constraint on X by L_i as follows:*

$$\mu_{\overset{\cup}{R}{}^{L_i}_X}(e) = \int_0^1 \chi_{\overset{\cup}{R}{}^{L_i}_X(\gamma)}(e)\, d\gamma, e \in E$$

Note that

$$\mu_{\overset{\cup}{R}{}^{L_i}_X}(e) = \mu_{L_i}(e), e \in E$$

3.3 Some Considerations

The normalization problem. Let $X \in \wp(E), L_i \in LA$. For bivalued cuts and non-normalized sets ($\mu_{L_i}(x) < 1, \forall x \in X$) then $m_{L_i(X)}(\varnothing) > 0$. Since $f(\varnothing)$ may not be defined (this happens for *mean*, for some quantifiers, ...) this is one of the main sources of non applicability of f. This problem [4] is generally solved in the literature by using some normalization technique [18, 19]. Although this may be a good choice for possibility distributions, normalization procedures may noticeably distort the information in fuzzy sets. For example, after normalization $R_X^{tall} = \left\{\frac{1}{3}/e_1, \ldots, \frac{1}{3}/e_4\right\}$ becomes $R_X^{tall\prime} = \{1/e_1, \ldots, 1/e_4\}$. Therefore, this situation would be treated exactly as if we initially had $R_X^{tall} = \{1/e_1, \ldots, 1/e_4\}$. In order to avoid non plausible behaviours that may be derived from this information distortion, in this paper we use a more general way of representing the non applicability of a function: a special element θ is defined for this purpose.

[4] A similar problem exists for trivalued cuts when $\mu_{L_i}(x) < 0.5\ \forall x$.

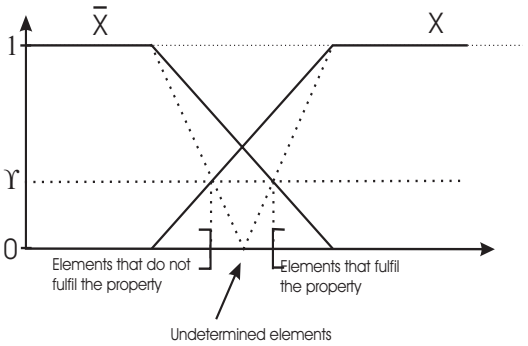


Fig. 1. Symmetry in trivalued cuts

Differences beetween bivalued and three-valued cuts. On the basis of definition (8) it can be seen that (see figure 1)

$$\overset{\cup}{\neg} X^{\gamma} = (\widetilde{\neg} X)^{\gamma} \tag{9}$$

where $\widetilde{\neg}$ and $\overset{\cup}{\neg}$ respectively denote the fuzzy and trivalued negation defined as $\widetilde{\neg}(x) = \overset{\cup}{\neg}(x) = 1 - x$.

For bivalued cuts it can only be guaranteed that

$$\neg X_{\geq \alpha} = (\widetilde{\neg} X_{>1-\alpha})$$

This is the principal difference between the use of trivalued and bivalued cuts. For trivalued cuts the use of the same cut level for one property and its negation is coherent; which does not hold for bivalued cuts.

(9) makes it possible to guarantee certain highly suitable behaviours in the evaluation of fuzzy quantified sentences [13, 14, 15].

4 Voting Profiles

In this section we explain how it is possible to establish reasonable relations between the levels of specificity (or cautiousness) used by the voters for labels $L_1 \ldots, L_s$ by means of probability functions. These probability functions are referred to as *voting profiles*[5].

– **Maximum dependence profile.** This scheme arises from supposing that voters choose focal elements of the same level for all labels. The interpretation of this profile is that voters are coherent in their decisions on different

[5] We will suppose that all the labels $L_1, \ldots, L_s$ are different because it is difficult to justify the use of different levels of specificity (or cautiousness) for a same label (e.g, in the evaluation of the quantified sentence "all tall people are tall"). In [8] a first attemp for avoiding this problem is developed.

labels; i.e., that voters are equally specific (or cautious) in all their decisions. The probability density P^{MD} that is associated to this case is a Dirac multidimensional function for which it is supposed that all marginal densities are uniform; i.e.,

$$P^{MD}(\alpha_1, \ldots \alpha_s) = \begin{cases} \uparrow : \alpha_1 = \ldots = \alpha_s \\ 0 : \quad otherwise \end{cases}, \alpha_i \in [0,1]$$

For $s = 2$, we define:

$$h(x_1, x_2) = \begin{cases} \frac{1}{1-a^2} : |x_1 - x_2| \leq 1 - a \\ 0 \quad : \quad otherwise \end{cases} \tag{10}$$

then

$$P^{MD}(\alpha_1, \alpha_2) = \lim_{a \to 1^-} h(\alpha_1, \alpha_2)$$

– **Independence profile.** This scheme arises from supposing that the levels of the focal elements that are chosen by voters are independent. Under this scheme, the specificity (or cautiousness) of a voter for one property does not condition his/her specificity (or cautiousness) for other properties.
The probability density P^I associated to this case is:

$$P^I(\alpha_1, \ldots \alpha_s) = 1, \alpha_i \in [0,1]$$

– **Other approaches.** It is possible to use other probability functions in order to condition the levels used by voters for the different labels. One possibility is to define an interpolation between the previously mentioned functions. Let $\lambda \in [0,1]$. We can define

$$P^{AD}_{\lambda}(\alpha_1, \ldots \alpha_s) = \lambda P^{MD}(\alpha_1, \ldots \alpha_s) + (1 - \lambda) P^I(\alpha_1, \ldots \alpha_s)$$

5 Extension Mechanisms

Let E be the finite referential of the domain of the problem, $f : \wp^s(E) \to \mathbb{R} \cup \{\theta\}$, $L_1, \ldots, L_s$ a set of labels, $Y_1, \ldots, Y_s \in \wp(E)$ and $X_1 = L_1(Y_1), \ldots, X_s = L_s(Y_s)$. In this section we formulate two mechanisms that allow to apply f to the fuzzy sets $X_1, \ldots, X_s$.

5.1 Extension of Function on the Basis of Bivalued Random Sets

In this section we explain a mechanism that is very similar to the one formulated in [7, 9, 10] for the problem of fuzzy quantification, but here generalized to the extension of the arbitrary function f.

On the basis of f we define the function $g^f : \wp^s(E) \times \mathbb{R} \cup \{\theta\} \to \{0,1\}$ as

$$g^f(X_1, \ldots, X_s)(r) = \begin{cases} 0 \; f(X_1, \ldots, X_s) \neq r \\ 1 \; f(X_1, \ldots, X_s) = r \end{cases}, r \in \mathbb{R} \cup \{\theta\}$$

and the probability of r being the result of f on $X_1, \ldots, X_s$ for the profile P^β (see section (4)) is defined as

$$\begin{aligned} & P\left(r|f, X_1, \ldots, X_s\right) \\ &= \int_0^1 \ldots \int_0^1 g^f \left(R_{Y_1}^{L_1}\left(\alpha_1\right), \ldots, R_{Y_s}^{L_s}\left(\alpha_s\right)\right)(r)\, P^\beta\left(\alpha_1, \ldots, \alpha_s\right) d\alpha_1 \ldots d\alpha_s \\ &= \int_0^1 \ldots \int_0^1 g^f \left(\left(X_1\right)_{\geq \alpha_1}, \ldots, \left(X_s\right)_{\geq \alpha_s}\right)(r)\, P^\beta\left(\alpha_1, \ldots, \alpha_s\right) d\alpha_1 \ldots d\alpha_s \end{aligned} \tag{11}$$

Let $H = \{r_1, \ldots, r_k\} \in \mathbb{R} \cup \{\theta\}$ be those values that have a probability greater than 0, and $p_1, \ldots, p_k$ their associated probabilities[6].

The mean of the evaluation of the function can be calculated as

- If $P\left(\theta\right) = 0$ then

$$\overline{H} = \sum_{r_i \in H} r_i p_i$$

- If $P\left(\theta\right) \neq 0$. We can calculate the mean by ignoring θ

$$\overline{H} = \frac{1}{1 - P\left(\theta\right)} \sum_{r_i \in H \setminus \theta} r_i p_i$$

  with which the mean of the extension of f will be $\overline{H}$ with probability $1 - P\left(\theta\right)$ and "not applicable" with probability $P\left(\theta\right)$.

In [12] a similar definition for unary functions is proposed, and an interval for the expected value of the extension of a function is defined. This interval is calculated on the basis of the representatives that are defined in the expression (6), which we do not consider to be suitable.

Here we do not take into account the recovery of a fuzzy set on the basis of the probability function, since the result of the extension of the function is not a consonant random set.

We can combine the result of the expression (11) with a label L defined on $\mathbb{R}$ by calculating the probability of the fuzzy event L [29].

Examples

Example 13. Let us consider the evaluation of the sentence "the last temporal point at which the temperature was high occurred around half an hour ago". Let $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$ be the temporal referential on which we are working, $e_1 < e_2 < \ldots < e_7$; and let

$$\begin{aligned} high\left(E\right) &= \{0.7/e_1, 0.9/e_2, 0.2/e_3, 0/e_4, 1/e_5, 0.9/e_6, 0.7/e_7\} \\ ar_half_hour_ago\left(E\right) &= \{0.3/e_1, 0.7/e_2, 1/e_3, 1/e_4, 1/e_5, 0.7/e_6, 0.3/e_7\} \end{aligned}$$

[6] It should be kept in mind that E is finite, and therefore the sequence $r_1, \ldots, r_k$ is also finite too.

Table 1. Representatives of $high\,(E)$ and $ar_half_hour_ago\,(E)$ in example 13

	$(high\,(E))_{\alpha_1}$		$(ar_half_hour_ago\,(E))_{\alpha_2}$
$\alpha_1 \in [1, 0.9)$	$\{e_5\}$	$\alpha_2 \in [1, 0.7)$	$\{e_3, e_4, e_5\}$
$\alpha_1 \in [0.9, 0.7)$	$\{e_2, e_5, e_6\}$	$\alpha_2 \in [0.7, 0.3)$	$\{e_2, e_3, e_4, e_5, e_6\}$
$\alpha_1 \in [0.7, 0.2)$	$\{e_1, e_2, e_5, e_6, e_7\}$	$\alpha_2 \in [0.3, 0)$	$\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$
$\alpha_1 \in [0.2, 0)$	$\{e_1, e_2, e_3, e_5, e_6, e_7\}$		

be the sets "high temperatures" and "half an hour ago". In the crisp case we could evaluate the sentence by formulating the function

$$f\,(X_1, X_2) = \begin{cases} max\,(X_1 \cap X_2) & X_1 \cap X_2 \neq \varnothing \\ \theta & X_1 \cap X_2 = \varnothing \end{cases} \quad , X_1, X_2 \in \wp\,(E)$$

where X_1 represents the "high temperatures" and X_2 "around half an hour ago". We define $g^f : \wp\,(E)^2 \to \{0, 1\}$ as

$$g^f\,(X_1, X_2)\,(r) = \begin{cases} 0 & f\,(X_1, X_2) \neq r \\ 1 & f\,(X_1, X_2) = r \end{cases} , r \in \mathbb{R} \cup \{\theta\}$$

We use the independence profile (see section (4)). The representatives of $high(E)$ and $ar_half_hour_ago(E)$ are described in table 1, and table 2 describes the crossed possibilities between $high(E)$ and $ar_half_hour_ago(E)$. The result is

$$P\,(e_7 | f, high\,(E), ar_half_hour_ago\,(E)) = 0.21$$
$$P\,(e_6 | f, high\,(E), ar_half_hour_ago\,(E)) = 0.42$$
$$P\,(e_5 | f, high\,(E), ar_half_hour_ago\,(E)) = 0.37$$

Example 14. Let us consider the quantified sentence "almost all tall women are blonde", where the semi-fuzzy quantifier **almost_all**, and the fuzzy sets *tall* and *blonde* take the following values:

$$tall\,(E) = \{0.8/e_1, 0.9/e_2, 1/e_3, 0.2/e_4\}$$
$$blonde\,(E) = \{1/e_1, 0.8/e_2, 0.3/e_3, 0.1/e_4\}$$
$$\textbf{almost_all}\,(X_1, X_2) = \begin{cases} \max\left(2\left(\frac{|X_1 \cap X_2|}{|X_1|}\right) - 1, 0\right) & X_1 \neq \varnothing \\ 1 & X_1 = \varnothing \end{cases}$$

Table 2. Table of crossed possibilities between $high\,(E)$ and $ar_half_hour_ago\,(E)$ in example 13

$f\left((high\,(E))_{\alpha_1}, (ar_half_hour_ago\,(E))_{\alpha_2}\right)$	$\alpha_2 \in [1, 0.7)$	$\alpha_2 \in [0.7, 0.3)$	$\alpha_2 \in [0.3, 0)$
$\alpha_1 \in [1, 0.9)$	$e_5 : 0.03$	$e_5 : 0.04$	$e_5 : 0.03$
$\alpha_1 \in [0.9, 0.7)$	$e_5 : 0.06$	$e_6 : 0.08$	$e_6 : 0.06$
$\alpha_1 \in [0.7, 0.2)$	$e_5 : 0.15$	$e_6 : 0.20$	$e_7 : 0.15$
$\alpha_1 \in [0.2, 0)$	$e_5 : 0.06$	$e_6 : 0.08$	$e_7 : 0.06$

Table 3. Representatives of the fuzzy sets $tall\,(E)$ and $blonde\,(E)$ in example 14

	$(tall\,(E))_{\alpha_1}$		$(blonde\,(E))_{\alpha_2}$
$\alpha_1 \in [1, 0.9)$	$\{e_3\}$	$\alpha_2 \in [1, 0.8)$	$\{e_1\}$
$\alpha_1 \in [0.9, 0.8)$	$\{e_2, e_3\}$	$\alpha_2 \in [0.8, 0.3)$	$\{e_1, e_2\}$
$\alpha_1 \in [0.8, 0.2)$	$\{e_1, e_2, e_3\}$	$\alpha_2 \in [0.3, 0.1)$	$\{e_1, e_2, e_3\}$
$\alpha_1 \in [0.2, 0)$	$\{e_1, e_2, e_3, e_4\}$	$\alpha_2 \in [0.1, 0)$	$\{e_1, e_2, e_3, e_4\}$

Table 4. Evaluation of the semi-fuzzy quantifier *almost_all* in example 14

$\mathbf{almost_all}_E\left((tall)_{\alpha_1}, (blond)_{\alpha_2}\right)$	$\alpha_1 \in [1, 0.9)$	$\alpha_1 \in [0.9, 0.8)$	$\alpha_1 \in [0.8, 0.2)$	$\alpha_1 \in [0.2, 0)$
$\alpha_2 \in [1, 0.8)$	$0 : 0.02$	$0 : 0.02$	$0 : 0.12$	$0 : 0.04$
$\alpha_2 \in [0.8, 0.3)$	$0 : 0.05$	$0 : 0.05$	$0.33 : 0.30$	$0 : 0.10$
$\alpha_2 \in [0.3, 0.1)$	$1 : 0.02$	$1 : 0.02$	$1 : 0.12$	$0.5 : 0.04$
$\alpha_2 \in [0.1, 0)$	$1 : 0.01$	$1 : 0.01$	$1 : 0.06$	$1 : 0.02$

In order to resolve this problem we use the independence profile (see section (4)). We show the representatives fo *tall* and *blonde* in table 3, and in table 4 is depicted the evaluation of **almost_all**. If we calculate the mean of the truth values we obtain

$$\overline{H} = 0.02 \times 0 + 0.05 \times 0 + 1 \times 0.02 + \ldots + 1 \times 0.02 = 0.379$$

The method that is formulated above was proposed in [7] for the evaluation of fuzzy quantified sentences. This proposal is analyzed in depth in [9, 10] .

5.2 Extension of Functions on the Basis of Trivalued Random Sets

In this section we formulate a mechanism that is based on the one explained in [13] for the evaluation of quantified sentences, but applied to the problem of the extension of an arbitrary function $f : \wp^s(E) \to \mathbb{R} \cup \{\theta\}$.

Firstly we deal with the problem of extending $f : \wp^s(E) \to \mathbb{R} \cup \{\theta\}$ to trivalued sets. Let $X_1, \ldots, X_s \in \overset{\cup}{\wp}(E)$. On the basis of f we define the function g^f as follows:

$$g^f(X_1, \ldots, X_s)(r) \tag{12}$$
$$= \begin{cases} 1 & f(Y_1, \ldots, Y_s) = r \text{ for all } Y_1, \ldots, Y_s \in \wp(E), X_i^{\min} \subseteq Y_i \subseteq X_i^{\max} \\ \frac{1}{2} & \left(\exists Y_1, \ldots, Y_s \in \wp(E), X_i^{\min} \subseteq Y_i \subseteq X_i^{\max} \ \wedge \ f(Y_1, \ldots, Y_s) = r\right) \wedge \\ & \left(\exists Y_1, \ldots, Y_s \in \wp(E), X_i^{\min} \subseteq Y_i \subseteq X_i^{\max} \ \wedge \ f(Y_1, \ldots, Y_s) \neq r\right) \\ 0 & otherwise \end{cases}$$

Interpretation of g^f is similar to the interpretation of trivalued sets. An alternative interpretation of g^f as a possibility distribution on $\mathbb{R} \cup \{\theta\}$ allows

defining:

$$g^f(X_1,\ldots,X_s) \tag{13}$$
$$= \left\{r \in \mathbb{R}\cup\{\theta\} : \exists Y_1,\ldots,Y_s \in \wp(E), X^{\min} \subseteq Y_i \subseteq X^{\max}, f(Y_1,\ldots,Y_s) = r\right\}$$

We define the degree to which $r \in \mathbb{R}\cup\{\theta\}$ is compatible with the evaluation of f for a given profile P^β as

$$C(r|f,X_1,\ldots,X_s) \tag{14}$$
$$= \int_0^1 \ldots \int_0^1 g^f\left(\overset{\cup}{RP}{}^{L_1}_{Y_1}(\gamma_1),\ldots,\overset{\cup}{RP}{}^{L_s}_{Y_s}(\gamma_s)\right)(r)\,P^\beta(\alpha_1,\ldots,\alpha_s)\,d\alpha_1\ldots d\alpha_s$$
$$= \int_0^1 \ldots \int_0^1 g^f\left((X_1)^{\gamma_1},\ldots,(X_s)^{\gamma_s}\right)(r)\,P^\beta(\alpha_1,\ldots,\alpha_s)\,d\alpha_1\ldots d\alpha_s$$

When $\theta \notin g^f(X_1,\ldots,X_s)$ expression (13) allows the calculation of a single value, using a suitable aggregation function (for example, the mean). When $\theta \in g^f(X_1,\ldots,X_s)$ it is necessary to study the meaning of θ regarding the function to be extended.

In [13] the aggregation of the values that are obtained in the extension of semi-fuzzy quantifiers is calculated by means of the fuzzy median. The fuzzy median $med_{\frac{1}{2}} : [0,1]\times[0,1] \to [0,1]$ is defined as

$$med_{\frac{1}{2}}(u_1,u_2) = \begin{cases} \min(u_1,u_2) & : \min(u_1,u_2) > \frac{1}{2} \\ \max(u_1,u_2) & : \max(u_1,u_2) < \frac{1}{2} \\ \frac{1}{2} & : else \end{cases} \tag{15}$$

The generalised fuzzy median $m_{\frac{1}{2}} : \wp([0,1]) \to [0,1]$ is defined by

$$m_{\frac{1}{2}}X = med_{\frac{1}{2}}(\inf X, \sup X)$$

for all $X \in \wp([0,1])$.

It can be interpreted that the fuzzy median "considers" $\frac{1}{2}$ as the most cautious value.

For the maximum dependence profile, the result of the extension of f is a trivalued set consonant by specificity, and then it can be interpreted as a fuzzy set.

Examples We now present two examples of the use of trivalued mechanisms in the extension of functions. In the first one the calculation of the cardinality of a fuzzy set is addressed. In the second a quantified sentence is evaluated.

Example 15. Let us suppose that we wish to calculate the cardinality of the fuzzy set $X = \{0.8/e_1, 0.9/e_2, 1/e_3, 0.2/e_4\}$. Using the crisp cardinality function $card(X) = |X|$ we define

$$g^{card}(X)(i) = \begin{cases} 0 & |X| \neq i \\ 1 & |X| = i \end{cases} \quad , i \in \mathbb{N}$$

Table 5. Representatives for calculating the fuzzy cardinality in example 15

	X	$X^{\min}$	$X^{\max}$
$\gamma \in [0, 0.6]$	$\{1/e_1, 1/e_2, 1/e_3, 0/e_4\}$	$\{e_1, e_2, e_3\}$	$\{e_1, e_2, e_3\}$
$\gamma \in (0.6, 0.8]$	$\{\frac{1}{2}/e_1, 1/e_2, 1/e_3, \frac{1}{2}/e_4\}$	$\{e_2, e_3\}$	$\{e_1, e_2, e_3, e_4\}$
$\gamma \in (0.8, 1]$	$\{\frac{1}{2}/e_1, \frac{1}{2}/e_2, 1/e_3, \frac{1}{2}/e_4\}$	$\{e_3\}$	$\{e_1, e_2, e_3, e_4\}$

Table 6. Extension of the cardinality for the different representatives in example 15

	$g^{card}(X^\gamma)$
$\gamma \in [0, 0.6]$	$\{3\}$
$\gamma \in (0.6, 0.8]$	$\{3, 4\}$
$\gamma \in (0.8, 1]$	$\{1, 2, 3, 4\}$

Then,

$$C(r|card, X) = \int_0^1 g^{card}(X^\gamma)(r)\, d\gamma, r \in \mathbb{N}$$

The trivalued representatives of X are shown in table 5. Following that, the calculation of $C(1)$ is performed as follows:

$$\begin{aligned} C(1|card, X) &= \int_0^1 g^{card}(X^\gamma)(1)\, d\gamma \\ &= \int_0^{0.6} g^{card}(X^\gamma)(1)\, d\gamma + \int_{0.6}^{0.8} g^{card}(X^\gamma)(1)\, d\gamma \\ &+ \int_{0.8}^1 g^{card}(X^\gamma)(1)\, d\gamma \\ &= \int_0^{0.6} 0 d\gamma + \int_{0.6}^{0.8} 0 d\gamma + \int_{0.8}^1 \frac{1}{2} d\gamma = 0.1 \end{aligned}$$

The result obtained is

$$C = \{0/0, 0.1/1, 0.2/2, 0.8/3, 0.2/4\}$$

It can be seen that the cardinality function defined by this mechanism coincides with FE_{Count} [25, 26, 28][7]. For this example, using (13) we would obtain results shown in table 6.

We can represent $g^{card}(X^\gamma)$ by means of trivalued sets, as is shown in table 7.

It should be noted that we have represented the image of the cardinality function by means of trivalued sets consonant with respect to specificity. Thus the answer to the problem can be interpreted as a fuzzy set.

[7] We think that using similar ideas to the ones explained in [25, 26] for the general problem of the extension of functions, the same results are obtained as when formulating the extension using trivalued mechanisms under the maximum dependence profile.

Table 7. Representation of the extension of the cardinality as a fuzzy set

	$g^{card}(X^\gamma)$
$\gamma \in [0, 0.6]$	$\{0/0, 0/1, 0/2, 1/3, 0/4\}$
$\gamma \in (0.6, 0.8]$	$\{0/0, 0/1, 0/2, \frac{1}{2}/3, \frac{1}{2}/4\}$
$\gamma \in (0.8, 1]$	$\{0/0, \frac{1}{2}/1, \frac{1}{2}/2, \frac{1}{2}/3, \frac{1}{2}/4\}$

Example 16. We now show the evaluation of a quantified sentence. Let us consider again example 14 and let us use the maximum dependence profile. The trivalued cuts of the sets $tall\,(E)$ and $blonde\,(E)$ are represented in table 8. Using the fuzzy median (15) to aggregate the images of g^Q for each level we obtain the results shown in table 9. The expected value of this result is:

$$C\,(Q|tall\,(E)\,, blonde\,(E)) = \int_0^1 m_{\frac{1}{2}}\left(g^{\mathbf{almost_all}}\left(\left(tall\,(E)\right)^\gamma, \left(blonde\,(E)\right)^\gamma\right)\right) d\gamma$$
$$= \frac{1}{3} \times 0.4 + \frac{1}{2} \times 0.6 = 0.433$$

This method for evaluating quantified sentences was described in [13] and has proven to be one of the best models for this task.

5.3 Discussion of the Mechanisms

As can be seen in the examples, the two extension mechanisms that have been described in this section exhibit satisfactory behaviour. Although this is far from being a formal proof, we expect that the results obtained on the basis of random set interpretations for fuzzy extension of functions may be better than the possibilistic based ones, since some relevant facts point to this direction. For example, both mechanisms provide excellent results [6, 9, 10, 13, 15, 16] when applied to semi-fuzzy quantifiers. The behaviour of trivalued models in fuzzy quantification is somewhat better, but the interpretation that supports bivalued mechanisms is highly consistent. Furthermore, it seems that problems with bivalued mechanisms can be avoided by means of a stricter interpretation of fuzzy sets [8].

Although the behaviour of trivalued mechanisms in the given examples is satisfactory, results may be somewhat questionable when extension of functions that depend on the order of the referential (e.g., *last*) is considered.

Table 8. Representatives of $tall\,(E)_\gamma^{\min}$ and $blonde\,(E)_\gamma^{\min}$

	$(tall\,(E))_\gamma^{\min}$	$(tall)_\gamma^{\max}$	$(blonde)_\gamma^{\min}$	$(blonde)_\gamma^{\max}$
$\gamma \in [0, 0.4]$	$\{e_1, e_2, e_3\}$	$\{e_1, e_2, e_3\}$	$\{e_1, e_2\}$	$\{e_1, e_2\}$
$\gamma \in (0.4, 0.6]$	$\{e_1, e_2, e_3\}$	$\{e_1, e_2, e_3\}$	$\{e_1, e_2\}$	$\{e_1, e_2, e_3\}$
$\gamma \in (0.6, 0.8]$	$\{e_2, e_3\}$	$\{e_1, e_2, e_3, e_4\}$	$\{e_1\}$	$\{e_1, e_2, e_3\}$
$\gamma \in (0.8, 1]$	$\{e_3\}$	$\{e_1, e_2, e_3, e_4\}$	$\{e_1\}$	$\{e_1, e_2, e_3, e_4\}$

Table 9. Evaluation of the semi-fuzzy quantifier *almost_all* in example 16

$g^{\mathbf{almost_all}}$	
$\gamma \in (0, 0.4]$	$m_{\frac{1}{2}}\left(g^{\mathbf{almost_all}}\left((tall\,(E))^{\gamma}, (blonde\,(E))^{\gamma}\right)\right) = m_{\frac{1}{2}}\left(\left\{\frac{1}{3}\right\}\right) = \frac{1}{3}$
$\gamma \in (0.4, 0.6]$	$m_{\frac{1}{2}}\left(\left\{\frac{1}{3}, 1\right\}\right) = \frac{1}{2}$
$\gamma \in (0.6, 0.8]$	$\frac{1}{2}$
$\gamma \in (0.8, 1]$	$\frac{1}{2}$

Table 10. Extension of *last* for the different representatives (example 17)

	$(high\,(E))^{\gamma}$	$last\,((high\,(E))^{\gamma})$
$\gamma \in [0, 0.5]$	$\{1/e_1, \frac{1}{2}/e_2, \frac{1}{2}/e_3, 1/e_4, 1/e_5\}$	$\{e_5\}$
$\gamma \in (0.5, 1]$	$\{1/e_1, \frac{1}{2}/e_2, \frac{1}{2}/e_3, \frac{1}{2}/e_4, \frac{1}{2}/e_5\}$	$\{e_1, e_2, e_3, e_4, e_5\}$

Example 17. Let us consider that we wish to evaluate the sentence "the last temporal instant at which the temperature was high" on the temporal referential $E = \{e_1, e_2, e_3, e_4, e_5\}$ $(e_1 < e_2 < \ldots < e_5)$. Let

$$high\,(E) = \{1/e_1, 0.5/e_2, 0.5/e_3, 0.75/e_4, 0.75/e_5\}$$

be the fuzzy set that indicates whether the temperature has been high. If we calculate the trivalued representatives of $high\,(E)$ we obtain results shown in table 10 and therefore

$$C\,(last|high\,(E)) = \{0.25/e_1, 0.25/e_2, 0.25/e_3, 0.25/e_4, 0.75/e_5\}$$

It should be noted that it seems counterintuitive to assign a truth value greater than 0 to points e_2 and e_3, since there exist instants later than e_3 with a higher truth value for "high temperature".

Although the issue is not discussed in this work, it is possible to solve this problem by using the order of the referential to correct the results of the expression (13).

One important aspect is that of computational complexity. In fuzzy quantification, complexity is polynomial for the models in the examples. For binary approaches it seems that once f has been calculated, extension can be computed in polynomial time. Unfortunately, for trivalued sets this is not the case, although it seems more "interesting" functions can be extended or approximated in polynomial time; further studies should be conducted.

6 Conclusions

In this work two mechanisms for the extension of functions have been described. Their underlying random set-based interpretation is explained, and a number of examples of application have been given. Among other interesting operators, these extension mechanisms frame plausible quantification models, fuzzy cardinality measures and fuzzy temporal operators.

The definition of mechanisms of this type is therefore fundamental for developing consistent semantic models that allow to perform a correctly interpretable computing with words.

References

[1] J. F. Baldwin, J. Lawry, and T. P. Martin. Mass assignment theory of the probability of fuzzy events. *Fuzzy Sets and Systems*, 83:353–367, 1996. 2, 7, 8, 9

[2] G. Bordogna and G. Pasi. Modeling vagueness in information retrieval. In M. Agosti, F. Crestani, and G. Pasi, editors, *ESSIR 2000, LNCS 1980*, pages 207–241. Springer-Verlag Berlin Heidelberg, 2000. 2

[3] P. Bosc, L. Lietard, and O. Pivert. Quantified statements and database fuzzy querying. In P.Bosc and J. Kacprzyk, editors, *Fuzziness in Database Management Systems*, volume 5 of *Studies in Fuzziness*, pages 275–308. Physica-Verlag, 1995. 1

[4] P. Cariñena, A. Bugarín, M. Mucientes, F. Díaz-Hermida, and S. Barro. *Technologies for Constructing Intelligent Systems*, volume 2, chapter Fuzzy Temporal Rules: A Rule-based Approach for Fuzzy Temporal Knowledge Representation and Reasoning, pages 237–250. Springer-Verlag, 2002. 1, 5

[5] M. Delgado, D. Sánchez, and M. A. Vila. Fuzzy cardinality based evaluation of quantified sentences. *International Journal of Approximate Reasoning*, 23(1):23–66, 2000. 2

[6] F. Díaz-Hermida, A. Bugarín, and S. Barro. Definition and classification of semi-fuzzy quantifiers for the evaluation of fuzzy quantified sentences. *International Journal of Approximate Reasoning*, 2003. In press. 1, 22

[7] F. Díaz-Hermida, A. Bugarín, P. Cariñena, and S. Barro. Evaluación probabilística de proposiciones cuantificadas borrosas. In *Actas del X Congreso Español Sobre Tecnologías y Lógica Fuzzy (ESTYLF 2000)*, pages 477–482, 2000. 16, 19

[8] F. Díaz-Hermida, A. Bugarín, P. Cariñena, and S. Barro. Un esquema probabilístico para el tratamiento de sentencias cuantificadas sobre fórmulas. In *Actas del XI Congreso Español Sobre Tecnologías y Lógica Fuzzy (ESTYLF 2002)*, pages 391–396, 2002. 10, 15, 22

[9] F. Díaz-Hermida, A. Bugarín, P. Cariñena, and S. Barro. Voting model based evaluation of fuzzy quantified sentences: a general framework. Technical Report GSI-02-01, Intelligent Systems Group. Univ. Santiago de Compostela, 2002. 2, 16, 19, 22

[10] F. Díaz-Hermida, P. Cariñena, A. Bugarín, and S. Barro. Probabilistic evaluation of fuzzy quantified sentences: Independence profile. *Mathware and Soft Computing*, VIII(3):255–274, 2001. 16, 19, 22

[11] D. Dubois and H. Prade. Fuzzy cardinality and the modeling of imprecise quantification. *Fuzzy Sets and Systems*, 16:199–230, 1985. 2, 4, 5, 6

[12] D. Dubois and H. Prade. Measuring properties of fuzzy sets: A general technique and its use in fuzzy query evaluation. *Fuzzy Sets and Systems*, 38:137–152, 1989. 2, 4, 9, 17

[13] I. Glöckner. DFS- an axiomatic approach to fuzzy quantification. TR97-06, Techn. Fakultät, Univ. Bielefeld, 1997. 2, 3, 10, 11, 12, 15, 19, 20, 22

[14] I. Glöckner. A framework for evaluating approaches to fuzzy quantification. Technical Report TR99-03, Universität Bielefeld, May 1999. 15

[15] I. Glöckner. Advances in DFS theory. TR2000-01, Techn. Fakultät, Univ. Bielefeld, 2000. 15, 22
[16] I. Glöckner and A. Knoll. A formal theory of fuzzy natural language quantification and its role in granular computing. In W. Pedrycz, editor, *Granular computing: An emerging paradigm*, volume 70 of *Studies in Fuzziness and Soft Computing*, pages 215–256. Physica-Verlag, 2001. 2, 3, 22
[17] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*. John Wiley and Sons Inc, 1994. 2, 7, 8
[18] J. Lawry. Possibilistic normalisation and reasoning under partial inconsistency. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(1):413–436, 2001. 10, 14
[19] E.S. Lee and Q. Zhu, editors. *Fuzzy and evidence reasoning*, volume 6 of *Studies in Fuzziness*. Physica-Verlag, 1995. 14
[20] A. L. Ralescu. Cardinality, quantifiers, and the aggregation of fuzzy criteria. *Fuzzy Sets and Systems*, 69:355–365, 1995. 2, 4
[21] A. L. Ralescu and D. A. Ralescu and K. Hirota. Evaluation of fuzzy quantified expresssions. In *Lecture Notes in Artificial Intelligence LNAI 1566*, pages 234–245. Springer-Verlag Berlin Heidelberg, 1999. 2
[22] D. Sánchez. *Adquisición de relaciones entre atributos en bases de datos relacionales. Tesis Doctoral*. PhD thesis, Universidad de Granada. E. T. S. de Ingeniería Informática, 1999. 1
[23] G. Shafer, editor. *A mathematical theory of evidence*. Princeton University Press, 1976. 8
[24] S. F. Thomas. *Fuzziness and Probability*. ACG Press, 1995. 2, 7
[25] M. Wygralak. Fuzzy cardinals based on the generalized equality of fuzzy subsets. *Fuzzy Sets and Systems*, 18:143–158, 1986. 2, 4, 6, 21
[26] M. Wygralak. Questions of cardinality of finite fuzzy sets. *Fuzzy Sets and Systems*, 102:185–210, 1999. 2, 4, 21
[27] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning, parts 1-3. *Information Sciences*, 8:199–279, 301–357; 9:43–80, 1975. 2, 4
[28] L. A. Zadeh. A theory of approximate reasoning. In J. E. Hayes, D. Michie, and L. I. Mikulich, editors, *Machine Intelligence*, volume 9, pages 149–194. Wiley, New York, 1979. 21
[29] L. A. Zadeh. Probability measures of fuzzy events. *J. Math. Anal. Appl.*, 23:421–427, 1968. 17
[30] L. A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Comp. and Machs. with Appls.*, 8:149–184, 1983. 2, 4, 5, 6
[31] L. A. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2):103–111, 1996. 1

A General Framework for Induction of Decision Trees under Uncertainty

Enric Hernández[1] and Jordi Recasens[2]

[1] Secció de Matemàtiques i Informàtica
ETSAB. Avda. Diagonal 649. 08028 Barcelona. Spain
Universitat Politècnica de Catalunya
enriche@lsi.upc.es
[2] Secció de Matemàtiques i Informàtica
ETSAV. Pere Serra 1-15. 08190 Sant Cugat. Spain
Universitat Politècnica de Catalunya
j.recasens@upc.es

Abstract. Real data is pervaded with uncertainty. Besides, nowadays it is widely accepted the existence of other kinds of uncertainty beyond the classical probabilistic approach. As a consequence, development and adaptation of automatic knowledge acquisition techniques under uncertainty is entirely advisable. Among them the decision tree paradigm is specially suitable to comprehension and readability concerns. This paper provides the definition of a general framework for the induction of decision trees in the presence of uncertainty. A novel approach based on the concept of observational entropy is also introduced.

1 Introduction

Recently, technological improvements and our growing ability in collecting information has raised the emergence of a great number of data sets about a very different kind of topics. But all this information will become useless unless some efficient mechanisms are developed in order to manage this huge amount of data.

This appreciation leads to the distinction between information and knowledge, thinking the latter as an arrangement or meta-structure that allows us a clear comprehension and affords efficient management procedures for the problem being considered.[1]

Moreover, available data sometimes happen to be pervaded with uncertainty either due to their inherent qualitative origin, or to limitations in measuring instruments. Nevertheless, it should be noted that lack of uncertainty may not be always a desirable property since its management could help us in obtaining simpler and more understandable representations by improving abstraction and generalization abilities. Therefore it seems reasonable asking the available methods and techniques for an adequate treatment of uncertainty.

[1] Indeed, emphasis should be put in the fact that knowledge should be suitable for the purpose at hand (classification, prediction, summarization ...) being, at the same time, comprehensible enough to avoid the "black-box" phenomenon.

J. Lawry, J. Shanahan, A. Ralescu (Eds.): Modelling with Words, LNAI 2873, pp. 26–43, 2003.
© Springer-Verlag Berlin Heidelberg 2003

Because of the comprehensibility requirement stated previously, we are mainly interested in knowledge representation systems which are close to human intuition. To this end, we will focus on what have been called linguistic rules. Such representations combine the advantage of the flexibility provided by "if-then" rules with the readability offered by the management of linguistic labels, since both their consequent and antecedent part are built from values taken by linguistic variables.

Zadeh [39] defined a linguistic variable as a fivefold structure $(V, T(V), U, G, M)$ where:

- V = name of the variable
- $T(V)$ = set of atomic terms
- U = universe of discourse
- G = syntactic rules for generating valid terms
- M = semantic rules for associating the proper "meaning" (fuzzy set on U) to each valid term.

Regarding the extraction of linguistic rules from data several approaches have been proposed in the literature. As illustrative, and by no means exhaustive, examples the following may be worth to be mentioned:

- The concept of linguistic summary [37, 35, 34, 36] was introduced by Yager [37] to describe, in a linguistic manner, the more relevant properties of a given set of instances. Summaries are instances of the following syntactical pattern "Q instances are S" , where S is a summarizer and Q is a linguistic quantifier.
- The family of methods known as sequential covering algorithms [6, 20] are intended to produce a disjunctive set of rules which "cover" a given subset of instances (target concept) of the training set by following the next scheme: first, a rule is generated which covers a "portion" of the target concept. We require this rule having high accuracy, but not necessarily high coverage. By high accuracy we mean the predictions it makes should be correct. By accepting low coverage, we mean it needs not make predictions for every training example [21]. Then instances which are covered by this rule are removed from the training set. This procedure can be iterated as many times as desired to learn a disjunctive set of rules that together cover any desired fraction of the training set. This is called sequential covering algorithm because it sequentially learns a set of rules that together cover the full set of instances representing the target concept. Generalizations to manage linguistic labels have also been devised [11].
- Linguistic functional dependencies. Based on the concept of fuzzy graph [40], the objective is grasping existing functional dependencies and expressing them in a qualitative manner. References [31, 8] are examples of this approximation.
- Other group is composed by those that can be called "biological-based" methods. In this case, use of some biological inspired mechanism (neural networks [16], genetic algorithms [12], ant colonies [5], ...) is made in order to induce a set of linguistic rules.

From among the catalogue of methods for extracting linguistic rules, in this paper we will be concerned with decision tree based methods.

Decision trees, since their formal appearance within the context of inductive learning [22] have become one of the most relevant paradigm of machine learning methods. The main reason for this wide-spreading success lies in their proved applicability to a broad range of problems, in addition to appealing features as the readability of the knowledge represented in the tree. Therefore, a lot of work have been carried out from Quinlan's TDID3 algorithm in order to extend the applicability to domains beyond the categorical ones and achieve further improvements. In this line, many approaches dealing with continuous-valued attributes have been proposed ([4, 24, 18]). Also, alternative measures to the classical Shannon's entropy measure [28] for attribute selection have been devised, like Gini's test [4], Kolmogorov-Smirnoff distance [30], distance between partitions [25], contrast measures [7], ...

Another important point has been providing decision tree induction algorithms with a more flexible methodology in order to cope with other sources of uncertainty beyond the probabilistic type. Indeed, when we face real problems we should overcome the limitations of the probabilistic framework by furnishing existing methods, so that other well-known types of uncertainty could be managed.

Some generalizations have been proposed to achieve this goal:

- Coping with missing or uncertain attribute values where uncertainty is represented by probability distributions led Quinlan to develop probabilistic decision trees [23].
- The generalization and interpolation properties of fuzzy sets together with their less sensitivity to small changes in input attribute values make them specially suitable for classification tasks. Fuzzy decision trees ([14], [32], [38], [29], [1]) benefit from those aspects to provide a flexible framework for inducing linguistic rules.
- Belief measures as they were introduced by Shafer [27] provide a mechanism to express and deal with subjective judgments in a much more flexible way than probability, offering tools for handle ignorance and combining several pieces of evidence. Hence it was advisable to integrate the advantages of belief functions and decision trees, resulting in the belief decision trees approach [10, 2].

Arriving at this point, a natural question arises: Could a general framework for induction of decision trees under uncertainty be defined so that those previously outlined methods and also potentially new ones constitute particularizations of this general framework?

The paper is devoted to develop further this idea and is organized as follows: in section 2 a detailed description of the general framework is given, section 3 provides a tabular characterization (in terms of the framework previously described) of representative methods belonging to families of procedures for inducing decision trees under uncertainty. Finally, section 4 introduces a novel approach for inducing decision trees based on the concept of observational entropy.

2 A General Framework

A decision tree can be viewed as a representation of a procedure to determine the classification of an object. Any specific decision tree based technique should deal basically with two main concerns, namely, how to build the tree out of a set of examples, and how it is going to be used, corresponding to the definition of a building procedure and an inference algorithm, respectively.

Regarding the building procedure it usually follows the basic scheme by Quinlan [22] based on a top down strategy (top down induction of decision tree (TDIDT)) which proceeds by successive partitioning of the training set as detailed in the procedure below:

1. Place the initial data on the root.
2. Select the best attribute from the set of non used attributes and mark it as used.
3. Create new child nodes according to the partition induced by the selected attribute.
4. For each newly generated child node iterate step 2 unless any stopping criterion holds. In this case mark current node as a leaf and compute its associated label.

On the other hand, the inference process aims at classifying a new instance by traversing down the proper branch of the tree until its corresponding leaf has been reached. In order to cope with uncertainty within this process some steps must be adapted. Indeed, since the partitioning strategy does not already define an equivalence relation, an instance can follow several paths down in the tree to a certain degree and, consequently, several leaves could be reached whose labels should be combined to produce a classification. Hence, the inference algorithm should involve the next two steps:

- Compute the set of leaves the instance to classify arrives to.
- Combine their associated labels to produce the output classification.

Our claim is that any decision tree based method admits a decomposition in terms of the points we are going to describe in the following subsections, so that a given method should be describable by means of a concrete configuration defined over them. Let us examine these points in more detail.

2.1 Structure of the Training Set

One major requirement when defining a general framework for induction of decision trees should be to integrate and manage different types and representations of uncertainty in an homogeneous way. So, the framework should allow us to deal with attributes pervaded with different kinds of uncertainty described in terms of the following training set structure.

Let A be the set of attributes, let $c \in A$ (class attribute) be a distinguished attribute providing information about the class to which each instance belongs

to. Thus we define the set of instances E where for all instance $e \in E$ and for all attribute $a \in A$, e_a is the available evidence (possibly uncertain) belonging to instance e about the value $v \in domain(a)$ taken by attribute a.

On the other hand, for each attribute $a \in A$ a set of linguistic labels $L_a = \{a_1, \ldots, a_{|L_a|}\}$ whose meaning are fuzzy sets in the corresponding domain ($domain(a)$) is defined.

These labels will "decorate" nodes and edges of the tree and shall make up the language for expressing the linguistic classification rules derived from the tree.

Now the question turns into how to manage different representations of evidences in a consistent way. A solution to this problem could be performing on the initial training set a transformation similar to the so called "binning" [3] in the classical setting. This transformation expands each attribute column of the initial training set in so many columns as the number of linguistic labels ($|L_a|$) defined for the attribute.

The cell of the expanded training set corresponding to linguistic label $a_i \in L_a$ and instance e will contain the compatibility degree between evidence e_a and linguistic label a_i given by a proper compatibility measure in such a way that these degrees could be interpreted as the approximation of evidence e_a in the linguistic label space of attribute a (L_a).

A more exhaustive study of these measures (properties, axiomatic characterization ...) falls out of the scope of this paper. Nevertheless it should be noted that the definition of such compatibility measures is possible because both evidence and linguistic label share a common base domain ($domain(a)$). Changing the family of compatibility measures results in different compatibility values and, therefore, in particular methods for computing the expanded training set.

2.2 Node Membership Function

As it was pointed out previously, TDIDT techniques rely on a "divide-and-conquer" paradigm by continuously partitioning the remaining set of instances as an effect of adding new constraints to be fulfilled. These new constraints come from the set of linguistic labels of the attribute selected to partition the set of instances.

In the classical setting, each partition defines an equivalence relation and the degree of membership of a given instance to each class of the quotient set is defined by the boolean conjunction of the set of constraints appearing when following the proper path up to the root.

When facing with uncertain evidences, "testing" by an attribute label does not usually produce a boolean answer. Instead of this, a compatibility degree between the evidence and the label should be managed.

Let N be the set of nodes, $n \in N$ be a given node and $R = \{r_1, \ldots, r_p\}$ be the set of constraints belonging to the path going from the root to node n. The fuzzy set $\mu_n : E \longrightarrow [0, 1]$ is defined over the set of instances as

$$\forall e \in E : \mu_n(e) = g(r_1(e), \ldots, r_p(e)) \tag{1}$$

where g is a conjunctive aggregation operator (usually a t-norm) and $r_i(e)$ the compatibility degree between instance e and the linguistic label corresponding to restriction r_i.

2.3 Attribute Selection

All along this section we will assume the following nomenclature:

- E , set of instances and $e \in E$ a particular instance
- N , set of nodes and $n \in N$ a particular node
- A , set of attributes and $a \in A$ a particular attribute
- L_a , set of linguistic labels defined for attribute a and $a_i \in L_a$ a particular linguistic label of attribute a
- Let $a_i \in L_a$ be a linguistic label of attribute a and $n \in N$, we note by $(n|a_i)$ the node whose associated set of constraints is the result of concatenating the constraints associated to node n with the constraint $a = a_i$
- Let $n \in N$, we note by μ_n the node n membership function as defined in 2.2.
- Let $c_i \in L_c$ be a linguistic label defined for the class attribute c, we define $\nu_{c_i} : E \longrightarrow [0, 1]$ as

$$\forall e \in E : \nu_{c_i}(e) = m_c(e_c, c_i) \tag{2}$$

  where m_c is the compatibility measure between evidences and labels for class attribute c.
- T , set of normalization functions (usually probabilistic or possibilistic normalization) and $t \in T$ a particular normalization function
- F , set of uncertainty measures[2] and $f \in F$ a particular uncertainty measure function
- G , set of aggregation operators[3] and $g \in G$ a particular aggregation operator. We will also use the notation g^{SET}_{NODE} indicating that SET is an index set referencing in a direct or indirect way the values to be aggregated; and $NODE$ is intended to help in providing the referential instances counting number to weigh up against to being, therefore, the basis for computing weight vectors in weighted aggregation operators (for example).

In the classical picture we have several alternatives for quantifying the uncertainty associated to a given node. When shifting to the uncertain setting, the repertory of measures goes even wider, reflecting the broadly accepted fact that there exist different kinds of uncertainty beyond the probabilistic one [15] and consequently, some well-established measures have been developed to cope with it.

Nevertheless, a concrete realization of the general framework is not only particularized by the uncertainty measure but also by the aggregation operators used and the order in which they are applied respect to the calculation of this

[2] A good compendium of uncertainty measures can be found in [15].
[3] For an in-depth study of aggregation operators the reader is referred to [19].

measure. We have basically two possibilities depending on whether we first apply the uncertainty measure (f) to each instance before the resulting values are aggregated at the corresponding node ("horizontal folding"), or we perform it the opposite way ("vertical folding"). Therefore, when considering attribute a as candidate to become selected as the current branching attribute[4] we have basically the two schemes below as options for computing uncertainty in node n.

1. Horizontal folding:

$$UNC(n) = g_n^{a_i \in A_L}(g_{(n|a_i)}^{e \in E}(f(t(\nu_{c_1}(e), \ldots, \nu_{c_{|L_c|}}(e))))) \tag{3}$$

2. Vertical folding:

$$UNC(n) = g_n^{a_i \in A_L}(f(t(g_{(n|a_i)}^{e \in E}(\mu_{(n|a_i|c_1)}(e)), \ldots, g_{(n|a_i)}^{e \in E}(\mu_{(n|a_i|c_{|L_c|})}(e))))) \tag{4}$$

2.4 Inference Algorithm

As it was already noted before, coping with uncertainty makes that in contrast to the classical case, neither a unique leaf is usually reached nor a single class label could be tagged on a given leaf, making it necessary to define a procedure for combining the classifications associated to the set of reached leaves.

Therefore, and arguing in a similar way that of the previous subsection, two basic schemes for combining these classifications are shown below, where u is the instance to classify, h a particular leaf and s a "collapsing" function (i.e, a function intended to provide a precise single value from an imprecise classification (majority policies, defuzzification methods, ...)).

1.

$$CLASIF(u) = s(g_u^{h \in H}(g_h^{e \in E}(\mu_{(h|c_1)}(e))), \ldots, g_u^{h \in H}(g_h^{e \in E}(\mu_{(h|c_{|L_c|})}(e)))) \tag{5}$$

2.

$$CLASIF(u) = g_u^{h \in H}(s(g_h^{e \in E}(\mu_{(h|c_1)}(e)), \ldots, g_h^{e \in E}(\mu_{(h|c_{L_c})}(e)))) \tag{6}$$

3 Characterization of Existing Families of Methods

Tables 1 and 2 contain the characterization of representative methods belonging to families of procedures for inducing decision trees in the presence of uncertainty. These families can be summarized in three main categories:

1. Fuzzy probabilistic decision trees (represented by [14]): adaptation of classical entropy based approach to the fuzzy case.

[4] Usually a maximum information gain criterion is used which, roughly speaking, is equivalent to proceeding by continuously diminishing uncertainty.

2. Possibilistic decision trees (represented by [38, 26]: use of possibility theory definitions and methods is extensively made in order to build the decision tree.
3. Belief decision trees (represented by [10]): exploit representational power and evidence combination rules provided by belief functions theory in order to model a subjective belief approach to the problem of growing decision trees.

4 A Novel Approach: Observational Decision Trees

In this section we will introduce a new approach to building a decision tree addressing the case when uncertainty arises as a consequence of having defined indistinguishability relations [13] on the domains of the attributes used to describe the set of instances. As far as we know, existing methods make the assumption that different events are perfectly distinguishable from each other when measuring, for instance, node's impurity (for entropy-based methods). In front of the above restrictive assumption we advocate for a more realistic setting in which decision maker's discernment abilities should be taken into account, and therefore, impurity should be measured accordingly to his frame of discernment. With this purpose in mind we introduce the notion of observational entropy which adapts the classical definition of entropy in order to incorporate such indistinguishability concerns. The main idea is that the occurrence of two different events but indistinguishable by the indistinguishability relation defined, will count as the occurrence of the same event when measuring the observational entropy.

4.1 Observational Entropy

In this subsection we will present the definition of observational entropy and conditioned observational entropy which will be used in later sections.

Definition 1 *Given a t-norm T, a T-indistinguishability operator E on a set X is a reflexive and symmetric fuzzy relation on X such that $T(E(x,y),E(y,z)) \leq E(x,z)$ (T-transitivity), for all $x,y,z \in X$.*

Throughout the section E and E' will denote T-indistinguishability operators on a given set X and P a probability distribution on X.

Definition 2 *The observation degree of $x_j \in X$ is defined by:*

$$\pi(x_j) = \sum_{x \in X} p(x)E(x,x_j).$$

This definition has a clear interpretation: the possibility of observing x_j is given by the probability that x_j really happens (expressed by the first term), plus the probability of occurrence of some element "very close" to x_j, weighted by the similarity degree.

Definition 3 *The observational entropy (HO) of the pair* (E, P) *is defined by:*

$$HO(E,P) = -\sum_{x\in X} p(x)\log_2 \pi(x).$$

The next step is to define the conditioned observational entropy. Informally, the conditioned observational entropy measures how do affect the observations performed by an observer "using" a T-indistinguishability operator E' in the variability degree of the potential observations (observational entropy) of some other observer using another T-indistinguishability operator E.

Definition 4 $\forall x \in X$ *we define:*

$$P^E_{x_j}(x) = \frac{p(x)\cdot E(x,x_j)}{\pi_E(x_j)} = \frac{p(x)\cdot E(x,x_j)}{\sum_{y\in X} p(y)\cdot E(y,x_j)}.$$

That is, $P^E_{x_j}(x)$ quantifies the contribution of x to the observation degree of x_j in (E, P).

Definition 5 *The conditioned observation degree of* $x_i \in X$ *having been observed* x_j *in* (E', P) *is defined as*

$$\pi^{E|E'}_{x_j}(x_i) = \sum_{x\in X} P^{E'}_{x_j}(x)\cdot E(x,x_i).$$

Definition 6 *The observational entropy of the pair* (E, P) *conditioned to the observation of* $x_j \in X$ *in* (E', P) *is defined as follows:*

$$HO_{x_j}(E \mid E', P) = -\sum_{x_i\in X} P^{E'}_{x_j}(x_i)\cdot \log_2 \pi^{E|E'}_{x_j}(x_i).$$

Definition 7 *The observational entropy of the pair* (E, P) *conditioned by the pair* (E', P) *is defined as*

$$HO(E \mid E', P) = \sum_{x_j\in X} p(x_j)\cdot HO_{x_j}(E \mid E', P).$$

In other words, the conditioned observational entropy of the pair (E, P) is the expected value of the observational entropy of (E, P) conditioned to the observation of all $x_j \in X$ in (E', P).

Due to the reflexivity of E, this expression can be rewritten as:

$$\pi(x_j) = p(x_j) + \sum_{x\in X \mid x\neq x_j} p(x)E(x,x_j).$$

Table 1. Methods characterization

	Classical id3 [22]	**Possibilistic id3 [38]**	**Possibilistic id3 [26]**
attribute evidences	nominal	fuzzy	nominal
class attribute evidences	nominal	fuzzy	not determined
attribute labels	nominal	fuzzy	nominal
class attribute labels	nominal	fuzzy	nominal
attribute compatibility measure (m)	crisp equality	not determined (values are given)	crisp equality
class attribute compatibility measure (m_c)	crisp equality	not determined (values are given)	not determined (values are given)
restrictions aggregation op $(g_e^{r_i \in R})$	boolean and	min t-norm	boolean and
node instance aggregation op $(g_n^{e \in E})$	sum	sum	max t-conorm
leaf instance aggregation op $(g_h^{e \in E})$	sum	sum	max t-conorm
leaves aggregation op $(g_u^{h \in H})$	(just one leaf reached)	max t-conorm	(just one leaf reached)
labels aggregation op $(g_n^{a_i \in A_L})$	weighted mean	weighted mean	max-min
normalization function (t)	probabilistic normalization	possibilistic normalization	not determined
collapsing function (s)	majority class	class with highest membership	class with less fuzziness
uncertainty measure (f)	entropy	U-uncertainty	U-uncertainty

Table 2. Characterization table (cont)

	Fuzzy probabilistic id3 [14]	**Belief id3 [10]**	**Observational id3 [9]**
attribute evidences	numerical	nominal	nominal + indistinguishability
class attribute evidences	numerical	bpa	nominal + indistinguishability
attribute labels	fuzzy set	nominal	nominal + indistinguishability
class attribute labels	fuzzy set	nominal	nominal + indistinguishability
attribute compatibility measure (m)	$Pos(label\|evid) = \mu_{label}(evid)$	crisp equality	conditioned observation degree
class attribute compatibility measure (m_c)	$Pos(label\|evid) = \mu_{label}(evid)$	pignistic probability	t-norm
restrictions aggregation op $(g_e^{r_i \in R})$	t-norm	boolean and	t-norm
node instance aggregation op $(g_n^{e \in E})$	sum	sum	sum
leaf instance aggregation op $(g_h^{e \in E})$	sum	bpa averaging function	sum
leaves aggregation op $(g_u^{h \in H})$	weighted mean	disjunctive rule combination	(just one leaf reached)
labels aggregation op $(g_n^{a_i \in A_L})$	weighted mean	weighted mean	weighted mean
normalization function (t)	probabilistic normalization	probabilistic normalization	probabilistic normalization
collapsing function (s)	defuzzification	pignistic decision	majority class
uncertainty measure (f)	entropy	entropy	observational entropy

Table 3. Original data set

instance	*outlook*	*temperature*	*windy*	*play*
1	sunny	hot	false	volley
2	sunny	hot	true	swimming
3	overcast	hot	false	tennis
4	rainy	mild	false	football
5	rainy	cool	true	football
6	overcast	cool	true	football
7	sunny	mild	false	tennis
8	sunny	mild	true	swimming
9	overcast	hot	false	tennis
10	rainy	mild	true	football

4.2 Induction Algorithm

We have already introduced the concept of observational entropy. Let us see how to use it for the task of building a decision tree from a set of examples. The problem could be posed as follows: Let $\{A_1, \ldots, A_n, C\}$ be a set of nominal[5] attributes (where the classes of C form the classification we want to learn), with domains $D_i = \{v_{i_1}, \ldots v_{i_{mi}}\}$ and $D_c = \{v_{c_1}, \ldots, v_{c_{mc}}\}$. Let $E \subseteq D_1 \times \ldots \times D_n \times D_c$ be the set of instances, and for each attribute A_i we consider a T-indistinguishability operator E_{A_i} and a probability distribution P_{A_i} defined on the domain of A_i. Let us illustrate the above definitions with the example of tables 3 and 4.

$$D_{Outlook} = \{sunny, overcast, rainy\}$$
$$D_{Temperature} = \{hot, mild, cool\}$$
$$D_{Windy} = \{true, false\}$$
$$D_{Play} = \{swimming, tennis, football, volley\}$$

In order to simplify, we will assume that the probability distribution associated to each attribute of the example will be defined as the uniform distribution on the corresponding domain. Generalizing this assumption is straightforward.

Next we present an algorithm for building a decision tree based on the observational entropy. The procedure could be summarized in the following points:

i) "Expanding" data set: from the original data set we create its associated "expanded" data set. For all instances, we compute the compatibility between each label and the evidence represented by the instance by computing the conditioned observational degree (5) between the given label and the proper component (evidence) of the instance.

[5] We consider nominal attributes for simplicity purposes, although the developed methodology can also deal with continuous domains.

Table 4. T-Indistinguishability operators (matrix representation)

$E_{Outlook}=$

	sunny	*overcast*	*rainy*
sunny	1	0	0
overcast	0	1	0.5
rainy	0	0.5	1

$E_{Temp}=$

	hot	*mild*	*cool*
hot	1	0.5	0.5
mild	0.5	1	0.5
cool	0.5	0.5	1

$E_{Play}=$

	swimming	*football*	*tennis*	*volley*
swimming	1	0	0	0
football	0	1	0.25	0.25
tennis	0	0.25	1	1
volley	0	0.25	1	1

$E_{Windy}=$

	true	*false*
true	1	0
false	0	1

As an example. let us detail how to compute the compatibility between the value *overcast* for the attribute *outlook* taken by instance 3 and the label *rainy*. We want to compute:

$$\pi_{overcast}(rainy) = \sum_{x\in\{sunny,overcast,rainy\}} P_{overcast}(x)\cdot E_{Outlook}(x, rainy).$$

where

$$P_{overcast}(sunny) = \frac{E_{Outlook}(sunny, overcast)}{\sum_{x\in\{sunny,overcast,rainy\}} E_{Outlook}(x, overcast)}$$

and

$$P_{overcast}(overcast) = \frac{E_{Outlook}(overcast, overcast)}{\sum_{x\in\{sunny,overcast,rainy\}} E_{Outlook}(x, overcast)}$$

and

$$P_{overcast}(rainy) = \frac{E_{Outlook}(rainy, overcast)}{\sum_{x\in\{sunny,overcast,rainy\}} E_{Outlook}(x, overcast)}$$

and therefore $\pi_{overcast}(rainy)$ equals $0 + \frac{1}{3} + \frac{1}{3}$.

The resulting "expanded" data set is depicted in table 5.

ii) Computing probabilities of observing events in a node n. Values contained in the expanded data set will be used to compute the compatibility degree (COM) between a conjunction of restrictions and the evidence represented by a given instance e:

$$COM(A_i = v_{i_j} \wedge \ldots \wedge A_k = v_{k_l}|e) = t(\pi(v_{i_j}|e_{A_i}), \ldots, \pi(v_{k_l}|e_{A_k}))$$

(where T is a t-norm) So, let n be a given node belonging to the current tree (the one which has been grown up to now) and let R be the conjunction of the restrictions found in the path going from the root of the tree to node n. We define the probability of observing label v_{i_j} of attribute A_i in node n as:

$$P_N(A_i = v_{i_j}) = \frac{\sum_{e\in E} COM((R \wedge A_i = v_{i_j})|e)}{\sum_{v_i\in D_i}\sum_{e\in E} COM((R \wedge A_i = v_i)|e)}$$

iii) Selecting branching attribute: in the previous point we have provided a method for computing the probabilities of observing the labels for all the attributes in a given node n. These values will allow us to select the best attribute in order to partition data "arriving" at node n (fulfilling the restrictions leading to node n). In this way, given a node n, we compute (for all non previously selected attributes) the observational entropy of class attribute (C) conditioned to a given remaining attribute A_i in the following manner:

$$HO(C|A_i) = \sum_{v_i \in D_i} P_n(A_i = v_i) \cdot HO(C|A_i = v_i)$$

where

$$HO(C|A_i = v_i) = -\sum_{v_c \in D_C} P_{n \wedge (A_i = v_i)}(c = v_c)$$
$$\cdot \log_2 \sum_{w_c in D_C} P_{n \wedge (A_i = v_i)}(C = w_c) \cdot E_C(w_c, v_c)$$

(where $P_{n \wedge (A_i = v_i)}$ are the probabilities measured in each one of child nodes of n induced by partition data arriving at node n accordingly with the labels of attribute A_i)

We select, as current branching attribute, the one which minimizes the conditioned observational entropy (which is equivalent to say that maximizes the observational information gain), and mark it as already used attribute.

Finally, for each newly generated child node the overall process should be iterated while all the following conditions hold:

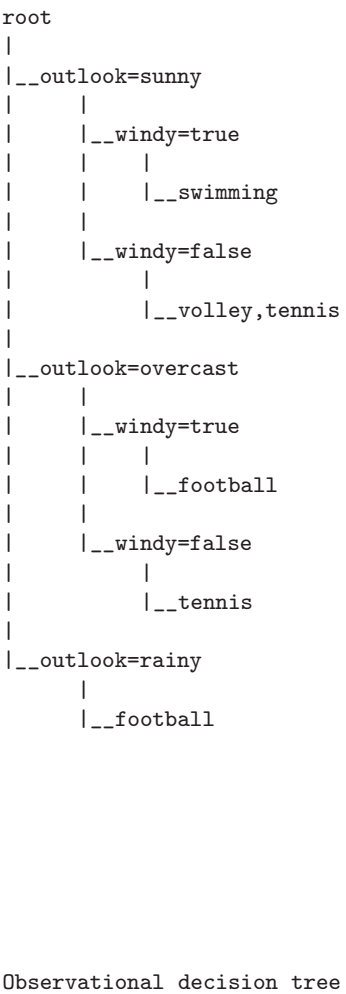
- There are remaining non used attributes.
- The set of instances arriving to that node is not the empty set.
- Observational entropy of current node is not below a predefined threshold value.

Table 5. Expanded data set

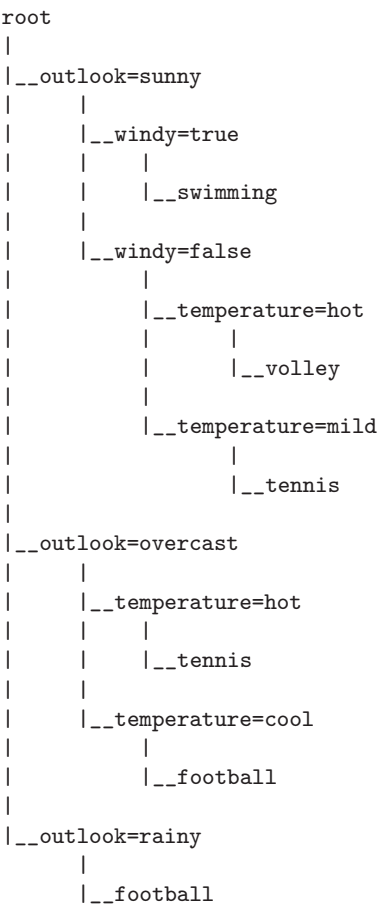
instance	*sunny*	*overcast*	*rainy*	*hot*	*mild*	*cool*	*true*	*false*	*swimming*	*tennis*	*football*	*volley*
1	1	0	0	0.7	0.6	0.6	0	1	0	0.9	0.9	0.1
2	1	0	0	0.7	0.6	0.6	1	0	1	0	0	0
3	0	0.8	0.6	0.7	0.6	0.6	0	1	0	0.9	0.9	0.1
4	0	0.6	0.8	0.6	0.7	0.6	0	1	0	0.5	0.5	0.7
5	0	0.6	0.8	0.6	0.6	0.7	1	0	0	0.5	0.5	0.7
6	0	0.8	0.6	0.6	0.6	0.7	1	0	0	0.5	0.5	0.7
7	1	0	0	0.6	0.7	0.6	0	1	0	0.9	0.9	0.1
8	1	0	0	0.6	0.7	0.6	1	0	1	0	0	0
9	0	0.8	0.6	0.7	0.6	0.6	0	1	0	0.9	0.9	0.1
10	0	0.6	0.8	0.6	0.7	0.6	1	0	0	0.5	0.5	0.7

This sketched procedure for building observational decision trees can be described according to the general framework detailed in section 2 as shown in *Observational Id3* column in table 2.

For data in table 3 the induced observational decision tree and classical decision tree are depicted below:



Observational decision tree



Classical decision tree

5 Concluding Remarks

As it has already been pointed out, real data is often pervaded with uncertainty. Therefore devising new techniques and adapting old ones in order to induce knowledge from data in the presence of uncertainty seems entirely advisable.

Among the existing methods, decision trees have become one of the most relevant paradigm within the machine learning community, mainly because of their proved

applicability to a broad range of problems in addition to features as the readability of the knowledge induced. Variants of the original scheme proposed by Quinlan have been developed, providing decision trees with a more flexible methodology in order to cope with different kind of uncertainty.

Nevertheless, it is our opinion that these methods would benefit from the definition of an homogeneous framework since most of their particularities could be easily described as particularizations of more general procedures.

This paper has been devoted to describe our proposal for such a common framework. Furthermore, characterizations (in terms of the proposed framework) of relevant existing methods for inducing decision trees in the presence of uncertainty are also provided.

Finally, a novel approach to building a decision tree is introduced, addressing the case when uncertainty arises as a consequence of considering a more realistic setting in which decision maker's discernment abilities are taken into account when computing impurity measures. This novel paradigm results in what have been called "observational decision trees" since the main idea stems from the notion of observational entropy as an adaptation of the classical definition of entropy in order to incorporate the mentioned indistinguishability concerns.

References

[1] J. Baldwin, J. Lawry, and P. Martin. Mass assignment based induction of decision trees on words. In *Proceedings IPMU 98*, 1998. 28

[2] M. Bjanger. Induction of decision trees from partially classied data using belief functions, 2000. 28

[3] Bratko, Cestnik, and Kononenko. *Assistant 86: A knowledge-elicitation tool for sophisticated users*. Sigma Press, 1986. 30

[4] L. Breiman et al. *Classification and regression trees*. Wadsworth International Group, 1984. 28

[5] J. Casillas, O. Cordon, and F. Herrera. Learning fuzzy rules using ant colony optimization algorithms. In *Proc. ANTS'2000*, pages 13–21, 2000. 27

[6] J. Cendrowska. Prism: an algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, (27):349–370, 1987. 27

[7] Van de Merckt. Decision trees in numerical attribute spaces. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1016–1021, 1993. 28

[8] D. Dubois, H. Prade, and E. Rannou. User driven summarization of data based on gradual rules. *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, (Vol2):839–844, 1997. 27

[9] Hernandez E. and Recasens J. A reformulation of entropy in the presence of indistinguishability operators. *Fuzzy sets and Systems*, 128(2):185–196, 2002. 36

[10] Z. Elouedi, k. Mellouli, and P. Smets. Decision trees using the belief function theory. In *Proceedings of IPMU'2000*, 2000. 28, 33, 36

[11] E. Hernandez and J. Recasens. A fuzzy sequential covering algorithm for the generation of rules. In *Actas del X Congreso sobre tecnologias y logica fuzzy (ESTYLF'00)*, pages 319–322, 2000. 27

[12] F. Herrera and M. Lozano. Fuzzy genetic algorithms: Issues and models. 27

[13] Jacas J. and Recasens J. Fuzzy t-transitive relations: eigenvectors and generators. *Fuzzy sets and systems*, (72):147–154, 1995. 33

[14] C. Janikow. Fuzzy decision trees: issues and methods. *IEEE Transactions on Systems, Man and Cybernetics*, 28(1):1–14, 1998. 28, 32, 36
[15] G Klir and M. Wierman. *Uncertainty based information. Elements of generalized information theory*. Physica-Verlag, 1999. 31
[16] R. Kruse and D. Nauck. Learning methods for fuzzy systems. In *Proc. Fuzzy–Neuro–Systeme'95*, pages 7–22, 1995. 27
[17] Liu, Wang, Hong, and Tseng. A fuzzy inductive learning strategy for modular rules. *Fuzzy Sets and Systems*, (103):91–105, 1999.
[18] P.E Maher and D. Saint-Clair. Uncertain reasoning in an id3 machine learning framework. In *2nd IEEE Conf. on Fuzzy Systems*, pages 7–12, 1993. 28
[19] J.L. Marichel. *Aggregation operators for multicriteria decision aid*. PhD thesis, Universite de Liege, 1998. 31
[20] M. Michalski. On the quasi-minimal solution of the general covering problem. In *Proc. of the first International Symposium on Information Processing*, pages 125–128, 1969. 27
[21] T. Mitchell. *Machine Learning*. Mc. Graw-Hill, 1997. 27
[22] J.R Quinlan. Induction of decision trees. *Machine Learning*, pages 81–106, 1986. 28, 29, 35
[23] J.R Quinlan. Probabilistic decision trees. In Morgan Kaufmann, editor, *Machine Learning*, pages 140–152, 1990. 28
[24] J.R Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann, 1993. 28
[25] Mantaras R. A distance-based attribute selection measure for decision tree induction. *Machine learning*, 6(1):81–92, 1991. 28
[26] J. Rives. Fid3: fuzzy decision tree. In *Proc. IEEE*, pages 457–462, 1990. 33, 35
[27] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976. 28
[28] C. Shannon and W. Weaber. *The mathematical theory of communication*. University of Illinois Press, 1964. 28
[29] M. Umano et al. Fuzzy decision trees by using fuzzy id3 algorithm and its application to diagnosis systems. In *Proceedings 3rd IEEE International Conference on Fuzzy Systems*, pages 2113–2118, 1994. 28
[30] P.E. Utgoff and J.A. Clouse. A kolmogorov-smirnoff metric for decision tree induction. Technical Report 96-3, University of Massachusets, 1996. 28
[31] L. Wang and J. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics*, (22(6)):1414–1427, 1992. 27
[32] R. Weber. Fuzzy-id3: a class of methods for automatic knowledge acquisition. In *Proceedings 2nd International Conference on Fuzzy Logic and Neural Networks*, pages 265–268, 1992. 28
[33] R. Yager. On linguistic summaries of data. In *Piatetsky-Shapiro and W.J.Frwaley (eds): Knowledge discovery in databases.MIT Press*, pages 347–363, 1991.
[34] R. Yager. On linguistic summaries of data. In *In Piatetsky-Shapiro and W.J.Frwaley (eds): Knowledge discovery in databases.MIT Press*, pages 347–363, 1991. 27
[35] R. Yager. Fuzzy summaries in database mining. In *Proc. FUZZ-IEEE*, pages 265–269, 1995. 27
[36] R. Yager and D. Rasmussen. Summary-sql: a fuzzy tool for data mining. *Intelligent Data Analysis*, ((1)1), 1997. 27
[37] R. Yager and Rubinson T. Linguistic summaries of databases. In *Proc. IEEE Conference on Decision and Control*, pages 1094–1097, 1981. 27
[38] Y. Yuan and Shaw M. Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, (69):125–139, 1995. 28, 33, 35

[39] L. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning i, ii and iii. In *Fuzzy Sets and Applications: selected papers by L.A.Zadeh*, pages 218–366, 1976. 27

[40] L. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, (4(2)):103–111, 1996. 27

Combining Rule Weight Learning and Rule Selection to Obtain Simpler and More Accurate Linguistic Fuzzy Models

Rafael Alcalá[1], Oscar Cordón[2], and Francisco Herrera[2]

[1] Department of Computer Science, University of Jaén
E-23071 Jaén, Spain
alcala@ujaen.es
[2] Department of Computer Science and A.I., University of Granada
E-18071 Granada, Spain
{ocordon,herrera}@decsai.ugr.es

Abstract. In complex multidimensional problems with a highly nonlinear input-output relation, inconsistent or redundant rules can be found in the fuzzy model rule base, which can result in a loss of accuracy and interpretability. Moreover, the rules could not cooperate in the best possible way.
It is known that the use of rule weights as a local tuning of linguistic rules, enables the linguistic fuzzy models to cope with inefficient and/or redundant rules and thereby enhances the robustness, flexibility and system modeling capability. On the other hand, rule selection performs a simplification of the previously identified fuzzy rule base, removing inefficient and/or redundant rules in order to improve the cooperation among them. Since both approaches are not isolated and they have complementary characteristics, they could be combined among them. In this work, we analyze the hybridization of both techniques to obtain simpler and more accurate linguistic fuzzy models.

1 Introduction

One of the problems associated with Linguistic Fuzzy Modeling is its lack of accuracy when modeling some complex systems. It is due to the inflexibility of the concept of linguistic variable, which imposes hard restrictions to the fuzzy rule structure [2]. Therefore, in this kind of modeling the *accuracy* and the *interpretability* of the obtained model are contradictory properties directly depending on the learning process and/or the model structure.

Furthermore, in complex multidimensional problems with highly nonlinear input-output relations many redundant, inconsistent and conflicting rules are usually found in the obtained rule base, which is detrimental to the linguistic fuzzy model performance and interpretability. In any case, these rules could not cooperate appropriately.

To overcome these drawbacks, many different possibilities to improve the Linguistic Fuzzy Modeling have been considered in the specialized literature [4].

J. Lawry, J. Shanahan, A. Ralescu (Eds.): Modelling with Words, LNAI 2873, pp. 44–64, 2003.
© Springer-Verlag Berlin Heidelberg 2003

All of these approaches share the common idea of improving the way in which the linguistic fuzzy model performs the interpolative reasoning by inducing a better cooperation among the rules in the learned model. This rule cooperation may be encouraged acting on three different model components: the data base, the rule base and the whole knowledge base (KB). Focusing on the rule base, there are different ways to induce rule cooperation acting on that component:

- Rule selection [6, 15, 17, 19, 20, 22, 28, 29, 30, 34]: It involves obtaining an optimized subset of rules from a previous rule base by selecting some of them.
- Multiple rule consequent learning [10, 24]: This approach allows the rule base to present rules where each combination of antecedents may have two or more consequents associated when it is necessary.
- Weighted linguistic rule learning [7, 21, 25, 35]: It is based on including an additional parameter for each rule that indicates its importance degree in the inference process, instead of considering all rules equally important as in the usual case.
- Rule cooperation [3, 31]: This approach follows the primary objective of inducing a better cooperation among the linguistic rules. To do so, the rule base design is made using global criteria that jointly consider the action of the different rules.

Two of the previous approaches, the weighted linguistic rule learning (accuracy purposes) and the rule selection (interpretability/simplicity purposes), present complementary characteristics. On the one hand, it is known that the use of rule weights as a local tuning of linguistic rules, enables the linguistic fuzzy models to cope with inefficient and/or redundant rules and thereby enhances the robustness, flexibility and system modeling capability. On the other hand, rule selection performs a simplification of the previously identified fuzzy rule base, removing inefficient and/or redundant rules in order to improve the cooperation among them. Furthermore, reducing the model complexity is a way to improve the system readability, i.e., a compact system with few rules requires a minor effort to be interpreted. Since both approaches are not isolated and they have complementary characteristics, they could be combined.

In this work, we analyze the hybridization of both techniques to obtain simpler and more accurate linguistic fuzzy models. To select the subset of rules with the best cooperation and the weights associated to them, different search techniques could be considered [26]. In this contribution, we will consider a Genetic Algorithm (GA) [18, 23] for this purpose. The proposal has been tested with two different real-world problems achieving good results.

This contribution proposes the use of weighted fuzzy rules and rule selection to improve simple linguistic fuzzy models. *This can be intended as a meta-method over any other linguistic rule generation method, developed to obtain simpler linguistic fuzzy models by only selecting the rules with a good cooperation while the use of rule weights improves the way in which they interact.* Depending on the combination of this technique with different fuzzy rule learning methods,

different learning approaches arise. In this work, we will consider the Wang and Mendel's method [32] and an extension of this method to obtain double-consequent fuzzy rules [10] as the initial linguistic rule generation methods.

This contribution is arranged in the following way. In Sections 2 and 3, the use of rule weights and rule selection is analyzed in depth, considering them as two complementary ways to improve the linguistic model performance. Sections 4 and 5 present the proposed learning strategy and the evolutionary optimization process performing the rule selection together with the rule weight derivation. Experimental results are shown in Section 6. In Section 7, some concluding remarks are pointed out. Finally, the double-consequent rule structure is presented in Appendix A.

2 Weighted Linguistic Rules

Using rule weights [7, 21, 25, 35] has been usually considered to improve the way in which the rules interact, improving the accuracy of the learned model. In this way, rule weights involve an effective extension of the conventional fuzzy reasoning that allows the tuning of the system to be developed at the rule level [7, 25]. It is clear that considering rule weights will improve the capability of the model to perform the *interpolative reasoning* and, thus, its performance. This is one of the most interesting features of fuzzy rule-based systems (FRBSs) and plays a key role in their high performance, being a consequence of the cooperative action of the linguistic rules existing in the fuzzy rule base.

Weighted linguistic models are less interpretable than the classical ones but, in any case, these kinds of FRBSs can be interpreted to a high degree, and also make use of human knowledge and a deductive process. When weights are applied to complete rules, the corresponding weight is used to modulate the firing strength of a rule in the process of computing the defuzzified value. For human beings, it is very near to consider this weight as an importance degree associated to the rule, determining how this rule interacts with its neighboring ones. We will follow this approach, since the interpretability of the system is appropriately maintained. In addition, we will only consider weight values in $[0, 1]$ since it preserves the model readability. In this way, the use of rule weights represents an ideal framework to extend the linguistic model structure when we search for a trade-off between accuracy and interpretability.

2.1 Weighted Rule Structure and Inference System

As we have said, rule weights will be applied to complete rules. In order to do so, we will follow the weighted rule structure and the Inference System proposed in [25]:

$$\textbf{IF } X_1 \textbf{ is } A_1 \textbf{ and } \ldots \textbf{ and } X_n \textbf{ is } A_n \textbf{ THEN } Y \textbf{ is } B \textit{ with } [w],$$

where X_i (Y) are the input (output) linguistic variables, A_i (B) are the linguistic labels used in the input (output) variables, w is the real-valued rule weight, and *with* is the operator modeling the weighting of a rule.

With this structure, the fuzzy reasoning must be extended. The classical approach is to infer with the FITA (First Infer, Then Aggregate) scheme [11] considering the matching degree of the fired rules. In this contribution, the *Center of Gravity weighted by the matching degree* will be used as the defuzzification strategy [11]:

$$y_0 = \frac{\sum_i h_i \cdot w_i \cdot P_i}{\sum_i h_i \cdot w_i},$$

with y_0 being the crisp value obtained from the defuzzification process, h_i being the matching degree of the i-th rule, w_i being the weight associated to it, and P_i being the characteristic value —*Center of Gravity*— of the output fuzzy set inferred from that rule, B'_i. On the other hand, we have selected the singleton fuzzification and the *minimum t-norm* playing the role of the implication and conjunctive operators.

A simple approximation for weighted rule learning would consists of considering an optimization technique (e.g., GAs [18, 23]) to derive the associated weights of a previously obtained set of rules.

2.2 Use of Rule Weights for Implicit Rule Selection

The use of rule weights as a local tuning of linguistic rules, enables the linguistic fuzzy models to cope with inefficient and/or redundant rules and thereby enhances the robustness, flexibility and system modeling capability [25]. Hence the ability of this technique to indicate the interaction level of each rule with the remaining ones is considered, improving the global cooperation. In this way, when we start from a previous set of rules, inefficient or redundant rules could be removed by assigning a zero weight to each of them, i.e., an *implicit rule selection* could be performed.

However, weights close to zero are usually obtained from the derivation process, practically avoiding the effects of such rules but maintaining them in the KB. It is due to the large search space tackled by this process, and can not be solved by removing these rules since in some cases they could be important rules with a low interaction level. Moreover, the obtained weights are highly dependent among them and to remove rules with low weight after applying the weighting algorithm (and then normalizing) provokes very bad accuracy. On the other hand, redundant, inconsistent and conflicting rules could be weighted as important rules if their neighbors are incorrectly weighted. In most cases, the algorithm has not the ability to avoid getting stuck at local optima removing (or weighting with low weights) these kinds of rules. Therefore, rule weighting processes could be improved considering any complementary technique that directly determines what rules should be removed.

2.3 An Example of a Learning Process for Weighted FRBSs

An example for weighted rule learning would consist of the following two steps —we will use this process in our experiments for comparison purposes—:

1. Firstly, a preliminary fuzzy rule set is derived considering a specific generation process. In this work, the generation process proposed by Wang and Mendel [32] is considered.
2. Then, a learning algorithm is used to derive the associated weights of the previously obtained rules. A real-coded GA where each gene indicates the corresponding rule weight may be considered as learning algorithm. The stochastic universal sampling procedure together with an elitist selection scheme —using the Mean Square Error (MSE) as fitness— and the max-min-arithmetical crossover [16] (see Sect. 5.3) together with the uniform mutation operator can be used.

3 Selecting Cooperative Rules

In complex multidimensional problems with highly nonlinear input-output relations many redundant, inconsistent and conflicting rules are usually found in the obtained rule base (especially in the case when they are generated by only considering expert's knowledge). On the other hand, in high-dimensional problems, the number of rules in the rule base grows exponentially as more inputs are added. A large rule set might contain many redundant, inconsistent and conflicting rules. These kinds of rules are detrimental to the model performance and interpretability.

Rule selection methods directly aggregate multiple rules and/or select a subset of rules from a given fuzzy rule set in order to minimize the number of rules while at the same time maintaining (or even improving) the system performance. Inconsistent and conflicting rules that degrade the performance are eliminated, thus obtaining a more cooperative fuzzy rule set and therefore involving a potential improvement of the system accuracy. Moreover, in many cases the accuracy is not the only requirement of the model and the interpretability becomes an important aspect. Reducing the model complexity is a way to improve the system readability, i.e., a compact system with few rules requires a minor effort to be interpreted.

Rule reduction methods have been formulated using Neural Networks, clustering techniques and orthogonal transformation methods, and algorithms based on similarity measures, among others [6, 15, 28, 29, 30, 34]. In [8], a different approach was proposed which attempted to reduce the growth of the rule base by transforming elemental fuzzy rules into DNF-form.

3.1 Considering a Genetic Approach

Using GAs to search for an optimized subset of rules is motivated in the following situations:

- the integration of an expert rule set and a set of fuzzy rules extracted by means of automated learning methods [17],
- the selection of a cooperative set of rules from a candidate fuzzy rule set [9, 10, 13, 19, 20, 22],

- the selection of rules from a given KB together with the selection of the appropriate labels for the consequent variables [5],
- the selection of rules together with the tuning of membership functions by coding all of them (rules and parameters) in a chromosome [14], and
- the derivation of compact fuzzy models through complexity reduction combining fuzzy clustering, rule reduction by orthogonal techniques, similarity driving simplification and genetic optimization [27].

Two of them are of particular interest in our case, the second and the fourth. In this work, we propose the selection of a cooperative set of rules from a candidate fuzzy rule set together with the learning of rule weights coding all of them (rules and weights) in a chromosome. This pursues the following aims:

- To improve the linguistic model accuracy selecting the set of rules best cooperating while a local tuning of rules is performed to improve the interaction among them.
- To obtain simpler, and thus easily understandable, linguistic models by removing unnecessary rules.

3.2 An Example of Rule Selection Process

A simple example to select the subset of rules best cooperating is the selection process proposed in [17] —we will use this process in our experiments for comparison purposes—. Of course, we are assuming the previous existence of a set of rules.

It is based on a binary coded GA where each gene indicates whether a rule is considered or not to belong to the final fuzzy rule base (alleles '1' or '0', respectively). The stochastic universal sampling procedure [1] together with an elitist selection scheme and the two-point crossover together with the uniform mutation operators are used, and the Mean Squared Error (MSE) is considered as fitness function. The MSE for a whole rule base RB, calculated over the example set E, is defined as:

$$MSE\,(E, RB) = \frac{\sum_{e^l \in E} (ey^l - s(ex^l))^2}{2 \cdot |E|}$$

with $s(ex^l)$ being the output value obtained considering RB when the input variable values are $ex^l = (ex^l_1, \ldots, ex^l_n)$ and ey^l is the known desired value.

In this way, considering the m rules contained in the preliminary/candidate rule set, the chromosome $C = (c_1, \ldots, c_m)$ represents a subset of rules composing the final rule base, such that:

$$IF\ c_i = 1\ THEN\ (R_i \in RB)\ ELSE\ (R_i \notin RB)\ ,$$

with R_i being the corresponding i-th rule in the candidate rule set and RB being the final rule base.

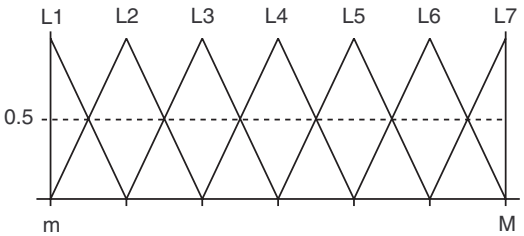


Fig. 1. Graphical representation of a possible fuzzy partition

4 Combining Rule Weight Derivation and Rule Selection

As discussed above, the hybridization of the rule weight derivation and the rule selection processes could result in important improvements of the system accuracy, obtaining simpler, and thus easily understandable, linguistic models by removing unnecessary rules. In this way, the interpretability is maintained to an acceptable level.

To generate linguistic models combining both approaches, we may follow an operation mode similar to the learning approach proposed in Section 2.3, by including the rule selection in the second step. Therefore, after performing the first step, where an initial set of numerous promising rules is generated, the two following tasks must be performed:

- Genetic selection of a subset of rules with good cooperation.
- Genetic derivation of the weights associated to these rules.

We will consider symmetrical fuzzy partitions of triangular-shaped membership functions (see Figure 1) to derive a candidate linguistic rule set (first step). With this aim, two different but similar approaches will be considered depending on the desired rule structure of the obtained models:

- *Based on the classical rule structure.* A preliminary fuzzy rule set based on linguistic rules with the usual structure is derived considering a specific generation process. In this work, the well-known *ad hoc data-driven generation method*[1] proposed by Wang and Mendel [32] is considered.
- *Based on the double-consequent rule structure.* Taking the first step of the Accurate Linguistic Modeling (ALM) methodology [10] and considering the generation process proposed by Wang and Mendel [32], the process involves dividing the input and output spaces into different fuzzy regions, generating the rule best covering each example, and finally selecting the two rules with the highest covering degree for each fuzzy input subspace (if there is more than a single rule on it). The double-consequent rule structure is presented in Appendix A. Notice that the preliminary rule base including

[1] A family of efficient and simple methods guided by covering criteria of the data in the example set

double-consequent rules will be preprocessed before giving it as input to the selection and rule weight derivation process. In this way, each double-consequent rule will be decomposed into two simple rules in order to allow the later process to assign rule weights to each consequent and to select the consequent/s best cooperating with the remaining rules. Thus, if one of the two simple rules obtained from decomposing a double-consequent rule is removed by the selection process, the corresponding fuzzy input subspace will have just a single consequent associated.

To select the subset of rules with the best cooperation and the weights associated to them (second step), we will consider a GA coding all of them (rules and weights) in a chromosome. The proposed algorithm is presented in the following section.

5 Genetic Weight Derivation and Rule Selection Process

The proposed GA must consider the use of binary (rule selection) and real values (weight derivation) in the same coding scheme. As we will see, a double coding scheme will be considered using integer and real coded genes, and therefore appropriate genetic operators for each part of the chromosome are considered.

In the following, the main characteristics of this genetic approach are presented.

5.1 Coding Scheme and Initial Gene Pool

A double coding scheme ($C = C_1 + C_2$) for both *rule selection* and *weight derivation* is used:

- For the C_1 part, the coding scheme generates binary-coded strings of length m (the number of single fuzzy rules in the previously derived rule set). Depending on whether a rule is selected or not, the alleles '1' or '0' will be respectively assigned to the corresponding gene. Thus, the corresponding part C_1^p for the p-th chromosome will be a binary vector representing the subset of rules finally obtained.
- For the C_2 part, the coding scheme generates real-coded strings of length m. The value of each gene indicates the weight used in the corresponding rule. They may take any value in the interval $[0, 1]$. Now, the corresponding part C_2^p for the p-th chromosome will be a real-valued vector representing the weights associated to the fuzzy rules considered.

Hence, a chromosome C^p is coded in the following way:

$$\begin{aligned} C_1^p &= (c_{11}^p, \ldots, c_{1m}^p) \mid c_{1i}^p \in \{0, 1\}, \\ C_2^p &= (c_{21}^p, \ldots, c_{2m}^p) \mid c_{2i}^p \in [0, 1], \\ C^p &= C_1^p C_2^p. \end{aligned}$$

The initial pool is obtained with an individual having all genes with value '1' in both parts, and the remaining individuals generated at random:

$$\forall k \in \{1, \ldots, m\},\ c_{1k}^1 = 1 \text{ and } c_{2k}^1 = 1.0\ .$$

5.2 Evaluating the Chromosome

To evaluate the p-th chromosome, we will follow an accuracy oriented policy by using the following fitness function, $F(C^p)$::

$$F(C^p) = MSE\,(E, RB(C^p)) = \frac{\sum_{e^l \in E} (ey^l - s(ex^l))^2}{2 \cdot |E|}.$$

with E being the set of training data and $s(ex^l)$ being the output value obtained from the rule base encoded in C^p when the input $ex^l = (ex_1^l, \ldots, ex_n^l)$ is presented, and ey^l being the known desired output. In this case, $s(ex^l)$ will be computed following the extended fuzzy reasoning process in order to consider the rule weight influence.

5.3 Genetic Components

The different components of the GA are introduced as follows:

Selection and Reproduction

The selection probability calculation follows linear ranking [1]. Chromosomes are sorted in order of raw fitness, and then the selection probability of each chromosome, $p_s(C^p)$, is computed according to its rank, $rank(C^p)$ —where $rank(C^{best}) = 1$—, by using the following non-increasing assignment function:

$$p_s(C^p) = \frac{1}{N_C} \cdot (\eta_{max} - (\eta_{max} - \eta_{min}) \cdot \frac{rank(C^p) - 1}{N_C - 1}),$$

where N_C is the number of chromosomes in the population and $\eta_{min} \in [0, 1]$ specifies the expected number of copies for the worst chromosome (the best one has $\eta_{max} = 2 - \eta_{min}$ expected copies). In the experiments developed in this paper, $\eta_{min} = 0.75$.

The classical **generational** [23] scheme has been used in this algorithm. In this way, linear ranking is performed along with *stochastic universal sampling* [1]. This procedure guarantees that the number of copies of any chromosome is bounded by the floor and by the ceiling of its expected number of copies. Together with the Baker's stochastic universal sampling procedure, an elitist mechanism (that ensures to maintain the best individual of the previous generation) has been considered.

Genetic Operators: Crossover and Mutation

Due to the different nature of the chromosomes involved in the rule base definition process, different operators working on each part, C_1 and C_2, are required. Taking into account this aspect, the following operators are considered.

The crossover operator will depend on the chromosome part where it is applied: in the C_1 part, the standard two-point crossover is used, whilst in the C_2 part, the max-min-arithmetical crossover [16] is considered.

The two-point crossover involves interchanging the fragments of the parents contained between two points selected at random (resulting two descendents). On the other hand, using the max-min-arithmetical crossover in the second parts, if $C_2^v = (c_{21}^v, \ldots, c_{2k}^v, \ldots, c_{2m}^v)$ and $C_2^w = (c_{21}^w, \ldots, c_{2k}^w, \ldots, c_{2m}^w)$ are going to be crossed, the resulting descendents are the two best of the next four offspring:

$$
\begin{aligned}
&O_2^1 = aC_2^w + (1-a)C_2^v,\\
&O_2^2 = aC_2^v + (1-a)C_2^w,\\
&O_2^3 \text{ with } c_{2k}^3 = \min\{c_{2k}^v, c_{2k}^w\},\\
&O_2^4 \text{ with } c_{2k}^4 = \max\{c_{2k}^v, c_{2k}^w\},
\end{aligned}
$$

with $a \in [0,1]$ being a constant parameter chosen by the GA designer. The max-min-arithmetical crossover was proposed to be used in real-coded spaces aiming to obtain a good balance between exploration and exploitation. This crossover operator obtains four well distributed descendents, one with the higher values of both parents, one with the lower values of both parents and two between the values of both parents (one nearest of the first parent and one nearest of the second parent). The two best are selected to replace the parents performing a good exploration/exploitation of the search space.

In this case, eight offspring are generated by combining the two from the C_1 part (two-point crossover) with the four ones from the C_2 part (max-min-arithmetical crossover). The two best offspring so obtained replace the two corresponding parents in the population.

As regards the mutation operator, it flips the gene value in the C_1 part and takes a value at random within the interval $[0,1]$ for the corresponding gene in the C_2 part.

Fig. 2 shows the application scope of these operators.

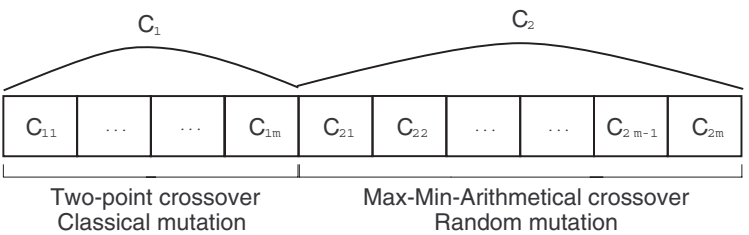


Fig. 2. Genetic representation and operators' application scope

Table 1. Methods considered for comparison

Ref.	Method	Description
[32]	**WM**	A well-known ad hoc data-driven method to obtain simple rules
[10]	**DC**	A method to obtain double-consequent rules (first step of ALM)
[10, 17]	**S**	Rule selection GA (second step of ALM or the WS C_1 part)
—	**W**	Weighted rule derivation GA (the WS C_2 part)
—	**WS**	The proposed GA performing weight derivation and rule selection

6 Experiments

In this section, we will analyze the performance of the linguistic fuzzy models generated from the proposed genetic weight derivation and rule selection process (see Section 5), when solving two different real-world problems [12]. The first presents *noise and strong nonlinearities* and the second presents four input variables, and therefore *a large search space*.

Two different approaches have been considered to obtain the initial set of candidate rules to be weighted and/or selected (see Section 4): the Wang and Mendel's method (WM) [32] and an extension of this method to obtain double-consequent fuzzy rules (DC) based on the ALM methodology [10]. In order to see the advantages of the combined action of the rule weight derivation and the rule selection, three different studies have been performed from both approaches: only considering rule selection (S), only considering rule weights (W) and considering both together, rule weights and rule selection (WS) —the algorithm proposed in this work—. Table 1 presents a short description of the methods considered for this study.

With respect to the fuzzy reasoning method used, we have selected the *minimum t-norm* playing the role of the implication and conjunctive operators, and the *center of gravity weighted by the matching* strategy acting as the defuzzification operator [11].

The values of the parameters used in all of these experiments are presented as follows[2]: 61 individuals, 1,000 generations, 0.6 as crossover probability, 0.2 as mutation probability per chromosome, and 0.35 for the a factor in the max-min-arithmetical crossover.

6.1 Estimating the Length of Low Voltage Lines

For an electric company, it may be of interest to measure the maintenance costs of its own electricity lines. These estimations could be useful to allow them to justify their expenses. However, in some cases these costs can not be directly calculated. The problem comes when trying to compute the maintenance costs of low voltage

[2] With these values we have tried to ease the comparisons selecting standard values for the common parameters that work well in most cases instead of searching very specific values for each specific method

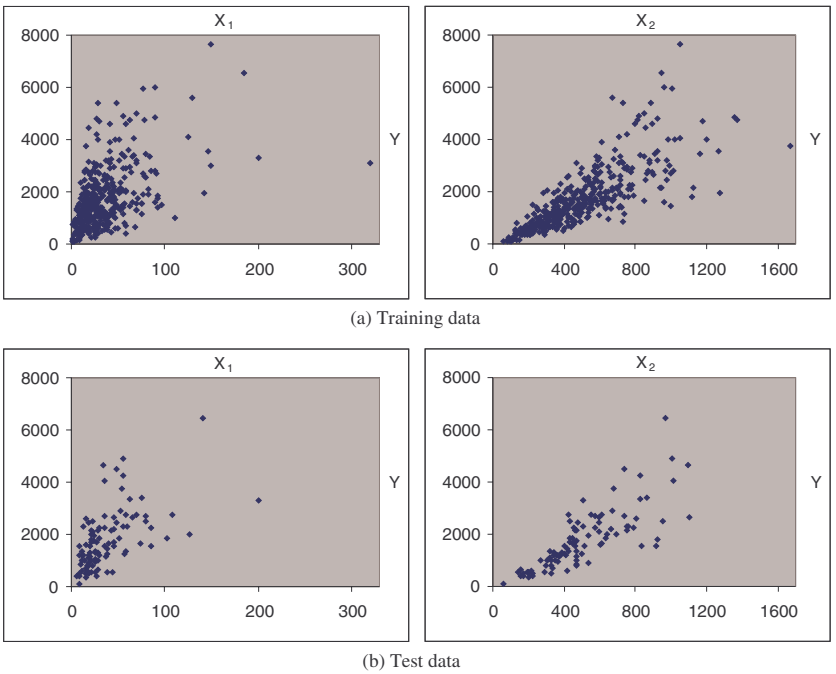


Fig. 3. (a) (X_1,Y) and (X_2,Y) dependency in the training data; (b) (X_1,Y) and (X_2,Y) dependency in the test data

lines and it is due to the following reasons. Although maintenance costs depend on the total length of the electrical line, the length of low voltage lines would be very difficult and expensive to be measured since they are contained in little villages and rural nuclei. The installation of these kinds of lines is often very intricate and, in some cases, one company can serve to more than 10,000 rural nuclei.

Due to this reason, the length of low voltage lines can not be directly computed. Therefore, it must be estimated by means of indirect models. The problem involves relating *the length of low voltage line of a certain village* with the following two variables: *the radius of the village* and *the number of users in the village* [12]. We were provided with the measured line length, the number of inhabitants and the mean distance from the center of the town to the three furthest clients in a sample of 495 rural nuclei.

In order to evaluate the models obtained from the different methods considered in this paper, this sample has been randomly divided into two subsets, the training set with 396 elements and the test set with 99 elements, the 80% and the 20% respectively. The existing dependency of the two input variables with the output variable in the training and test data sets is shown in Fig. 3 (notice that they present strong non-linearities). Both data sets considered are

Table 2. Results obtained in the length of low voltage lines estimation problem

Method	#R	← (SC+DC)	MSE_{tra}	MSE_{tst}
WM	24	–	222,654	239,962
WM-S	**17**	–	214,177	265,179
WM-W	24	–	191,577	221,583
WM-WS	20	–	**191,565**	**219,370**
Considering double consequent rules				
DC	24	(14+10)	231,132	259,973
DC-S (ALM)	**17**	(14+ 3)	155,898	178,534
DC-W	24	(14+10)	144,983	191,053
DC-WS	18	(15+ 3)	**144,656**	**177,897**

SC = Single Consequent, DC = Double Consequent.

available at `http://decsai.ugr.es/~casillas/fmlib/`. The linguistic partitions considered are comprised by seven linguistic terms with triangular-shaped fuzzy sets giving meaning to them (see Figure 1). The corresponding labels, $\{l_1, l_2, l_3, l_4, l_5, l_6, l_7\}$, stand for extremely small, very small, small, medium, large, very large, and extremely large, respectively.

The results obtained by the four methods analyzed are shown in Table 2, where $\#R$ stands for the number of rules, and MSE_{tra} and MSE_{tst} for the error obtained over the training and test data respectively. The best results are shown in boldface in each table. These results were obtained for a PENTIUM III with clock rate of 550 MHz and 128 MB of RAM. The run times for the different algorithms do not exceed 20 minutes.

Focusing on the WM approach, the model obtained from WM-WS presented the best performance, with improvements of a 14% in training and a 9% in test respect to the basic WM approach and, presenting a similar performance to the one obtained from WM-W. However, although accuracy and simplicity are contradictory requirements, four rules were eliminated respect to WM and WM-W (the second more accurate model), with this number representing the 17% of the candidate set of rules obtained from WM. The model obtaining the lowest number of rules was obtained from WM-S, but its performance was even worse than the original model obtained from WM.

Considering the DC approach, the proposed algorithm, DC-WS, obtains again the best performance, with improvements of about a 37% and a 31% in training and test, respectively respect to DC. In this case, there are significant differences between the results obtained by the approaches considering a single optimization (i.e., only rule selection, DC-S, or only rule weight derivation, DC-W) and our two-fold process. On the one hand, DC-S is only able to achieve a similar performance to DC-WS on the test error, but training is significantly worse. On the other hand, DC-W is only able to achieve a similar performance to DC-WS on training, but the test error is significantly worse. However, six rules were removed from the initial model obtained from DC considering WS. It represents a 25% of the total number of rules in DC. Furthermore, our model

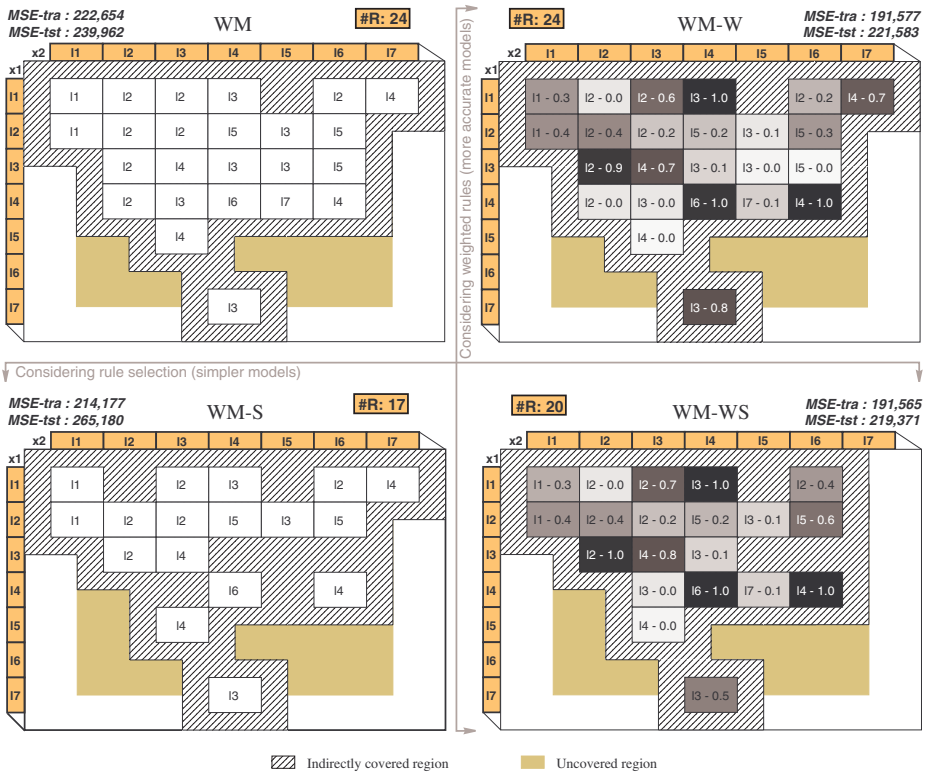


Fig. 4. Decision tables of the obtained models considering the WM approach

only presents 3 double-consequent rules respect to the 10 considered in DC and DC-W.

From the results presented in Table 2, we can say that the proposed technique is more robust than only considering weight derivation or rule selection in isolation. On the one hand, W only achieves good results by considering the WM approach. On the other hand, S only achieves good results by considering the DC approach.

The decision tables of the models obtained by the studied methods for the WM approach are presented in Figure 4. Each cell of the tables represents a fuzzy subspace/rule and contains its associated output consequent(s) —the primary and/or the secondary in importance when the DC approach is considered—, i.e., the corresponding label(s) together with its(their) respective rounded rule weight(s) when they are considered. These weights have been graphically by means of the grey colour scale, from black (1.0) to white (0.0). In this way, we can easily see the importance of a rule *with respect to their neighbors* which could help the system experts to identify important rules.

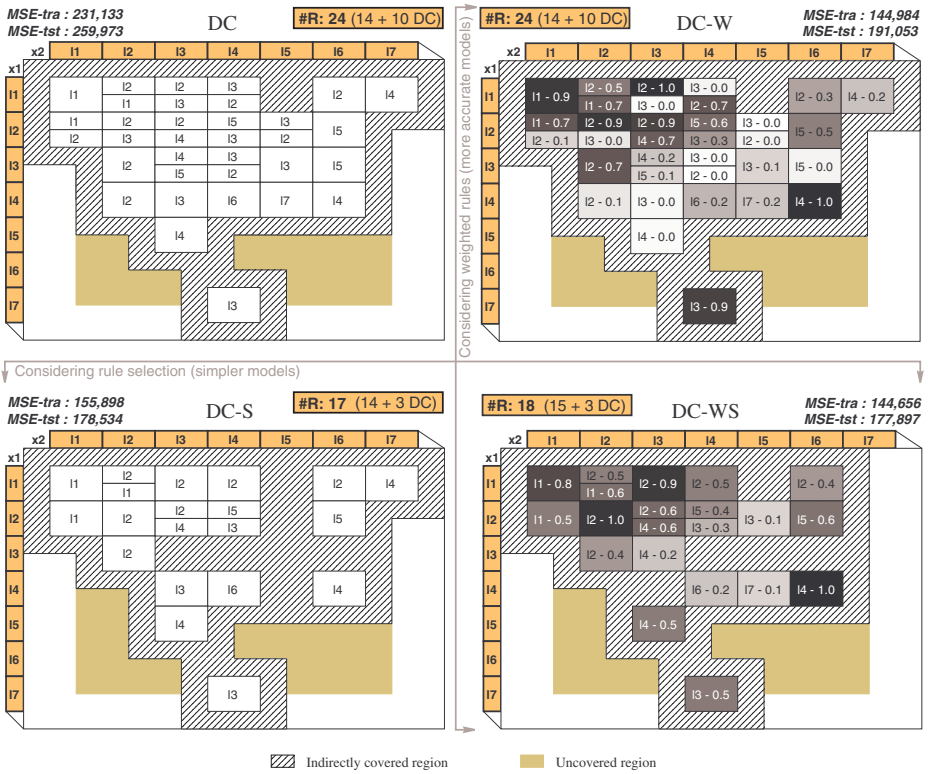


Fig. 5. Decision tables of the obtained models considering the DC approach

Notice that, the model obtained from WM-WS presents practically the same weights than that obtained from WM-W. Moreover, WM-WS and WM-S practically coincide in the selected rules to be considered in the final rule base. We can observe as some rules presenting weights very close to zero were removed in WM-WS respect to WM-W, those in the subspaces l_3-l_5, l_3-l_6 and l_4-l_2. However, as we said in Section 2.2, some rules were not removed since they were the one of their regions (that located in l_5-l_3) or since they improve the system performance interacting at low level (l_1-l_2 and l_4-l_3).

Figure 5 shows the decision tables of the models obtained by the studied methods when the DC approach is considered. Once again, there are similarities between DC-WS and DC-W (in terms of weights). However, in this case we can found some differences due to the large number of double-consequent rules considered in DC-W. On the other hand, strong similarities are found respect to the selected rules from DC-WS and DC-S. Taking into account this fact in both, WM and DC, we could say that WS inherits the accuracy characteristics of the rule weighting and the simplicity characteristics of the rule selection.

Table 3. Results obtained in the maintenance costs of medium voltage lines estimation problem

Method	#R ← (SC+DC)		MSE_{tra}	MSE_{tst}
WM	66	–	71,294	80,934
WM-S	**43**	–	57,025	59,942
WM-W	66	–	33,639	33,319
WM-WS	**43**	–	**32,476**	**32,638**
Considering double consequent rules				
DC	66	(49+17)	217,808	212,966
DC-S (ALM)	**47**	(44+ 3)	51,714	58,806
DC-W	66	(49+17)	26,377	28,637
DC-WS	51	(47+ 4)	**25,657**	**28,513**

SC = Single Consequent, DC = Double Consequent.

6.2 Estimating the Maintenance Costs of Medium Voltage Lines

Estimating the maintenance costs of the optimal installation of medium voltage electrical network in a town [12] is an interesting problem. Clearly, it is impossible to obtain this value by directly measuring it, since the medium voltage lines existing in a town have been installed incrementally, according to its own electrical needs in each moment. In this case, the consideration of models becomes the only possible solution. Moreover, the model must be able to explain how a specific value is computed for a certain town. These estimations allow electrical companies to justify their expenses. Our objective will be to relate the *maintenance costs of medium voltage line* with the following four variables: *sum of the lengths of all streets in the town*, *total area of the town*, *area that is occupied by buildings*, and *energy supply to the town*. We will deal with estimations of minimum maintenance costs based on a model of the optimal electrical network for a town in a sample of 1,059 towns.

To develop the different experiments in this contribution, the sample has been randomly divided into two subsets, the training and test ones, with an 80%-20% of the original size respectively. Thus, the training set contains 847 elements, whilst the test one is composed of 212 elements. These data sets used are available at `http://decsai.ugr.es/~casillas/fmlib/`. Five linguistic terms with triangular-shaped fuzzy sets giving meaning to them are considered for each variable (see Figure 1). In these case, the corresponding labels, $\{l_1, l_2, l_3, l_4, l_5\}$, stand for very small, small, medium, large, and very large, respectively.

The results obtained by the analyzed methods are shown in Table 3, where the same equivalences in Table 2 remain. Again, these results were obtained for a PENTIUM III with clock rate of 550 MHz and 128 MB of RAM. In this case the run times for the different methods do not exceed 65 minutes.

Considering the WM approach, the results obtained by WM-WS are the best in accuracy, with improvements of a 55% and a 60% in training and test, respect to the original simple linguistic model obtained by WM. Similar results were

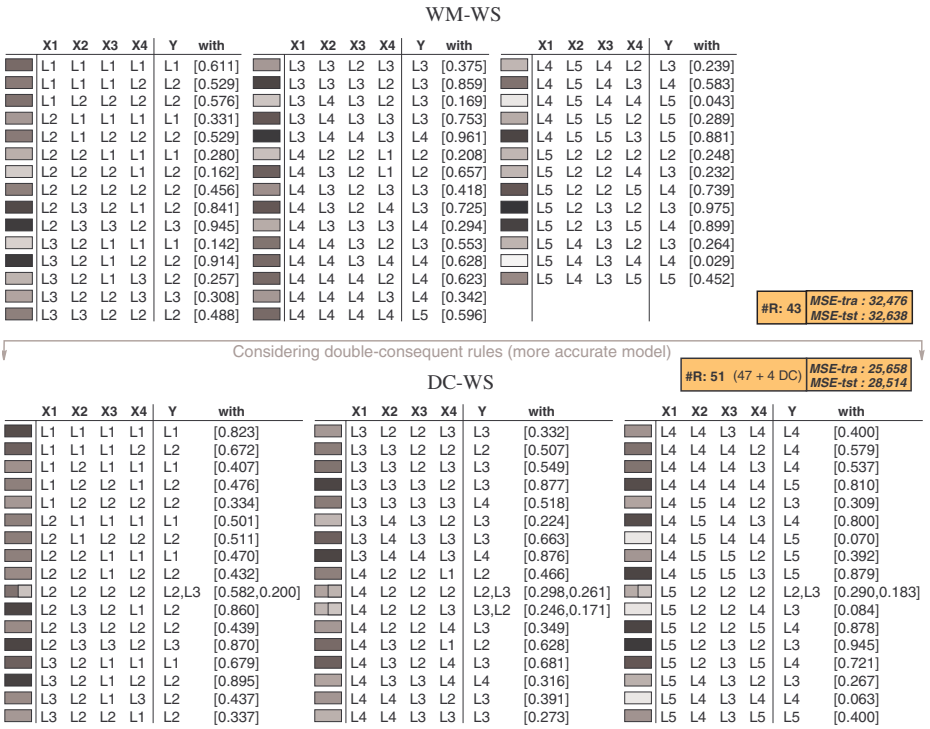


Fig. 6. Rule set of the linguistic models obtained from the proposed technique when both the WM and the DC approaches are considered

obtained by only considering rule weights, WM-W. However, the proposed algorithm presents the simplest model (in terms of the number of rules) together with WM-S, removing 23 rules (a 35%) respect to WM and WM-W, and improving WM-S about a 50% in training and test, respectively.

Focusing on the DC approach, similar results were obtained respect to DC, DC-S and DC-W. Notice that, DC-WS does not only remove 15 rules more than DC-W but it also achieves a reduction of the number of double-consequent rules, obtaining only four rules of this type.

Figure 6 represents the rule set of the linguistic models obtained from the proposed technique. In this case, each row represents a fuzzy subspace/rule and contains its associated output consequent(s) —the primary and/or the secondary in importance when the DC approach is considered—, i.e., the corresponding label(s) together with its(their) respective rounded rule weight(s). Once again, the absolute importance weight for each fuzzy rule has been graphically shown by means of the grey colour scale, from black (1.0) to white (0.0).

From the 625 (5^4) possible fuzzy rules, the obtained linguistic fuzzy models are composed of only 43 and 51, respectively. In the case of DC-WS, it only con-

tains four double-consequent rules. Notice that, all the double-consequent rules are very near in the four-dimensional space, representing a zone with a high complexity. Moreover, rules with weights close to 1 represent groups of important rules and do not usually appear alone. As in the previous problem, some similarities can be observed between the obtained models in terms of the derived weights and the selected rules, even considering different rule structures.

7 Concluding Remarks

In this work, the use of weighted linguistic fuzzy rules together with rule selection to obtain more simple and accurate linguistic fuzzy models has been proposed. To do so, a GA coding rules and weights in each chromosome has been developed with the main aim of improving the accuracy of simple linguistic fuzzy models and maintaining their interpretabiliy to an acceptable level (i.e., to obtain compact but powerful models).

In view of the obtained results, the proposed approach seems to inherit the accuracy characteristics of the rule weighting and the simplicity characteristics of the rule selection, obtaining simple but powerful linguistic fuzzy models. This is due to the following reasons:

- The ability of rule weights to indicate the interaction level of each rule with the remainder, improving the global performance of the weighted fuzzy model.
- The complementary characteristics that the use of weights and the rule selection approach present. The ability of rule selection to reduce the search space by only choosing the rules presenting a good cooperation is combined with an improvement of the rule cooperation capability by determining the appropriate interaction levels among the selected rules by the use of weights.

References

[1] Baker, J. E.: Reducing bias and inefficiency in the selection algorithm. In: Grefenstette, J. J. (Ed.), Proceedings of the 2nd International Conference on Genetic Algorithms, Lawrence Erlbaum Associates (Hillsdale, NJ, USA, 1987) 14–21 49, 52

[2] Bastian, A.: How to handle the flexibility of linguistic variables with applications. International Journal of Uncertainty, Fuzziness and Knowlegde-Based Systems **2:4** (1994) 463–484 44

[3] Casillas, J., Cordón, O., Herrera, F.: COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics **32:4** (2002) 526–537 45

[4] Casillas, J., Cordón, O., Herrera, F., Magdalena, L.: Accuracy improvements to find the balance interpretability-accuracy in fuzzy modeling: an overview. In: Casillas, J., Cordón, O., Herrera, F., Magdalena, L. (Eds.), Fuzzy modeling and the interpretability-accuracy trade-off. Part II, accuracy improvements preserving the interpretability (Physica-Verlag, Heidelberg, Germany, 2002) 3–24 44

[5] Chin, T. C., Qi, X. M.: Genetic algorithms for learning the rule base of fuzzy logic controller. Fuzzy Sets and Systems **97:1** (1998) 1–7 49
[6] Chiu, S.: Fuzzy model identification based on cluster estimation. Journal of Intelligent and Fuzzy Systems **2** (1994) 267–278 45, 48
[7] Cho, J. S., Park, D. J.: Novel fuzzy logic control based on weighting of partially inconsistent rules using neural network. Journal of Intelligent Fuzzy Systems **8** (2000) 99–110 45, 46
[8] Combs, W. E., Andrews, J. E.: Combinatorial rule explosion eliminated by a fuzzy rule configuration. IEEE Transactions on Fuzzy Systems **6:1** (1998) 1–11 48
[9] Cordón, O., Herrera, F.: A three-stage evolutionary process for learning descriptive and approximative fuzzy logic controller knowledge bases from examples. International Journal of Approximate Reasoning **17:4** (1997) 369–407 48
[10] Cordón, O., Herrera, F.: A proposal for improving the accuracy of linguistic modeling. IEEE Transactions on Fuzzy Systems **8:4** (2000) 335–344 45, 46, 48, 50, 54, 63, 64
[11] Cordón, O., Herrera, F., Peregrín, A.: Applicability of the fuzzy operators in the design of fuzzy logic controllers. Fuzzy Sets and Systems **86:1** (1997) 15–41 47, 54, 63
[12] Cordón, O., Herrera, F., Sánchez, L.: Solving electrical distribution problems using hybrid evolutionary data analysis techniques. Applied Intelligence **10** (1999) 5–24 54, 55, 59
[13] Cordón, O., del Jesús, M. J., Herrera, F.: Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods. International Journal of Intelligent Systems **13:10-11** (1998) 1025–1053 48
[14] Gómez-Skarmeta, A. F., Jiménez, F.: Fuzzy modeling with hybrid systems. Fuzzy Sets and Systems **104** (1999) 199–208 49
[15] Halgamuge, S., Glesner, M.: Neural networks in designing fuzzy systems for real world applications. Fuzzy Sets and Systems **65:1** (1994) 1–12 45, 48
[16] Herrera, F., Lozano, M., Verdegay, J. L.: Tuning fuzzy logic controllers by genetic algorithms. International Journal of Approximate Reasoning **12** (1995) 299–315 48, 53
[17] Herrera, F., Lozano, M., Verdegay, J. L.: A learning process for fuzzy control rules using genetic algorithms. Fuzzy Sets and Systems **100** (1998) 143–158 45, 48, 49, 54
[18] Holland, J. H.: Adaptation in natural and artificial systems. Ann arbor: The University of Michigan Press (1975), The MIT Press, London (1992) 45, 47
[19] Ishibuchi, H., Nozaki, K., Yamamoto, N., Tanaka, H.: Selecting fuzzy if-then rules for classification problems using genetic algorithms. IEEE Transactions on Fuzzy Systems **9:3** (1995) 260–270 45, 48
[20] Ishibuchi, H., Murata, T., Türksen, I. B.: Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. Fuzzy Sets and Systems **89** (1997) 135–150 45, 48
[21] Ishibuchi, H., Takashima, T.: Effect of rule weights in fuzzy rule-based classification systems. IEEE Transactions on Fuzzy Systems **3:3** (2001) 260–270 45, 46
[22] Krone, A., Krause, H., Slawinski, T.: A new rule reduction method for finding interpretable and small rule bases in high dimensional search spaces. Proceedings of the 9th IEEE International Conference on Fuzzy Systems (San Antonio, TX, USA, 2000) 693–699 45, 48
[23] Michalewicz, Z.: Genetic algorithms + data structures = evolution programs. Springer-Verlag (1996) 45, 47, 52

[24] Nozaki, K., Ishibuchi, H., Tanaka, H.: A simple but powerful heuristic method for generating fuzzy rules from numerical data. Fuzzy Sets and Systems 86 (1997) 251–270 45, 63, 64
[25] Pal, N. R., Pal, K.: Handling of inconsistent rules with an extended model of fuzzy reasoning. Journal of Intelligent Fuzzy Systems **7** (1999) 55–73 45, 46, 47
[26] Pardalos, P. M., Resende, M. G. C.: Handbook of applied optimization. Oxford University Press, NY (2002) 45
[27] Roubos, H., Setnes, M.: Compact fuzzy models through complexity reduction and evolutionary optimization. Proceedings of the 9th IEEE International Conference on Fuzzy Systems **2** (San Antonio, Texas, USA, 2000) 762–767 49
[28] Rovatti, R., Guerrieri, R., Baccarani, G.: Fuzzy rules optimization and logic synthesis. Proceedings of the 2nd IEEE International Conference on Fuzzy Systems **2** (San Francisco, USA, 1993) 1247–1252 45, 48
[29] Setnes, M., Babuska, R., Kaymak, U., van Nauta-Lemke, H. R.: Similarity measures in fuzzy rule base simplification. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics **28** (1998) 376–386 45, 48
[30] Setnes, M., Hellendoorn, H.: Orthogonal transforms for ordering and reduction of fuzzy rules. Proceedings of the 9th IEEE International Conference on Fuzzy Systems **2** (San Antonio, Texas, USA, 2000) 700–705 45, 48
[31] Thrift, P.: Fuzzy logic synthesis with genetic algorithms. Proceedings of the 4th International Conference on Genetic Algorithms (ICGA'91) (Morgan Kaufmann, San Mateo, CA, 1991) 509–513 45
[32] Wang, L. X., Mendel, J. M.: Generating fuzzy rules by learning from examples. IEEE Transactions on Systems, Man, and Cybernetics **22** (1992) 1414–1427 46, 48, 50, 54
[33] Yam, Y., Baranyi, P., Yang, C. T.: Reduction of fuzzy rule base via singular value decomposition. IEEE Transactions on Fuzzy Systems **7** (1999) 120–132
[34] Yen, J., Wang, L.: Simplifying fuzzy rule-based models using orthogonal transformation methods. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics **29** (1999) 13–24 45, 48
[35] Yu, W., Bien, Z.: Design of fuzzy logic controller with inconsistent rule base. Journal of Intelligent Fuzzy Systems **2** (1994) 147–159 45, 46

A Double-Consequent Rule Structure

More flexible linguistic models may be obtained by allowing them to present fuzzy rules where each combination of antecedents may have two consequents associated [10, 24]:

$$\textbf{IF } X_1 \text{ is } A_1 \text{ and } \ldots \text{ and } X_n \text{ is } A_n \textbf{ THEN } Y \text{ is } \{B_1, B_2\},$$

with X_i (Y) being the linguistic input (output) variables, A_i being the linguistic label used in the i-th input variable, and B_1 and B_2 the two linguistic terms associated to the output variable.

Since each double-consequent fuzzy rule can be decomposed into two different rules with a single consequent, the usual plain fuzzy inference system can be applied. The only restriction imposed is that the defuzzification method must consider the matching degree of the rules fired, for example, the *center of gravity weighted by the matching degree* defuzzification strategy [11] may be used.

The consideration of this structure to generate advanced linguistic models was initially proposed in [24]. Another approach, according to the ALM methodology, was also introduced in [10].

Semantics-Preserving Dimensionality Reduction in Intelligent Modelling

Qiang Shen

Centre for Intelligent Systems and their Applications
School of Informatics
The University of Edinburgh
Appleton Tower, Crichton Street
Edinburgh EH8 9LE, UK
qiangs@inf.ed.ac.uk

Abstract. The last two decades have seen the exponential increase of powerful model-building methods being called upon to solve large-scale real-world problems. However, for all their accuracy, many of the systems built using these methods tend to be black boxes. Inspecting the model is difficult, and they are often unable to provide explanations about their reasoning. This limits the trust human users can put in such systems. Modelling methods that preserve their semantics in human-readable, linguistic terms are thus very desirable. This, however, is a moot point when very complex domains are involved. Even linguistic rules can become too complicated for humans to follow if the domain they model is more complex than the human mind can handle. This chapter discusses an approach for semantics-preserving dimensionality reduction, or feature selection, that simplifies domains in the context of fuzzy or neural modelling, all the while retaining the accuracy of the respective model. The approach is described with respect to a number of example applications.

1 Introduction

With the improvement of the performance of modern computer systems, automated domain modelling methods are rapidly gaining popularity for wide ranges of applications. They have found their way into all manner of application domains: the stock market, financial customer profiling and risk assessment, industrial monitoring and control, assembly robotics, global and personal information retrieval and filtering, and even computer games. This success is easily explained by the fact that intelligent modelling systems are cost-effective when they are applicable.

The applicability of most modelling systems is, however, limited by the curse of dimensionality that imposes a ceiling on the complexity of the application domain. It is becoming more and more important to find ways to overcome this limitation, to allow the solution to more complex application problems to be automated. Dimensionality reduction is also required to improve the runtime performance of the system. For example, in industrial plant monitoring [12],

J. Lawry, J. Shanahan, A. Ralescu (Eds.): Modelling with Words, LNAI 2873, pp. 64–79, 2003.
© Springer-Verlag Berlin Heidelberg 2003

by requiring less observations per variable, the dimensionality reduced model becomes more compact and its response time decreases. The cost of obtaining data drops accordingly, as fewer connections to instrumentation need be maintained. In the meantime, the overall robustness of the system can increase, since, with fewer instruments, the chances of instrumentation malfunctions leading to spurious readings are reduced dramatically.

Inspired by such observations, numerous different dimensionality reduction methodologies have been proposed in recent years. Unfortunately, many of them remove redundancy by irretrievably destroying the original meaning of the data given for learning. This tends to create black-box models, significantly reducing, if not completely losing, the potential expressive power of the learned models for computing with clear semantics. This, in turn, leads to a lack of trust in such models. The knowledge engineers are unable to inspect the model building process, while operators feel they cannot trust apparently arbitrary decisions without being made aware of the reasoning behind them.

The work on rough set theory [7] offers an alternative, and formal, methodology that can be employed to reduce the dimensionality of datasets, as a preprocessing step to assist any chosen modelling method for learning from data. It helps select the most information rich features in a dataset, without transforming the data, all the while attempting to minimise information loss during the selection process [13]. Computationally, the approach is highly efficient, relying on simple set operations, which makes it suitable as a preprocessor for techniques that are much more complex. Unlike statistical correlation-reducing approaches [2], it requires no human input or intervention. Most importantly, it also retains the semantics of the data, which makes the resulting models more transparent to human scrutiny.

Combined with an automated intelligent modeller, say a fuzzy system or a neural network, the feature selection approach based on rough set theory can not only retain the descriptive power of the learned models, but also allow simpler system structures to reach the knowledge engineer and field operator. This helps enhance the interoperability and understandability of the resultant models and their reasoning. Drawing on the initial results previously presented in [1, 11, 12, 13, 14], this chapter demonstrates the applicability of this approach in supporting transparent fuzzy or neural modelling, with respect to a number of application domains.

The remainder of this chapter is structured as follows. The rough set-assisted feature selection mechanism is briefly summarised in section 2, followed by a series of four example applications, demonstrating how different modelling tasks can benefit from semantics-preserving dimensionality reduction. The chapter is concluded in section 4.

2 Rough Feature Selection

This section shows the basic ideas of rough sets [7] that are relevant to the present work and describes an efficient computational algorithm, named Rough Set Attribute Reduction (RSAR), for feature selection.

2.1 Rough Sets

A rough set is an approximation of a vague concept by a pair of precise concepts, called lower and upper approximations. The lower approximation is a description of the domain objects which are known with absolute certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects which possibly belong to the subset.

Rough sets have been employed to remove redundant conditional attributes from discrete-valued datasets, while retaining their information content. Central to this work is the concept of indiscernibility. Without losing generality, let $I = (U, A)$ be an information system, where U is a non-empty set of finite objects (the universe of discourse), and A is a non-empty finite set of variables such that $a : U \rightarrow V_a \ \forall a \ \in A$, V_a being the value set of variable a. In building a classification system, for example, $A = \{C \cup D\}$ where C is the set of input features and D is the set of class indices. Here, a class index $d \in D$ is itself a variable $d : U \rightarrow \{0, 1\}$ such that for $a \in U, d(a) = 1$ if a has class d and $d(a) = 0$ otherwise.

With any $P \subseteq A$ there is an associated equivalence relation $IND(P)$:

$$IND(P) = \{(x, y) \in U \times U \mid \forall\, a \in P,\ a(x) \ = \ a(y)\} \tag{1}$$

Note that this corresponds to the equivalence relation for which two objects are equivalent if and only if they have vectors of attribute values for the attributes in P. The partition of U, determined by *IND(P)* is denoted U/P, which is simply the set of equivalence classes generated by $IND(P)$.

If $(x, y) \in IND(P)$, then x and y are indiscernible by features in P. The equivalence classes of the P-indiscernibility relation are denoted $[x]_P$. Let $X \subseteq U$, the P-*lower* and P-*upper* approximations of a classical crisp set are respectively defined as:

$$\underline{P}X = \ \{x \mid [x]_P \ \subseteq \ X\} \tag{2}$$

$$\overline{P}X = \ \{x \mid [x]_P \cap X \neq \emptyset\} \tag{3}$$

Let P and Q be subsets of A, then the important concept of *positive region* is defined as:

$$POS_P(Q) = \bigcup_{X \in U/Q} \underline{P}X \tag{4}$$

Taking feature pattern based classification as an example, the positive region contains all objects of U that can be classified into classes of U/Q using the knowledge in the features of P.

2.2 Feature Dependency and Significance

The important issue here is to discover dependencies of object classes upon given features. Intuitively, a set of classes Q depends totally on a set of features P, denoted $P \Rightarrow Q$, if all class indices from Q are uniquely determined by values of features from P. Dependency can be measured in the following way [13]:

For $P,Q \subseteq A$, Q depends on P in a degree k $(0 \leq k \leq 1)$, denoted $P \Rightarrow_k Q$, if

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|U|} \tag{5}$$

where $|S|$ stands for the cardinality of set S.
If $k = 1$, Q depends totally on P; if $0 < k < 1$, Q depends partially (in a degree k) on P; and if $k = 0$, Q does not depend on P.

By calculating the change in dependency when a feature is removed from the set of considered possible features, an estimate of the significance of that feature can be obtained. The higher the change in dependency, the more significant the feature is. If the significance is 0, then the feature is dispensible. More formally, given P,Q and a feature $x \in P$, the significance of feature x upon Q is defined by

$$\sigma_P(Q, x) = \gamma_P(Q) - \gamma_{P-\{x\}}(Q) \tag{6}$$

2.3 Reducts and Feature Selection Algorithm

The selection of features is achieved by reducing the dimensionality of a given feature set, without destroying the meaning conveyed by the individual features selected. This is, in turn, achieved by comparing equivalence relations generated by sets of features with regard to the underlying object classes, in the context of classification.

Features are removed so that the reduced set provides the same quality of classification as the original. A *reduct* is formally defined as a subset R of the initial feature set C such that $\gamma_R(D) = \gamma_C(D)$. A given dataset may have many feature reduct sets, and the collection of all reducts is denoted by

$$R = \{X \mid X \subseteq C, \gamma_X(D) = \gamma_C(D)\} \tag{7}$$

The intersection of all the sets in R is called the *core*, the elements of which are those features that cannot be eliminated without introducing more contradictions to the representation of the dataset. Clearly, for feature selection, a reduct with minimum cardinality is searched for. That is, an attempt is to be made to locate a single element of the minimal reduct set $R_{min} \subseteq R$:

$$R_{min} = \{X \mid X \in R, \forall Y \in R, |X| \leq |Y|\} \tag{8}$$

A basic way of achieving this is to calculate the dependencies of all possible subsets of C. Any subset X with $\gamma_X(D) = 1$ is a reduct; the smallest subset with this property is a minimal reduct. However, for large datasets with a large feature set this method is impractical and an alternative strategy is required.

1. $R \leftarrow \{\}$
2. do
3. $\quad T \leftarrow R$
4. $\quad \forall x \in (C - R)$
5. $\quad\quad$ if $\gamma_{R \cup \{x\}}(D) > \gamma_T(D)$
6. $\quad\quad\quad T \leftarrow R \cup \{x\}$
7. $\quad R \leftarrow T$
8. until $\gamma_R(D) = \gamma_C(D)$
9. return R

Fig. 1. The RSAR feature selection algorithm

The RSAR feature selection algorithm given in Figure 1 attempts to calculate a minimal reduct without exhaustively generating all possible subsets. It starts off with an empty set and adds in turn, one at a time, those features that result in the greatest increase in $\gamma_P(Q)$, until the maximum possible value of $\gamma_P(Q)$, usually 1, results for the given dataset. Note that this method does not always generate a *minimal* reduct, as $\gamma_P(Q)$ is not a perfect heuristic. However, it does result in a close-to-minimal reduct, which is still useful in greatly reducing feature set dimensionality.

Note that RSAR works in a greedy manner, not compromising with a set of features that contains a large part of the information of the initial set. It attempts to reduce the feature set without loss of information significant to solving the problem at hand. The way it works is clearly dependent upon features being represented in nominal values. However, this does not give rise to problems in the use of the resulting learned models because the real feature values are only required to be temporarily discretised for feature selection. The learned model will use the original real-valued features directly. In this regard, it is independent of the modelling methods adopted. When used in conjunction with an explicit descriptive modelling tool, the resulting model will be defined in terms of only the significant features of the data, retaining the desirable transparency. The training process is accelerated, while the runtime operation of the system is sped up since fewer attributes are required.

3 Example Applications

To emphasise the generality of the presented approach and its independence from any specific domain, four application case studies are given here. These example experiments were carried out using different intelligent learning methods, including fuzzy and neural network modelling.

3.1 Industrial Plant Monitoring

Problem Case. This application concerns the task of monitoring a water treatment plant [12]. In the present example, the Water Treatment database comprises a set of historical data obtained over a period of 521 days, with one series

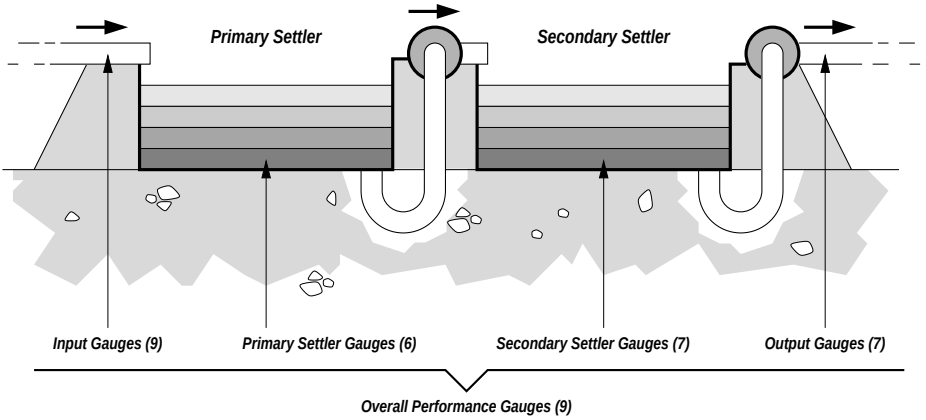


Fig. 2. Schematic diagram of the water treatment plant, indicating the number of measurements sampled at various points

of measurements per day. Thirty eight different feature values are measured per day, with one set of such measurements forming one datum. All measurements are real-valued. The goal is to implement a fuzzy reasoning system that, given this dataset of past measurements and without the benefit of an expert in the field at hand, will monitor the plant's status and produce human comprehensible explanations of identified malfunctions. This domain was chosen because of its realism. A large plant is likely to involve a number of similar features, not all of which will be essential in determining the operational status. Interrelations between features are unavoidable as the plant is a single system with interconnections, leading to a fair degree of redundancy.

The thirty eight features account for the following five aspects of the water treatment plant's operation (see Figure 2 for an illustration of this): input to plant; input to primary settler; input to secondary settler; output from plant; and overall plant performance. The operational state of the plant is represented by a boolean categorisation representing the detection of a fault. The point is to draw the operator's attention to an impending fault.

Fuzzy Modelling. In this experimental study, to obtain a model that will entail classification of the plant's operating status, the fuzzy induction algorithm first reported in [4] is used. The resulting model is represented in a set of fuzzy production rules. However, any other fuzzy rule induction method may be utilised as an alternative for model building. For the sake of completeness, an outline of the induction algorithm adopted is given below.

The algorithm generates a hyperplane of candidate fuzzy rules by fuzzifying the entire training dataset using all permutations of the input features. Thus, a system with M inputs, each of which has a domain fuzzified by f_j fuzzy sets ($1 \leq j \leq M$), the hyperplane is fuzzified into $\prod_{j=1}^{M} f_j$ M-dimensional

clusters, each representing one vector of rule preconditions. Each cluster $\underline{p} = \langle D^1, D^2, \ldots, D^M \rangle$ may lead to a fuzzy rule, provided that dataset examples support it.

To obtain a measure of what classification applies to a cluster, fuzzy min-max composition is used. The input feature pattern of each example object is fuzzified according to the fuzzy sets $\{\mu_{D^1}, \mu_{D^2}, \ldots, \mu_{D^M}\}$ that make up cluster $\underline{p}$. For each object $\underline{x} = \langle x_1, x_2, \ldots, x_M \rangle$, the following t-norm of it, with respect to cluster $\underline{p}$ and classification c, is calculated:

$$T_c^{\underline{p}} \underline{x} = \min \left(\mu_{D^1}(x_1), \mu_{D^2}(x_2), \ldots, \mu_{D^M}(x_M) \right) \tag{9}$$

To give a measure of the applicability of a classification to cluster $\underline{p}$, the maximum of all t-norms with respect to $\underline{p}$ and c is then calculated and this is dubbed an s-norm:

$$S_c^{\underline{p}} = \max \left\{ T_c^{\underline{p}} \underline{x} \mid \underline{x} \in C_c \right\} \tag{10}$$

where C_c is the set of all examples that can be classified as c. This is iterated over all possible classifications to provide a full indication of how well each cluster applies to each classification.

A cluster generates at most one rule. The rule's preconditions are the cluster's M co-ordinate fuzzy sets connected conjunctively. The conclusion is the classification attached to the cluster. Since there may be s-norms for more than one classification, it is necessary to decide on one classification for each of the clusters. Such contradictions are resolved by using the *uncertainty margin*, ε ($0 \leq \varepsilon < 1$). An s-norm assigns its classification on its cluster if and only if it is greater by at least ε than all other s-norms for that cluster. If this is not the case, the cluster is considered undecidable and no rule is generated. The uncertainty margin introduces a trade-off in the rule generation process between the size and the accuracy of the resulting model. In general, the higher ε is, the less rules are generated, but classification error may increase. A fuller treatment of this algorithm in use for descriptive modelling can be found in [4, 12].

Results. Running the RSAR algorithm on the Water Treatment dataset provided a significant reduction, with merely two features selected from the total of 38. Testing on previously unseen data resulted in a classification accuracy of 97.1%, using the fuzzy model generated by the above-mentioned rule induction method.

A comparison against a widely recognised benchmark method should help in establishing the success of the system. C4.5 [8] is a widely accepted and powerful algorithm that provides a good benchmark [6] for modelling by learning from data. The decision trees it generates allow for rapid and efficient interpretation. Yet, C4.5's decision tree for the present problem involves a total of three attributes from the dataset, as opposed to two chosen by the RSAR algorithm. In terms of classification performance, C4.5 obtains a compatible accuracy of around 96.8%.

Note that training a fuzzy system on all 38 features would be computationally prohibitive with the adopted learning algorithm. As stated previously, the benefits do not limit themselves to the learning phase; they extend to the runtime use of the learned model. By reducing the dimensionality of the data, the dimensionality of the ruleset is also decreased. This results in fewer measured features, which is very important for dynamic systems where observables are often restricted. This in turn leads to fewer connections to instrumentation and faster system responses in emergencies.

The most important benefit of using RSAR is, however, derived from its conjunctive use with the linguistically expressive fuzzy model. With the learned rules, it can provide explanations of its reasoning to the operator. This leads to increased trust in the system, as its alarms can be understood meaningfully. A model consisting of rules involving 38 features, even though they are all directly measurable and hence individually interpretable, is very difficult to understand, whilst one involving only two features is very easy to interpret.

3.2 Algae Population Estimation

Problem Case. The task of this application is to provide a system to estimate populations of various alga species (families of ubiquitous single-celled plants), say, for environment protection purposes. To derive the rules required for estimation, training samples were taken from different European rivers over the period of one year. These samples involve the following feature measurements [3]: the time of year the sample was taken, river size, water flow rate, and eight chemical concentrations known to influence alga growth patterns.

It is relatively easy to locate relations between one or two of these quantities and a species of alga. However, the process of identifying relations between different chemical elements and the population of different alga species requires expertise in chemistry and biology. It also involves well-trained personnel and microscopic examination that cannot be automated given the state of the art. Thus, the process becomes expensive and slow, even for a subset of the quantities involved here. There are complex relations at work between the variables of this application domain: algae may influence one another, as well as be influenced by the concentration of chemicals. As such, there is expected to be some redundancy in the data. This forms an important reason for the present explanation of the RSAR technique.

The dataset available for training includes 200 instances. The first three features of each instance (season, river size and flow rate) are represented as fuzzy linguistic variables. Chemical concentrations and algae population estimates are represented as continuous quantities, which are later fuzzified to generate the fuzzy model. The dataset includes a few samples with missing values. Of the 200 instances, two exhibiting mostly unknown values were removed from the dataset because of their extremely low quality.

The same modelling method as used in the last case study is employed here. For convenience and easy interpretability, each of the seven alga species were processed separately to produce seven independent fuzzy models.

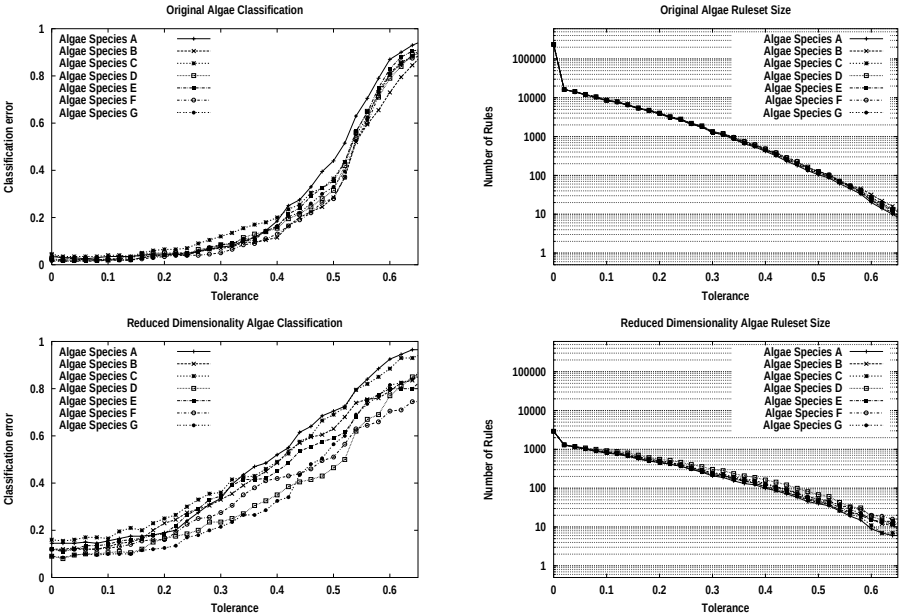


Fig. 3. Algae estimation accuracy before (top) and after (bottom) dimensionality reduction. Note that ruleset sizes are given on a logarithmic scale

Results. It is, first of all, interesting to investigate what effects dimensionality reduction may have on the runtime performance of this particular application. To show whether feature reduction has an impact on overall accuracy, fuzzy rulesets were induced from the entire, unreduced algae dataset [3], one per species. The results are shown on the top row of Figure 3. Then, RSAR was employed to reduce the dimensionality of the dataset. This resulted in a seven-feature dataset selected from the original, eleven-feature one. The results of testing the rulesets induced from this reduced dataset are illustrated on the bottom row of Figure 3.

The exact selected features were different for each alga species, although certain ones were present in all seven reduct sets, namely the Season and Concentrations 1, 4 and 7. There is a certain drop in accuracy (approximately 10%) after feature selection, which may indicate that the attribute reduction process has removed some of the necessary information. Two empirical conclusions can be drawn from the conducted experiments: first, not all features contribute the same information; second, the results obtained from random sets of features are worse than those obtained from the reduct set. The latter conclusion demonstrates that RSAR does indeed locate a relatively high-quality reduct. At the same time, by retaining the original semantics of the data and by selecting important features, the readability of the generated models is increased significantly, allowing for rules to be inspected and understood by humans.

3.3 Automated E-mail Categorisation

Problem Case. Many techniques involving electronic text categorisation describe documents by using vectors of real numbers that exhibit extremely high dimensionality: typically one value per word or pair of words in a document or corpus of documents [1]. These vector ordinates are used as preconditions to rules or similarity metrics which decide what category the document belongs to. There are several tens of thousands of such ordinates in all but the simplest applications, making effective text categorisation an extremely hard, if not intractable, problem for even the most powerful computers, unless the task is simplified.

Applying rough set attribute reduction to this problem domain provides assistance in locating those parts of text datasets that possess the necessary information, thereby reducing the amount of data to be handled by text classifiers. Additionally, it allows humans to inspect the models generated, be they fuzzy or conventional, thus understanding the rules underlying the categorisation task. This provides much needed transparency to a domain that has traditionally been characterised by opaqueness.

Data for this case study consists of folders (classes) of human-classified email messages used as training examples. In this context, a feature is the perceived importance of words and phrases within messages. The concept of importance is defined by different metrics [9], though they all are rather simple in order to keep the overall computational complexity under control. Different types of model, including fuzzy models, were built using the training data and metrics.

Results. RSAR was applied to reduce the dimensionality of the training datasets, and models using reduced features were generated to allow for comparison. The dimensionality reduction is dramatic, decreasing the dimensionality of datasets by approximately 3.5 orders of magnitude (i.e., a reduction by a factor of around 3,162). As expected, this indicates that the text categorisation domain has a very high degree of redundancy. This can be taken advantage of in an effort to improve the efficiency of the categorisation systems by using the present approach. Systems that previously needed to base decisions on thousands of features can now be reduced to those that use ten or twenty important features. This makes the results readily understandable by humans who can immediately glean the presence or absence of the selected, essential features.

The simplified, transparent fuzzy models provided a very high classification accuracy of around 90% to 98%, depending on the training data and the number of classes. This is very close to the results obtained from the unreduced data, but with the added benefit of understandability of the resulting model, and highly improved time and space requirements.

3.4 Medical Image Classification

Problem Case. Comparing normal and abnormal blood vessel structures, via the analysis of cell images, plays an important role in pathology and

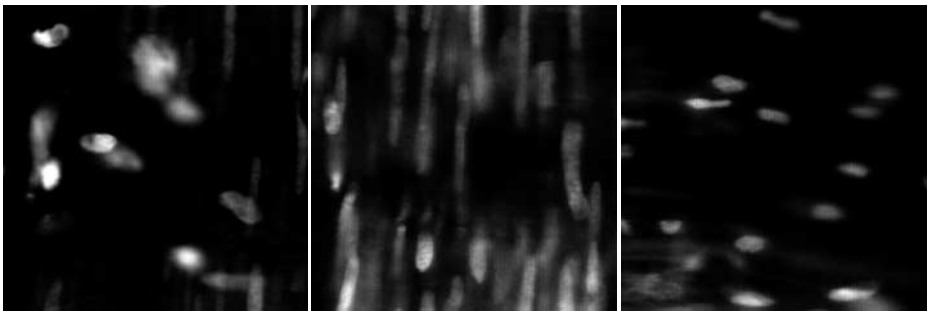
medicine [11]. Central to this analysis is the capture of the underlying features of such images. Many feature extraction methods are available to yield various kinds of characteristic descriptions of a given image. However, little knowledge is available as to what features may be most helpful to provide the discrimination power between normal and abnormal cells and between their types, while it is computationally impractical to generate many features and then to perform classification based on these features for rapid diagnosis. Generating a good number of features and selecting from them the most informative ones off-line, and then using those selected on-line is the usual way to avoid this difficulty. The problem is that the features produced ought to have an embedded meaning and such meaning should not be altered during the selection process. Therefore, this problem presents a challenging case to test the potential of RSAR.

The samples of subcutaneous blood vessels used in this work were taken from patients suffering critical limb ischaemia immediately after leg amputation. The level of amputation was always selected to be in a non-ischaemic area. The vessel segments obtained from this area represented internal proximal (normal) arteries, whilst the distal portion of the limb represented ischaemic (abnormal) ones. Images were collected using an inverted microscope, producing an image database of 318 cell images, each sized 512×512 pixels with grey levels ranging from 0 to 255. Examples of the three types of cell image taken from non-ischaemic and ischaemic resistance arteries are shown in Figure 4. Note that many of these images seem rather similar to the eye. It is therefore a difficult task for visual inspection and classification.

Neural Network Modelling. In this work, each image classifier is implemented using a traditional multi-layer feedforward artificial neural network (MFNN). To capture and represent many possible and essential characteristics of a given image, fractal models [5] are used. Note that, although these particular techniques are herein adopted to perform their respective task, the work described does not rely on them, but is generally applicable when other classification and feature extraction methods are employed.

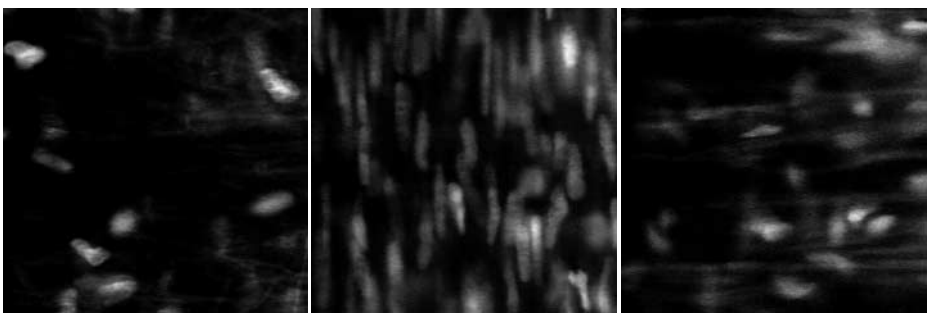
An MFNN-based classifier accomplishes classification by mapping input feature patterns onto their underlying image classes. The design of each MFNN classifier used for the present work is specified as follows. The number of nodes in its input layer is set to that of the dimensionality of the given feature set (before or after feature reduction), and the number of nodes within its output layer is set to the number of underlying classes of interest. The internal structure of the network is designed to be flexible and may contain one or two hidden layers.

The training of the classifier is essential to its runtime performance, and is here carried out using the back-propagation algorithm [10]. For this, feature patterns that represent different images, coupled with their respective underlying image class indices, are selected as the training data, with the input features being normalised into the range of 0 to 1. Here, each feature pattern consists of 9 fractal features (including 5 isotropic fractals measured on the top five finest resolutions and 4 directional fractals [11]) and the mean and standard deviation



(1) adventitial (2) SMC (3) endothelial

(a) from proximal, non-ischaemic blood vessels



(1) adventitial (2) SMC (3) endothelial

(b) from distal, ischaemic blood vessels

Fig. 4. Section cell images, where the first, second and third columns respectively show adventitial, smooth muscle and endothelial cells in proximal non-ischaemic and distal ischaemic subcutaneous blood vessels, taken from a human lower limb

(STD), with their reference numbers listed in Table 1. Note that when applying the trained classifier, only those features selected during the learning phase are required to be extracted and that no discretisation is needed but real-valued features are directly fed to the classifier.

Results. Eighty-five images selected from the image database are used for training and the remaining 233 images are employed for testing. For simplicity, only MFNNs with one hidden layer are considered.

Table 2 lists the results of using RSAR and the original full set of features. The error rate of using the five selected features is lower than that of using the full feature set. This improvement of performance is obtained by a structurally much simpler network of 10 hidden nodes, as opposed to the classifier that requires 24 hidden nodes to achieve the optimal learning. This is indicative of the power of

Table 1. Features and their reference number

Feature No.	Feature Meaning	Feature No.	Feature Meaning
1	0° direction	7	3rd finest resolution
2	45° direction	8	4th finest resolution
3	90° direction	9	5th finest resolution
4	135° direction	10	Mean
5	Finest resolution	11	STD
6	2nd finest resolution		

RSAR in helping reduce not only redundant feature measures but also the noise associated with such measurement. Also, the classifier using those five RSAR-selected features considerably outperforms those using five randomly selected features, with the average error of the latter reaching 19.1%.

Again, a comparison against a widely recognised benchmark method should help reflect the success of the system. For this, the results of rough feature selection are systematically compared to those obtained via the use of Principal Component Analysis (PCA) [2], as summarised in Table 3. Note that PCA is perhaps the most adopted dimensionality reduction technique. Although efficient, it irreversibly destroys the underlying semantics of the feature set. Therefore, in this table, for the results of using PCA, feature number $i, i \in \{1, 2, ..., 11\}$, stands for the ith principal component, i.e. the transformed feature that is corresponding to the ith largest variance.

The advantages of using RSAR are clear. Of the same dimensionality (i.e., 5), the classifier using the features selected by the rough set approach has a substantially higher classification accuracy, and this is achieved via a considerably simpler neural network. When increasing the dimensionality of principal features, the error rate generally gets reduced, but the classifier generally underperforms until almost the full set of principal features is used. The overall structural complexity of all these classifiers are more complex than that of the classifier using the five RASR-selected features. In addition, the use of those classifiers that use PCA-selected features would require many more feature measurements to achieve comparable classification results.

Table 2. Results of using rough-selected and the original full set of features

Method	Dimensionality	Features	Structure	Error
Rough	5	1,4,9,10,11	$5\times10 + 10\times6$	7.55%
Original	11	1,2,3,4,5,6,7,8,9,10,11	$11\times24 + 24\times6$	9.44%

Table 3. Results of using rough and PCA-selected features

Method	Dimensionality	Features	Structure	Error
Rough	**5**	**1,4,9,10,11**	**5×10 + 10×6**	**7.7%**
PCA	1	1	1×12 + 12×6	57.1%
	2	1,2	2×12 + 12×6	32.2%
	3	1,2,3	3×12 + 12×6	31.3%
	4	1,2,3,4	4×24 + 24×6	28.8%
	5	**1,2,3,4,5**	**5×20 + 20×6**	**18.9%**
	6	1,2,3,4,5,6	6×18 + 18×6	15.4%
	7	1,2,3,4,5,6,7	7×24 + 24×6	11.6%
	8	1,2,3,4,5,6,7,8	8×24 + 24×6	13.7%
	9	1,2,3,4,5,6,7,8,9	9×12 + 12×6	9.9%
	10	1,2,3,4,5,6,7,8,9,10	10×20 + 20×6	7.3%
	11	1,2,3,4,5,6,7,8,9,10,11	11×8 + 8×6	7.3%

4 Conclusion

Producing human-comprehensible solutions to complex real-world problems has become particularly important for many reasons, including proper validation of mission-critical computing systems and the system's ability to generate explanations and otherwise ratify their reasoning. Computing with models represented in terms of linguistic features with a prescribed meaning helps address this important point.

However, the applicability of most intelligent modelling approaches is limited by the curse of dimensionality that imposes a ceiling on the complexity of application domains. It is becoming more and more desirable to find ways to overcome this limitation. Semantics-preserving dimensionality reduction makes learned models much more transparent and comprehensible to humans, who have inherent trouble understanding high-dimensionality domains. Also, it allows systems to cope with more complex problems by lessening the obstacles of the dimensionality ceiling.

This chapter has demonstrated such an approach. It employs rough set theory to reduce the dimensionality of datasets as a preprocessing step to building an intelligent modeller that learns from historical data. Rough Set Attribute Reduction (RSAR) selects the most information rich attributes in a dataset, without transforming the data, all the while attempting to minimise information loss as regards the classification task at hand. At the same time, it provides transparency, a much-needed trait for more successful co-operation with other computing techniques.

Associated with intelligent modellers (e.g., fuzzy systems and neural networks), by simplifying the problem domain, RSAR helps enhance the transparency and maintain the accuracy of the learned models. It allows relatively simple model structures to be presented to the knowledge engineer and field operator, facilitating the examination of the quality of the results inferred by the

use of such models. This has been demonstrated in applications to four rather different problem domains, with very promising results.

The present work uses a crisp measure of feature dependency, with the dependency function returning values for sets of features in the range of 0 to 1. Work is being carried out on a fuzzified dependency function, which will return qualitative linguistic labels for use in the search for a minimal, reduced feature set. With this mechanism in place, several informative features may be chosen at one time, speeding up the feature selection process. More importantly, this would increase the explanation power of the learned model since only linguistic labels on feature dependency are used.

Acknowledgements

The author is very grateful to Alexios Chouchoulas, Knox Haggie, Richard Jensen and Changjing Shang, for their contribution, whilst taking full responsibility for the views expressed in this chapter. Thanks also go to the anonymous referees for their constructive comments which are useful in revising this work.

References

[1] Chouchoulas, A. and Shen, Q. *Rough set-aided keyword reduction for text categorisation.* Applied Artificial Intelligence **15** (2001) 843–873. 65, 73

[2] Devijver, P. and Kittler, J.. Pattern Recognition: a Statistical Approach. Prentice Hall, 1982. 65, 76

[3] ERUDIT, European Network for Fuzzy Logic and Uncertainty Modelling in Information Technology. *Protecting Rivers and Streams by Monitoring Chemical Concentrations and Algae Communities (Third International Competition)* http://www.erudit.de/erudit/activities/ic-99/problem.htm (1999). 71, 72

[4] Lozowski, A. Cholewo, T. and Zurada, J. Crisp rule extraction from perceptron network classifiers. In *Proceedings of International Conference on Neural Networks*, volume Plenary, Panel and Special Sessions, pages 94–99, 1996. 69, 70

[5] Mandelbrot, B. The Fractal Geometry of Nature. San Francisco: Freeman, 1982. 74

[6] Mitchell, T. *Machine Learning.* McGraw-Hill (1997). 70

[7] Pawlak, Z. *Rough Sets: Theoretical Aspects of Reasoning About Data.* Kluwer Academic Publishers, Dordrecht (1991). 65, 66

[8] Quinlan, J. R. *C4.5: Programs for Machine Learning.* The Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann Publishers (1993). 70

[9] van Rijsbergen, C. J. *Information Retrieval.* Butterworths, London (1979). 73

[10] Rumelhant, D. Hinton, E. and Williams, R. Learning internal representations by error propagating. In: E. Rumelhant and J. McClelland (Eds.), Parallel Distributed Processing. MIT Press, 1986. 74

[11] Shang, C. and Shen, Q. Rough feature selection for neural network based image classification. *International Journal of Image and Graphics* **2** (2002) 541-555. 65, 74

[12] Shen, Q. and Chouchoulas, A. A modular approach to generating fuzzy rules with reduced attributes for the monitoring of complex systems. *Engineering Applications of Artificial Intelligence* **13** (2000) 263–278. 64, 65, 68, 70

[13] Shen, Q. and Chouchoulas, A. A fuzzy-rough approach for generating classification rules. *Pattern Recognition*, **35** (2002) 341–354. 65, 67

[14] Shen, Q. and Chouchoulas, A. FuREAP: A fuzzy-rough estimator of algae population. *Artificial Intelligence in Engineering*, **15** (2001) 13–24. 65

Conceptual Graphs for Modelling and Computing with Generally Quantified Statements

Tru H. Cao

Faculty of Information Technology
Ho Chi Minh City University of Technology
Viet Nam
tru@dit.hcmut.edu.vn

Abstract. Conceptual graphs have been shown to be a logic that has a smooth mapping to and from natural language, in particular generally quantified statements, which is one of its advantages over predicate logic. However, classical semantics of conceptual graphs cannot deal with intrinsically vague generalized quantifiers like *few*, *many*, or *most*, which represent imprecise quantities that go beyond the capability of classical arithmetic. In this paper, we apply the fuzzy set-theoretic semantics of generalized quantifiers and formally define the semantics of generally quantified fuzzy conceptual graphs as probabilistic logic rules comprising only simple fuzzy conceptual graphs. Then we derive inference rules performed directly on fuzzy conceptual graphs with either relative or absolute quantifiers.

1 Introduction

Natural language is a principal and important means of human communication. It is used to express information as inputs to be processed by human brains then, very often, outputs are also expressed in natural language. How humans process information represented in natural language is still a challenge to science, in general, and to Artificial Intelligence, in particular. However, it is clear that, for a computer with the conventional processing paradigm to process natural language, a formalism is required. For reasoning, it is desirable that such a formalism be a logical one.

The object of natural language (specifically English) that our attention is focused on in this paper is one of *generalized quantifiers*, which can be classified into *absolute quantifiers* and *relative quantifiers*, where the quantities expressed by the latter are relative to the cardinality of a set. Examples of absolute quantifiers are *only one*, *few*, or *several*, while ones of relative quantifiers are *about 9%, half*, or *most*. In practice, there are quantifying words, e.g. *few* and *many*, that may be used with either meaning depending on the context. For instance, *few* in "*Few* people in this conference are from Asia" may mean a small number of people, while *few* in "*Few* people in the United Kingdom are from Asia" may mean a small percentage of population.

J. Lawry, J. Shanahan, A. Ralescu (Eds.): Modelling with Words, LNAI 2873, pp. 80-101, 2003.
© Springer-Verlag Berlin Heidelberg 2003

Classical predicate logic with only the existential quantifier, equivalent to the absolute quantifier *at least 1*, and the universal quantifier, equivalent to the relative quantifier *all* or *every*, cannot deal with general quantification in natural language. In fact, the logic of generalized quantifiers has been the quest and focus of significant research effort.

In [18], for instance, the semantics of generally quantified statements were defined relatively to each other, e.g., "*Few A*'s are *B*'s" if and only if "*Not many A*'s are *B*'s". As such, that work does not define the primary meaning of generalized quantifiers, which is the quantity that they express, and thus is not adequate for quantitative reasoning with them. A path-breaking work in formalizing generally quantified statements was [4], where a generally quantified set was interpreted as a set of sets of individuals. For example, "*only N A*'s" was interpreted as the set $\{X \subseteq U \mid |X \cap A| = N\}$, where U is a universe of discourse and $|S|$ denotes the cardinality of S, whence "*Only N A*'s are *B*'s" means $B \in \{X \subseteq U \mid |X \cap A| = N\}$. Still, no definition was given for words expressing imprecise quantities, like *several* or *most*; only some semantic postulates stating their relations to each other were suggested as in [18].

Meanwhile, fuzzy logic ([27]) based on fuzzy set theory ([26]) has been developed for approximate representation of, and reasoning with, imprecise information often encountered in the real world as reflected in natural language. While there are still many unresolved theoretical issues regarding the uncertainty management problem in general, and fuzzy logic in particular, fuzzy logic has been successfully applied to several areas, such as expert systems, knowledge acquisition and fusion, decision making, and information retrieval, among others.

In particular, fuzzy arithmetic has provided a basis for defining and computing with generalized quantifiers, e.g. [15, 16, 28], where they are treated in a unified way as fuzzy numbers, which are fuzzy sets on the set of real numbers. A relative quantifier Q in a statement "Q *A*'s are *B*'s" can be interpreted as the proportion of objects of type A that belong to type B, i.e., $Q = |A \cap B|/|A|$, which is a fuzzy number. Equivalently, as discussed recently in [15], it can also be interpreted as the fuzzy conditional probability, which is a fuzzy number on [0, 1], of $B(x)$ being **true** given $A(x)$ being **true** for an object x picked at random uniformly.

Although the interpretation of relative quantifiers as fuzzy probabilities was also mentioned in [29], the cardinality-based interpretation was used instead for deriving inference rules on relatively quantified statements. Meanwhile, [15] has shown that the probability-based interpretation lends a new perspective of reasoning with generalized quantifiers as probabilistic logic programming. For instance, on the basis of Jeffrey's rule ([13]), it allows one to have an answer as a fuzzy probability to a question like "How likely is it that John is *not fat*?" given that "*Most* people who are *tall* are *not fat*" and "John is *fairly tall*".

However, a logic for handling natural language should have not only a capability to deal with the semantics of vague linguistic terms, but also a structure of formulas close to that of natural language sentences. Significantly, while Zadeh has regarded a methodology for computing with words as the main contribution of fuzzy logic ([28], [30]), Sowa has regarded a smooth mapping between logic and natural language as the main motivation of conceptual graphs (CGs) ([20], [21]). It shows that fuzzy logic and conceptual graphs are two logical formalisms that emphasize the same target of

natural language, each of which is focused on one of the two mentioned desired features of a logic for handling natural language.

Conceptual graphs, based on semantic networks and Peirce's existential graphs, combine the visual advantage of graphical languages and the expressive power of logic. On the one hand, the graphical notation of conceptual graphs is advantageous for the representation of nested information whose representation in linear notations is difficult to follow. On the other hand, the formal order-sorted logic foundation of conceptual graphs provides a rigorous basis not only for reasoning processes performed directly on them, but also for justifying the soundness and the completeness of a reasoning procedure.

Conceptual graphs have been used for solving problems in several areas such as, but not limited to, natural language processing, knowledge acquisition and management, database design and interface, and information systems. This language has been proposed as a normative conceptual schema language by the ANSI standards committee on Information Resource Dictionary Systems (ANSI Report No. X3H4/93-196), and as a knowledge representation language in conceptual models by the ANSI standards committee on Information Processing Systems (ANSI Report No. X3T2/95-019r2).

In [19] and [20], Sowa showed that natural language expressions with generalized quantifiers could be smoothly represented in conceptual graphs but not in predicate logic, yet did not adequately formalize them. A substantial effort that has been made in formalizing generally quantified CGs was [23] and [24]. Therein, the authors extended predicate logic with equality by adding to it numerical quantifiers, then used it to define semantics and inference rules for conceptual graphs with numerical quantifiers. However, limited by classical arithmetic, vague quantifiers like *few*, *many*, or *most* were not considered in that work.

In this paper, we aim to combine the advantages of both conceptual graphs and fuzzy logic into a formalism for representing and reasoning with generally quantified statements in particular, and linguistic information in general. In [25], while extending fuzzy conceptual graphs (FCGs) ([17]) with fuzzy conceptual relations, the authors also introduced generalized quantifiers into them. However, that work actually just adapted inference rules in [29] to the conceptual graph notation, without formally defining the semantics of fuzzy conceptual graphs with generalized quantifiers. In contrast, here we use the probability-based interpretation to expand generally quantified FCGs into probabilistic FCG rules, formally defining their semantics in terms of simple FCGs without generalized quantifiers. Then, on the basis of this semantics, we derive inference rules performed directly on FCGs with either relative quantifiers or absolute quantifiers; reasoning with the latter was not considered in [29] or [15].

Firstly, for the paper being self-contained, Section 2 and Section 3 summarize the basic notions of fuzzy arithmetic (cf. [11, 14]) and conceptual graphs (cf. [10, 19]). Section 4 presents the expansion rule that formally defines the semantics of generally quantified FCGs in terms of simple FCGs. Then, Section 5 derives inference rules for relative quantifiers and absolute quantifiers, and Jeffrey's rule on generally quantified FCGs. Finally, Section 6 concludes the paper and suggests future research.

2 Fuzzy Arithmetic

2.1 Fuzzy Sets and Fuzzy Numbers

For a classical set, an element is to be or not to be in the set or, in other words, the membership grade of an element in the set is binary. Fuzzy sets are to represent classes of objects where the boundary for an object to be or not to be in a class is not clear-cut due to the vagueness of the concept associated with the class. Such vague concepts are frequently encountered in the real world as reflected in natural language, like *young* or *old*, *small* or *large*. So the membership grade of an element in a fuzzy set is expressed by a real number in the interval [0, 1].

Definition 2.1 A *fuzzy set A* on a domain U is defined by a membership function μ_A from U to [0, 1]. It is said to be a *fuzzy subset* of a fuzzy set B also on U, denoted by $A \subseteq B$, if and only if $\forall u \in U$: $\mu_A(u) \leq \mu_B(u)$.

In this work we apply the voting model interpretation of fuzzy sets ([3, 12]) whereby, given a fuzzy set A on a domain U, each voter has a subset of U as his/her own crisp definition of the concept that A represents. For example, a voter may have the interval [0, 35] representing human ages from 0 to 35 years as his/her definition of the concept *young*, while another voter may have [0, 25] instead. The membership function value $\mu_A(u)$ is then the proportion of voters whose crisp definitions include u. As such, A defines a probability distribution on the power set of U across the voters, and thus a fuzzy proposition "x is A" defines a family of probability distributions of the variable x on U.

Operations on fuzzy sets also generalize those on classical sets, computing with real numbers in [0, 1] instead.

Definition 2.2 Let A and B be two fuzzy sets on a domain U. Then the *fuzzy intersection* of A and B is a fuzzy set denoted by $A \cap B$ and defined by $\forall u \in U$: $\mu_{A \cap B}(u) = min\{\mu_A(u), \mu_B(u)\}$. The *fuzzy union* of A and B is a fuzzy set denoted by $A \cup B$ and defined by $\forall u \in U$: $\mu_{A \cup B}(u) = max\{\mu_A(u), \mu_B(u)\}$. The *fuzzy complement* of A is a fuzzy set denoted by $\tilde{A}$ and defined by $\forall u \in U$: $\mu_{\tilde{A}}(u) = 1 - \mu_A(u)$.

As a special category of fuzzy sets, fuzzy numbers defined as fuzzy sets on the set **R** of real numbers are to represent imprecise numeric values. Using fuzzy numbers is one step further than intervals in representing imprecise numeric values, whereby intervals can be considered as special fuzzy numbers. For example, an interval $[a, b]$ can be represented as the fuzzy number I defined by $\mu_I(x) = 1$ if $x \in [a, b]$, or $\mu_I(x) = 0$ otherwise.

2.2 Operations on Fuzzy Numbers

One method for defining operations on fuzzy numbers is based on the *extension principle*. According to this principle, any function f: $U_1 \times U_2 \times \ldots \times U_n \rightarrow V$ induces a function g: $X_1 \times X_2 \times \ldots \times X_n \rightarrow Y$ where X_1, X_2, ..., X_n and Y are respectively sets of

fuzzy sets on U_1, U_2, ..., U_n and V and, for every $(A_1, A_2, ..., A_n) \in X_1 \times X_2 \times ... \times X_n$, $g(A_1, A_2, ..., A_n) \in Y$ is defined by:

$$\forall v \in V: \mu_{g(A_1, A_2, ..., A_n)}(v) = sup\{min\{\mu_{A_1}(u_1), \mu_{A_2}(u_2), ..., \mu_{A_n}(u_n)\} \mid$$

$$(u_1, u_2, ..., u_n) \in U_1 \times U_2 \times ... \times U_n \text{ and } v = f(u_1, u_2, ..., u_n)\}.$$

From now on, for simplicity, we use the same notation for a real number function (e.g. f above) and its fuzzy extension (e.g. g above).

For example, using this method, the product of two fuzzy numbers A and B is the fuzzy number $A.B$ defined as follows:

$$\forall z \in \mathbf{R}: \mu_{A.B}(z) = sup\{min\{\mu_A(x), \mu_B(y)\} \mid (x, y) \in \mathbf{R} \times \mathbf{R} \text{ and } z = x.y\}.$$

Figure 2.1 illustrates this multiplication operation on fuzzy numbers.

Similarly, the method can be applied to extend *min* and *max*, which are functions from $\mathbf{R} \times \mathbf{R}$ to $\mathbf{R}$, for fuzzy numbers as defined below:

$$\forall z \in \mathbf{R}: \mu_{min\{A, B\}}(z) = sup\{min\{\mu_A(x), \mu_B(y)\} \mid (x, y) \in \mathbf{R} \times \mathbf{R} \text{ and } z = min\{x, y\}\}$$
$$\forall z \in \mathbf{R}: \mu_{max\{A, B\}}(z) = sup\{min\{\mu_A(x), \mu_B(y)\} \mid (x, y) \in \mathbf{R} \times \mathbf{R} \text{ and } z = max\{x, y\}\}.$$

A fuzzy version of the less-than-or-equal-to relation ≤ on real numbers can then be defined for fuzzy numbers such that $A \leq B$ if and only if $min\{A, B\} = A$ or, equivalently, $max\{A, B\} = B$. However, we note that, while ≤ is a total order on real numbers, its fuzzy version is just a partial order on fuzzy numbers, because $min\{A, B\}$ and $max\{A, B\}$ may equal to neither A nor B. For illustration, Figure 2.2 shows two comparable fuzzy numbers and two incomparable ones with respect to ≤.

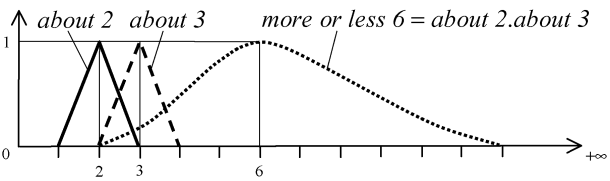
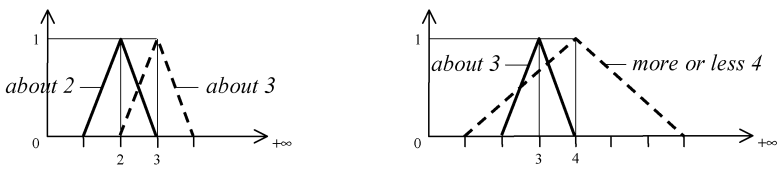


Fig. 2.1. Fuzzy multiplication



comparable fuzzy numbers: *about 2* ≤ *about 3* incomparable fuzzy numbers

Fig. 2.2. Comparable and incomparable fuzzy numbers

2.3 Fuzzy Lower and Upper Bounds

For a real number x, the notion "at least x" is represented by the interval $[x, +\infty]$, which includes all the real numbers y such that $x \le y$. Similarly, "at most x" is represented by $[-\infty, x]$. For a fuzzy number A, we denote the least specific fuzzy number that is greater than or equal to A by *at least A*. That is, $A \le$ *at least A* and $B \subseteq$ *at least A* for every B such that $A \le B$, whence the membership function of *at least A* can be specified as follows:

$$\forall x \in \mathbf{R}: \mu_{at\ least\ A}(x) = sup\{\mu_A(y) \mid y \in \mathbf{R} \text{ and } y \le x\}.$$

Similarly, the fuzzy number *at most A*, such that *at most A* $\le A$ and $B \subseteq$ *at most A* for every B where $B \le A$, is defined by:

$$\forall x \in \mathbf{R}: \mu_{at\ most\ A}(x) = sup\{\mu_A(y) \mid y \in \mathbf{R} \text{ and } x \le y\}.$$

Figure 2.3 illustrates such lower bound and upper bound fuzzy numbers.

We call a fuzzy number whose membership function is non-decreasing a *non-decreasing fuzzy number*, and a fuzzy number whose membership function is non-increasing a *non-increasing fuzzy number*. According to the above definition of fuzzy lower bounds, if A is a non-decreasing fuzzy number, then *at least A* = A. Similarly, if A is a non-increasing fuzzy number, then *at most A* = A. The following property also holds, where A/B denotes the fuzzy division of A by B, and B is said to be *positive* if and only if $\mu_B(x) = 0$ for every $x \le 0$.

Proposition 2.1 For any fuzzy number A and positive fuzzy number B:

1. at least (A/B) = at least A / B.
2. at most (A/B) = at most A / B.

Proof.

1. Since B is positive, only $\mu_B(y)$ where $y > 0$ need to be counted in the following:
   $\forall t \in \mathbf{R}: \mu_{at\ least\ (A/B)}(t) = sup\{\mu_{A/B}(z) \mid z \le t\} = sup\{sup\{min\{\mu_A(x), \mu_B(y)\} \mid z = x/y\} \mid z \le t\} = sup\{min\{\mu_A(x), \mu_B(y)\} \mid x/y \le t\} = sup\{sup\{min\{\mu_A(z), \mu_B(y)\} \mid z \le x\} \mid t = x/y\} = sup\{min\{sup\{\mu_A(z) \mid z \le x\}, \mu_B(y)\} \mid t = x/y\} = sup\{min\{\mu_{at\ least\ A}(x), \mu_B(y)\} \mid t = x/y\} = \mu_{at\ least\ A\ /\ B}(t)$, whence *at least* (A/B) = *at least A* / *B*.
2. The proof is similar.

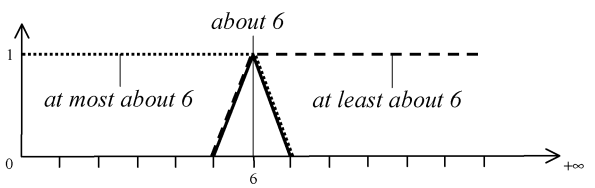


Fig. 2.3. Fuzzy lower bound and upper bound

3 Conceptual Graphs

3.1 Syntax

A conceptual graph is a bipartite graph of *concept* vertices alternate with (conceptual) *relation* vertices, where edges connect relation vertices to concept vertices. Each concept vertex, drawn as a box and labelled by a pair of a *concept type* and a *concept referent*, represents an entity whose type and referent are respectively defined by the concept type and the concept referent in the pair. Each relation vertex, drawn as a circle and labelled by a *relation type*, represents a relation of the entities represented by the concept vertices connected to it. For brevity, we may call a concept or relation vertex a concept or relation, respectively. Concepts connected to a relation are called *neighbour concepts* of the relation. Each edge is labelled by a positive integer and, in practice, may be directed just for readability.

For example, the CG in Figure 3.1 says "John is a student. There is a subject. Computer Science is a field of study. The subject is in Computer Science. John studies the subject", or briefly, "John studies a subject in Computer Science". In a textual format, concepts and relations can be respectively written in square and round brackets as follows:

[STUDENT: John]→(STUDY)→[SUBJECT: ∗]→(IN)→[FIELD: Computer Science]

Here, for simplicity, the labels of the edges are not shown.

In this example, [STUDENT: John], [SUBJECT: ∗], [FIELD: Computer Science] are concepts with STUDENT, SUBJECT and FIELD being concept types, whereas (STUDY) and (IN) are relations with STUDY and IN being relation types. The referents John and Computer Science of the concepts [STUDENT: John] and [FIELD: Computer Science] are *individual markers*. The referent * of the concept [SUBJECT: ∗] is the *generic marker* referring to an unspecified entity. In this paper, two concepts with two different individual markers are assumed to refer to two different entities, while concepts with the same individual marker are assumed to refer to the same entity.

To denote that two concepts with the generic marker refer to the same unspecified entity, a *coreference link* drawn as a dotted line connecting them is used. For example, the CG in Figure 3.2 adds to the CG in Figure 3.1 a fact that "There is a person who studies the same subject as John does". In this example, the CG can be transformed into a semantically equivalent one without a coreference link by joining the two concepts [SUBJECT: *] into a single one. However, coreference links are more appealing when concepts occur in different contexts, such as *negative contexts*, where the concepts cannot be joined. We say two concepts are *coreferent* if they refer to the same entity, that is, they have the same individual marker or they are connected by a coreference link.

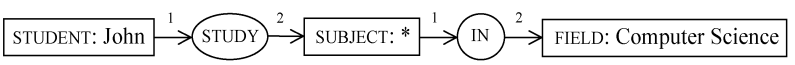


Fig. 3.1. An example CG

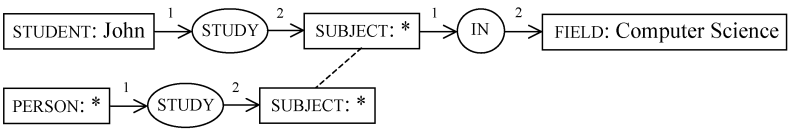


Fig. 3.2. A CG with a coreference link

Corresponding to the notion of sorts in order-sorted predicate logic, concept types are partially ordered by the concept subtype order. This order can be regarded as an information or specificity order in the sense that, given two concept types t_1 and t_2 where t_2 is a concept subtype of t_1, a fact "Object x is of type t_2" is more informative than "Object x is of type t_1". So we write $t_1 \leq_t t_2$ to denote that t_2 is a concept subtype of t_1. For example, one may have PERSON $\leq_t$ STUDENT. Relation types can also be partially ordered. For example, with the CG in Figure 3.1, one may have STUDY as a subtype of a relation type ACT, written as ACT $\leq_t$ STUDY.

For a partial order on concept referents, which are basically individual markers and the generic marker only, it is simply that, for every individual marker i, $* \leq_t i$, and all individual markers are pairwise incomparable. However, a partial order on concept referents is more appealing when they are extended to contain more information such as descriptors represented by CGs themselves, as in [19].

3.2 Semantics

In the semantic aspect, Sowa gave two formal semantics of conceptual graphs ([19]). One is the first-order predicate logic semantics defined through the operator Φ that maps a CG to a first-order predicate logic formula. The other is the CG-based semantics that uses CG projection to define the denotations of CGs with respect to models represented by CGs themselves.

Basically, Φ maps each vertex of a CG to an atomic formula of first-order predicate logic, and maps the whole CG to the conjunction of those atomic formulas with all variables being existentially quantified. Each individual marker is mapped to a constant, each generic marker is mapped to a variable, and each concept or relation type is mapped to a predicate symbol.

Each concept of type t and referent m is mapped to:

$$p(\Phi(m))$$

where $p = \Phi(t)$.

Each relation of type t and neighbour concept referents $m_1, m_2, ..., m_n$ is mapped to:

$$p(\Phi(m_1), \Phi(m_2), ..., \Phi(m_n))$$

where $p = \Phi(t)$.

For example, let G be the CG in Figure 3.1, then $\Phi(G)$ is:

$$\exists x \, (\ \text{student(John)} \wedge \text{subject}(x) \wedge \text{field(Computer Science)} \wedge \text{study(John, } x) \wedge \text{in}(x, \text{Computer Science}))$$

Partially ordered sets of concept and relation types are also mapped to formulas of first-order predicate logic. Each pair of concept types t_1 and t_2 such that $t_1 \leq_t t_2$ is mapped to:

$$\forall x \, (p_2(x) \rightarrow p_1(x))$$

where $p_1 = \Phi(t_1)$ and $p_2 = \Phi(t_2)$.

Each pair of relation types t_1 and t_2 of arity n such that $t_1 \leq_t t_2$ is mapped to:

$$\forall x_1 \forall x_2 \ldots \forall x_n \, (t_2(x_1, x_2, \ldots, x_n) \rightarrow t_1(x_1, x_2, \ldots, x_n))$$

where $p_1 = \Phi(t_1)$ and $p_2 = \Phi(t_2)$.

For example, PERSON $\leq_t$ STUDENT and ACT $\leq_t$ STUDY are respectively mapped to the following formulas:

$$\forall x \, (\text{student}(x) \rightarrow \text{person}(x))$$

and

$$\forall x \forall y \, (\text{study}(x, y) \rightarrow \text{act}(x, y)).$$

For the CG-based semantics, a closed world model is defined to be a set of CGs such that no individual marker occurs in more than one concept in the set. The denotation of a CG with respect to a model, which is either **true** or **false**, is then defined through a fundamental operation in CG theory that is *CG projection*, whereby it is **true** if and only if (iff) there is a projection from that CG to a CG in the model.

A projection maps a CG to another more or equally specific one, by mapping each vertex of the former to a vertex of the latter that has a more or equally specific label. The label (t_1, m_1) of a concept is said to be more or equally specific than the label (t_2, m_2) of another concept iff $t_2 \leq_t t_1$ and $m_2 \leq_t m_1$. The label t_1 of a relation is said to be more or equally specific than the label t_2 of another relation iff $t_2 \leq_t t_1$. The mapping must also preserve the adjacency of the neighbour concepts of a relation, that is, if a relation r_2 of type t_2 of arity n is mapped to a relation r_1 of type t_1 of the same arity then, for every i from 1 to n, the neighbour concept connected to r_2 by the edge labelled i must be mapped to the neighbour concept connected to r_1 by the edge labelled i. Figure 3.3 illustrates a CG projection from G to H.

CG projection is reflexive and transitive. That is, every CG has a projection to itself and, given CGs G, H and F, if G has a projection to H and H has a projection to F, then G has a projection to F. As such, CG projection defines a preorder on CGs, which can also be regarded as an information or specificity order, whereby if G has a projection to H, then H is more informative or more specific than G. Logically, if a CG G has a projection to a CG H, then H implies G.

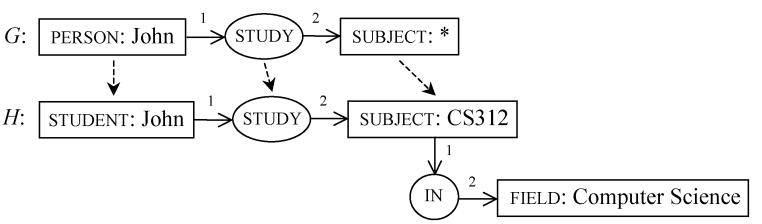


Fig. 3.3. A CG projection

3.3 Nested Conceptual Graphs

A proposition can be represented as a concept of type PROPOSITION whose referent is a CG describing that proposition. The negation of a proposition is then represented by

a relation of type NEG connected to the concept representing that proposition, as illustrated in Figure 3.4.

Nested CGs can be used to express arbitrary first-order logic sentences and, furthermore, modal and higher-order logic ones. For examples, in Figure 3.5, *G* says "Every student studies some subject", *H* says "Julian couldn't fly to Mars", and *F* says "Tom believes that Mary wants to marry a sailor", where ¬ denotes a NEG relation.

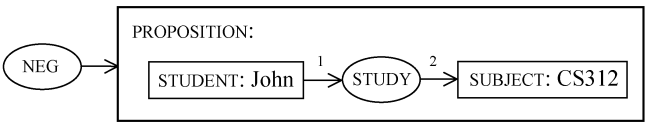


Fig. 3.4 A negated CG

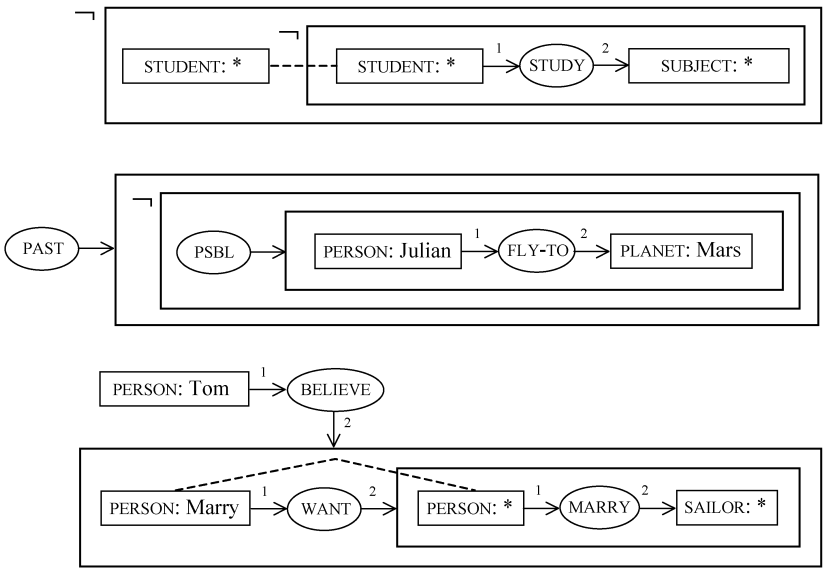


Fig. 3.5. Nested CGs

4 Representation of Generalized Quantifiers

4.1 Quantification as Conditional Probability

Firstly, in reasoning with quantifiers, *absolute quantifiers* and *relative quantifiers* on a set have to be distinguished, where the quantities expressed by the latter are relative to the cardinality of the set. Examples of absolute quantifiers are *only one*, *few*, or *several*, while ones of relative quantifiers are *about 9%, half*, or *most*.

In the crisp case, absolute quantifiers can be defined by natural numbers, and relative quantifiers by non-negative rational numbers that are not greater than 1

measuring a proportion of a set, where 0 means 0% and 1 means 100%. Correspondingly, in the fuzzy case, absolute quantifiers can be defined by fuzzy sets on the set **N** of natural numbers, i.e., fuzzy numbers whose domain is restricted to **N**, and relative quantifiers by fuzzy numbers whose domain is restricted to the set of rational numbers in [0, 1]. For simplicity without rounding of real numbers, however, we assume absolute quantifiers to be defined by fuzzy numbers on [0, +∞] and relative quantifiers by fuzzy numbers on [0, 1].

The existential quantifier in classical logic corresponds to *at least 1* in natural language, which is an absolute quantifier whose membership function is defined by $\mu_{at\ least\ 1}(x) = 1$ if $x \geq 1$, or $\mu_{at\ least\ 1}(x) = 0$ otherwise. Meanwhile, the universal quantifier, which corresponds to *all* or *every* in natural language, is a relative quantifier and its membership function is defined by $\mu_{all}(1) = 1$ and $\mu_{all}(x) = 0$ for every $0 \leq x < 1$.

Arithmetic operations for fuzzy numbers as presented in Section 2 are also applicable to absolute quantifiers with [0, +∞] being used in place of **R**, and to relative quantifiers with [0, 1] being used instead. Also, on the basis of the extension principle, each absolute quantifier Q on a type T whose denotation set in a universe of discourse has the cardinality $|T|$ corresponds to the relative quantifier $Q_T = Q/|T|$.

As mentioned in the introduction section, a relative quantifier Q in a statement "Q A's are B's" can be interpreted as the conditional probability of $B(x)$ being **true** given $A(x)$ being **true** for an object x picked at random uniformly. That is, it can be represented by the probabilistic logic rule $B(x) \leftarrow A(x)$ [Q] where $Q = Pr(B(x) \mid A(x))$. For example, "*Most* Swedes are *tall*" can be represented by $tall(x) \leftarrow \text{Swede}(x)$ [*most*] where $most = Pr(tall(x) \mid \text{Swede}(x))$.

In general, $A(x)$ or $B(x)$ can also be represented in conceptual graphs or any other logical formalism. In [5], for dealing with vagueness and imprecision in the real world as reflected in natural language, fuzzy conceptual graphs were developed with fuzzy concept or relation types ([8]) and fuzzy attribute values defined by fuzzy sets. Here, focusing on representing and reasoning with generalized quantifiers, we consider *simple FCGs* with only fuzzy attribute values. For example, the simple FCG in Figure 4.1 expresses "John is *fairly tall*", where *fairly tall* is a linguistic label of a fuzzy attribute value, and [PERSON: John] is called an entity concept whereas [HEIGHT: *@*fairly tall*] an attribute concept.

We now formulate generally quantified FCGs as logical expressions that smoothly match with generally quantified statements in natural language.

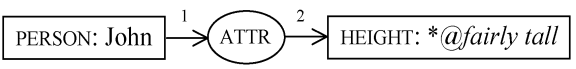


Fig. 4.1. A simple FCG

4.2 Generally Quantified FCGs

In [19], set referents and numeric quantifiers were introduced into concepts to represent plural noun phrases and quantification in natural language, where the semantics of such an extended CG was defined by its expansion into a CG without

numeric quantifiers. For example, Figure 4.2 shows an extended CG *G* and its defining expansion *E*, which literally says "There exists a set of two persons who see John".

However, that defining expansion does not capture the actual meaning of generalized quantifiers, because a quantifier on a type does not simply mean that there exists a set of objects of that type that have some property, and the cardinality of the set is defined by the quantifier. Rather, a quantifier on a type constrains the number of objects of that type that can have some property. For example, "*Only one* person is the President of the United States" means not only "There exists a set of *only one* person who is the President of the United States", but also "There is no more than one person who is the President of the United States", as the former alone does not exclude "There are two different sets each of which has *only one* person who is the President of the United States".

Here we apply the conditional probability interpretation of relative quantifiers presented above to define an FCG with a relative quantifier as a probabilistic FCG rule, where the quantifier is the conditional probability of the head given the body of the rule, which are both represented by simple FCGs. Meanwhile, an FCG with an absolute quantifier is semantically equivalent to the one with the relative quantifier converted from that absolute quantifier as noted above.

In this paper, we consider only FCGs that contain only one generally quantified concept, excluding ones with the generic referent *, whose quantifiers are implicitly the existential quantifier. For example, Figure 4.3 shows a generally quantified FCG *G* and its defining expansion *E*, expressing "*Most* Swedes are *tall*", where *most* and *tall* are linguistic labels of fuzzy sets. We note that this defining expansion rule can be seen as a generalization of the one for universally quantified CGs defined in terms of CG rules as introduced in [19] and studied in [7], as the universal quantifier is a special relative quantifier.

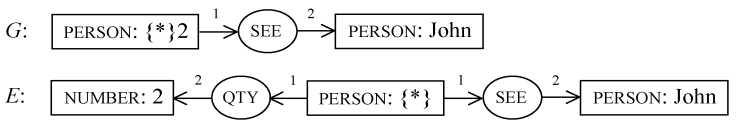


Fig. 4.2. A CG with a numeric quantifier and its defining expansion

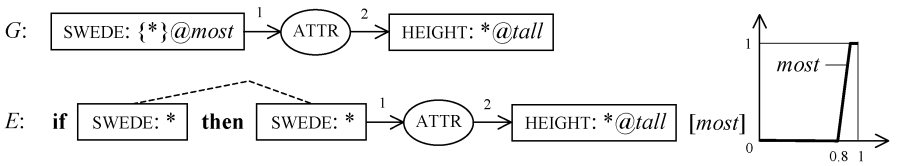


Fig. 4.3. A generally quantified FCG and its defining expansion

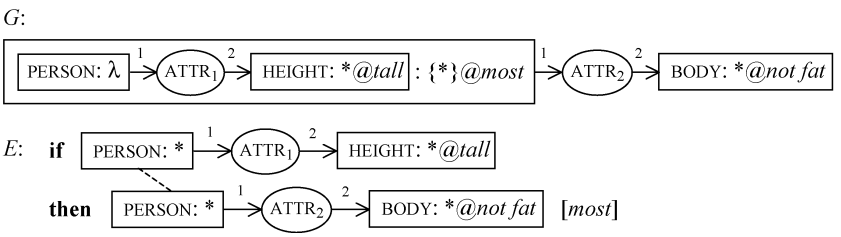


Fig. 4.4. Quantification on a type defined by a lambda FCG

Furthermore, the type in a generally quantified concept can be represented by a simple FCG as a lambda expression defining that type as for CGs ([19, 22]). We call such a simple FCG a *lambda FCG*, which is like a simple FCG except that it has one concept, which we call a *lambda concept*, whose referent is denoted by λ to be distinguished from the generic and individual referents. For example, Figure 4.4 illustrates a generally quantified FCG *G* and its defining expansion *E*, expressing "*Most* people who are *tall* are *not fat*". As such, a lambda FCG corresponds to a relative clause in natural language.

We now formally define generally quantified FCGs and the expansion rule for their semantics exemplified above. Since a generally quantified concept [T:{*}*Q*] is semantically equivalent to [[T: λ]:{*}*Q*], where [T: λ] is a special lambda FCG with only the lambda concept, one can always assume the type of a generally quantified concept to be represented by a lambda FCG.

Definition 4.1 A *generally quantified concept* is defined by a triple of a concept type represented by a lambda FCG, the set referent {*}, and a generalized quantifier. A *generally quantified FCG* is a simple FCG with one generally quantified concept. It is said to be a *relatively quantified* or an *absolutely quantified* FCG if the generalized quantifier in it is a relative quantifier or an absolute quantifier, respectively.

Definition 4.2 The defining expansion of a relatively quantified FCG *G* is the probabilistic FCG rule **if** *F* **then** *H* [*Q*] where:

1. *F* is obtained from the lambda FCG in *G* by replacing its lambda concept [T: λ] with [T: *], *H* is obtained from *G* by replacing its generally quantified concept with [T: *], and there is a coreference link between these two concepts [T: *] of *F* and *H*.
2. *Q* is the relative quantifier in *G* and $Pr(H \mid F) = Q$.

5 Reasoning with Generalized Quantifiers

For the inference rules presented below, generalizing an FCG means replacing its concept or relation types and concept referents with less specific ones, as for a CG, and its fuzzy attribute values with ones whose defining fuzzy sets are fuzzy supersets of those defining the former. It is dually similar for specializing.

5.1 Inference Rules for Relative Quantifiers

Proposition 5.1 A generally quantified FCG G with a relative quantifier Q entails a generally quantified FCG G^* obtained from G by generalizing it except for its generally quantified concept, and replacing Q with *at least Q*.

Proof.

Let the expansion of G be **if** F **then** H [Q]. Then the expansion of G^* is **if** F **then** H^* [*at least Q*], where H^* is a generalization of H. One has $Pr(H^* \mid F) \geq Pr(H \mid F)$, whence $Pr(H \mid F) = Q$ entails $Pr(H^* \mid F) =$ *at least Q*. Thus **if** F **then** H [Q] entails **if** F **then** H^* [*at least Q*], i.e., G entails G^*.

For example, in Figure 5.1, G expressing "*Most* people who are *tall* are *not fat*" entails G^* expressing "*Most* people who are *tall* are *not very fat*", provided that *not fat* $\subseteq$ *not very fat*, and *most* is a non-decreasing fuzzy number whereby *at least most* = *most* as noted in Section 2.

Proposition 5.2 A generally quantified FCG G with a relative quantifier Q entails a generally quantified FCG G^* obtained from G by specializing it except for its generally quantified concept, and replacing Q with *at most Q*.

Proof.

Let the expansion of G be **if** F **then** H [Q]. Then the expansion of G^* is **if** F **then** H^* [*at most Q*], where H^* is a specialization of H. One has $Pr(H^* \mid F) \leq Pr(H \mid F)$, whence $Pr(H \mid F) = Q$ entails $Pr(H^* \mid F) =$ *at most Q*. Thus **if** F **then** H [Q] entails **if** F **then** H^* [*at most Q*], i.e., G entails G^*.

For example, in Figure 5.2, G expressing "*About 9%* people who are *tall* are *fat*" entails G^* expressing "*At most about 9%* people who are *tall* are *very fat*", provided that *very fat* $\subseteq$ *fat*.

Proposition 5.3 A universally quantified FCG G entails a universally quantified FCG G^* obtained from G by specializing its lambda FCG.

Proof.

Let the expansion of G be **if** F **then** H [1]. Then the expansion of G^* is **if** F^* **then** H^* [1], where F^* and H^* are specializations of F and H, respectively, with their two coreferent concepts corresponding to the lambda concept in G. More precisely, if the type of the lambda concept in G is specialized in G^*, then H^* is only different from H in that type specialization; otherwise, H^* and H are identical. As such, $H^* \wedge F^*$ is semantically equivalent to $H \wedge F^*$, whence $Pr(H^* \mid F^*) = Pr(H^* \wedge F^*)/Pr(F^*) = Pr(H \wedge F^*)/Pr(F^*) = (Pr(F^*) - Pr(\neg H \wedge F^*))/Pr(F^*)$. Furthermore, one has $Pr(\neg H \wedge F^*) \leq Pr(\neg H \wedge F) = Pr(\neg H \mid F).Pr(F) = (1 - Pr(H \mid F)).Pr(F) = 0$, as $Pr(H \mid F) = 1$,whence $Pr(H^* \mid F^*) = 1$. That is, $Pr(H \mid F) = 1$ entails $Pr(H^* \mid F^*) = 1$ or, in other words, **if** F **then** H [1] entails **if** F^* **then** H^* [1], i.e., G entails G^*.

For example, in Figure 5.3, *G* expressing "*All* people who are *tall* are *not fat*" entails *G** expressing "*All* males who are *very tall* are *not fat*", provided that *very tall* ⊆ *tall*.

In [7], CG projection, which also defines the entailment relation between two CGs, was extended for universally quantified CGs, where each CG could have more than one universally quantified concept but quantification was only on concept type labels. Here a universally quantified FCG is assumed to have only one universally quantified concept whose concept type however can be a lambda FCG. Proposition 5.3 above is in agreement with the extended CG projection in [7], with respect to the entailment relation between two involved CGs or FCGs.

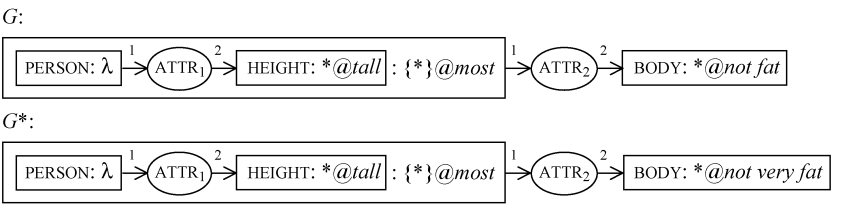


Fig. 5.1. Generalization inference rule for relative quantifiers

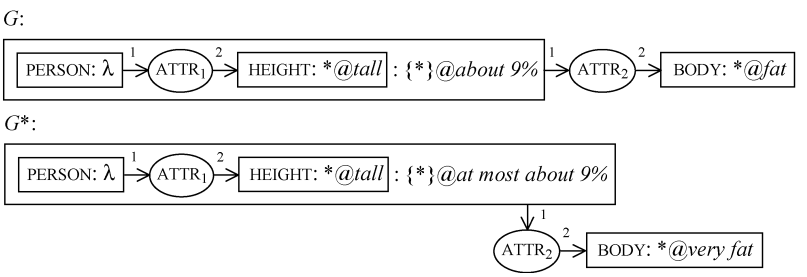


Fig. 5.2. Specialization inference rule for relative quantifiers

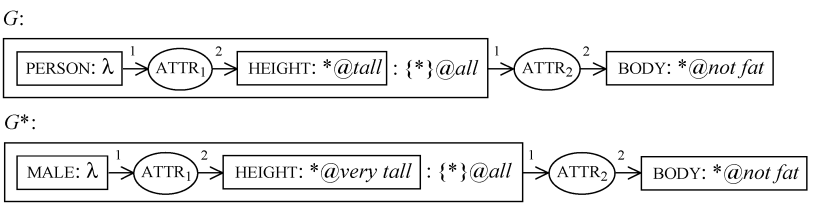


Fig. 5.3. Inference rule for the universal quantifier

5.2 Inference Rules for Absolute Quantifiers

The following propositions are obtained on the basis as noted in Section 4 that an absolutely quantified FCG is semantically equivalent to a relatively quantified one with the corresponding relative quantifier, whose defining expansion is given by Definition 4.2.

Proposition 5.4 A generally quantified FCG G with an absolute quantifier Q entails a generally quantified FCG G^* obtained from G by generalizing it, including its lambda FCG, and replacing Q with *at least* Q.

Proof.

Let the expansion of G be **if** F **then** H $[Q_F]$, where F corresponds to G's lambda FCG, and $Q_F = Q/|F|$ with $|F|$ being the cardinality of the denotation set in a universe of discourse of the type that F represents, as noted in Section 4. Then the expansion of G^* is **if** F^* **then** H^* [*at least* Q / $|F^*|$], where F^* and H^* are respectively generalizations of F and H. One has $Pr(H^* \mid F^*) = Pr(H^* \wedge F^*)/Pr(F^*) \geq Pr(H \wedge F)/Pr(F^*) = (Pr(H \mid F).Pr(F))/Pr(F^*) = ((Q/|F|).Pr(F))/Pr(F^*)$, and $Pr(F) = Pr(F \wedge F^*) = Pr(F \mid F^*).Pr(F^*) = (|F|/|F^*|).Pr(F^*)$, whence $Pr(H^* \mid F^*) \geq Q/|F^*|$. Thus $Pr(H \mid F) = Q/|F|$ entails $Pr(H^* \mid F^*) = at\ least\ (Q/|F^*|)$. According to Proposition 2.1, *at least* $(Q/|F^*|)$ = *at least* Q / $|F^*|$, whence **if** F **then** H $[Q/|F|]$ entails **if** F^* **then** H^* [*at least* Q / $|F^*|$], i.e., G entails G^*.

For example, in Figure 5.4, G expressing "*Few* people who are *tall* are *fat*" entails G^* expressing "*At least few* people who are *fairly tall* are *fairly fat*", provided that *tall* $\subseteq$ *fairly tall*, *fat* $\subseteq$ *fairly fat*, and *few* is used as an absolute quantifier.

Proposition 5.5 A generally quantified FCG G with an absolute quantifier Q entails a generally quantified FCG G^* obtained from G by specializing it, including its lambda FCG, and replacing Q with *at most* Q.

Proof.

Let the expansion of G be **if** F **then** H $[Q/|F|]$, where F corresponds to G's lambda FCG, and $|F|$ is the cardinality of the denotation set in a universe of discourse of the type that F represents. Then the expansion of G^* is **if** F^* **then** H^* [*at most* Q / $|F^*|$], where F^* and H^* are respectively specializations of F and H. One has $Pr(H^* \mid F^*) = Pr(H^* \wedge F^*)/Pr(F^*) \leq Pr(H \wedge F)/Pr(F^*) = (Pr(H \mid F).Pr(F))/Pr(F^*) = ((Q/|F|).Pr(F))/Pr(F^*)$, and $Pr(F^*) = Pr(F^* \wedge F) = Pr(F^* \mid F).Pr(F) = (|F^*|/|F|).Pr(F)$, whence $Pr(H^* \mid F^*) \leq Q/|F^*|$. Thus $Pr(H \mid F) = Q/|F|$ entails $Pr(H^* \mid F^*) = at\ most\ (Q/|F^*|)$. According to Proposition 2.1, *at most* $(Q/|F^*|)$ = *at most* Q / $|F^*|$, whence **if** F **then** H $[Q/|F|]$ entails **if** F^* **then** H^* [*at most* Q / $|F^*|$], i.e., G entails G^*.

For example, in Figure 5.5, G expressing "*Few* people who are *tall* are *fat*" entails G^* expressing "*At most few* people who are *very tall* are *very fat*", provided that *very tall* $\subseteq$ *tall*, *very fat* $\subseteq$ *fat*, and *few* is also used as an absolute quantifier.

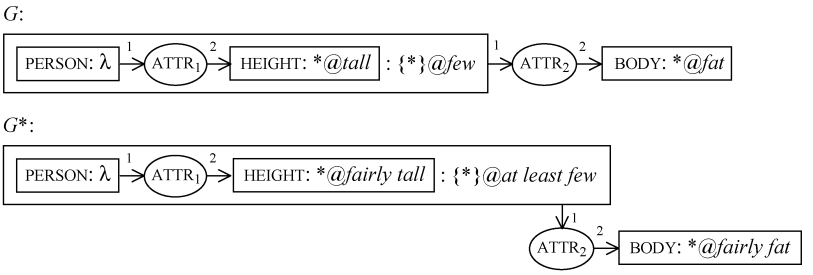


Fig. 5.4. Generalization inference rule for absolute quantifiers

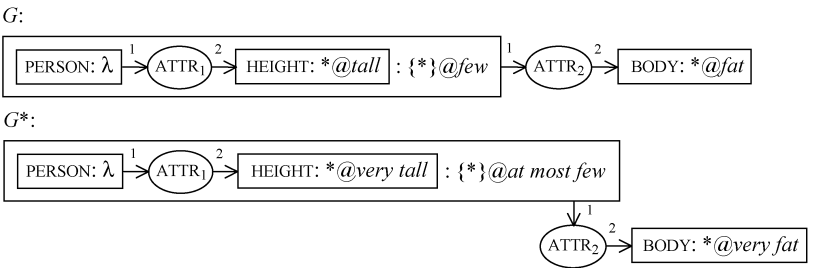


Fig. 5.5. Specialization inference rule for absolute quantifiers

5.3 Jeffrey's Rule

In practice, it is often the case that a condition probability $v = Pr(B(x) \mid A(x))$ is obtained from statistical data meaning that, if an object x is *randomly* picked up such that $A(x)$ is **true**, then the probability for $B(x)$ being **true** is v. As such, that conditional probability value may not be applicable to a *specific* object in a universe of discourse. For example, one may have $Pr(\text{fly}(x) \mid \text{bird}(x)) = 0.9$ expressing that 90% of birds can fly, but the probability for a penguin, a specific bird, being able to fly is 0.

Therefore, for decision making, Jeffrey introduced a rule assuming such a conditional probability like $Pr(B(x) \mid A(x))$ or $Pr(B(x) \mid \neg A(x))$ to be applicable to any specific object o or, in other words, $Pr(B(o) \mid A(o)) = Pr(B(x) \mid A(x))$ and $Pr(B(o) \mid \neg A(o)) = Pr(B(x) \mid \neg A(x))$, when only the probability for $A(o)$ being **true** is known. Then, on the basis of the total probability theorem, the probability for $B(o)$ being **true** could be obtained as follows:

$$\begin{aligned} Pr(B(o)) &= Pr(B(o) \mid A(o)).Pr(A(o)) + Pr(B(o) \mid \neg A(o)).Pr(\neg A(o)) \\ &= Pr(B(x) \mid A(x)).Pr(A(o)) + Pr(B(x) \mid \neg A(x)).Pr(\neg A(o)). \end{aligned}$$

In [5], FCG projection was defined, which matches a simple FCG with another and compute the relative necessity degree of the former given the later. Here, for applying Jeffrey's rule to reasoning with generally quantified FCGs, we introduce *probabilistic FCG projection* that computes the conditional probability of a simple FCG given another one. For its definition, we apply the definition of conditional probability of fuzzy events in [3], namely, $Pr(x \text{ is } B \mid x \text{ is } A)$ whose value is a subinterval of [0, 1]

treated here as a fuzzy number on [0, 1], where A and B are two fuzzy sets on the same domain.

In the following definition, $\mathbf{V}_{\mathbf{C}_G}$ and $\mathbf{V}_{\mathbf{R}_G}$ respectively denote the set of all concepts and the set of all relations in a simple FCG G. For a concept c, *referent*(c) and *type*(c) are respectively the referent and the type of c. For a relation r, *type*(r) is the type of r, *arity*(*type*(r)) is the arity of *type*(r), and *neighbour*(r, i) is the neighbour concept connected to r by the edge labelled i.

Definition 5.1 Let G and H be two simple FCGs. A *probabilistic FCG projection* from G to H is a mapping π: $G \to H$ such that:

1. $\forall c \in \mathbf{V}_{\mathbf{C}_G}$: *referent*($c$) $\leq_t$ *referent*(πc) and *type*(c) $\leq_t$ *type*(πc).
2. $\forall r \in \mathbf{V}_{\mathbf{R}_G}$: *neighbour*($\pi r$, i) = π*neighbour*(r, i) for every $i \in \{1, 2, ..., arity(type(r))\}$, and *type*($r$) $\leq_t$ *type*(πr).

Then $\varepsilon_\pi = Pr(G \mid \pi G)$ is defined to be the product of the conditional probabilities of all fuzzy attribute values pairs in π.

For example, Figure 5.6 shows a probabilistic FCG projection from G expressing "Some person is *tall*" to H expressing "John is *fairly tall*", where $Pr(G \mid H) = Pr(G \mid \pi G) = Pr(tall \mid fairly\ tall)$.

For a probabilistic FCG projection from a lambda FCG to a simple FCG, we assume the referent λ in the lambda concept to be replaced with the generic referent *, as in the following proposition.

Proposition 5.6 Let G be a generally quantified FCG with a relative quantifier Q, and G* be a simple FCG such that there is a probabilistic FCG projection π from the lambda FCG in G to G*. Then Jeffrey's rule derives the simple FCG H* with the probability (*at least* ($Q.\varepsilon_\pi$))$\cap$(*at most* ($Q.\varepsilon_\pi + (1 - \varepsilon_\pi)$)) where H* is obtained from G by replacing its generally quantified concept with its lambda concept c whose referent λ is replaced with *referent*(πc).

Proof.

Let the expansion of G be **if** F **then** H [Q], where F corresponds to G's lambda FCG. Let F* be the simple FCG obtained from G's lambda FCG by replacing the referent of its lambda concept c with *referent*(πc). As such, F and F* are different only in that referent, whence $\pi F^* = \pi F$ and $Pr(F^* \mid \pi F^*) = Pr(F \mid \pi F) = \varepsilon_\pi$. Jeffrey's rule gives $Pr(H^*) = Pr(H|F).Pr(F^*) + Pr(H|\neg F).Pr(\neg F^*)$, from which one has $Pr(H|F).Pr(F^*) \leq Pr(H^*) \leq Pr(H|F).Pr(F^*) + (1 - Pr(F^*))$ as proved in [9], assuming that $Pr(H|\neg F)$ is totally unknown, i.e., only $0 \leq Pr(H|\neg F) \leq 1$ is known. Here $Pr(H \mid F) = Q$, and $Pr(F^*) = Pr(F^*|\pi F^*) = \varepsilon_\pi$ as $Pr(\pi F^*) = Pr(G^*) = 1$. Thus $Pr(H^*)$ is *at least* ($Q.\varepsilon_\pi$) and *at most* ($Q.\varepsilon_\pi + (1 - \varepsilon_\pi)$), whence $Pr(H^*) =$ (*at least* ($Q.\varepsilon_\pi$))$\cap$(*at most* ($Q.\varepsilon_\pi + (1 - \varepsilon_\pi)$)) on the basis of the principle of minimum specificity ([11]).

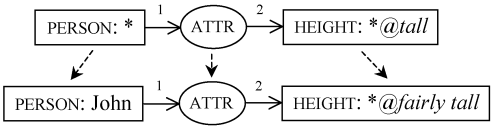


Fig. 5.6. A probabilistic FCG projection

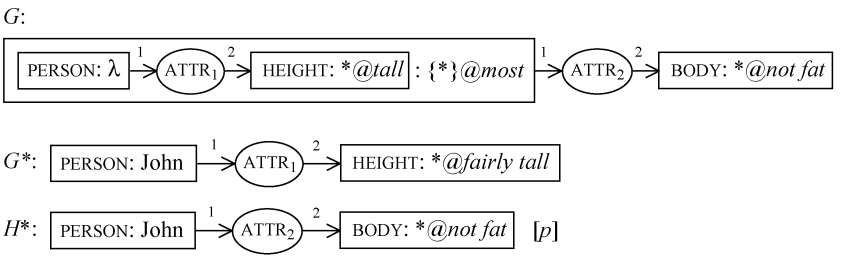


Fig. 5.7. Jeffrey's rule on generally quantified FCGs

For example, in Figure 5.7, *G* expresses "*Most* people who are *tall* are *not fat*" and *G** expresses "John is *fairly tall*". Then *H** expressing "John is *not fat*" can be derived with the probability $p = (at\ least\ (most.\varepsilon_\pi)) \cap (at\ most\ (most.\varepsilon_\pi + (1 - \varepsilon_\pi)))$, where $\varepsilon_\pi = Pr(tall \mid fairly\ tall)$, as an answer to the query "How likely is it that John is *not fat*?".

6 Conclusion

We have formulated generally quantified FCGs where generalized quantifiers are defined by fuzzy numbers to capture their vagueness and imprecision often encountered in natural language. A generally quantified FCG with a relative quantifier, defined by a fuzzy number on [0, 1], has been interpreted as a probabilistic FCG rule with that relative quantifier as the conditional probability of the head given the body of the rule. Meanwhile, an absolutely quantified FCG is semantically equivalent to a relatively quantified one with the corresponding relative quantifier. On the basis of this semantics, we have derived generalization, specialization, and Jeffrey's inference rules performed directly on generally quantified FCGs.

This is our first step in formally integrating generalized quantifiers into conceptual graphs. The presented inference rules are not meant to be complete, but to show how some basic patterns of human reasoning on generally quantified statements in natural language could be performed as machine operations on corresponding generally quantified FCGs. In the general case, one could apply the proposed probabilistic interpretation of generally quantified FCGs to study complete inference rules on them in the framework of probabilistic logic programming.

We however believe that human reasoning is more heuristic and plausible than sound and complete. That is because one often has to make additional assumptions on given information in order to infer useful new information, and thus the inference is

actually not sound with respect to the given information. Meanwhile the high complexity of a problem makes complete inference impractical. Therefore, as the theme of this work is to combine fuzzy logic and conceptual graphs, both of which emphasize the target of natural language, into a logical formalism for Artificial Intelligence approaching human expression and reasoning, our attention is focused on heuristic and plausible inference rules on FCGs.

In this paper, we considered FCGs with only one generally quantified concept whose referent is the generic set referent {*}. FCGs with more than one generally quantified concept whose set referent can include individual referents as outlined in [6] are required to represent more complex natural language expressions. Also, we did not discuss from where generally quantified statements with the proposed semantics could be obtained. For example, the statement "*Most* Swedes are *tall*" is actually a summarization of statistical data about Swedes' heights, which is a difficult induction problem for machine learning. These are among the topics that we are currently investigating.

References

[1] ANSI Report No. X3H4/93-196, IRDS Conceptual Schema (1993).

[2] ANSI Report No. X3T2/95-019r2, Information Interchange and Interpretation (1995).

[3] Baldwin, J.F., Martin, T.P., Pilsworth, B.W.: Fril - Fuzzy and Evidential Reasoning in Artificial Intelligence. Research Studies Press (1995).

[4] Barwise, J., Cooper, R.: Generalized Quantifiers and Natural Language. Linguistics and Philosophy **4** (1981) 159-219.

[5] Cao, T.H.: Foundations of Order-Sorted Fuzzy Set Logic Programming in Predicate Logic and Conceptual Graphs. PhD Thesis, University of Queensland (1999).

[6] Cao, T.H.: Fuzzy Conceptual Graphs: A Language for Computational Intelligence Approaching Human Expression and Reasoning. In: Sincak, P. et al. (eds.): The State of the Art in Computational Intelligence. Physica-Verlag (2000) 114-120.

[7] Cao, T.H., Creasy, P.N.: Universal Marker and Functional Relation: Semantics and Operations. In: Lukose, D. et al. (eds.): Conceptual Structures: Fulfilling Peirce's Dream. Lecture Notes in Artificial Intelligence, Vol. 1257. Springer-Verlag (1997) 416-430.

[8] Cao, T.H., Creasy, P.N.: Fuzzy Types: A Framework for Handling Uncertainty about Types of Objects. International Journal of Approximate Reasoning **25** (2000) 217-253.

[9] Cao, T.H., Rossiter, J.M., Martin, T.P., Baldwin, J.F.: On the Implementation of Fril++ for Object-Oriented Logic Programming with Uncertainty and Fuzziness. In: Bouchon-Meunier, B. et al. (eds.): Technologies for Constructing Intelligent Systems, Studies in Fuzziness and Soft Computing, Physica-Verlag (2002) 393-406.

[10] Chein, M., Mugnier, M.L.: Conceptual Graphs are also Graphs. Research Report No. 95003, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (1995).
[11] Dubois, D., Prade, H.: Fuzzy Sets in Approximate Reasoning, Part 1: Inference with Possibility Distributions. International Journal for Fuzzy Sets and Systems **40** (1991) 143-202.
[12] Gaines, B.R.: Fuzzy and Probability Uncertainty Logics. Journal of Information and Control **38** (1978) 154-169.
[13] Jeffrey, R.: The Logic of Decision. McGraw-Hill (1965).
[14] Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic Theory and Applications. Prentice Hall (1995).
[15] Lawry, J.: An Alternative Interpretation of Linguistic Variables and Computing with Words. In: Proceedings of the 8th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (2000) 1743-1750.
[16] Liu, Y., Kerre, E.: An Overview of Fuzzy Quantifiers - (I) Interpretations (II) Reasoning and Applications. International Journal for Fuzzy Sets and Systems **95** (1998) 1-21, 135-146.
[17] Morton, S.K.: Conceptual Graphs and Fuzziness in Artificial Intelligence. PhD Thesis, University of Bristol (1987).
[18] Peterson, P.L.: On the Logic of Few, Many and Most. Notre Dame Journal of Formal Logic **XX** (1979) 155-179.
[19] Sowa, J.F.: Conceptual Structures - Information Processing in Mind and Machine. Addison-Wesley Publishing Company (1984).
[20] Sowa, J.F.: Towards the Expressive Power of Natural Language. In: Sowa, J.F. (ed.): Principles of Semantic Networks - Explorations in the Representation of Knowledge. Morgan Kaufmann Publishers (1991) 157-189.
[21] Sowa, J.F.: Matching Logical Structure to Linguistic Structure. In: Houser, N., Roberts, D.D., Van Evra, J. (eds.): Studies in the Logic of Charles Sanders Peirce. Indiana University Press (1997) 418-444.
[22] Sowa, J.F.: Conceptual Graphs: Draft Proposed American National Standard. In: Tepfenhart, W., Cyre, W. (eds.): Conceptual Structures: Standards and Practices. Lecture Notes in Artificial Intelligence, Vol. 1640. Springer-Verlag (1999) 1-65.
[23] Tjan, B.S., Gardiner, D.A., Slagle, J.R.: Direct Inference Rules for Conceptual Graphs with Extended Notation. In: Proceedings of the 5th Annual Workshop on Conceptual Structures (1990).
[24] Tjan, B.S., Gardiner, D.A., Slagle, J.R.: Representing and Reasoning with Set Referents and Numerical Quantifiers. In: Nagle, T.E. et al. (eds.): Conceptual Structures - Current Research and Practice. Ellis Horwood (1992) 53-66.
[25] Wuwongse, V., Manzano, M.: Fuzzy Conceptual Graphs. In: Mineau, G.W., Moulin, B., Sowa, J.F. (eds.): Conceptual Graphs for Knowledge Representation. Lecture Notes in Artificial Intelligence, Vol. 699. Springer-Verlag (1993) 430-449.
[26] Zadeh, L.A.: Fuzzy Sets. Journal of Information and Control **8** (1965) 338-353.
[27] Zadeh, L.A.: Fuzzy Logic and Approximate Reasoning. Synthese **30** (1975) 407-428.

[28] Zadeh, L.A.: PRUF - A Meaning Representation Language for Natural Languages. International Journal of Man-Machine Studies **10** (1978) 395-460.
[29] Zadeh, L.A.: A Computational Approach to Fuzzy Quantifiers in Natural Languages. Computers and Mathematics with Applications **9** (1983) 149-184.
[30] Zadeh, L.A.: Fuzzy Logic = Computing with Words. IEEE Transactions on Fuzzy Systems **4** (1996) 103-111.

Improvement of the Interpretability of Fuzzy Rule Based Systems: Quantifiers, Similarities and Aggregators

Anne Laurent[1], Christophe Marsala[1], and Bernadette Bouchon-Meunier[2]

[1] LIP6, Dept. of Artificial Intelligence, Université Pierre et Marie Curie
8 rue du Capitaine Scott, 75015 Paris, France
{Anne.Laurent,Christophe.Marsala}@lip6.fr
[2] CNRS, LIP6, Dept. of Artificial Intelligence, Université Pierre et Marie Curie
8 rue du Capitaine Scott, 75015 Paris, France
Bernadette.Bouchon-Meunier@lip6.fr

Abstract. The automatic generation of fuzzy rules is a well-known task based either on the construction of a fuzzy decision tree or on the direct generation of fuzzy rules (*e.g.* association rules, gradual rules, or fuzzy summaries). In this paper, fuzzy rules obtained from fuzzy decision trees are compared to fuzzy summaries. In both approaches, the knowledge is presented to users using linguistic terms. In this framework, means to improve the interpretability of such fuzzy rules are necessary. The aim of this paper is to reinforce each method by taking advantage of the qualities of the other one. Several methods are proposed, mainly based on the use of fuzzy quantifiers, similarities and aggregations.

1 Introduction

Rule based systems are well-known frameworks that provide users with knowledge about their data. These systems are based either on descriptive methods (such as association rules or fuzzy summaries) or on predictive methods (such as decision trees). The users of such systems are often decision makers or experts from the domain addressed by the data, but they are not supposed to be computer scientists. In this framework, the introduction of Fuzzy Set Theory provides tools to handle linguistic terms. Knowledge is expressed in natural language, which is very interesting from the user point of view. However, rule based systems suffer from the problem of the interpretability of knowledge displayed to users. This problem arises because a huge amount of rules is displayed to users, and some rules are not as relevant as required.

This paper addresses the problem of interpretability of rule based systems and studies means to improve it. This study is led with regard to two main methods of generating rules: fuzzy decision trees and fuzzy summaries. The means proposed in this paper are mainly based on the following methods: use of fuzzy terms and quantifiers, reduction of the number of rules, and ordering of rules.

The first method is based on the use of fuzzy summaries and the use of fuzzy quantifiers and fuzzy terms from fuzzy partitions. The second method is mainly

J. Lawry, J. Shanahan, A. Ralescu (Eds.): Modelling with Words, LNAI 2873, pp. 102–123, 2003.
© Springer-Verlag Berlin Heidelberg 2003

based on level-wise algorithms (A-Priori-like) or on the merging of rules. The third method is based on the use of quality measures and aggregators that allow us to order rules as a post-treatment.

The paper is organized as follows: Section 2 describes the two rule generation methods studied in this paper: fuzzy decision rules and fuzzy summaries. Section 3 compares these two methods and highlights the similarities and differences between them. Section 4 proposes methods to improve rule based systems by reinforcing each method by taking advantage of the qualities of the other one. Finally, a summary is provided and future work is described.

2 Two Methods to Construct Fuzzy Rules

2.1 Fuzzy Decision Trees

The principle of the construction of decision trees is based on the successive partitioning of the database by choosing the most discriminant attribute.

Let C be a set of classes representing a physical or a conceptual phenomenon. This phenomenon is described by means of a set of attributes $\mathcal{A} = \{A_1, ..., A_N, A_{N+1}\}$. Each attribute A_j can take a value v_j^l in a set $\{v_j^1, ..., v_j^{m_j}\}$ of possible values (symbolic, numerical, or fuzzy). A *description* is a N-tuple of attribute-value pairs (A_j, v_j^l). Each description is associated with a value c_k of a particular attribute $C \in \mathcal{A}$, called the class C taking values $\{c_1, ..., c_K\}$, to make up an *instance* (or *example*, or *case*) e_i of the phenomenon. In data mining, the construction of a decision tree is done automatically from a *training set* $\mathcal{E} = \{e_1, ..., e_n\}$ of examples, each one with values $e_i(A_j)$ of attributes, $1 \leq i \leq n$, $1 \leq j \leq N$. Algorithms to build decision trees generalize $\mathcal{E}$ to a set of rules (*i.e.* a decision tree) to bring out relations between descriptions and classes in C.

Construction of Decision Trees. Most algorithms to construct decision trees proceed in the same way: the so-called *Top Down Induction of Decision Tree* (TDIDT) method. They build a tree from the root to the leaves, by successively partitioning the training set into subsets. Each partition is done through of a test on an attribute and leads to the definition of a node of the tree. An attribute is selected thanks to a measure of discrimination H. Such a measure enables us to order the attributes according to an increasing accuracy related to the class when splitting the training set. For each attribute A_j, the adequation of its values with the values of C is valued in order to select the attribute the most linked to C. The partitioning is done by means of a splitting strategy P. A stopping criterion T enables us to stop splitting a set and to construct a leaf in the tree. Methods to construct decision trees differ from each other by their choices for H, P and T. The TDIDT algorithm to build decision trees is described hereafter.

TDIDT Algorithm. *Let $\mathcal{E}$ be a training set, $\mathcal{A}$ be a set of $N+1$ attributes. Let $C \in \mathcal{A}$ be a particular attribute called the class.*

(step 1) Select *an attribute* $A_j \in \mathcal{A}$, $A_j \neq C$, *with* m_j *values* $\{v_j^1, ..., v_j^{m_j}\}$ *using the measure of discrimination* H. *The chosen attribute minimizes* $H(C|A_j)$. *This attribute will be used to split the training set* $\mathcal{E}$ *into several subsets. A node* $\eta(A_j)$ *is created in the tree. It is labeled with a test on the value of* A_j.

(step 2) Split *the training set* $\mathcal{E}$ *by means of the selected attribute through the strategy P. For instance, as many subsets* $\mathcal{E}_j^l$ *as values* v_j^l *are created for the previously chosen attribute* A_j*:*

$$\mathcal{E} = \bigcup_{l=1,...,m_j} \mathcal{E}_j^l \quad with \quad \mathcal{E}_j^l = \{e_i \in \mathcal{E} \mid e_i(A_j) = v_j^l\}$$

and $\forall i, k,\ i \neq k,\ \mathcal{E}_j^i \cap \mathcal{E}_j^k = \emptyset$. *Each value* v_j^l *of the attribute labels an edge from the node* $\eta(A_j)$, *this edge leads to the corresponding subset* $\mathcal{E}_j^l$.

(step 3) Check *each subset* $\mathcal{E}_j^l$ *to determine if the stopping criterion* T *is fulfilled. A subset* $\mathcal{E}_j^l$ *that fulfills the stopping criterion enables the creation of a leaf associated with the majority class in* $\mathcal{E}_j^l$. *The size of the leaf is the cardinality of* $\mathcal{E}_j^l$.

(step 4) Start again in step 2.1 with all the subsets that do not fulfill the stopping criterion T.

Classical algorithms, like the ID3 algorithm [22], work well when handling *symbolic* attributes (*i.e.* attributes the values of which are symbolic). However, problems arise when considering *numerical* attributes (*i.e.* attributes having numerical values), or *imprecise* or *fuzzy* attributes (*i.e.* attributes the values of which are fuzzy). Most of these methods enable us to construct decision trees, given a set of symbolic attributes. A few of them enable us to construct them when some attributes are numerical. In the ID3 algorithm, numerical attributes are processed as symbolic ones, each numerical value being considered as a symbolic value. However, with such a method, attributes with a high number of values are often favored by the measure of discrimination. Thus, generated decision trees are generally wide with such a method.

Another method is to discretize the numerical set of values into a set of symbolic values [9, 21]. A threshold labels a test on the values of the attribute in a node of the tree. The threshold can be either a simple value or a fuzzy answer associated with it. Another approach to handle numerical values is to complexify the structure of the node of the tree as a function [7, 10]. Finally, this discretization step can occur either before the construction of the decision tree, as a pre-treatment [6] or during the construction itself [5].

Construction of Fuzzy Decision Trees. To take into account fuzzy values, classical decision trees have been enhanced by the use of fuzzy set theory (FST) into *fuzzy decision trees* (FDT). Thus, FST is an efficient way to take into account imprecision in the description of a case, produced by its measurement for instance, and to set up an interface between numerical values and symbolic values. In a fuzzy decision tree, nodes are associated with fuzzy attributes, edges

are labeled by fuzzy values and leaves can be associated with a fuzzy subset of the class. Methods to construct decision trees in presence of fuzzy attributes are less common and have been developed for few years. FST has been introduced to handle numerical attributes when constructing decision trees [4, 11, 23, 24, 26].

Measures of discrimination must be adapted to enable the construction of fuzzy decision trees from a set of examples described by means of fuzzy attributes. Two kinds of methods to construct fuzzy decision trees can be brought out [20]. The first one deals with methods based on a generalized Shannon entropy measure as a measure of discrimination. The second one deals with methods based on another family of fuzzy measures.

The *entropy of fuzzy events* (see Equation 1), also called *star-entropy* [23], corresponds to the Shannon entropy extended to fuzzy events by substituting Zadeh's probability of fuzzy events P^* to classical probabilities [27].

$$H_S(A_j) = -\sum_{l=1}^{m_j} P^*(v_j^l)\log(P^*(v_j^l)) \tag{1}$$

This entropy has been extended to conditional fuzzy events in order to be used in the process of construction of a fuzzy decision tree [23]:

$$H_S(C|A_j) = -\sum_{l=1}^{m_j} P^*(v_j^l)\sum_{k=1}^{K} P^*(c_k|v_j^l)\log(P^*(c_k|v_j^l)) \tag{2}$$

The SAFI system [23] was one of the first methods of construction of FDT in the domain of inductive learning in presence of fuzzy data. In this software, the star-entropy is used as a measure of discrimination. Fuzzy values on the universe of values of numerical attributes are considered as provided by experts of the given domain.

In the same way, several systems have been introduced to construct FDT by means of the star-entropy, for instance Weber's [24] or Janikow's [11] systems. All these systems construct FDT in presence of given fuzzy values defined on the universe of values of numerical attributes.

The star-entropy is one of the measures of discrimination implemented in the *Salammbô* software [4, 17]. In this software, once the FDT constructed, it is regarded as a fuzzy rule base and the classification process is done by means of the Generalized Modus Ponens [18].

Determination of Fuzzy Partitions. Fuzzy decision trees are often very interesting because they also provide tools to build automatically fuzzy partitions on numerical attributes [17]. This way, rules are more understandable and they provide a better generalization.

Various methods exist to infer such a fuzzy partition [19]. A lot of them are based on statistics to find a fuzzy partition or they are based on questionnaires to construct a membership function by means of answers to basic questions asked to experts or users of the domain. Automatic methods to construct a fuzzy partition are also used.

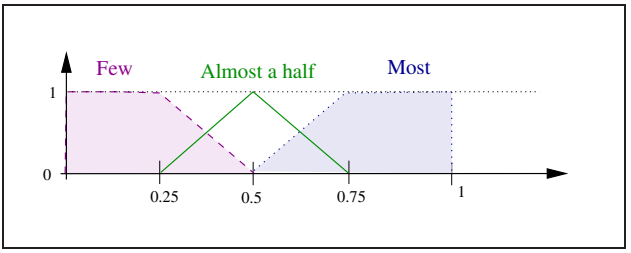


Fig. 1. Examples of quantifiers

In the Salammbô software, the fuzzy partition is constructed by an algorithm based on the use of mathematical morphology operators that are used to filter the given set of values and highlight kernels of fuzzy subsets.

2.2 Fuzzy Summaries

Principle. The principle of the generation of fuzzy summaries is a well-known task [12, 13]. A database $D = \{y_1, \ldots, y_n\}$ is considered. Each object y_i $(i = 1, \ldots, n)$ is described by means of attributes. For instance the attributes (PRODUCT, DISTRICT, MONTH, LEVEL OF SALES) may be considered in the sales database. Attributes are associated with domains which are set of values from which objects are built. For instance, objects like (canoes, New-York, January, bad) may exist in the sales database.

Fuzzy summaries can be built from such databases. A fuzzy summary is a rule such as Q L y_i*'s are* R: τ, where Q is a fuzzy quantifier, y_i is a database object, L and R are fuzzy criteria, and τ is a degree of truth. For instance, in a database related to results of sport sales in USA, *most sales of canoes are bad*. The summaries are built using a set of quantifiers defined by the user and fuzzy criteria, which are fuzzy sets. Quantifiers are described in the universe of frequencies $[0, 1]$ and fuzzy criteria are defined on the universe of the $y's$. Q, L and R are represented respectively by μ_Q, μ_L and μ_R.

For instance, the quantifiers *few, almost a half, most* are considered, as shown in Fig. 1.

Given a set of n objects y_i $(i = 1, \ldots, n)$, the degree τ is computed as follows:

$$\tau = \mu_Q \left(\frac{1}{n} \sum_{i=1}^{n} \top(\mu_L(y_i), \mu_R(y_i)) \right) \tag{3}$$

For each fuzzy summary, five measures of quality are proposed in [13]:

- the degree of truth τ, as described previously, describing to which extent the summary is true regarding the data,
- the degree of imprecision describing the level of precision of the criteria in the summary,

- the degree of coverage describing the number of objects concerned with the summary,
- the degree of appropriateness describing the extent to which the fuzzy summary is characteristic,
- the length describing the number of criteria in the summary.

For a given fuzzy summary, these five measures can be merged into a single value in order to evaluate its global quality, for instance by using OWA operators [25].

In this paper, we are concerned with the automatic generation of such kind of rules from a database (for data mining applications). Due to the very high number of possible rules, existing systems do not generate all rules and rather validate user propositions. For this reason, a new approach based on fuzzy multidimensional databases has been proposed in [15].

The Multidimensional Framework. The OLAP (*On-Line Analytical Processing*) framework corresponds to a set of tools devoted to the storing and analysis of voluminous aggregated multidimensional data [8]. Multidimensional databases have emerged in order to support the OLAP framework. A multidimensional database is a set of *hypercubes* (or simply *cubes*). Each cube is defined on a set of *dimensions* and the cell content corresponds to the *measure* of the cube. Dimensions may be organized into hierarchies in order to represent data at different levels of granularity. For instance, the *sale* cube may be considered. It is defined on three dimensions *PRODUCT*, *DISTRICT* and *MONTH*. Each dimension is provided with a hierarchy. For example, months may be grouped into semesters, and years. In this example, cell values describe the number of units sold for the corresponding product, district and month.

The multidimensional model has been recently extended to the handling of fuzzy data and flexible queries [14]. This extension is used to build fuzzy multidimensional summaries. This approach is mostly based on the principle of association rule generation [1], where only rules having a high support are considered. Fuzzy summaries are close to (fuzzy) association rules. For a detailed comparison, see [15].

An association rule is a rule of the form $L \rightarrow R$. It is a combination of values of attributes consisting in two parts: the left part L (or *antecedent*) and the right part R (or *consequent*), which are both sets of attribute values.

The main measures to evaluate the quality of an association rule are:

- the *covering*: proportion of examples of the database matching the left part.
- the *support*: proportion of examples matching both the left and right part. For instance, the support of the rule *Most sales of canoes are bad* is the number of examples in the sales database where canoes *and* bad sales are considered.
- the *confidence*: proportion of examples matching the left part that are also matched by the right part. This value corresponds to the probability that an example matches the right part of the rule if it matches the left part.

Algorithms of Construction of Fuzzy Multidimensional Summaries. Many algorithms to discover association rules have been proposed since the first definition by Agrawal et al. [1].

As for association rules, the generation of fuzzy multidimensional summaries is performed in two steps. Let us consider a database D where each object is described by means of a set of values for a given set of attributes (or dimensions in the multidimensional case).

The first step consists in building all frequent itemsets (all combinations of values occurring frequently in the database). An itemset is *frequent* if the number of examples matching the corresponding values (*i.e.* its *support*) is greater than a given threshold. This step is performed level by level, considering first single values (to build all 1-itemsets) occurring more frequently among the objects of the database than a user-defined threshold. Then combinations are considered to build 2-itemsets (combination of 2 frequent 1-itemsets) and so on. It can be easily proven that all subsets of itemsets of a frequent itemset are frequent (anti-monotonic property). Thus combining only frequent itemsets guarantees that all frequent itemsets are discovered very efficiently. This method is scalable since it reduces the number of scans of the database and the number of itemsets (and thus of rules) to be built.

The second step of the construction of fuzzy summaries is the generation of the rules themselves, from the frequent itemsets. Most methods consider only rules having a single consequent. However, in [15], the consequent may be either single (constituted by a single value) or complex (with a set of attribute values). For each itemset, every possible rule is considered excepting the rule having an empty consequent. This means that for an itemset of n attribute values, $2^n - 1$ rules are considered. In practice, n is small because of the first step of the algorithm (generation of frequent itemsets). Thus the evaluation of the rules is efficient. The degree of confidence is first computed. Then, given a set of user-defined quantifiers, the degree of truth of the summary is the membership degree taken from the quantifier that best fits the degree of confidence.

For instance, let us consider the frequent itemset ($CANOE$, $CHICAGO$, BAD) built on the three attributes $PRODUCT$, $DISTRICT$ and $LEVEL\ OF\ SALES$ (the size of this itemset is 3). Then 7 rules are considered, among which the rule

$$rule : CANOE \rightarrow CHICAGO\ and\ BAD$$

The confidence of this rule is computed as

$$conf(rule) = \frac{support(CANOE\ and\ CHICAGO\ and\ BAD)}{support(CANOE)}$$

The quantifier having the best degree of membership for *conf(rule)* is chosen. With the examples of fuzzy sets given in Fig. 1, the degrees $\mu_{few}(conf(rule))$, $\mu_{almost\ a\ half}(conf(rule))$ and $\mu_{most}(conf(rule))$ are computed. For instance, the fuzzy summary *Most sales of canoes in Chicago are bad* is obtained.

Note that this method extends the classical level-wise algorithms since it considers multidimensional data whereas classical methods consider mostly binary attributes.

Formal Approaches of Fuzzy Summaries Construction. More formally, let us consider a set of n attributes $\mathcal{A} = \{A_1, \ldots, A_n\}$ where the attribute A_j has values $\{v_j^1, \ldots, v_j^{m_j}\}$.

A summary of size k built on the itemset I is constituted by k attributes $\mathcal{A}' \subseteq \mathcal{A}$ with $\mathcal{A}' = \{A_{i_1}, \ldots, A_{i_k}\}$. For each attribute A_l ($l \in [i_1, i_k]$), a value v_l^j ($j \in [1, m_l]$) is considered.

Thus the itemset I has the following type: $I = \{v_{l_1}^{j_{l_1}}, \ldots, v_{l_k}^{j_{l_k}}\}$. The fuzzy summary is built on a rule of type $L \rightarrow R$ where $L \subset I$ and $R = I \setminus L$.

The support is the number of examples from the database where $A_l = v_l^{j_l}$ for each $l \in [i_1, i_k]$. We have

$$conf(L \rightarrow R) = \frac{support(L \text{ and } R)}{support(L)} = P(R|L)$$

For each quantifier Q, the degree τ_Q of truth of the summary *Q objects that are L are R* is computed as $\tau_Q = \mu_Q(conf(L \rightarrow R))$.

The generation cost can be reduced by several ways. For instance, the user may be allowed to select *a priori* the attributes, the quantifiers, and the description criteria to be used.

Means to refine summaries (from a high level of hierarchy to a lower one) are also considered in the proposed approach. This allows the user to select the part of the data that is relevant for him regarding to first results on aggregated data. For instance, rules are generated at the level of years and then the user decides to refine a discovered rule at the level of months.

This approach is implemented in *FWB* Miner [14], by integrating fuzzy features within Oracle Express Server.

3 Comparison of Both Approaches

This section details fuzzy decision tree based methods and fuzzy summaries by highlighting the construction process and the interpretability of the obtained rules. Both methods are then compared in detail.

3.1 Fuzzy Decision Trees

Fuzzy decision tree based methods are often used in machine learning. In this domain, a model (or, in this case, a rule base) has to be generated automatically in order to constitute the knowledge of a computer system. By induction, from a set of cases (or observations), described by means of a set of values for a given set of attributes, these methods enable the construction of this kind of model.

A principal property which is required for such a model lies in its predictive power. The aim here is to enable an automatic system to decide what happens, thus, any training case must be taken into account during the learning. As a consequence, the final model must cover all training cases in order to ensure its full capability of generalization.

Learning methods as FDT ones are used when classes are known to guide the training phase. Here, the learning process is said *supervised* because a *teacher* is required to label each training case with a class.

Learning algorithms are evaluated in order to be compared on each kind of encountered problems. The principal method of comparison is done by splitting the set of all known cases into two disjoint subsets: the training set used to construct the model, and the test set used to evaluate the model. The evaluation by the test set is done by calculating the *error rate* of the model on the cases of this set. A counting is done for test cases whose class is perfectly predicted (which constitutes the good predicted), and the test cases whose class is not predicted (which constitutes the false predicted). The ratio of good predicted by false predicted will constitute the *error rate* of the model.

3.2 Fuzzy Summaries

Fuzzy multidimensional summaries aim at *describing* the data, either to *validate* user hypothesis, or to *highlight* relevant knowledge on the data. The summaries are not supposed to cover all the data and all possibilities. Contrarily to (fuzzy) decision trees, the set of fuzzy summaries automatically generated from a database may be totally inadequate to evaluate a new example. This case appears when the database do not contain enough data supporting the example.

When generated automatically, as in [15], fuzzy summaries are not built with an idea of the conclusion of the rules. For this reason, this method is said *unsupervised*.

Fuzzy summaries are generated using the whole database, no split between a training and a test set is performed. Thus, their accuracy in future generalization steps for untrained data is not valued.

Scalability is guaranteed since the *support* constitutes an *anti-monotonic* property important for the reduction of the search space.

The quality of the rules is computed by means of several criteria, especially the support and the confidence. User-defined thresholds provide means to reduce the number of rules. Some other criteria are available, that may be used to order the rules (they can be merged, for instance using OWA operators, as mentioned previously).

The good interpretability of rules is due to the fact that these criteria (support, confidence, ...) are involved in the rule set reduction. Moreover, the use of linguistic terms is a good way to provide the user with more understandable rules. Quantifiers are used in order to describe the confidence, and fuzzy partitions of numerical attributes lead to the linguistic description of such values, without the drawback of crisp cuts between terms.

3.3 Comparison

Fuzzy summaries and fuzzy decision trees are mainly different because decision trees concern supervised learning whereas fuzzy summaries are involved in unsupervised learning, and because the first method aims at classifying all examples while the second one aims at describing the relevant data. For these reasons, the construction methods are hardly comparable.

However, these methods are based on counting, and on conditional probabilities. In the two cases, rules can be obtained. With decision trees, all rules have a single conclusion which is a value from the class, while fuzzy summaries consider all possible values as a consequent, including complex consequents built by using several attribute values.

The main property used for the scalability in the automatic generation of fuzzy multidimensional summaries proposed in [15] is based on the support, which guarantees that a sufficient number of examples is available for the rule. As decision trees aim at improving the quality of learning, scalability is hardly considered. However, the support may be compared to the stopping criterion based on the number of examples in the construction of the decision tree.

Table 1. Comparison Between Fuzzy Decision Trees and Fuzzy Summaries

	Fuzzy Decision Trees	**Fuzzy Summaries**
Domain	Machine learning	Database
Type	supervised	unsupervised
Data Management	training set vs. test set	whole set
Construction Criterion	entropy	support
Goal	predictive must cover all cases generalization	descriptive must provide interesting rules relevance, usefulness
Degrees of Quality		
Support	possible stopping criterion	quality and construction criterion
Coverage	-	quality measure
Confidence	leaf impurity	quality measure
Appropriateness	-	quality measure
Imprecision	-	quality measure
Length	optimized by construction	quality measure
Precision/Error rate	to be optimized	-
Recall	to be optimized	-
Complexity/size	to be optimized	reduced by support
Interpretability	good	good
Quantifiers	-	used to express the confidence
Aggregations	used to merge rules	-
Type of Rules	atomic consequent (class)	all types

In decision trees, the fact that the entropy takes into account all values from an attribute in order to select relevant attributes may be a problem since it is possible for an attribute to be very unbalanced in the represented values, for instance.

The confidence in the association rules entails the choice of the best quantifier in the case of fuzzy summaries. In the case of decision trees, confidence may be compared to the degree of impurity in leaves. Table 1 sums this comparison up.

4 Improving the Interpretability

This section addresses the improvement of the interpretability of rule based systems. Considering fuzzy decision trees and fuzzy summaries, several ways of using properties from one approach in the second one are possible. Moreover, methods based on the use of quantifiers, similarities and aggregations are proposed.

4.1 Introducing Fuzzy Summary Methods in the Fuzzy Decision Tree Construction

Fuzzy summaries are very interesting because knowledge is expressed in linguistic expressions, by means of user natural language. The main feature that may be included in this framework is the use of fuzzy quantifiers in order to express the confidence of the rule. The confidence value is given by the impurity in the leaf considered to build the rule. The use of modifiers may enhance this method. Considering a set of quantifiers and modifiers, the combination of a modifier and a quantifier that best fits the degree of impurity is considered. An instance of such a method is described hereafter.

In a FDT, a leaf can be labeled by a set of classes $\{c_1, \ldots, c_K\}$, where each c_j is associated with a weight $P^*(c_j|(v_{l_1}^{j_{l_1}}, \ldots, v_{l_p}^{j_{l_p}}))$ computed during the learning phase. This weight is the probability of belonging to c_j for training examples with values $(v_{l_1}^{j_{l_1}}, \ldots, v_{l_p}^{j_{l_p}}))$ weighted by their membership degree to the leaf. This weight does not exist in classical decision trees because training values are crisp, and a training example only belongs to one leaf. So, in this case, $P^*(c_j|(v_{l_1}^{j_{l_1}}, \ldots, v_{l_p}^{j_{l_p}}))$ is equal to 1 for each c_j.

Thus, a path of a FDT is equivalent to a rule such that:
if $A_{l_1} = v_{l_1}^{j_{l_1}}$ and $\ldots$ and $A_{l_p} = v_{l_p}^{j_{l_p}}$ then $C = c_1$ with the degree $P^*(c_1|(v_{l_1}^{j_{l_1}}, \ldots, v_{l_p}^{j_{l_p}}))$ and $\ldots C = c_K$ with the degree $P^*(c_K|(v_{l_1}^{j_{l_1}}, \ldots, v_{l_p}^{j_{l_p}}))$.

The weights $P^*(c_j|(v_{l_1}^{j_{l_1}}, \ldots v_{l_p}^{j_{l_p}}))$ can be regarded as the strength of the rule. The higher the weight, the more confident the rule. Moreover, this weight belongs to $[0, 1]$ and thus, in the same way as for fuzzy summaries, the fuzzy quantifiers of Fig. 1 can be used to qualify the rules. A fuzzy summary Q A y_i *'s are* C*:* τ, can thus be deduced from the rule:

Q $A_{l_1} = v_{l_1}^{j_{l_1}}$ and $\ldots$ and $A_{l_p} = v_{l_p}^{j_{l_p}}$ *are* $C = c_j : \tau$.

where

$$\tau = \mu_Q \left(P^*(c_j | (v_{l_1}^{j_{l_1}}, \ldots v_{l_p}^{j_{l_p}})) \right) \tag{4}$$

Another approach is to consider the number of examples in the successive partitions as a stopping criterion. This method guarantees that a sufficient number of examples in the database is available to support the rule being built and it guarantees the scalability of the algorithm.

In this case, the TDIDT algorithm of construction of decision trees is adapted considering that the stopping criterion is the number of examples.

A third method aims at building complex rules by means of decision trees. The class may be selected automatically among the attributes (as for unsupervised algorithms). A decision tree is built for each possible class.

In this case, a preliminary step is introduced in the TDIDT algorithm:

(step 0^1*)* *Select* an attribute $C \in \mathcal{A}$.

And another step is added at the end of the algorithm:

(step 5^1*)* Start again in *(step* 0^1*)* and select another attribute.

Moreover, in a way similar to fuzzy summaries, we can construct the class attribute as a Cartesian product of attributes in order to obtain rules with complex consequents (aggregation of values):

(step 0^2*)* Let k, $1 \leq k \leq N+1$, be given. *Select* k attributes $A_{\rho_1}, \ldots, A_{\rho_k}$ in $\mathcal{A}$. $C = \{A_{\rho_1}, \ldots, A_{\rho_k}\}$.

Finally, the fuzzy summaries can be used in a preliminary step of the construction of the decision tree, in order to find candidate attributes that can potentially be used as classes.

(step 0^3*)* Find a set of fuzzy summaries E_{FS}. Let $A_{\rho_1}, \ldots, A_{\rho_k}$ be attributes from $\mathcal{A}$ whose values appear in the most confident fuzzy summaries from E_{FS}. *Select* an attribute $C \in \{A_{\rho_1}, \ldots, A_{\rho_k}\}$.

4.2 Introducing Fuzzy Summary Methods in the Fuzzy Decision Tree Interpretation

Fuzzy summaries can be also introduced in order to enhance the interpretability of fuzzy decision trees. First of all, we have already seen in the last section that a fuzzy decision tree can be transformed in a set of fuzzy summaries.

Moreover, the fuzzy decision tree can be interpreted in the light of the fuzzy summary construction methods. The measure of discrimination (for instance, the entropy of fuzzy events (Equation 1)) that enables us to select attributes associated with a set of fuzzy quantifiers.

For instance, if the measure of discrimination is normalized, the value $H(C|A_j)$ will belong to $[0, 1]$ and can be quantified according to Fig. 2. With

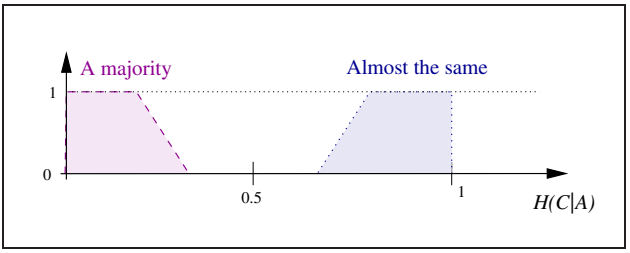


Fig. 2. Quantifiers for $H(C|A_j)$

these quantifiers, if the value $H(C|A_j)$ of a subset $\mathcal{E}$ of examples is near 0, it can be said that *a majority of* examples of $\mathcal{E}$ have the same class. Moreover, if $H(C|A_j)$ is near 1, it can be said that in $\mathcal{E}$ there is *almost the same* number of examples in each class.

This quantification can also be used during the process of construction of the decision tree to give interpretable information to the user. Thus, the selection of attributes can be understood more easily than with the numerical value of $H(C|A)$.

4.3 Introducing Fuzzy Decision Tree Methods in the Fuzzy Summary Generation

This section addresses the possible enhancement of the automatic generation of fuzzy summaries by means of decision tree-like methods.

The main interesting point is that the use of the entropy provides a tool to handle all values from an attribute whereas the current implementation of the multidimensional fuzzy summaries use supports that do not take into account the distribution of values. It may be interesting to consider the entropy in order to select interesting attributes. In this framework, the goal is to select relevant attributes to generate summaries. Several possibilities are considered depending on whether a class (described by an attribute or a combination of attributes) is supposed to be known or not.

More formally, let $C \subset \mathcal{A}$ be the class[1]. An order $\prec$ is defined on the set $\mathcal{A} \backslash C$ of the remaining attributes such that: $A_j \prec A_k$ iff $H(C|A_j) < H(C|A_k)$.

Considering this order, the attributes having the lowest values are the attributes discriminating the class whereas the knowledge of the value of an attribute having a high place in the order do not give much information about the value of the class.

Thus, the attributes having either the lowest values or the highest ones may be kept in order to build fuzzy summaries. Choosing the attributes having the lowest values of entropy with respect to the class will provide fuzzy summaries describing the data, whereas choosing the attributes having the highest values of entropy will provide descriptions of exceptions.

[1] C may be complex, defined by a combination of attributes.

In this framework, the *information gain* of an attribute A_j, $I(A_j, C) = H(C) - H(C|A_j)$ may also be considered. The previous remarks still hold in this case.

It is important to mention that the number of attributes is much lower than the total number of values. So it is still tractable to compute the entropy of each attribute with respect to a class defined as a combination of several other attributes.

If no class is considered, then attributes are ordered by considering their entropy $H_S(A_j)$. This value is low if the attribute has a single value, and high if all values are uniformly distributed. Thus an attribute A_j is said *relevant* if $H(A_j)$ is near 0 (but not null) since a single value in the attribute would lead to well-known fuzzy summaries (even if the support of this value is maximum)[2].

The other method we propose aims at validating the rule bases obtained by means of fuzzy summaries using a test set. The database may then be cut into a training set and a test set. There is no possibility to consider the same validation process as for decision trees. However, each summary built on the training set may be validated in the test set in order to evaluate the rule base. Quality measures may be defined by considering the error on the degree of truth. Considering the ν rules $(r_1, \ldots, r_\nu)$ built from the training set, we denote τ_i $(i = 1, \ldots, \nu)$ the degree of truth of the rule r_i in the training set and τ_i' the degree of truth in the test set. The error ϵ is defined as:

$$\epsilon = \sum_{i=1}^{\nu} (\tau_i - \tau_i')^2$$

4.4 Similarities and Aggregators

The size of a rule base may be reduced by using methods to simplify or merge rules, either by means of *intra-rule* methods, or by means of *inter-rule* methods:

- the same attribute appears several times in the rule,
- several rules are very close.

As described in [3], rules can be merged when the same attribute appears several times in the rule. This case occurs in the framework of fuzzy decision trees since the fuzzy partition is built automatically during the construction of the tree by successive refinements.

The second case appears when several rules share the same values on all attributes but one. The values on this last attribute may then be merged in order to obtain a single rule. Possible merging functions for this method are studied in [2].

In the case of fuzzy summaries, such methods may help the user to refine or redefine the fuzzy partitions he works on.

[2] The same remark is also considered in the association rule generation systems where rules having very high support are not considered.

4.5 Multidimensional Databases and OLAP

Multidimensional databases, and especially fuzzy multidimensional databases provide a very interesting framework to deal with huge volumes of data from the real world. Methods to build fuzzy decision trees and fuzzy summaries from such databases are proposed in [16]. Data are dealt with at different levels of granularity, which enhances the interpretability of the obtained rules.

Amount of Data Scalability is guaranteed in the framework of multidimensional databases. We argue that database management systems are required in order to efficiently manage data and rule bases as shown in [16].

Hierarchies and Granularities Data in multidimensional databases are handled at different levels of granularity. This feature enhances the interpretability of rules because attributes are described in the level of granularity chosen by the user, and because it reduces the size of rule bases.

The user may choose to work at any level of granularity. He has the choice to refine the obtained summaries from a high level of granularity to a lower one.

5 Experimental Results

A set of experiments has been conducted on a database in order to evaluate the methods proposed here. The database concerns the results to a French diploma. In this experiment, a set of 1404 examples is considered, describing the number of students that passed their exam depending on their age, their diploma speciality, their sex, their nationality, the professional category of their parents, and rate of admissibility to the diploma (see Table 2).

The AGE attribute is numerical. The NATIONALITY attribute is either French nationality (1) or foreign one (2). The SEX attribute is either man (1) or woman (2). The speciality (DIPLOMA) attribute and the professional category (called PCS) are coded as labels. The ADMISSIBILITY attribute takes symbolic

Table 2. Description of the set of examples

Attribute	Type	Set of values
ADMISSIBILITY	symbolic	ZERO, NULL, VERY-WEAK, WEAK, HIGH, VERY-HIGH
AGE	continuous	[14, 77]
DIPLOMA	symbolic	111, 112, 113, 211, 212, 213, 214, 215, 216, 217
NATIONALITY	symbolic	1, 2
PCS	symbolic	21, 33, 55, 61
SEX	symbolic	1, 2

Table 3. Number of summaries for various support values

Support	**# rules** *most*	**# rules** *few*	**Total**
0.1	20	2	22
0.14	1	0	1
0.18	2	0	2
0.19	1	0	1
0.29	0	1	1
0.31	1	0	1
0.42	2905	361	3266
0.44	25	0	25
0.47	1	0	1
0.57	1	0	1
0.97	61	0	61
Grand-total	**3018**	**364**	**3382**

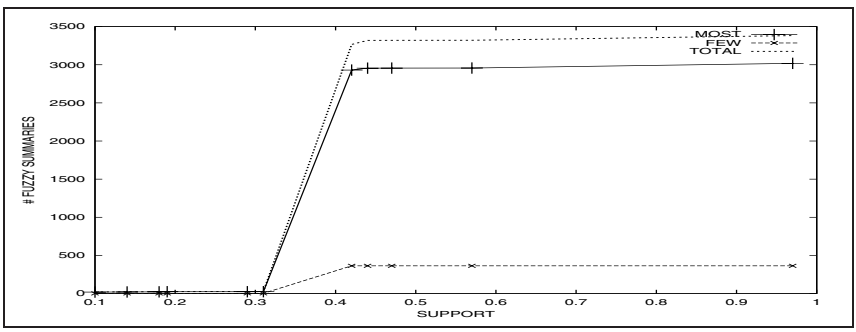


Fig. 3. Cumulative number of fuzzy summaries

values constructed from a numerical rate. ZERO means 0% of students, NULL from 0% to 0.01% of students, VERY-WEAK from 0.01% to 0.12%, WEAK from 0.12% to 1.12%, HIGH from 1.12% to 4.25%, and VERY-HIGH more than 4.25% of students.

Rules obtained from the decision tree construction and from the generation of fuzzy summaries are considered upon their interpretability.

5.1 Fuzzy Summaries

First of all, Table 3 presents the various support values. Fig. 3 illustrates this result.
For instance, some summaries that can be found are the following:

– Summaries with a very high support:
most of 18-year-old candidates belong to the professional category 21.

- Summaries having a conjunctive left part (as for decision-tree rules):
  most of candidates for diploma 111 being 18-year-old belong to the professional category 52.
- Summaries with a complex consequent:
  most of 19-year-old candidates apply for diploma 111 and belong to the professional category 48.
- Summaries expressing exceptions (fuzzy quantifier *few*):
  few 18-year-old candidates are girls.

Some summaries may be merged into a single rule. For instance, the three summaries:

- *when age is 18, PCS is 21,*
- *when age is 19, PCS is 21,*
- *when age is 20, PCS is 21.*

are merged into: *most of the candidates being 18-year-old OR 19-year-old OR 20-year-old belong to the professional category 21.* In this case, a disjunction is used for the left part.

5.2 Fuzzy Decision Trees

To construct a fuzzy decision tree, the attribute ADMISSIBILITY has been chosen as the class among the set of attributes.

A part of the whole fuzzy decision tree that can be constructed from the training set is given in Fig. 4. The whole tree has 194 paths. The longest path has 9 internal nodes (questions on the value of an attribute). In the average, paths have 5.9 internal nodes.

In Table 4, the various sizes of the obtained leaves of the tree are presented.

From this fuzzy decision tree, a set of reduced rules can be obtained by merging questions about the same attribute appearing several times on a given path of the tree. This pruning step does not reduce the whole number of paths

Table 4. Size of the leaves

Size # examples	**# leaves**
4	110
8	52
12	9
16	14
20	2
24	5
28	2
Grand-total	**194**

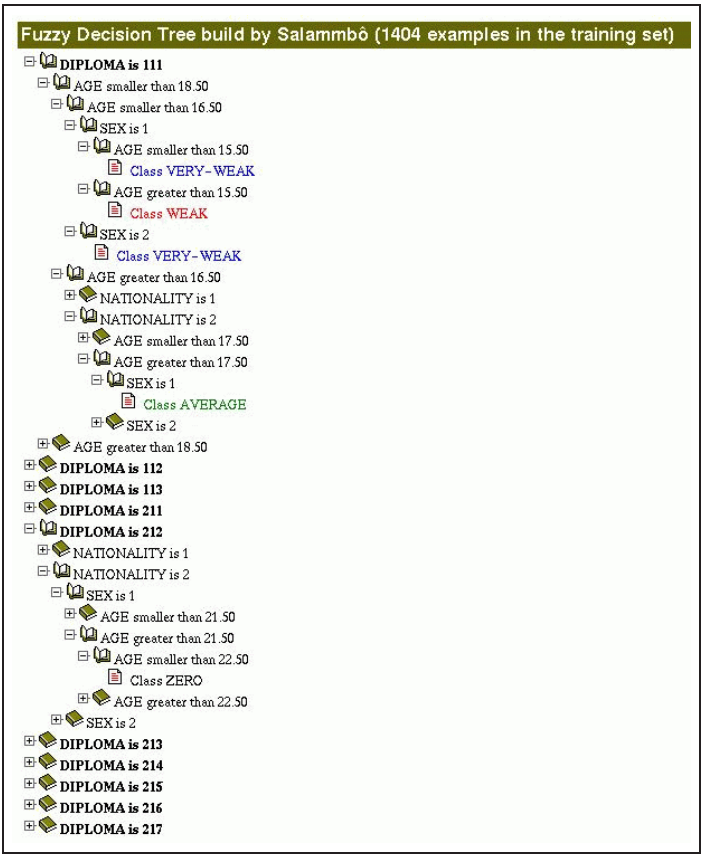


Fig. 4. A part of the induced fuzzy decision tree

of the tree, but it highly reduces the number of internal nodes of paths (less or equal to 4 nodes).

For instance, some instances of the obtained reduced rules are the following:

```
Reduced rule 6:
  IF     DIPLOMA IS 111
    AND  AGE IS around 17 (fuzzy number: core {17}, support [16,18])
    AND  NATIONALITY IS 2
  THEN   CLASS IS VERY-WEAK
Reduced rule 13:
  IF     DIPLOMA IS 111
    AND  AGE IS around 21 (fuzzy number: core {21}, support [20,23])
    AND  NATIONALITY IS 1
    AND  SEX IS 2
  THEN   CLASS IS WEAK
```

```
Reduced rule 114:
  IF     DIPLOMA IS 213
   AND   AGE IS around 20 (fuzzy number: core {20}, support [17,22])
   AND   NATIONALITY IS 1
   AND   SEX IS 1
  THEN   CLASS IS HIGH
```

The fuzzy numbers are induced by the Salammbô software during the process of construction of the fuzzy decision tree (for more details on this method see [19]).

Another way to reduce the size of the tree is to prune it by means of the measure of discrimination. A threshold, from 0 to $\log(K)$ for the Shannon entropy value can be given to stop the construction of subpaths in the tree. For instance, if the threshold is 0.6, the previous tree is composed by only 126 paths. The longest path has 8 internal nodes, and, in average, paths have 5.5 internal nodes.

In this kind of pruned tree, leaves do not contain necessarily only one value of the class, but they may contain a set of values, each one associated with a probability as described in Section 4.1. For instance, such a rule may be:

```
Pruned rule 1:
  IF     DIPLOMA IS 111
    AND  AGE IS under 18 (fuzzy set: core [0,16], support [0,18])
  THEN   CLASS IS NULL with P(NULL|('diploma 111','age under 18'))= 0.86
    OR   CLASS IS WEAK with P(WEAK|('diploma 111','age under 18'))= 0.14
```

Here, fuzzy quantifiers may be used in order to describe the rules more properly (see Fig. 2). Exceptions may then be handled in heterogeneous leaves. The rules are then of the following type (for pruned rule 1 for instance):

- *For* **MOST** *candidates, if diploma is 111 and age is under 18 then class is* **NULL**
- *For* **FEW** *candidates, if diploma is 111 and age is under 18 then class is* **NOT NULL**

5.3 Entropy-Based Rules

In Table 5, partial values of the measure of discrimination that can be obtained during the construction of the tree are given. In this Table, the value of the measure of discrimination has been normalized and its values belongs to $[0, 1]$. The best attribute, selected to split the training set at this stage of the construction of the FDT, is the AGE attribute which minimizes $H(C|A)$.

These measures enable us to generate the following rules about the training set at this step of the construction:

- *most of the examples of this training set are discriminated by attribute AGE,*
- *there are almost the same number of examples for each class when partitioning with attributes PCS or SEX.*

Table 5. Partial results: values of the measure of discrimination

Attribute	$H(C\|A)$ **(normalized)**
AGE	0.26
PCS	0.61
SEX	0.61

These rules are very interesting for the user in order to know how data are dispatched according to the class. The user can understand more easily the process of the construction of the tree.

6 Conclusion

This paper introduces means to improve the interpretability of fuzzy rule based systems. A comparison between methods is proposed to adapt advantages of each approach in order to reduce the drawbacks of the other approach. Moreover, OLAP tools are proposed as a way to improve the interpretability of the generated rules.

These methods are applicable to the classical case, where the same issues appear. The solutions proposed here may improve the interpretability of voluminous rule bases generated by association rule mining algorithms. In this framework, we do not consider ways to reduce the size in memory (as works in the database community do with condensed representations) but rather the number of rules presented to users.

References

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining Association Rules between Sets of Items in Large Databases. In P. Buneman and S. Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D. C., 1993. 107, 108

[2] U. Bodenhofer. Binary ordering-based modifiers. In *IPMU*, pages 1953–1959, Annecy, France, 2002. 115

[3] B. Bouchon-Meunier and C. Marsala. Improvement of the interpretability of fuzzy rules constructed by means of fuzzy decision tree based systems. In *Proceedings of the FSTA'2002 conference*, 2002. extended abstract. 115

[4] B. Bouchon-Meunier, C. Marsala, and M. Ramdani. Learning from imperfect data. In D. Dubois, H. Prade, and R. R. Yager, editors, *Fuzzy Information Engineering: a Guided Tour of Applications*, pages 139–148. John Wileys and Sons, 1997. 105

[5] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification And Regression Trees*. Chapman and Hall, New York, 1984. 104

[6] J. Catlett. On changing continuous attributes into ordered discrete attributes. In Y. Kodratoff, editor, *Machine Learning – Proceedings of EWSL-91*, number 482

in Lecture notes in Artificial Intelligence, pages 164–178. Springer-Verlag, 1991. 104

[7] K. J. Cios and N. Liu. A machine learning method for generation of a neural network architecture: A continuous ID3 algorithm. *IEEE Transactions on Neural Networks*, 3(2):280–290, March 1992. 104

[8] E. F. Codd, S. B. Codd, , and C. T. Salley. Providing olap (on-line analytical processing) to user-analysts: An it mandate. Technical report, Arbor Software White Paper, 1993. 107

[9] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. of the 13th Intl Joint Conf. on Artificial Intelligence*, volume 2, pages 1022–1027, 1993. 104

[10] J.-S. R. Jang. Structure determination in fuzzy modeling: a fuzzy CART approach. In *Proceedings of the 3rd IEEE Int. Conf. on Fuzzy Systems*, volume 1, pages 480–485, Orlando, 6 1994. IEEE. 104

[11] C. Z. Janikow. Fuzzy decision trees: Issues and methods. *IEEE Transactions on Systems, Man and Cybernetics*, 28(1):1–14, February 1998. 105

[12] J. Kacprzyk, R. R. Yager, and S. Zadrozny. A fuzzy logic based approach to linguistic summaries of databases. *Int. Journal of Applied Mathematics and Computer Science*, 10:813–834, 2000. 106

[13] J. Kacprzyk and S. Zadrozny. Computing with words: Towards a New Generation of Linguistic Querying and Summarization in Databases. In P. Sinčák, J. Vaščák, V. Kvasnička, and R. Mesiar, editors, *Proc. of the Euro-International Symposium on Computational Intelligence (ISCI)*, volume 54, Kosice, Slovaquie, 2000. Springer-Verlag. 106

[14] A. Laurent. *Bases de données multidimensionnelles floues et leur utilisation pour la fouille de données*. PhD thesis, Université P. et M. Curie, Paris, France, September 2002. 107, 109

[15] A. Laurent. A new Approach for the Generation of Fuzzy Summaries based on Fuzzy Multidimensional Databases. *International Journal of Intelligent Data Analysis*, 7(2), 2003. 107, 108, 110, 111

[16] A. Laurent, B. Bouchon-Meunier, and A. Doucet. Towards fuzzy-olap mining. In *Proc. of the PKDD Workshop "Database Support for KDD"*, pages 51–62, Fribourg, Allemagne, 2001. 116

[17] C. Marsala. *Apprentissage inductif en présence de données imprécises: Construction et utilisation d'arbres de décision flous*. PhD thesis, Université Pierre et Marie Curie - Paris 6, France, 1998. 105

[18] C. Marsala. Fuzzy decision trees to help flexible querying. *Kybernetika*, 36(6):689–705, 2000. 105

[19] C. Marsala. Fuzzy partitioning methods. In W. Pedrycz, editor, *Granular Computing: an Emerging Paradigm*, Studies in Fuzziness and Soft Computing, pages 163–186. Springer-Verlag, 2001. 105, 120

[20] C. Marsala and B. Bouchon-Meunier. Construction methods of fuzzy decision trees. In *Proceedings of the JCIS'98 Conference*, volume 4, pages 17–20, October 1998. 105

[21] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, Ca, 1993. 104

[22] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):86–106, 1986. 104

[23] M. Ramdani. Une approche floue pour traiter les valeurs numériques en apprentissage. In *Journées Francophones d'apprentissage et d'explication des connaissances*, 1992. 105

[24] R. Weber. Fuzzy-ID3: A class of methods for automatic knowledge acquisition. In *IIZUKA'92 Proceedings of the 2nd International Conference on Fuzzy Logic*, volume 1, pages 265–268, July 1992. 105

[25] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. on Systems, Man and Cybernetics*, 18(1):183–190, 1988. 107

[26] Y. Yuan and M. J. Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and systems*, 69:125–139, 1995. 105

[27] L. A. Zadeh. Probability measures of fuzzy events. *Journal Math. Anal. Applic.*, 23, 1968. reprinted in "*Fuzzy Sets and Applications: selected papers by L. A. Zadeh*", R. R. Yager, S. Ovchinnikov, R. M. Tong and H. T. Nguyen eds, pp. 45–51. 105

Humanist Computing: Modelling with Words, Concepts, and Behaviours

Jonathan Rossiter

AI Research Group, Department of Engineering Mathematics, University of Bristol
Bristol, BS8 1TR, UK
Jonathan.Rossiter@bris.ac.uk

Abstract. In this paper we present a new approach to the modelling of data and knowledge called Humanist Computing. We describe how the general principle of Humanist Computing, namely the modelling with words, concepts and behaviours defines a hierarchy of methods which extends from the low level data-driven modelling with words to the high level fusion of knowledge in the context of human behaviours. We explore this hierarchy and focus on a number of levels in the hierarchy. For each level we describe the general Humanist Computing approach and give specific examples based either on approximations to the real-world or the real-world itself.

1 Introduction: Motivating Humanist Computing

Humanist computing seeks to combine the following three elements of soft computing; 1. modelling with words, 2. conceptual specifications based on these words and their associated perceptions of natural concepts, and 3. simple models of human behaviour based on these conceptual definitions. Humanist computing is a methodology inspired by observations of human perceptions and behaviours. The general principle is that humans are good at solving problems, summarising information and making decisions with incomplete information. It seems natural therefore that when we are faced with the problem of building artificial systems to perform similar tasks we take some inspiration from the observation of humans. Although humans are not infallible (far from it!) we must surely have some faith in our own behaviours and reasoning processes.

Of course, human behaviour is a function of its environment and consequently a high degree of specialisation has evolved in some of our behaviour. We should therefore say at the start that mimicking human behaviour has an impact on the type of problems humanist computing can solve. A very simple analogy is in the realm of hard computing. The arithmetic processing of the human brain can in no way match the raw power and determinism of modern computers. We would therefore not wish to model arithmetic computing using humanist computing techniques. On the other hand, humans are particularly good at operating in new, unfamiliar, and uncertain environments. These environments would seem to provide good problem domains in which humanist computing could give us some advantages.

J. Lawry, J. Shanahan, A. Ralescu (Eds.): Modelling with Words, LNAI 2873, pp. 124–152, 2003.
© Springer-Verlag Berlin Heidelberg 2003

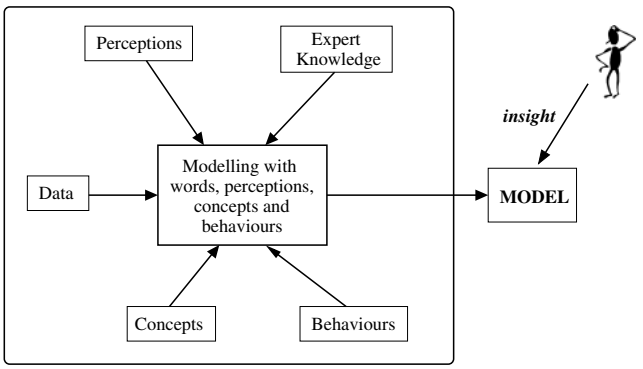


Fig. 1. Humanist Computing

We can also motivate our development of humanist computing from the point of transparency. In an increasing number of critical applications it is important for us to be able to understand and evaluate the computer model that has been developed. Examples include automated medical diagnosis and engineering risk assessment. It would seem natural therefore to base these models on forms that are natural to humans. What could be more natural for us to interpret than concepts and behaviours that we all exhibit? This neatly leads to the emerging characteristic of transparency. Transparency in artificial systems enables us to verify the system and to build up confidence in its representation and reasoning mechanisms. In developing a humanist model our assumption of transparency can be extended so that even non-expert observers will be able to understand the underlying reasoning processes. The key tenet here is that any motivation for transparency in artificial systems can be logically extended to modelling in a human-like manner. In other words, when building artificial systems we can most readily understand those that behave in some way similar to ourselves. As illustrated in figure 1, humanist computing generates models from a combination of domain data, perceptions, concept definitions, behaviours and expert knowledge through uncertain machine learning, modelling with words and information fusion. The resulting models give us some new (and clear) *insight* into the problem domain. As such these models are *glass-box* rather than *black-box* in nature.

As a final consideration with respect to this glass-box or transparency property, we might argue that any fully glass-box representation should provide the most clear, flexible and rich description of the model. In reality any computationally tractable calculus claiming to model semantics on words is likely to be severely restricted. Consequently the flexibility and richness of the representation suffers. In the case of modelling with words we may wish to represent our words by fuzzy granules, and to subset these into just those granules representing simple perceptions. Given such a restriction our corresponding vocabulary of words will be much reduced. As a result we can only go as far towards true

transparency as a *murky-box*, or *foggy-box* model. That said, we would advocate that in practical situations, a murky insight into a problem domain is better than no insight at all.

2 A Hierarchy of Methods for Humanist Computing

In order to fully describe a framework that will enable humanist computing we must consider how we can relate computing with words, perceptions, concepts and behaviours. To this end we define a simple ordering of these soft methods based on their modelling forms. We can thus illustrate how a mechanism for humanist computing can be built up from these soft computing paradigms.

At the lowest level in this humanist computing model ordering we will take the primitive to be the uncertain definition of a word. This is taken to be the granular definition put forward by Zadeh [24]. Such granular words are defined with labels such as *small*, *medium*, or *large*. We then subsume this form with the linguistic definition of perception also proposed by Zadeh [25]. Here the granular words are defined by labels such as *small_weight* or *high_cost*. We now extend the ordering with a higher level form defining a concept. At this level a conceptual definition defines a natural, observable concept which can be described in terms of granular words and granular perceptions. Concepts also allow us to introduce rules linking perceptions and words. For example the concept *tall man* can be modelled by a soft concept that not only includes the granular perceptions *tall_height* but also the perception *large_feet* and the implication rule $large_feet \rightarrow tall_height$.

Finally we subsume our uncertain concepts inside the form of an uncertain behavioural model. A behavioural model extends a concept with some representation of human behaviour. These behaviours can range from the extremely objective and specific, such as driving above the speed limit, to the extremely subjective and unspecific, such as feeling jealousy or fear.

Lets us formalise this a little by constructing the hierarchy H of humanist computing defined using the ordering discussed above. Consider the set of granular words W. Using W we can perform the machine learning operation of inferring relations between this words. We term this process modelling with words (represented by the symbols M_W) and it, together with the set W, defines the lowest level class C_W in our humanist computing hierarchy H. Now consider the immediate superclass of C_W which defines perceptual computing. Here we define the class C_P which contains a set of perceptions P defined on granular words W. The operation on C_P is termed modelling with perceptions (i.e. M_P), that is, the inference of relations between perceptions. We next extend this class into the class of conceptual definitions C_C which contain richer conceptual descriptions of the real world than single perceptions allow. These richer descriptions typically define some structured conceptual relationships through notions of dependence or causality. Where M_P and M_W can be defined, for example, using basic Bayesian operations the class C_C requires a richer representation and also a more complicated reasoning process such as those used in

Table 1. The Humanist Computing Hierarchy

Class	*Focus*	*Operations*	*Operands*	*Examples*
C_B	Behaviours	M_B	$\{B, C, P, W\}$	decision making
$\downarrow$				
C_C	Concepts	M_C	$\{C, P, W\}$	$\{manager,\ car\}$
$\downarrow$				
C_P	Perceptions	M_P	$\{P, W\}$	$\{hot,\ fast\}$
$\downarrow$				
C_W	Words	M_W	$\{W\}$	$\{small,\ medium\}$

Bayesian networks [18] or uncertain conceptual graphs [6]. C_C therefore contains the set of conceptual descriptions and operations on these can be termed modelling with concepts, or M_C. Likewise the operation M_C on C_C is the inference of relations between concepts. Finally we extend our class of conceptual definitions with simple descriptions of human behaviour to give the top level class C_B. C_B therefore contains the set of simple behavioural models B and operations on C_B can be termed modelling with behaviours, or M_B. It should be clear that we have formed a hierarchy defined by a simple subsumption relation.

Table 1 illustrates the four classes in the humanist computing hierarchy, their scope of operation, and the operands that they act upon. The symbol $\downarrow$ is taken to mean that the class above subsumes the class below.

Let us take for example the uncertain modelling of components of a car. At the lowest level measurements of individual components within the car will involve numerical uncertainty and this can be modelled using words (i.e. *modelling with words*, or M_W). Now let us consider a specific temperature sensor in the engine of the car. Here we are dealing with a perceptual quantity, namely temperature. At this level we would model the sensor using perceptions (i.e. *modelling with perceptions*, or M_P). At the next level, we could consider the whole car. here the concept of a car starts to come into play and we must consider the typical components that describe a car. For example, cars usually have four wheels, they usually have at least two seats, and so on. Here the concept of a car is a structured definition based on the components and corresponding uncertainties that go to make it up. At this level we need to model the uncertain concept that defines a car (i.e. *modelling with concepts*, or M_C). At the very highest level of humanist computing we also model behaviours, such as driving the car. Here we need to model not just the concept, perceptions, and words that describe the car, but also the behaviours that humans exhibit when performing the operation of driving. In this example we may also wish to divide driver behaviour into a number of prototypical classes such as *safe driver*, *aggressive driver*, and so on.

This hierarchy of modelling forms in some way shares some common components with one view of the knowledge induction mechanisms in humans. This perspective [22] links perceptions of the real world through a series of inference

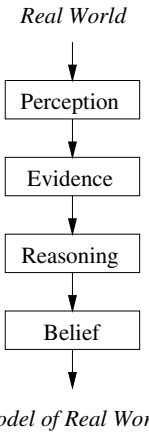


Fig. 2. Conceptual dependencies

steps to the models of the real world that humans naturally construct. There are four key stages in this process; *belief*, *reasoning*, *evidence* and *perception*. The inference path from real world to model is shown in figure 2.

Here the real world provides the measurable quantities (or data) with which we can perform modelling with words. Perceptions in figure 2 enable us to start to interpret the data from the real world. In addition, perceptions themselves can give us some measure of conceptual structure between words (for example we perceive that a *big* person has *large* shoes and is *tall*). Processing at the higher levels (reasoning and belief) involves the processing of concepts and behaviours.

This process of generating a model from perceptions of the real world assumes a linear flow through the four concepts above. More complex models can include other links, such as from belief to reasoning. In such a case an existing belief may colour how new sensory information is reasoned with.

The one key similarity between the dependencies in figure 2 and the ordering in table 1 is the inclusion of some model of perception. Since this is a focal point in both our interaction with the real world and our attempts to model it with humanist computing it bears some more careful examination.

2.1 Perception

In our daily lives we humans constantly rely on our perception systems. Through familiarity we rarely ponder on how sophisticated they are. The significant point to make is that, no matter how we view our perception systems they are the only route that information can take from the outside world into the brain.

At the lowest level our perception systems are merely measurement devices, or transducers. The eyes turn photons into electrical signals and the ears turn sound energy into similar electrical signals. Experiments have shown that this transducing is not the complete picture. The eyes and the optic nerve, for ex-

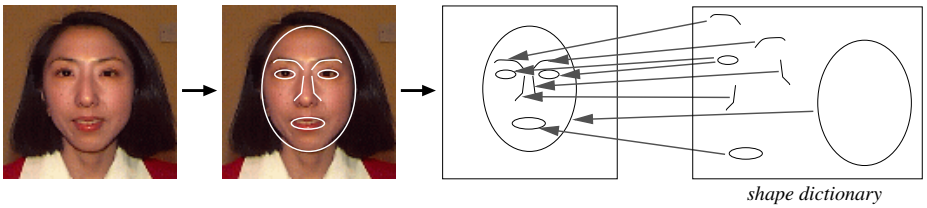


Fig. 3. Simple dictionary of perceptual concepts

ample, contain some processing elements which detect important features. This processing also serves to reduce the amount of raw information fed to the brain.

At a higher level the perception system reaches what Goldman [12] calls a *terminus*. The terminus of the perception system is the final high level perceptual representation we use. This terminus of perception can involve complex structuring and organisation within the brain.

Take for example a face. The raw information entering the eye is simple colour-point information. The perceptual processing performed in the eye and brain may lead to a terminus of perception based on lines, points, regions of colour, relative position etc.

Figure 3 shows how a dictionary of perceptual concepts can be built up at the terminus of perception using simple two-dimensional shapes.

Clearly in this example the perceptions that may be present at the terminus of perception in humans are somewhat more complicated than this typically modelling using computing with perceptions. In other words, the rigid ordering in table 1 is far less well defined in the human perception system and we could view a perception as a somewhat higher level concept than previously discussed. It is with this in mind that in later sections we give examples of modelling with concepts and words which, to some degree, encompass modelling with perceptions without explicitly referring to perceptions. In effect we have a reduced ordering where perceptions can be modelled either with simple words or with more complex concepts. This is shown in table 2.

Table 2. The Humanist Computing Hierarchy

Class	*Focus*
C_B	Behaviours
$\downarrow$	
C_C	Concepts $\cup$ Perceptions
$\downarrow$	
C_W	Words $\cup$ Perceptions

2.2 Practical Approaches to Humanist Computing

In the next sections we will illustrate three approaches to modelling and learning from data using humanist computing which all involve modelling with words, concepts and behaviours:

1. An object oriented approach to modelling with words.
2. A method for structured information fusion based on specificity.
3. A behaviour based approach to uncertain belief updating.

3 An Object Oriented Approach to Modelling with Words

Here we review our recent work on a framework for uncertain taxonomical knowledge representation and reasoning. We discuss how the implementation of these theories in Fril++ [3] has enabled the development of a toolkit for objected oriented machine learning with words. We focus here on machine learning by fuzzy rule induction. This object oriented approach to model induction using words is in keeping with humanist computing because of the emphasis on learning fuzzy rules which naturally lend themselves to a linguistic interpretation.

As presented in [20] an object oriented approach to modelling with words has the following features:

1. **Clear and Natural Representation**
   We apply class hierarchies to many parts of our daily lives. Indeed one might argue that uncertain taxonomies are *necessary* for us to function in, and understand, a world with so many varied objects. The shear impossibility of developing special responses to every perception has led us to develop a higher level taxonomical view of the world which does no require specific scientific knowledge. For example, we may class trees into 'large trees' and 'small trees'. We may then split 'large trees' into 'quite large trees' and 'very large trees', as shown in figure 4. The important thing to see here is that the linguistic terms commonly used in computing with words (large, small, very large, etc) may also be integral to class descriptions. Clearly we need underlying mechanisms for handling uncertainty in words in these hierarchies. This will be discussed in the next section.
2. **Scalability of Knowledge Representation**
   Object oriented modelling with words naturally has a measure of scale through different perspectives of the hierarchy. If we build a model that has hundreds of classes in our hierarchy we can focus on the appropriate level of the hierarchy for the appropriate linguistic description of the model we are looking for.
   In effect we can summarise the model at as many levels as there are levels in the hierarchy. A *complex* summary involves knowledge from *lower down* the hierarchy while a more *general* summary involves knowledge from just the *top* of the hierarchy.

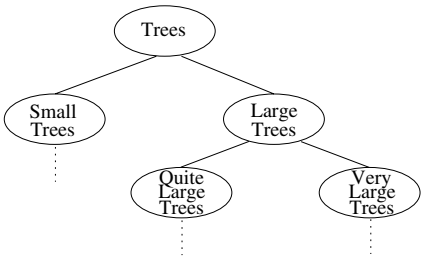


Fig. 4. A hierarchy of trees

3. **Power of Inheritance, Overriding, Encapsulation and Information Hiding**
   From a knowledge representation point of view inheritance helps reduce unnecessary repetition in a class hierarchy. This aids model conciseness. Overriding, on the other hand enables us to form a richer hierarchical model, and in extreme cases to implement forms of non-monotonic reasoning.
   Encapsulation (the grouping of methods and attributes within a class) and information hiding (restricting access to properties with a class) are features that can make the final model more robust. Although these are essentially programming aids they are important in modelling in words where the models are produced to solve real-world problems.

3.1 The Underlying Framework

We have described a new approach to uncertain object oriented representation and reasoning in [8] and have implemented these theories in Fril++, the uncertain object oriented logic programming language. Fril++ enables us to define hierarchies where classes can contain uncertain properties (attributes and methods) and objects can have uncertain memberships in classes. Interval probabilities are used to represent uncertainty in class memberships and property applicability. Intervals across the [0, 1] domain give us a richer representation than point probabilities.We adopt a subsumption partial ordering in place of a fuzzy subset ordering for the sub class relationship. This avoids the problem of a symmetric sub class relation which could generate a network rather than a hierarchy, and thus reduces computational complexity. Membership of an object to a class can be calculated by matching uncertain properties followed by the application of an interval probability form of Jeffrey's rule [15].

3.2 An Extendible Object Oriented Data Browser

We will now give an example of modelling with words in our uncertain object oriented language, Fril++. We overview an implementation of a tool for dataset modelling and data mining. The theory behind the object oriented data browser builds on research into a non-object oriented data browser in [2]. The data

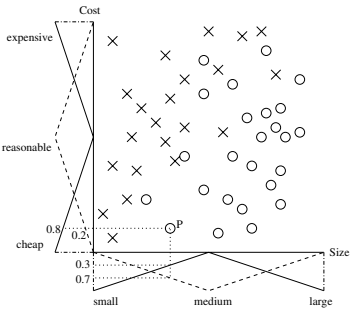


Fig. 5. A Dataset object

browser tool is required to approximate functions and to learn classification relationships from example databases.

To provide a clear and extendible hierarchy for the data browser we consider the general case of a machine learner. We split a typical machine learner into a number of parts; the inducer, the tester, and the data. We abstract these three components into high level Fril++ classes as follows.

- **The Data**
  The data class is an abstraction of all forms of data into a two dimensional table. Tuples are ordered and attributes can be named. Each attribute can have a corresponding universe specified for it. Each of these universes can be split into any number of partitions. Partitions can be crisp or fuzzy. In the case of fuzzy partitions the partitions can define a list of words. In such a way the 3-attribute data object shown in figure 5 is described by the individual data points and the fuzzy words on the axes.
  We can represent any point value by a fuzzy set on words. Take for example the point P in figure 5. Within the Size domain the point P can be represented by the fuzzy set on labels, $\{small : 0.3, \quad medium : 0.7\}$.
- **The Model**
  The model is the basic class for all objects that predict or classify from data. The model object, once built by an inducer, models or summarises the data that was used to build it. In this paper we concentrate on fuzzy rule models that are linguistic in nature and are expressions on words.
- **The Inducer**
  The inducer class defines all the objects that perform induction from data. An inducer operates on data and generates models. These models can then be used to classify or predict from new data. For all examples in this paper we use a simple example of an inducer that constructs fuzzy rulesets from data. The inducer class can be subclassed to implement a more complicated induction process.
- **The Tester**
  Once the model object (a fuzzy rulebase in our examples) has been built it can be tested using an object of the tester class. A tester object takes

the model object and applies data to the model. Results of classification or prediction can be aggregated and presented as rms errors, classification errors and so on.

3.3 Data Modelling Examples

We present some examples of modelling with words using the extendible object oriented data browser. Each attribute in both datasets are partitioned into triangular fuzzy sets representing words.

2D Shape Databases Here we learn two 2d shapes; a doughnut, and a figure-eight shape. Using the learnt models we attempt to reclassify points to determine if they are legal (part of the shape) or illegal (outside the shape).

- Figure-Eight
  This is a classification problem based on a simple two dimensional figure eight shape. This is effectively the XOR problem, which is particularly difficult for decomposing machine learning algorithms like this one. Figure 6 shows the form of the shape and figure 7 shows the points from a regular grid that have been classified as legal. Notice the symmetry in the points across both X and Y while the actual shape is not symmetrical in X or Y. This is due to the decomposing nature of our simple rule induction algorithm. Of course, there are many more effective machine learning algorithms available but we choose this one here for its simplicity and for its use in the following section on information fusion. Even so this model correctly classifies points in the figure eight with an accuracy of 84%.
- Doughnut
  This is similar to the figure-eight above, again generating decomposition errors when learning using X and Y independently. Figure 8 shows the form of the shape and figure 9 shows the points from a regular grid that have been classified as legal. The decomposition error causes the shape to deform about the sides. This is simply because of the number of illegal class points in those projected regions swamping the legal points. The learnt model of the doughnut classifies points correctly with an accuracy of 85.5%.

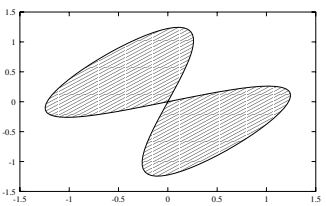


Fig. 6. Figure eight shape

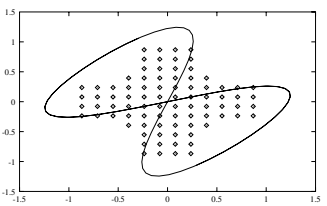


Fig. 7. Learnt figure eight shape: 84%

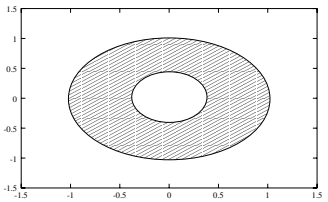


Fig. 8. Doughnut shape

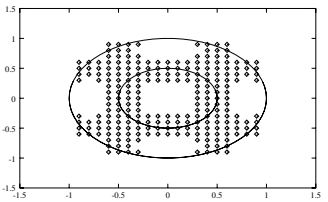


Fig. 9. Learnt doughnut shape: 85.5%

Facial Feature Database This database contains 168 instances of 18 measurements on human faces. The goal is to classify instances into sex. See [5] for more information about this dataset. The dataset was split randomly into a training set of 138 samples and a test set of 30 samples. Each of the 18 attributes was partitioned into 20 triangular fuzzy sets, with 50% overlap, representing 20 words. The generated model was represented by two Fril fuzzy rules, one for male and one for female. The generated model classified the test dataset with an result of 83% correct.

Lung Image Region Classification Recently the clinical diagnosis of lung disease has concentrated on non-invasive tests such as chest x-rays and CT and MRI scanning. For a single patient these methods produce a large number of cross sectional scans of the lungs. For example, a typical CT scan data set consists of over 12.5MB of data. The radiologist must examine all this data and generate a report that is typically 100 words in length and a diagnosis of typically 10 words in length. In information terms this is equivalent to an astonishing compression ratio of approximately 200,000:1. This is possible because the radiologist utilises two skills. Firstly, he uses his perceptual skills to process the low-level data in front of him, and secondly he uses his high-level expert knowledge and experience to guide his review and to help interpret the low-level data. Any approach to automating the review process must therefore involve a combination of low-level perceptual processing and high-level reasoning. This problem is therefore ideally suited to humanist computing

The first stage in this process is to extract lung fields (figure 10b) from lung CT scans (figure 10a). An image processing algorithm has been developed which will process individual 2d images and detect lung fields. (For details of the background to this work see [13] and [14]). Unfortunately solely using image processing is not wholly effective and expert judgement is again required to sift correct lung fields from incorrect fields. This is shown in figures 11a and 11b where the trachea has been incorrectly included as part of the lung field. We therefore require a machine classifier which will determine if a candidate lung field is truly part of the lung.

We learn a model from a database of 1203 samples taken from 10 patients with varying degrees of lung disease. Each sample consists of the following eight

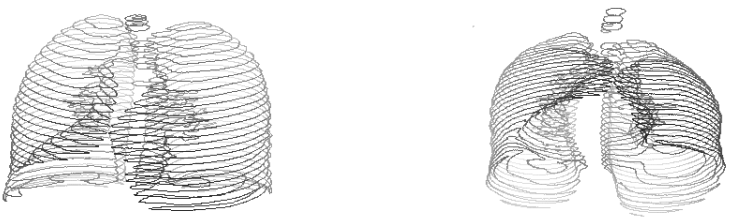
attribute values; 3 coordinates for the centre of the region (X, Y, Z), the width and length of the region (H, W) and the number of pixels inside the region (SZ). Figure 12 shows 3d plots of the data projected into the 3-spaces $[X, Y, Z]$ and $[H, W, SZ]$. It is quite clear from figure 12 how the legal points (shown as diamonds) roughly form the structure of the lung shown in figure 11.

After learning a model for this data using our object oriented data browser we reclassify the database using the learnt model. The decomposing learner achieved an accuracy score of 87.4%. The misclassified points are shown in figure 13 as diamonds.

Given the decomposing nature of this machine learner and by inspection of figure 12 it is clear that this problem must be decomposable. Even so, in

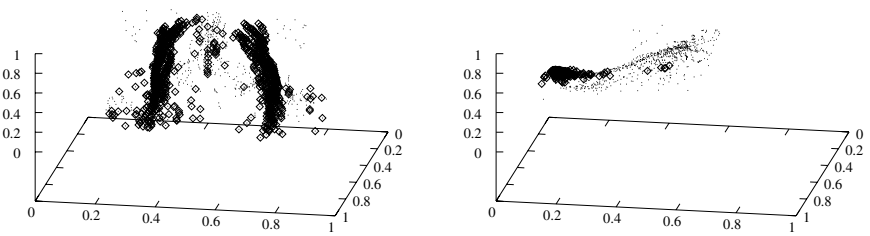
a. Typical CT scan b. Corrected extracted lung fields

Fig. 10. Processing a 2d lung image



a b

Fig. 11. Lung volume reconstruction



$[X, Y, Z]$ $[H, W, SZ]$

Fig. 12. 3d lung data visualisations

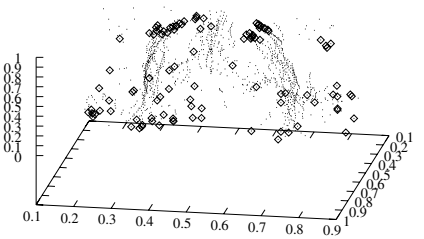


Fig. 13. Misclassifications (large points)

a medical environment the accuracy of a diagnosis tool must be extremely high in order to be put into everyday use. In the next section we attempt to improve on the accuracy of such models by fusing in high level, expert knowledge.

4 A Method for Structured Information Fusion Based on Specificity

In the previous section we showed how an object oriented implementation of machine learning can be used to learn models based on words from data. Let us now consider how we can merge such learnt models with higher level expert-derive knowledge. This is particularly relevant to humanist computing where humans can not only provide the inspiration for model construction but they may also have important high level knowledge to impart that may be difficult to learn directly from data. The goal for any fusion methodology is to perform better at decision making (classification, regression, etc.) using the fused model then using either of the component models on their own.

4.1 Deriving High-Level Knowledge and Low-Level Information

Hierarchies are, in themselves, consistent with expert and real world information. Humans naturally categorise and rank the world around them and this impinges on the way we handle and communicate information. Lung disease for example can be split into sub classes such as “cancer” and “emphysema”. The class of “cancer” can be split up further into more specific sub classes, and so on. Clearly information higher up the hierarchy is more general and information lower down the hierarchy is more specific.

Given our chosen framework of uncertain class hierarchies for the representation of both expert and learnt information we must now consider how the two can be merged, or ‘fused’. Any approach to fusion must accept an importance ordering on the information to be fused. Such an importance ordering gives us three base cases:

1. expert knowledge is given more weight than learnt information,
2. expert information has a lower weight than learnt information,
3. learnt and expert information are equally weighted.

We must also consider how expert or learnt information provides prior information that can be used in the construction of the class hierarchy, the transcription of expert information or the way in which learning is performed. Some circumstances may demand that we take expert knowledge to be background (i.e. prior) information which then influences the learning process. Alternatively we might say that learnt information has roots in empirical analysis and results from machine learning should be fed back to experts who can then revise the transcription of their expert knowledge. In this paper we give equal weight to expert and learnt knowledge.

Let us consider the concept T which we are required to model. The expert transcribes his knowledge of T and this is used to construct an uncertain class C_1. At the same time we perform some machine learning from a database of exemplars of T. The learnt knowledge is used to construct a second uncertain class C_2. We now require that C_1 and C_2 be fused. Here we crucially assume equal weight to expert and learnt information and ensure that C_1 and C_2 are constructed independently. Since, in our example, we know that C_1 and C_2 are related through the common concept T we rank the two classes at the same level in the class hierarchy, i.e. they have the same immediate superclass C_s. We can now use the mechanisms of uncertain hierarchical reasoning to query the fused hierarchy. Figure 14 illustrates this simple fusion of classes defining the same concept T.

4.2 Hierarchical Structure Fusion

Where in the previous section we fused a single class into a hierarchy at a known point we now consider the problem of fusing hierarchies where we have no prior knowledge of where the fusion should occur. In the simplest case we can just focus on the maximal superclass of each hierarchy. Let us take two simple hierarchical structures. Depending on prior bias and on the some measure of matching and/or consistency we may find that the most appropriate fusion is that shown in figure 15a. Notice that in this case the sub-hierarchies retain their structures. A more complicated form of hierarchical fusion may result in the structure shown in figure 15b. Here consideration has been given not only to the place at which

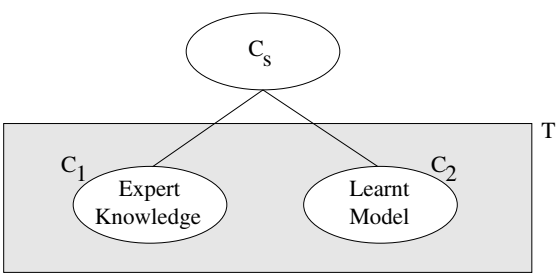


Fig. 14. Fusing two, conceptually related, classes

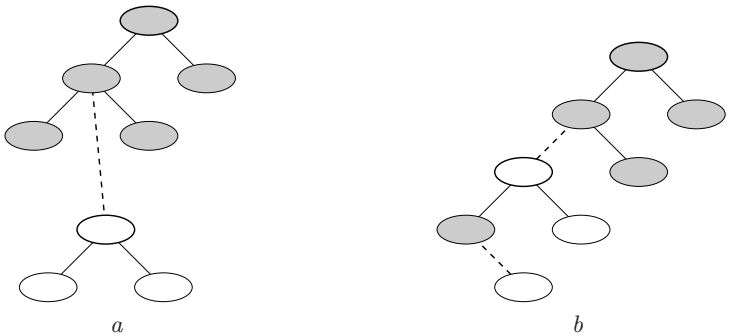


Fig. 15. Example fusions

to fuse the maximal superclass, but also the most appropriate places at which to fuse subclasses. Clearly this later approach is considerably more complicated.

We now suggest a number of ways in which the most appropriate fusion point for a maximal superclass could be determined. This is the simplest structural fusion task as illustrated in figure 15a.

1. **Concept Matching.** By concept matching we mean the matching of all properties within the classes that contribute to the definition of the relevant ontology. For example, let us consider classes C_1 and C_2 where C_1 contains properties $\psi_{1,1}$, $\psi_{1,2}$ etc. and C_2 contains properties $\psi_{2,1}$, $\psi_{2,2}$ etc. By comparing $\psi_{1,1}$ with $\psi_{2,1}$ and $\psi_{1,2}$ with $\psi_{2,2}$ and so on, we can determine a match of C_1 to C_2. If properties are crisp values or fuzzy sets we may use semantic unification [4] to determine values for $Pr(\psi_{1,1}|\psi_{1,2})$, $Pr(\psi_{2,1}|\psi_{2,2})$, and so on. We can then insert the maximum superclass at the same level as the maximum matching class.
2. **Cross Entropy.** Here we consider probability distributions across all possible objects that are, to some degree, consistent with a class definition. We use the principle of minimising cross-entropy [23] to compare a class definition from one hierarchy with a class definition from the other hierarchy. In this way we can determine and select the class having the minimal cross entropy, and hence having the maximal similarity.
3. **Empirical Testing.** Perhaps the simplest comparison method is to force a classification or prediction test upon each class and determine the classes which are most consistent in test results. This may be computationally very expensive and, we would argue, is rather inelegant and brutish.

4.3 Default Reasoning for Inconsistency Resolution

Unfortunately, although the expert knowledge and learnt knowledge could be entirely consistent it is common for there to be some inconsistency between the two. This may be the result of errors or generalisations in expert knowledge, or there may be some problem with the learnt model, either through generalisation

or noise. These inconsistencies are especially important in the context of uncertain class hierarchies. Here an object can have a graded membership in a class depending on property matching. It holds that since properties can be uncertain, an object may have a finite (although often very small) membership in any or all classes. In this case any reasoning about a property ψ must take into account *all* definitions for ψ in the hierarchy. Any reasonably large and diverse hierarchy is likely to incorporate some inconsistent information with respect to ψ. Of course there are mechanisms for dealing with independent and conflicting information (such as Dempster's rule) but these are often blunt instruments that do not take into account the information contained in the hierarchical structure itself. Another approach is to isolate the most inconsistent information and eliminate it from the reasoning until a consistent conclusion is produced. The problem with this approach is that we would effectively be searching for all possible consistent subsets of contributing evidence and selecting the most consistent. This is an exponential time complexity problem.

A better approach is to focus on a fundamental characteristic of our uncertain hierarchy: information lower down the hierarchy is more specific. Cao described a default reasoning algorithm for uncertain property inheritance using such a specificity ranking assumption which is polynomial in time complexity [7]. In this section we apply a modified form of Cao's algorithm to the problem of consistency resolution in fused uncertain hierarchies.

4.4 Two Simple Examples

We illustrate uncertain hierarchical information fusion using the two simple example class hierarchies of expert and learnt information shown in figures 16 and 17 which define the conceptual shapes shown in figures 6 and 8 respectively. The learnt models of the figure eight and doughnut are those that have been learnt using the object oriented data browser described previously in this paper. In this way we show how object oriented modelling with words and information fusion fit into the proposed humanist computing framework. The expert knowledge is coded as simple fuzzy rules on a vocabulary of five words, $\{very_small, small, medium, large, very_large\}$, with the following linguistic equivalent forms:

Figure Eight

Class is **illegal** if x is *small* or *very_small* and y is *small* or *very_small* or if x is *large* or *very_large* and y is *large* or *very_large* or if x is *very_small* and y is *very_large* or if x is *very_large* and y is *very_small*.

Class is **legal** if x is *large* and y is *small* or if x is *small* and y is *large*.

Doughnut

Class is **illegal** if x is *medium* and y is *medium* or if x is *very_large* or *very_small* or if y is *very_large* or *very_small* or if y is *large*

and x is *large* or if y is *large* and x is *small* or if y is *small* and x is *small* or if y is *small* and x is *large*.
Class is **legal** if x is *large* or *small* and y is *medium* or if y is *large* or *small* and x is *medium*.

Figures 18 and 19 show the classification results using just the expert hierarchies. The accuracy values show the percentage of test points classified correctly. Notice how the expert knowledge, represented using simple linguistic fuzzy rules, captures the general shapes but, especially in the doughnut case, tends to over-generalise the structure of the concepts.

When we fuse the expert and learnt classes and apply Cao's default reasoning algorithm to resolve inconsistencies we generate the classification results shown in figures 20 and 21. It is interesting to observe the effects of the fusion and reasoning processes on the structure of the classification results. Fusion gives slightly worse results for the figure eight problem and slightly better results for the doughnut problem. The important point here is that we have fused learnt and expert knowledge in a common hierarchical representation and have been able to draw measured conclusions about the shape of the concepts being represented through the stratified selection of consistent knowledge. Further to this, we can examine the linguistic rules that have been combined and this may be of great use to experts who wish to learn more about a problem domain from the learnt model.

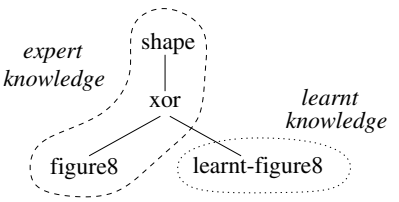


Fig. 16. Figure eight hierarchy

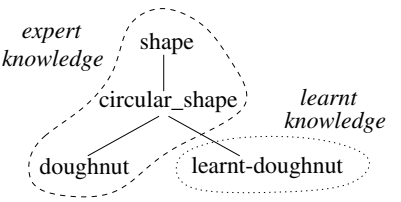


Fig. 17. Doughnut hierarchy

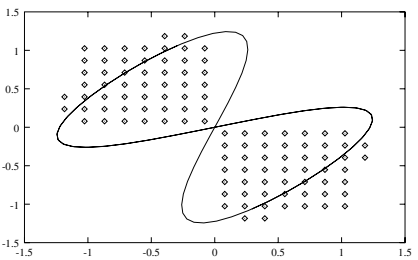


Fig. 18. Expert's figure eight: 81.5%

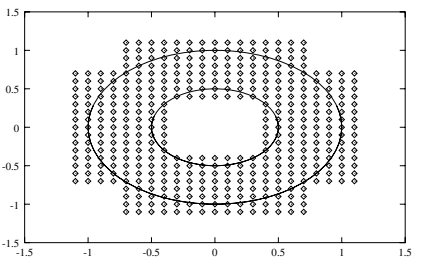


Fig. 19. Expert's doughnut: 81.3%

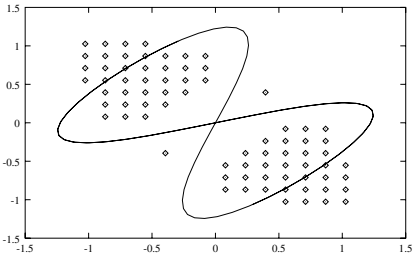


Fig. 20. Fused figure eight: 79.5%

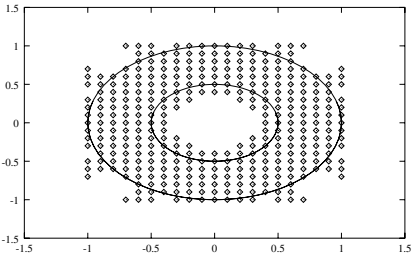


Fig. 21. Fused doughnut: 88.8%

4.5 An Alternative Default Fusion Strategy

In Cao's inconsistency resolution algorithm the final step is to evaluate the following equation:

$$S(\psi|O) = \bigcup_{p_i \in P'} (\bigcap x \mid x \in p_i) \tag{1}$$

where $S(\psi|O)$ is the support for property ψ being true given object O and P' is a set of consistent subsets selected from the class hierarchy.

While the aggregation function shown in equation 1 is formally reasonable it is questionable in practice. The tendency in the union operation is to widen the support region without taking into account the intervals of uncertainty *within* the supports. Take for example the support intervals $[l_1, u_1]$ and $[l_2, u_2]$ which we combine by interval union to yield $[\text{MIN}(l_1, l_2), \text{MAX}(u_1, u_2)]$. If the union operation gives us $[l_1, u_1]$ then this clearly has not taken into account the specificity of the narrower interval $[l_2, u_2]$. A better approach may be to join these supports in a more considerate way, such as through an interval disjunction. Another alternative is to defuzzify the supports since we commonly require the final support for a property to be expressed as a singleton. In the following alternative aggregation the interval conjunction is implemented as in equation 2 and the interval disjunction is implemented as in equation 3.

$$[l, u] \wedge [n, p] = [l \times n, \quad u \times p] \tag{2}$$

$$[l, u] \vee [n, p] = [l + n - l \times n, \quad u + p - u \times p] \tag{3}$$

Motivation for the use of logical rather than set operations is based on the selective nature of the default reasoning algorithm. The algorithm selects a set of consistent subsets which form a *theory* for the applicability of property ψ. Determining the applicability of ψ can therefore be thought of as the resolution of this theory into a single conclusion. Logical operations are a natural alternative for resolving this theory. Our new aggregation operator is shown in equation 4.

$$S(\psi|O) = \bigvee_{p_i \in P'} (\bigwedge x \mid x \in p_i) \tag{4}$$

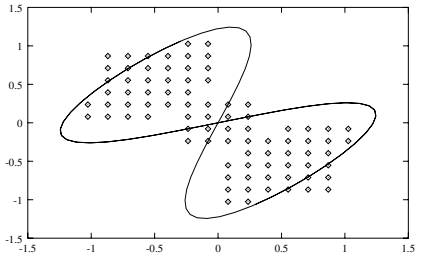


Fig. 22. Improved fused figure eight: 85%

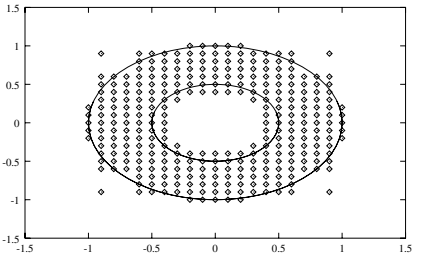


Fig. 23. improved fused doughnut: 92.1%

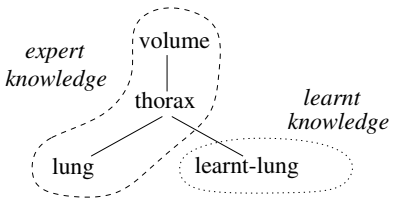


Fig. 24. Lung hierarchy

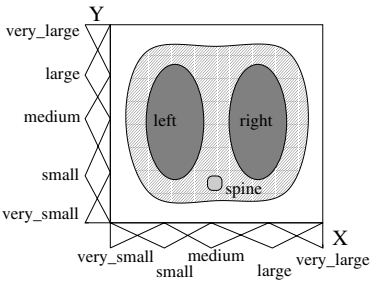


Fig. 25. Simplified model of 2d lung image

Figures 22 and 23 show the results of classifying the figure eight and doughnut shape using this alternative aggregation operator. Notice the improvement in both the classification accuracy and also in the general shape being modelled.

4.6 A Problem of Priorities

We have shown in the previous section how combining expert and learnt knowledge can yield a more accurate model, while at the same time retaining some degree of transparency. Now let us look at a case where such fusion is less successful.

Previously we showed how a model of the lung region database can be learnt using object oriented modelling with words. Given the requirement for very high success rates in medical applications we desire that the accuracy of 87.4% be improved upon. By examining the misclassified points in figure 13 we could conclude that by adding some expert knowledge defining the valid lung volume we could classify quite a number of these points correctly. To this effect we define the uncertain class hierarchy in figure 24.

We now define the expert knowledge in classes volume, thorax and lung in terms of five fuzzy words defined on each of the x and y domains shown in figure 25. For the volume class we simple define both legal and illegal to be true

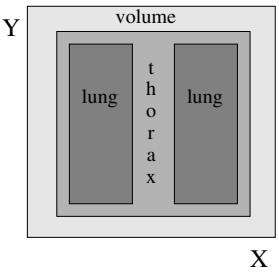


Fig. 26. Restrictions on expert knowledge through the hierarchy

for any point in the unit cube. For the thorax class we define the legal points as those in the cube further restricted by the condition $x = y = \{very_small : 0,\ small : 0.5,\ medium : 1,\ large : 0.5,\ very_large : 0\}$. For the lung class the rules for legal and illegal are further restricted to remove the points between the left and right lung regions. These restrictions are shown graphically in figure 26.

Unfortunately the expert classes perform badly (68%, the same as the majority class) when called upon to classify test points. This is because the specification of expert knowledge in the domain x and y is not enough to fully capture the shape of the volume. Consequently, when we perform the information fusion with this class and the learnt model from the object oriented data browser we get a result of just 68.5%.

The reason for the poor fusion result stems from the equal priority that the default reasoning algorithm puts on both the expert and the learnt models. Clearly in this instance the expert knowledge, though at first seeming reasonable, should have a much lower weighting in the fusion. At present no mechanism is included in this approach for weighting knowledge from different sources. This is an important next step in the development of this research.

5 A Behaviour Based Approach to Uncertain Belief Updating

In this final section we briefly describe recent work on modelling human behaviours in the context of user modelling. Since we are modelling humans, this is an area where humanist computing is a natural approach to adopt.

The problem of user recognition centres on the temporal aspect of user behaviour. We have some set of known user prototypes $\{U_1, \ldots, U_n\}$, the behaviours of which we know and to which we provide a corresponding set of services. An unknown user u at time t behaves in the fashion b_t, where behaviour is commonly the outcome of some crisp or fuzzy choice, such as whether or not to buy expensive wine. We wish to determine the similarity of u to each of $\{U_1, \ldots, U_n\}$ in order that we provide the appropriate service to u at time t. We must repeat this process as t increases.

In an object oriented environment we construct a hierarchy of n user classes, $\{c_1, \ldots, c_n\}$, and we try to determine the support $S_t(u \in c_m)$ for user u belonging to user class c_m at time t. This support is some function f of the current behaviour b_t and the history of behaviours $\{b_1, \ldots, b_{t-1}\}$. This is shown more generally in equation 5.

$$S_t(u \in c_m) = f(\{b_1, \ldots, b_t\}) \tag{5}$$

$$S_t(u \in c_m) = g(S_{t-1}(u \in c_m), b_t) \tag{6}$$

We can solve this problem at time t if we have the whole behaviour series up to t. Unfortunately at time $t+1$ we will have to do the whole calculation again. Where t is very large the storage of the whole behaviour series and the cost of the support calculation may be too expensive. An alternative approach is to view the support $S(u \in c_m)$ as some belief in the statement *"user u belongs to class c_m"* and this belief is updated whenever a new behaviour is encountered. This belief updating approach is more economical in space since the whole behaviour series no longer needs to be stored. In computation this approach is more efficient since we now must calculate the function g of just the previous support $S_{t-1}(u \in c_m)$ and the latest behaviour b_t. This belief updating approach is shown more generally in equation 6.

In this paper we examine the case where belief is used to represent the membership of a user object in one or more prototype user classes. As implemented in our uncertain hierarchy, memberships are represented by a support interval $[n, p]$ which is a sub interval of $[0, 1]$.

We now examine two approaches to uncertain interval belief updating. The first method defines a simple moving average which takes new evidence to be *absolute* and updates belief to a degree defined entirely by the new evidence. The second method is more interesting in the light of humanist computing since it is inspired by studied of real human behaviour. This second method, Einhorn and Hogarth's anchor and adjustment belief revision, assesses new evidence in the context of our current beliefs and updates our beliefs *relatively*.

5.1 A Moving Average Update Function

The generalised moving average update function [20] calculates the current support S_{n+1} for a hypothesis given the previous support S_n and new evidence x_n+1 as shown in equation 7.

$$S_{n+1} = \frac{n^{\lambda} S_n + n^{1-\lambda} s(x_{n+1})}{n^{\lambda} + n^{1-\lambda}} \tag{7}$$

Where λ would typically lie in the interval $[0, 1]$. This function is applied to the upper and lower bounds of the belief support interval $[l, u]$ independently. The averaging nature of this function ensures that the validity of the support interval is maintained (i.e. $l \leq u$). If $\lambda = 1$ we have a simple moving average expression where current belief is n times as important as new evidence. If $\lambda = 0$

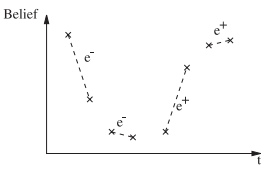


Fig. 27. Order effects in anchor and adjustment

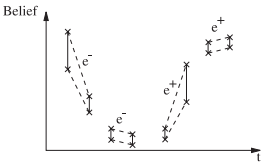


Fig. 28. Order effects in interval anchor and adjustment

we have an expression that weights new evidence n times as important as current belief. This flexibility may be important in cases where we know that users change their style of behaviour and must therefore be re-classified quickly. For example, a user p is a student for a number of years and therefore cannot afford expensive food or wine. Upon graduation p gets a well paid job and starts to spend their earnings on expensive groceries. From their behaviour we should detect that p has moved from a class of low income to a class of high income. If $\lambda \approx 1$ and n is large this averaging function will react very slowly to such a change.

Notice that we have some freedom in the λ parameter to vary the function's updating from a *recency* bias (the greater influence of recent evidence) to a *primacy* bias (the greater influence of initial evidence). This flexibility is an important characteristic to be taken into account when considering the suitability of this approach, and the value of λ, for a particular application.

5.2 Anchor and Adjustment Belief Revision

If we are to classify human users we should look at how humans might perform this classification task. Hogarth and Einhorn have done much work on models of belief updating that bear some relation to human behaviour [10] [11] . They have suggested that the strength of current belief can have a major effect on how new evidence updates that belief. For example, the stronger the belief a person has in the trustworthiness of a friend, the greater the reduction in this belief when the friend commits an act of dishonesty. This is a kind of 'knee jerk' or 'shock' reaction. The typical patterns of behaviour is shown in figure 27. Here the negative evidence e^- has two differing effects depending on how large the belief was before e^- was presented. Likewise there are two differing effects of the equal and opposite positive evidence e^+.

Hogarth and Einhorn's anchor and adjustment belief revision model [11] updates a belief given new evidence through two processes. Equation 8 shows

how belief S_k is updated given new *negative* evidence. Equation 9 shows how the same belief S_k is updated given new *positive* evidence.

$$S_k = S_{k-1} + \alpha S_{k-1}[s(x_k) - R] \qquad \text{for } s(x_k) \leq R \tag{8}$$

$$S_k = S_{k-1} + \beta(1 - S_{k-1})[s(x_k) - R] \qquad \text{for } s(x_k) > R \tag{9}$$

R is a reference point for determining if the evidence $s(x_k)$ is positive or negative, and typically $R = 0$ or $R = S_{k-1}$. α and β are constants which define how sensitive the model is to negative or positive evidence respectively.

Since evidence in our uncertain environment can be presented as a support interval we must consider the implications of an interval representation of evidence on the anchor and adjustment model. For a piece of evidence e with the associated support interval $[l, u]$ we can view l as the positive evidence associated with e and $1-u$ as the negative evidence associated with e. The general principle is that, given a current belief $[n, p]$ and a piece of evidence with support $[l, u]$, belief increases by an amount proportional to $1 - p$ and belief decreases by an amount proportional to n.

We can apply equations 8 and 9 to the support interval to yield equations 10 to 13 where S^- and S^+ are the lower bound and the upper bound of belief respectively.

$$S_k^- = S_{k-1}^- + \alpha S_{k-1}^-[s^-(x_k) - R^-] \qquad \text{for } s^-(x_k) \leq R^- \tag{10}$$

$$S_k^- = S_{k-1}^- + \beta(1 - S_{k-1}^+)[s^-(x_k) - R^-] \qquad \text{for } s^-(x_k) > R^- \tag{11}$$

$$S_k^+ = S_{k-1}^+ + \alpha S_{k-1}^-[s^+(x_k) - R^+] \qquad \text{for } s^+(x_k) \leq R^+ \tag{12}$$

$$S_k^+ = S_{k-1}^+ + \beta(1 - S_{k-1}^+)[s^+(x_k) - R^+] \qquad \text{for } s^+(x_k) > R^+ \tag{13}$$

R^- is a reference point for determining if the lower bound of the presented evidence is positive or negative with respect to the lower bound of belief and R^+ is the corresponding reference point for the upper bound of belief, and where $R^- = S_{k-1}^-$ and $R^+ = S_{k-1}^+$ and $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$. The precise effects of negative evidence e^- and positive evidence e^+ is determined by α and β respectively. The typical pattern of belief updating behaviour for the interval case is shown in figure 28. Notice how the support interval may contract or expand depending in the interval, magnitude and sign of the new evidence and also on the previous belief.

This new interval version of Hogarth and Einhorn's belief updating model has a number of advantages over the previous generalised averaging method. Recency characteristics allow the anchor and adjustment model to reclassify users quickly. The order effects of this model are related to human behaviour and this seems an important consideration where we are recognising human users and where we are working towards humanist computing. In addition, this method allows us to control the effects of positive and negative evidence separately. This last feature may be especially important in medical applications where false negative classifications have far more serious consequences than false positive classifications.

5.3 A User Recognition Example: The Iterated Prisoner's Dilemma

The n player iterated prisoner's dilemma problem [1] [16] is a good testbed for user recognition due to the production of streams of human-like behaviour that are a result of the pair-wise interaction of simulated prisoners. The problem is most easily understood by looking at the non-iterated problem with $n = 2$. Two prisoners are due to be sentenced. They each have the choice to cooperate together or to defect. If the players both cooperate they will both serve 3 years. If they both defect they will both serve 1 year. If they choose to behave differently then the defecting player will serve 0 years but the cooperating player will serve 5 years. The iterated problem simply continues the game after each round.

A wide range of strategies are possible, ranging from trusting behaviour (always cooperate) to defective behaviour (always defect) and including more complex strategies such as conditional cooperation (cooperating unless the opponent's last m behaviours were defect).

5.4 The User Recognition Example in Fril++

If we were to construct a class hierarchy of prisoners in Fril++ it could resemble figure 29. The subclasses of *prisoner* are the classes that define prototypical prisoners and their behaviours. For each of the prisoner classes we construct at least one instance. A tournament is then initiated between these instances. After each round of the tournament the memberships of each instance in each prototypical class is determined. The supports for these memberships, i.e. $\{(\alpha_T\ \beta_T)$, $(\alpha_D\ \beta_D)$, $(\alpha_R\ \beta_R)$, $(\alpha_P\ \beta_P)\}$, are calculated using the belief updating methods discussed previously. [8] explains how object recognition is performed, and hence how memberships are calculated, in Fril++.

As an example let us consider the population of ten prisoners $\{p_1, \ldots, p_{10}\}$ where there are two instances of each of the classes {*cooperative*, *defective*, *tit-for-tat*, *random*, *respd*} and whose behaviours are defined in table 3.

A population of ten prisoners was created and a game of 75 rounds was initiated. Each round involved picking pairs of prisoners at random from the population until non were left and for each pair recording the behaviours they exhibit (defect, cooperate, etc). From the past history of each player, and using the techniques described earlier ($\alpha = \beta = 0.3,\ \lambda = 1$), they were classified into the five behaviour classes. The winning class was taken as the class in

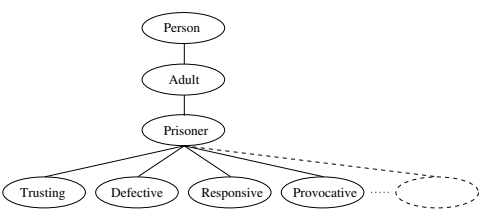


Fig. 29. A class hierarchy for the prisoner's dilemma problem

Table 3. Behaviour classes

Behaviour	Description
cooperative	always cooperate with opponent
uncooperative	always defect against opponent
tit-for-tat	cooperate unless last opponent defected
random	equal random chance of defect or cooperate
respd	defect unless the last 6 opponents chose to cooperate

Table 4. The prisoner population

Individual	Behaviour before 60^{th} round	Behaviour after 60^{th} round
1	random	cooperative
2	random	uncooperative
3	cooperative	tit-for-tat
4	cooperative	respd
5	uncooperative	random
6	uncooperative	respd
7	tit-for-tat	cooperative
8	tit-for-tat	random
9	respd	tit-for-tat
10	respd	uncooperative

Table 5. Classification results before 60^{th} round

Interval a-a method	Generalised average method
63.6%	63.3%

which minimum membership (i.e. the lower bound of the membership interval) was greatest. If the winning class matched the actual class in table 4 then the classification was recorded as a success.

To recreate the situation where user behaviour changes, after 60 rounds the behaviours of all 10 prisoners was changed, as shown in the third column of table 4. After this point the game was continued for 15 rounds. We compared classification results using the interval anchor and adjustment belief updating method with the generalised moving average method described above. The whole process was repeated five times and the mean of the results was take.

As can be seen from table 5, classification results before the 60^{th} round (the point of behaviour change) are quite similar between the two methods. After the 60^{th} round, however, there is a marked difference in the results, with a large fall in the performance of the generalised average approach. The effect can be seen most clearly in figure 30 where the generalised average approach is slow to react

Table 6. Classification results after 60^{th} round

Interval a-a method	Generalised average method
57.3%	22.2%

to the change of behaviour of prisoner number 3 from cooperative to tit-for-tat. For clarity figure 30 shows only the lower bound of the belief interval defining the support (current belief) for membership in each class.

6 Further Applications of Humanist Computing

6.1 Autonomous Robots in Dangerous Environments

A robot must learn how to move in dangerous environments in order to perform some helpful task - such as searching for humans in fires. This will involve learning rules from sensor input such as audio, visual and motion. (For an example of the low-level fusion of such sensor data see [17]). Unfortunately, when training the robot we cannot teach it using real hazardous situations because, firstly, these situations are difficult to manage and expensive to set up, and secondly, the robot could be damaged before it learns how to behave safely.

We need, therefore, to give the robot some expert knowledge and human-like behaviour concerning dangerous environments and the tasks it must perform. This knowledge may be common-sense in nature such as 'do not walk on soft ground if you have thin legs and are quite heavy' or may be more specialised information such as 'move toward badly injured humans, especially those that are not moving and are in rooms with dense smoke'.

6.2 Recreating Human Behaviour in Robots

In the near future human-like robots will have to operate in the world of the human. Their behaviours will have to be complementary and integrated with

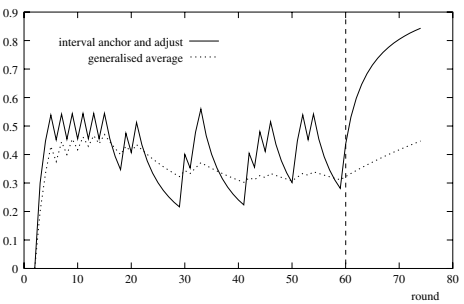


Fig. 30. Membership of prisoner 3 in class tit-for-tat

ours. On the other hand, we require that intelligent machines perform tasks more efficiently and more rapidly than humans. Unfortunately speed and pleasant behaviours at not always compatible. For example, when two people are about to walk through a door together it is common for the first person to hold the door open for the second. This is a simple courtesy. On the other hand, when the first person is very busy he may just rush through the door on his own and not be courteous. If a robot has been trained with a reinforcement learning algorithm it may learn that the fastest way to complete its tasks is to never hold the door open for anyone else. This will be particularly unhelpful when the robot fails to help someone who is carrying a large load with both hands.

Here the robot needs not only to sense the environment and work towards its immediate goal but also to take into account expert high-level knowledge and human behaviour such as 'unless you are very busy, always hold the door open for someone else'.

7 Conclusions

In this paper we have proposed Humanist Computing as an intuitive extension to modelling with words, perceptions, concepts and behaviours. We have given three concrete examples of humanist computing in application to data modelling, information fusion and user classification. While models constructed using humanist computing may not necessarily perform classification or regression tasks better than alternatives, the advantage of the humanist computing approach is in the natural representations of knowledge and behaviour. Using such natural, human-inspired, representations and behaviours we generate transparent glass-box models (in reality these are more murky-box in nature) that we can inspect and query and in which we can have some intuitive confidence.

In the simplest terms we have motivated humanist computing by the assumption that a model which reasons in the same way as us and perceives the world in the same way as us is likely to make the same decisions as us. Clearly, where humans are not good at making the correct decisions we should not use humanist computing. On the other hand there are a large number of applications where humans work naturally and efficiently with vague and uncertain perceptions and concepts. It is in these environments that humanist computing offers the most promise.

Acknowledgements

I would like to thank James Luckraft and Paul Brassnett for their contributions in the respective problem areas of lung region extraction and information fusion. I would also like to thank John Malone for his work on hand labelling the large number of extracted lung regions generated by James' extraction program.

References

[1] Axelrod R.: The evolution of cooperation New York: Basic Books, (1985) 147
[2] Baldwin J. F.: Fuzzy Rule Automation from Data using Mass Assignment Theory. Proceedings of FUZZ-IEEE/IFES 1995, no. 4, pp. 674-679, (1995) 131
[3] Baldwin J. F., Cao T. H., Martin T. P., Rossiter J. M.: Towards Soft Computing Object Oriented Logic Programming. Proc. of the Ninth IEEE Int. Conf. on Fuzzy System, FUZZ-IEEE 2000, 768–773, (2000). 130
[4] Baldwin J. F., Lawry J., Martin T. P.: Efficient Algorithms for Semantic Unification. Proceedings of IPMU 1996, (1996) 138
[5] Brunelli R., Poggio T.: Caricatural Effects in Automated Face Perception. Biological Cybernetics, Vol. 69, pp. 235-241, (1993). 134
[6] Cao T. H.: Fuzzy Conceptual Graphs: A Language for Computational Intelligence Approaching Human Expression and Reasoning. In Sincak, P. et al. (Eds): The State of the Art in Computational Intelligence, Advances in Soft Computing, Physica-Verlag, pp. 114-120, (2000). 127
[7] Cao T. H.: Uncertain Inheritance and Reasoning as Probabilistic Default Reasoning. Int. J. of Intelligent Systems, **16**, pp. 781–803, (2001). 139
[8] Cao T. H., Rossiter J. M., Martin T. P., Baldwin J. F.: Inheritance and recognition in uncertain and fuzzy object oriented models Proceedings of NAFIPS 2001, (2001). 131, 147
[9] Cao T. H., Rossiter J. M., Martin T. P., Baldwin J. F.: Inheritance and Recognition in Uncertain and Fuzzy Object Oriented Models. Proc. of the 1st Int. Joint Conf. of the Int. Fuzzy Systems Ass. and the North American Fuzzy Information Processing Soc., IFSA/NAFIPS, (2001)
[10] Einhorn H. J., Hogarth R. M.: Ambiguity and uncertainty in probabilistic inference. Psychological Review, Vol 93, pp. 433-461, (1985). 145
[11] Einhorn H. J., Hogarth R. M.: Order effects in belief updating: the belief-adjustment model. Cognitive Psychology, Vol 24, pp. 1-55, (1992). 145
[12] Goldman A. I.: Epistemology and cognition. Harvard University Press, pp 344-358, (1986). 129
[13] Horwood A., Hogan S. J., Goddard P. R., Rossiter J. M.: Computer-assisted Diagnosis of CT Pulmonary Images. Applied Nonlinear Mathematics Technical Report #200109, University of Bristol, (2001). 134
[14] Horwood A., Hogan S. J., Goddard P. R., Rossiter J. M.: Image Normalization, a Basic Requirement for Computer-based Automatic Diagnostic Applications. ANM Technical Report #200110, University of Bristol, (2001). 134
[15] Jeffrey R. C.: The logic of decision. McGraw-Hill, (1965). 131
[16] Martin T. P.: Incremental Learning of User Models - an Experimental Testbed. Proceedings of IPMU2000, (2000). 147
[17] Mukai T., Ohnishi N.: Sensor Fusion of a CCD Camera and an Acceleration-Gyro Sensor for the Recovery of Three-Dimensional Shape and Scale. Proc. of the 2nd Int. Conf. on Information Fusion (FUSION'99), pp.221-228, (1999). 149
[18] Pearl J.: Probabilistic reasoning in intelligent systems: Networks of plausible inference. San Francisco, CA: Morgan Kaufmann (1988). 127
[19] Rossiter J. M., Cao T. H., Martin T. P., Baldwin J. F.: User Recognition in Uncertain Object Oriented User Modelling Proceedings of the Tenth IEEE International Conference on Fuzzy System, FUZZ-IEEE 2001, (2001)
[20] Rossiter J. M., Cao T. H., Martin T. P., Baldwin J. F.: object oriented modelling with words Proceedings of IEEE Workshop on Modelling with Words, FUZZ-IEEE 2001, (2001).. 130, 144

[21] Rossiter J. M., Cao T. H., Martin T. P., Baldwin J. F.: object oriented modelling with words. Proc. of the Tenth IEEE Int. Conf. on Fuzzy System, FUZZ-IEEE 2001, (2001).
[22] Rossiter J. M.: Humanist Computing for Knowledge Discovery from Ordered Datasets. University of Bristol, (2000). 127
[23] Shore J. E., Johnson R. W.: Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy. IEEE Trans. Inform. Theory, IT-26(1), 26-37, (1980). 138
[24] Zadeh L. A.: Fuzzy Logic = Computing With Words. IEEE Transactions on Fuzzy Systems, Vol. 4, No. 2, (1996). 126
[25] Zadeh L. A.: From Computing With Numbers to Computing with Words - From Manipulation of Measurement to Manipulation of perceptions. IEEE Transactions on Circuits and Systems, Vol. 45, No. 1, (1999). 126

A Hybrid Framework Using SOM and Fuzzy Theory for Textual Classification in Data Mining

Yi-Ping Phoebe Chen

Centre for Information Technology Innovation
Faculty of Information Technology
Queensland University of Technology
Brisbane QLD, Australia
E-mail: p.chen@qut.edu.au

Abstract. This paper presents a hybrid framework combining self-organising map (SOM) and fuzzy theory for textual classification. Clustering using self-organizing maps is applied to produce multiple targets. In this paper, we propose that an amalgamation of SOM and association rule theory may hold the key to a more generic solution, less reliant on initial supervision and redundant user interaction. The results of clustering stem words from text documents could be utilised to derive association rules which designate the applicability of documents to the user. A four stage process is consequently detailed, demonstrating a generic example of how a graphical derivation of associations may be derived from a repository of text documents, or even a set of synopses of many such repositories. This research demonstrates the feasibility of applying such processes for data mining and knowledge discovery.

1 Introduction

Organized collections of data support a new dimension in retrieval, the possibility to locate part of related or similar information that the user was not obviously in search of. The information age has witnessed a flood of massive digital libraries and knowledge bases which have evolved dynamically and continue to do so. Therefore, there is a well documented necessity for intelligent, concise mechanisms that can organise, search and classify such overwhelming well-springs of information. The more fundamental, traditional methods for collating and searching collections of text documents lack the ability to manage such copious resources of data. Data mining is the systematic derivation of trends and relationships in such large quantities of data [1]. Previous work also discusses the use of such data mining principles instead of the more "historic" information retrieval mechanisms [2].

There has been much recent research in the data mining of sizeable text data stores for knowledge discovery. The self-organising map (SOM) is a data mining innovation which is purported to have made significant steps in enhancing knowledge discovery

J. Lawry, J. Shanahan, A. Ralescu (Eds.): Modelling with Words, LNAI 2873, pp. 153-167, 2003.
© Springer-Verlag Berlin Heidelberg 2003

in very large textual data stores [3, 4]. The latter describes a SOM as a normally unsupervised neural-network process that produces a graph of similarity of the data. A finite set of models are derived which approximate the open set of input data. Each model is a neural node in the map and a recursive process clusters similar models on this map (thus the term self-organising).

The fundamental benefit of the SOM is that it derives a lower dimensionality mapping in situations of high dimensionality [5]. The clusters are arranged in this low dimensional topology to preserve the neighbourhood relations in the high dimensional data [5, 6]. This relationship consequently holds true for clusters themselves as well as nodes within a single cluster. Whilst self-organising maps have proven a demonstrated usefulness in text classification, the primary deficiency is the time required to initially train them [6, 7]. Even though the preliminary training can be unsupervised (hence, not requiring manual intervention or seeding by an "expert" user), the time required to perform the operation limits the applicability of SOM as a potential solution in real-time searches.

The extent of this paper is to analyse several approaches at hybridising the SOM methodology in an attempt to circumvent the above-mentioned deficiency. These techniques come from individual research carried out by external researchers and documented in appropriate journals. The papers chosen all propose a form of hybridisation of self-organising maps which extends the fundamental SOM concept to include concepts external to the base SOM theory. This tenet is closely related to the author's own preliminary work and thus of interest.

Previous research work highlights the importance of pre-processing the text documents for simplification [8-12, 30]. In all work examined, an implicit assumption is made that *all* text documents are in the same language (i.e. English). Whilst the author will maintain this inherent assumption, it may hold bearing on such pre-processing, especially with respect to document sources with documents in different languages, or indeed, multi-lingual records.

The ordinary denominator in textual pre-processing follows the two steps of: removing stop-words and stemming keywords. Stop-words are words which hold little to no intrinsic meaning within themselves and purely serve to connect language in semantic and syntactic structure (e.g. words such as conjunctions and "of", "the", "is", etc). Secondly, stemming is the process of deriving the root word from words through removal of superfluous prefixes and suffixes which would only serve to produce redundant items in a derived index of terms. A artificial example could be the stemming of "mis-feelings" to "feel" by removal of a prefix, suffix and pluralisation. The underlying process of mapping text documents relies upon indexation of keywords and utilisation of such pre-processing serves to maximise both precision and recall in this process [8].

In this paper we focus on textual classification, the knowledge discovery and data mining using the self organization maps and fuzzy theory. It proceeds through the following sections: an outline of preceding, generally background study; a more focussed investigation of several SOM hybrid approaches, including comparison against our defined set of criteria; the preliminary description of our approach to self-organising map hybridisation, once again, contrasted on the same set of criteria; the

results from SOM can be improved if combined with other techniques such as fuzzy logic and, a conclusion covering future research yet to be achieved.

2 Related Work, Evaluation and Outline Criteria

SOM is a method that can detect unexpected structures or patterns by learning without supervision. Whilst there has been abundant literature on SOM theory, as mentioned previously, we have focused on the following as preliminary reading. SOM's are often related to document collation [13]. It is in this document that it seems that SOM was first widely purported to be an exceptionally viable method in classifying large collections of text documents and organizations of such into maps. Although this document was more preliminary and focused on a more interactive map for visualisation, other research provided a corollary with more formal evidence of SOM usage on very large document stores [4]. The example provided scaled up the SOM algorithm to derive a map with more than a million nodes from input categorised into 600 dimensions. Over 6.9 million documents were subsequently mapped. Parallelism and several optimising short-cuts were utilised to expedite the initial training. Such steps may not be feasible in a more generic solution where the documents may be more loosely formatted and not as uniform in layout and/or contents (the research [4] examined patent applications).

Other work, published at a similar time, proposed utilisation of clustering of the SOM output rather through the SOM itself and directly relates to the papers analysed more comprehensively, below [14, 15]. The latter journal article provides discourse on hierarchical clustering techniques through SOM technology and relates to the topic through that perspective. Such a hierarchical approach allows for a faster initial solution and then more detailed mappings to drill down and fine tune the data mining.

A hierarchical perspective is also examined where the output of a first SOM is used as the input for a second one [16]. This derivation of associations which are fed into the supplementary SOM is the key interest of the authors. Furthermore, two of the documents analysed more thoroughly in this paper implement a more sophisticated variant of this methodology [7, 17]. Another proposes a similar concept with an approach designated as multiple self-organising maps [18]. The base tenet of the mechanics of applying distance measures to string inputs for a SOM is examined [19]. Although more related to symbol parsing, rather than true text document mapping, it warrants a higher level examination for the overall “philosophy”. On the other hand, previous work specifically focussed on knowledge acquisition from internet date repositories [20].

A contrast of self-organising map and auto-associative feed forward network methodologies, amongst others, provides informative background on different techniques for dimensionality reduction [21]. Finally, whilst an interpolating self-organising map which can enhance the resolution of its structure without the requirement of reintroducing the original data may hold implications for hierarchical SOM usage [22], a relation between Bayesian learning and SOM fundamentals [23] is the focus of more intrinsic analysis in this document.

2.1 Evaluation of Existing Hybridised Self-organising Map Approaches

In here we have presented recent techniques have warranted a more detailed analysis and comparison: utilisation of SOM approximation of probability densities of each class to "exploit" Bayesian theory to minimise the average rate of misclassifications [17]; amalgamation of an interactive associative search instigated by the user with SOM theory [6]; and, development of a hierarchical pyramid of SOM's in which a higher level tier allows subsequent drilling down into and derivation of a more fine-tuned map [7]. All of these methods share a commonality with respect of utilisation of a hybridised SOM in text-based data mining and information retrieval.

2.2 Applicability in Connection with Criteria

In our investigation we will include the following sub-topics: utilization of hierarchy, intensity of hybridisation, degree of generality, requirement for supervision and/or interaction and a brief synopsis of the deficiencies of each methodology is presented.

2.2.1 Utilization of Hierarchy

Earlier research found that document collections inherently lend themselves to a hierarchical structure [7]. This hierarchy originated directly from the subject matter. Their research resulted in a mechanism for producing a hierarchical feature map – each element in a SOM expands into a SOM of its own until sufficient detail has been explored. This approach allows for user interaction so that the desired level of detail can by chosen; thereby preventing unnecessary mining sophistication and delay but still maintaining the desirable degree of precision.

Similar in technique, the proposal to integrate Bayesian classification maintains the fundamental principal of training an independent SOM for each class derived from the training set. The difference is that once the basis for the collection of SOM's is formed, they are merely kept in an array-like structure and there is no utilisation of a hierarchy other than the two tiers of the classes and the reliant self-organising maps.

Certainly, even though the interactive associative search method is relatively simplistic in its explanation of SOM usage, it shows a more advanced use of hierarchical principles. The proposed extension would subsequently achieve similar documents derived as near neighbours on a SOM. This involves a comparatively simple hierarchy between a contemporary solution and SOM technology. A preliminary search through contemporary mechanisms would produce a result set.

2.2.2 Intensity of Hybridisation

Cervera's implementation utilises neuron proximity in the self-organising map as approximations of the class probability densities. Bayesian classification theory is subsequently applied to minimise error in the classification process and to optimise the resultant clusters [22]. Both the formal hierarchical SOM pyramid and integration with Bayesian theory method implement a potential solution via multiple SOM's.

The methodology initiated by Merkl ordinarily entails a very high-dimensional feature set correlated to the number of index terms required. This feature set declares the vector representation of a text document for mapping (and hence the derived dis-

tance between two or more of them which corresponds to their similarity). It is stated that to overcome this "vocabulary problem", auto-associative feed forward neural network technology is subsequently used to optimise compression of this feature space. This reduces patterns of frequently co-occurring words to a single word [21].

On the contrary, other work relied upon an interactive associative search instigated by the user [6]. The underlying dimensionality reduction is dependent on word category map construction, just like *WEBSOM* [13]. On the other hand, some research implemented fundamental information retrieval mechanisms for stemming superfluous word segments and filtering "noise" stop-words [6, 7]. Of course, the former document purports that even current, contemporary search engines should readily adopt this principle. In either implementation scenario, the resultant hierarchy of results is presented to the user through the graphical nature of SOM output. These two papers also share a commonality with respect to exploiting the ready visualisation capabilities of SOM technology [5, 20, 24, 25].

2.2.3 Degree of Generality

Similarly, even though the hierarchical SOM pyramid methodology was originally designed specifically for text document classification, the base tenets of constructing a pyramid of hierarchically tiered self-organising maps would be equally applicable in other SOM scenarios. Even as Cervera, et al, really documented experiments on sonar signals and audio recognition of vowel sounds from several speakers, the principles maintained in the related research would remarkably hold true for all features of SOM unsupervised learning [17].

The interactive associative search hybrid, nevertheless, is heavily reliant on user interaction and the design principle weighs heavily on this interactivity and visual exploration. While the SOM's "natural" applicability to visualisation does hold true for other applications, the hybridisation aspects of this research is dependent on document map queries. There are probably only minor facets which could be utilised if symbol strings, such as through [19], were derived. Furthermore, the actual SOM manufacture is a static process in this implementation where clustering is pre-processed for a finite and discrete suite of documents earlier. This is the opposite of the design of the other two proposals where the infrastructure would conceptually handle all generic text document clustering situations. Therefore, there is a possibility that the on the whole architecture is not generic for all cases of text documents.

2.2.4 Requirement for Supervision and/or Interaction

Even though the unsupervised associative search is definitely instigated by a user, it has the least in that SOM production is completely unsupervised and an initial pre-processing stage is performed before the results are viewed. Every method holds a differing degree of supervisory dependence. The user intervention is the corollary to this, at a later time, and it is only then when the user performs interactive associative searching that results are tailored.

The hierarchical pyramid of SOM's is dependent on user interaction to decide which higher level tier requires drilling down into and consequent derivation of a more fine-tuned map. However, conceptually, the entire process could be also all pre-processed for n tier levels. Such a scenario would be infeasible due to constraints on

user expectation for timely results (or suffer the same deficiency as the unsupervised associative search with respect to only being applicable to a static set of documents from a source). The hybridisation with Bayesian theory was inherently designed with a supervised component. Production of probability density approximations through Bayesian classification is reliant on supervised training. By itself, although the interaction with users after mapping is not relevant, the pre-requisite for an expert to initially train the system is.

2.3 Overall Deficiencies

The fundamental inadequacy in the Bayesian classification hybrid is the reliance on a supervised component for expediting the classification. In the proposed scenario for dynamic implementation on text-based documents provided on an a priori basis, there is little scope for initial pre-training with a specialised training set provided by an expert. Another approach must be fashioned to provide more accurate SOM mappings.

The base methodology of an interactive associative search is relatively simplistic and reliant on pre-processing the entire suite of documents to be examined before initiating an interactive associative search with the end user. Once again, this would be inadequate in the scenario in question of a priori-based documents from (possibly) unknown data sources and repositories (such as an internet search, e.g. [20]).

A pyramid of hierachical SOM's is perceived to be the closest to the target of such ad hoc, dynamic interactive searches with the hierarchical organization of SOM's. The example provided in the experiment, nonetheless, is not rich enough to provide comprehensive feedback on timeliness in a “real-world” scenario. The resultant feature space on a much higher dimensionality caused by a large set of index terms would be time prohibitive.

3 A New Framework in SOM Hybridisation

The inherent barrier to efficient knowledge acquisition in such circumstances is the underlying feature space. This resultant very high dimensionality is directly originates from the requirement of a high set of index terms. Even with advanced information retrieval stemming and stop-word filtering, this will cause both loss of precision accuracy and effect recall. Without dimensionality reduction on this feature space, results will furthermore not be available in a timely manner.

There several features of self-organising map theory which are adaptable to hybridisation and cross-pollination of new ideas. In terms of classification of text-based documents (notably a priori and from unknown originating sources) such hybridisation must expedite the mapping process yet remain unsupervised. It is also most likely that the user will wish to alter the output resolution.

A hierarchically-based solution is desirable but a simple approach of merely forming a pyramid of SOM's would not facilitate optimisation of the feature space. Preprocessing the document repository/s is, of course, not a viable option. Further re-

search is required to form another hybrid proposal, most likely cross-pollinating further ideas from other data mining theory.

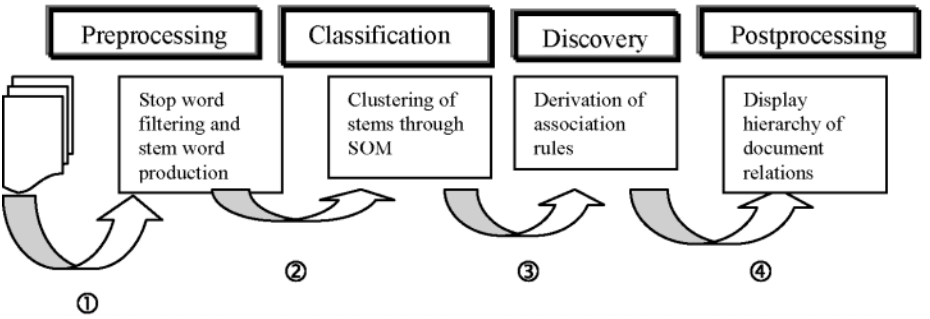


Fig. 1. Procedure Phases in the Amalgamation of SOM and Association Rule Theory

3.1 Hybridisation of SOM Theory with Association Rules

A generic framework may be proposed in which the concepts of episodes and episode rules are developed [2, 34, 35]]. These are derived from the basic concept of association rules and frequent sets, when applied to sequential data. Text, as sequential data, can be perceived as a sequence of *(feature vector, index)* pairs where the former is an ordered set of features (such as words, stems, grammatical features, punctuation, etc) and the latter information about the position of the word in the sequence. Consequently, episode rule may be derived with respect to these episode pairs, as per association rule theory. Episode rule can be described by association rules and co-occurrence patterns.

In our approach, simplifying the paradigm further, the clusters derived from a SOM could be equated to frequent itemsets. As such, these may, in turn, be fed to an association rule based engine which would consequently develop association rules. In order to simplify the initial classification process, redundant stop-words would be removed and only the word stems of the text documents considered. This helps alleviate the verbosity of natural language and allows rule derivation to take place on the simplest, core level. The previous work on episodes and their derived rules was too intrinsically bound to grammatical context and punctuation; we wish to develop a more holistic perspective about the overall document context. This will, in turn, derive more fundamental associations, rather than similar rules bound by the original grammatical context. Figure 1 displays a framework overview on how the hybridisation will supplement knowledge discovery in text mining. The representation of initial text documents could equally be applicable to either a text datastore or a repository of meta-data on text datastores themselves. It is the relation of SOM classification results to association rule derivation which is the fundamental focus, not the underlying source of SOM input.

The consequential variant is proposed to expedite the text mining procedure on large manuscript collections [4, 13, 20] .The inherent extent for research relates to the application of two data mining methodologies into a hybridised amalgamation.

3.2 Case of Processing

A brief generic example will be supplied to enhanced illustrate the processing which occurs in each step.

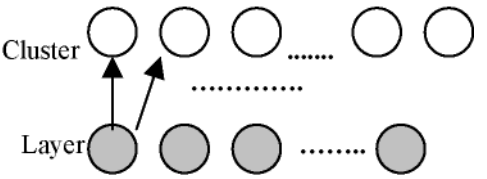


Fig. 2. SOM

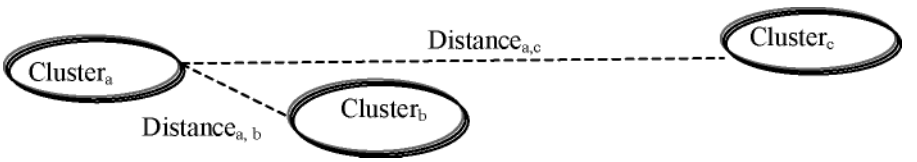


Fig. 3. Distances Between Clusters a, b and a, c

3.2.1 Preprocessing Stage

In the former instance, the input to the preprocessing stage ① in Figure1 is a set of "records" where each record is a document comprised of previously unparsed text. In the latter scenario, each record is a synopsis of a text document datastore with meta-data, such as repository key terms and/or abstracts, etc. The initial set of documents may equally represent either a single text document repository or a result set detailing the meta-data derived from a collection of such repositories.

The real preprocessing phase reduces the textual component (either in terms of document contents or meta-data information) into a set of common stem words ② (Figure 1). This procedure relies upon relatively standardised information retrieval processes for detection and removal of redundant stop words, as well as the discovery of pertinent stem words (e.g. [6, 7]).

3.2.2 Classification *in SOM*

The second phase recognizes the stem word sets ② Figure 1 and utilises them as input to a standard self-organising map of relatively small size. It is envisaged that the common two-dimensional SOM will be implemented as there is deemed to be no rationale or circumstances which warrant a more complex model to map to. The range of each dimension will be low to expedite clustering as it is by far more likely that the order of magnitude of input sets would be in the order of hundreds rather than thousands.

In a comparable layer to the initial phase (Figure 2), the fundamental mechanics of processing the inputs relies upon existing SOM technology, which is well documented in a huge number of sources. The only specialisation made is the extraction of the

resultant SOM output array en masse as an input to the next phase ③ of Figure 1. The operation of the initial preprocessing and classification steps may occur in parallel as the outputs of the former are merely "fed" to the latter asynchronously. All stem word sets must be processed, however, before the transfer of the refined SOM contents to the next, discovery, and phase.

3.2.3 Discovery in SOM

In this processing step, there is a realistic expectation that the contents of each cluster in the input, ③ of Figure 1, may be equated to frequent itemsets. Taking a Euclidean perspective as the basis of supposition, for the sake of simplicity, the distance measures taken form neighbourhood functions are perceived to provide a firm foundation for derivation of association rules. Therefore, there is a quantifiably higher degree of associativity between $Cluster_a$ and $Ckuster_b$ than $Cluster_a$ and $Cluster_c$. In figure 3, below, we can see that $Distance_{a,b}$ is much smaller than $Distance_{a,c}$ ($Distance_{a,c}$ >= $Distance_{a,b}$).

Although more formal association rule techniques, such as *Apriori* [26-29], will most likely need to be implemented to derive confidence and support levels, it is envisaged that the process may be expedited by examining the distance measures for initial item set suitability. The derived association rules may be represented as two-dimensional vectors designating the key stem words which imply other, related stems ④. Without a doubt, once a physical prototype of the process is developed, it may demonstrate an empirically derived correlation between distance and confidence.

3.2.4 Postprocessing Phase

The associations derived in the discovery phase ④ may be indicated to the user as a mathematical graph (i.e. linked nodes). This provides feedback on which "records" (either text documents or entire repositories, depending on context) relate to which others and through what key (stem) words. Of course, links may be incremental in such that $Node_a$ is indirectly related to $Node_d$ through $Node_c$ (see figure 4).

In the end context of associations for whole text document datastores, it is perceived to be feasible to subsequently "drill down" into a desired repository and perform the same associative process on it, discovering more detailed rule derivations through classification of its more thorough set of stem keywords.

3.3 Applicability in Connection with Criteria

With the aim of contrast our approach with those discussed previously, it is ideal to analyse it with respect to the criteria we detailed above.

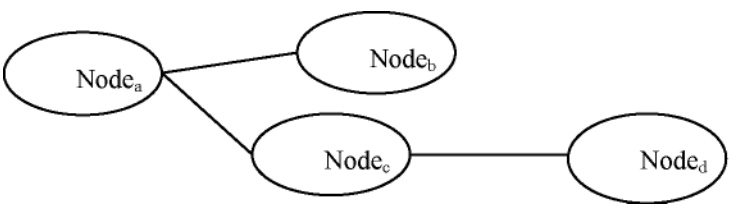


Fig. 4. Associations Represented as Graph Nodes

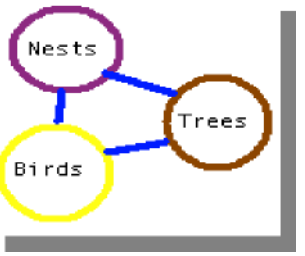


Fig. 5. Nests, Birds and Trees Associations

3.3.1 Utilization of Hierarchy

In general, hierarchy is simply a two tiered approach where the SOM clustering is a foundation for association rule data mining. On the other hand, there is an intrinsic hierarchy to the rules themselves in that the derived associations can lead to subsequent ones. For instance (Figure 5), an association that there is a high degree of support that documents with reference to nests also reference birds can point to a corollary association in which it was discovered that documents referring to birds also referred to trees.

3.3.2 Intensity of Hybridisation

The amalgamation of association rule data mining theory with that of SOM classification involves a relatively high level of hybridisation.

3.3.3 Degree of Generality

Although specifically designed for text mining applications, the initial source of stemmed source terms may be from a repository of text documents or a datastore of meta-terms relating to multiple text document sources. Indeed, there is no perceivable limitation on the holistic approach to even non-text based originating sources, as long as the rule derivation maintains a base level of "common sense" to the original data (i.e. the information derived from association rules derived from audio data may or may not be useful, depending on the context of investigation).

3.3.4 Requirement for Supervision and/or Interaction

There is no perceived requirement for supervision in this model; also, there is deemed to be little necessity of pre-requisite user interaction, except in terms of defining the initial document source. At the completion of the postprocessing stage, it is feasible that user interaction may occur with the model's results. If the initial original data source was a pool of meta-data on lower level document stores, the user may wish to repeat the process on a lower level of abstraction on the actual text document repositories deemed applicable.

The inherent methodology utilised in the SOM hybrid is unsupervised, whilst associative user interaction may be introduced to exploit the underlying architecture, the base model is foreseen to be relatively independent of supervision requirements.

4 Fuzzy Theory within SOM

The results from SOM can be improved if combined with other techniques such as fuzzy logic. This fuzzy theory with SOM provides a perceptual indexing and retrieval mechanism; hence the users can query these using higher level descriptions in a natural way. The fuzzy theory we have applied here is a possibility-based approach. We use the possibility and necessity degree which is proposed by Prade and Testemale [31] to perform shape retrieval. The query form *attribute = value* in conventional database can be extended to *attribute = value with degree* θ, where θ can be possibility or necessity degree.

Given two possibility distributions on the same domain, they can be compared according to the possibility and necessity measures. This comparison leads to two types of degrees: the possibility degree and the necessity degree which mean to what extent two fuzzy sets are possibly and necessarily close. The possibility degree Π represents the extent of the intersection between the pattern set and a datum set. It is the maximum membership value of the intersection set. The necessity degree N represents the extent of semantic entailment of a pattern set to a given datum set. It is the minimum membership value of the union set of the pattern set and the complement of the datum set. The interval defined by $[N, \Pi]$ represents the lower and upper bounds of the degree of matching between such pattern and datum sets. Since what is necessary must be possible, the possibility degree is always not less than the necessity degree. The proof of this property and the detailed formulas for calculating the possibility and necessity degrees can be found in [31, 32]. Fig. 6 shows the fuzzy condition, fuzzy data and the corresponding possibility and necessity degrees. Possibility and necessity are two related measures.

The possibility-based framework uses possibility distribution to represent imprecise information including linguistic terms. This distribution acts as a soft constraint on the values that may be assigned to an attribute. In this framework, the relation is an ordinary relation yet available imprecise information about the value of an attribute for a tuple is represented by a possibility distribution. Hence, this representation is more flexible and expressive than the similarity-based framework and the fuzzy-relation-based framework. As the possibility-based fuzzy model associates imprecise information directly to data items, it satisfies the need for storing whose parameters are represented as possibility distributions. Both possibility measure and necessity measure belong to a more general class of fuzzy measures (Figure 7).

Figure 8 and 9 shows the relationships among information. The possibility and necessity degrees are employed to perform data retrieving.

A possibility-based fuzzy theory has been constructed within this framework. A fuzzy term is represented by a set of descriptors and parameters. It is indexed and retrieved by fuzzy descriptors. Since the fuzzy set approach groups crisp value into partitions according to their similarity, the fuzzy set representation indicates the merging of opinions of different terms. For example, when we try to insert a term with a parameter *p1=1.0*, if there is another word with parameter *p1=1.001* and we know that two words with these two different parameters have almost the same appearance; we do not need to save the new word into the database. The same principle can be extended

to fuzzy meaning with fuzzy set values. If two fuzzy meanings are nearly the same, we store only one of them in the database.

5 Conclusions and Future Work

This paper has presented a hybrid framework combining self-organising map (SOM) and fuzzy theory for textual classification. Existing hybrids of SOM theory are not perceived to effectively address the requirements in data mining text documents: a high degree of generality for generic text mining without the requisite initial supervised training. Furthermore, it is extremely desirable that multiple levels of abstraction be catered for; in which more generic original sources with less granularity may be drilled down to obtain more focused dependent repositories. An optimum solution would efficiently flag applicable documents without a necessity for a phase of preparation through training or high level of user interaction. For example, a datastore with meta-information on many document sources may be mined for "leads" to the more applicable sources and then these subsequently mined in a more detailed manner.

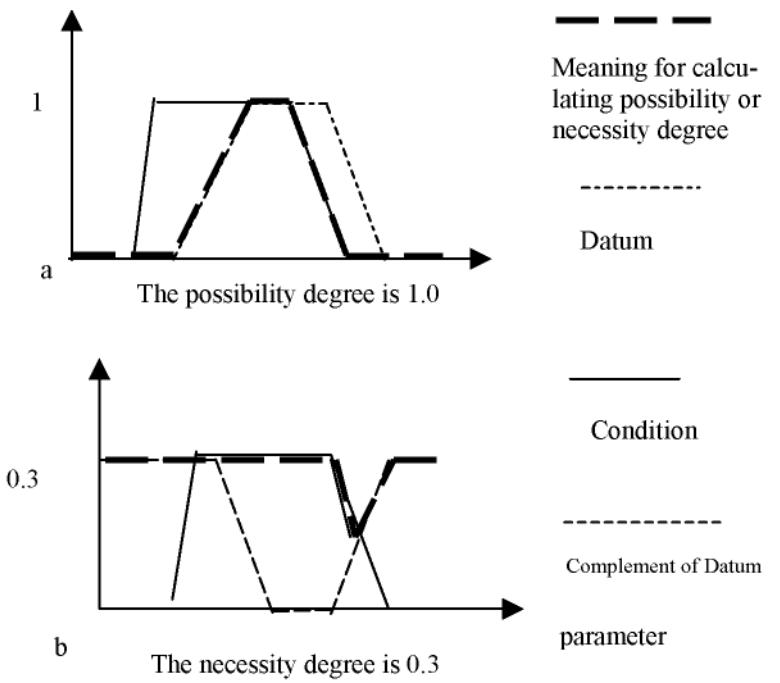


Fig. 6. The possibility and necessity degree

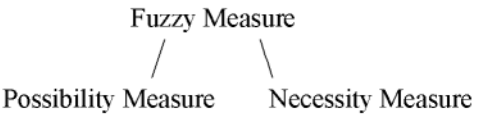


Fig. 7. SOM with Possibility-Based approach (1)

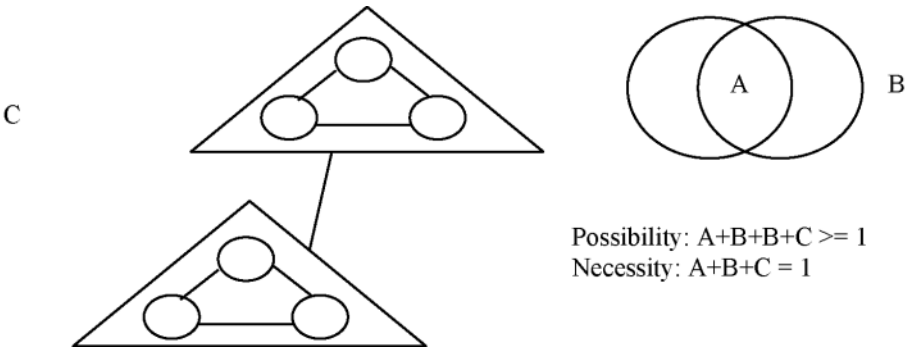


Fig. 8. SOM with Possibility-Based approach (1)

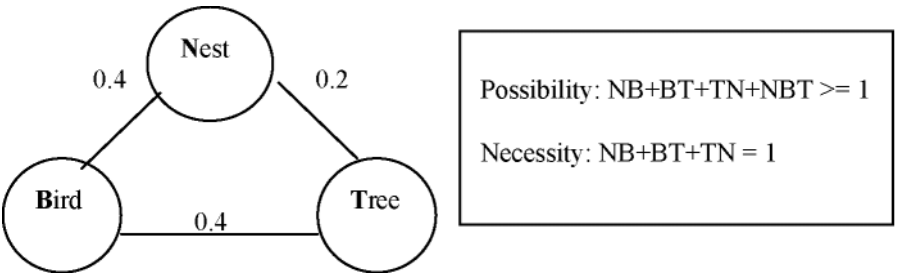


Fig. 9. SOM with Possibility-Based approach (2)

The proposed amalgamation of SOM, association rule theories and fuzzy theories is identified as a feasible model for such text mining applications. It is envisaged that the viability of this technique may be proven with respect to efficiency and precision of results. Further scope also exists in implementing corollary SOM, association rule theory and fuzzy theory to expedite the output delivery, but the underlying, fundamental methodology is viewed to be a sound platform for relatively expeditious and efficient text mining. From our investigation of several SOM hybrid approaches, including comparison against our defined set of criteria; the preliminary description of our approach to self-organising map hybridisation, once again, contrasted on the same set of criteria; the results from SOM can be improved if combined with other techniques such as fuzzy logic and, each node can be extended into a multimedia-based document.

Acknowledgments

The author thanks Paul Gunther for his input about this work.

References

[1] Pendharkar, P.C., et al., Association, statistical, mathematical and neural approaches for mining breast cancer patterns. Expert Systems with Applications, 1999. **17**(3): p. 223-232.
[2] Ahonen, H., et al., *Applying Data Mining Techniques in Text Analysis*. 1997, University of Helsinki: Helsinki. p. 1-12.
[3] Kaski, S., *The Self-organizing Map (SOM)*. 1999, Helsinki University of Technology: Helsinki. p. 1.
[4] Kohonen, T., et al., *Self Organization of a Massive Document Collection.* IEEE Transactions on Neural Networks, 2000. **11**(3): p. 574-585.
[5] Vesanto, J., *SOM-based data visualization methods.* Intelligent Data Analysis, 1999. **3**(2): p. 111-126.
[6] Klose, A., et al., *Interactive Text Retrieval Based on Document Similarities.* Phys. Chem. Earch (A), 2000. **25**(8): p. 649-654.
[7] Merkl, D., Text classification with self-organizing maps: Some lessons learned. Neurocomputing, 1998. **21**(1-3): p. 61-77.
[8] Savoy, J., *Statistical Inference in Retrieval Effectiveness Evaluation.* Information Processing and Management, 1997. **33**(4): p. 495-512.
[9] Riloff, E. and W. Lehnert, *Information extraction as a basis for high-precision text classification.* ACM Transactions on Information Systems, 1994. **12**(3): p. 296-333.
[10] Chang, C.-H. and C.-C. Hsu, *Enabling Concept-Based Relevance Feedback for Information Retrieval on the WWW.* IEEE Transactions on Knowledge and Data Engineering, 1999. **11**(4): p. 595-608.
[11] O'Donnell, R. and A. Smeaton. A Linguistic Approach to Information Retrieval. in 16th Research Colloquium of the British Computer Society Information Retrieval Specialist Group. 1996. London: Taylor Graham Publishing.
[12] Srinivasan, P., et al., *Vocabulary mining for information retrieval: rough sets and fuzzy sets.* Information Processing and Management, 2001. **37**(1): p. 15-38.
[13] Kaski, S., et al., *WEBSOM - Self-organizing maps of document collections.* Neurocomputing, 1998. **21**(1-3): p. 101-117.
[14] Vesanto, J. and E. Alhoniemi, *Clustering of the Self-Organizing Map.* IEEE Transactions on Neural Networks, 2000. **11**(3): p. 586-600.
[15] Alahakoon, D., S.K. Halgamuge, and B. Srinivasan, *Dynamic Self Organizing Maps with Controlled Growth for Knowledge Discovery.* IEEE Transactions on Neural Networks, 2000. **11**(3): p. 601-614.
[16] De Ketelaere, B., et al., *A hierarchical Self-Organizing Map for classification problems*. 1997, K.U. Leuven: Belgium. p. 1-5.

[17] Cervera, E. and A.P. del Pobil, *Multiple self-organizing maps: A hybrid learning scheme.* Neurocomputing, 1997. **16**(4): p. 309-318.
[18] Wan, W. and D. Fraser, *Multisource Data Fusion with Multiple Self-Organizing Maps.* IEEE Transactions on Geoscience and Remote Sensing, 1999. **37**(3): p. 1344-1349.
[19] Kohonen, T. and P. Somervuo, *Self-organizing maps of symbol strings.* Neurocomputing, 1998. **21**(1-3): p. 19-30.
[20] Chen, H., et al., *Internet Browsing and Searching: User Evaluations of Category Map and Concept Space Techniques.* Journal of the American Society for Information Science, 1998. **49**(7): p. 582-603.
[21] De Backer, S., A. Naud, and P. Scheunders, *Non-linear dimensionality reduction techniques for unsupervised feature extraction.* Pattern Recognition Letters, 1998. **19**(8): p. 711-720.
[22] Yin, H. and N.M. Allinson, *Interpolating self-organising map (iSOM).* Electronics Letters, 1999. **35**(19): p. 1649-1650.
[23] Hämäläinen, T., et al., Mapping of SOM and LVQ algorithms on a tree shape parallel computer system. Parallel Computing, 1997. **23**(3): p. 271-289.
[24] Walter, J. and H. Ritter, *Rapid learning with parametrized self-organizing maps.* Neurocomputing, 1996. **12**(2-3): p. 131-153.
[25] Kangas, J. and T. Kohonen, *Developments and applications of the self-organizing map and related algorithms.* Mathematics and Computers in Simulation, 1996. **41**(1-2): p. 3-12.
[26] Joshi, K.P., *Analysis of Data Mining Algorithms*. 1997, http://www.gl.umbc.edu/~kjoshi1/data-mine/proj_rpt.htm. p. 1-19.
[27] Zaki, M.J., *Scalable Algorithms for Association Mining.* IEEE Transactions on Knowledge and Data Engineering, 2000. **12**(3): p. 372-390.
[28] Boley, D., et al., *Partioning-based clustering for Web document categorization.* Decision Support Systems, 1999. **27**(3): p. 329-341.
[29] Pudi, V. and J.R. Haritsa, *Quantifying the Utility of the Past in Mining Large Databases.* Information Systems, 2000. **25**(5): p. 323-343.
[30] Gunther, P. and Chen P., *A Framework to Hybrid SOM Performance for Textual Classification.* Proceedings of the 10th International IEEE conference on Fuzzy Systems, 2001, IEEE CS Press. P.968-971.
[31] Prade H. and Testemale C., 1984 "Generalizing Database Relational Algebra for the Treatment of Incomplete/Uncertain Information and Vague Queries," *Information Sciences*, vol. 34, pp. 115-143, 1984.
[32] Bosc P. and Galibourg M., 1989 "Indexing Principles for a Fuzzy Data Base," Information Systems, vol. 14, pp. 493-499, 1989.
[33] Peter Pirolli, Patricia Schank, Marti A. Hearst, and Christine Diehl, Scatter/Gather Browsing Communicates the Topic Structure of a Very large Text Collection, Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI), May 1996.
[34] Mario Drobics, Ulrich Bodenhofer, Werner Winiwarter, Interpretation of Self-Organizing Maps with Fuzzy Rules (2000), In Proceedings of ICTAI'00, pages 304-311, Vancouver, 2000.

Combining Collaborative and Content-Based Filtering Using Conceptual Graphs

Patrick Paulson[1] and Aimilia Tzanavari[2]

[1] MIAMI UNIVERSITY
Department of Computer Science & Systems Analysis
Oxford, OH-45056, USA
paulsop@muohio.edu
[2] UNIVERSITY OF CYPRUS
Department of Computer Science, P.O. Box 20537
Nicosia, CY-1678, Cyprus
aimilia@ucy.ac.cy

Abstract. Collaborative Filtering and Content-Based Filtering are techniques used in the design of Recommender Systems that support personalization. Information that is available about the user, along with information about the collection of users on the system, can be processed in a number of ways in order to extract useful recommendations. There have been several algorithms developed, some of which we briefly introduce, which attempt to improve performance by maximizing the accuracy of their predictions. We describe a novel algorithm in which user models are represented as Conceptual Graphs and report on results obtained using the EachMovie dataset. We compare the algorithms based on the average error of prediction and standard deviation and discuss our method's strengths and advantages.

1 Introduction

The necessity of system-originated recommendation becomes imperative as human computer interaction becomes more complicated. Software packages are becoming more complex, the number of provided services is increasing, the range of selection is widening, and the user, in general is confronted with numerous dilemmas. *Recommender Systems* [1] can guide the user through these processes by recommending paths, solutions, alternatives and new ideas. Recommendation in most cases is regarded as an extension of the prediction process that frequently takes place in *User Modeling Systems* [2, 3] – instead of predicting a single item of a user's profile based on other information, a whole set of items is predicted in a similar fashion. The predicted items will thus play the role of items to be recommended to the user.

In the remainder of the article we will explore two key approaches to designing recommender systems: *Content-Based* and *Collaborative Filtering*. The first

J. Lawry, J. Shanahan, A. Ralescu (Eds.): Modelling with Words, LNAI 2873, pp. 168-185, 2003.
© Springer-Verlag Berlin Heidelberg 2003

category's philosophy can be summarized in that the user is mainly characterized by the content of the items he shows interest in and any recommendations should be guided by these pre-known interests. On the contrary, in Collaborative Filtering the user is seen as part of a group - in relation to the other users on the system. This view leads to recommendations based on other, similar users' interests. We will subsequently describe a number of *Hybrid Systems* that have combined these two methods, each taking a relatively different route. Finally, we will introduce a novel hybrid system that combines them through the use of *Conceptual Graphs* for knowledge representation. The strengths and advantages of this approach will be demonstrated through the presentation of a set of experimental results, experiments in which we will compare it to other competing algorithms.

2 Recommender Systems

2.1 Definition

A common situation for computer system users, whether they are browsing the Web to find some information, trying to purchase something on-line, or just trying to gain experience in the use of a software package, is finding themselves wondering about the right decision to make. A user might look for the correct keywords for queries in order to find particular items. A computer system may be requested for the book, for example, that is most appropriate given a user's personal needs and abilities. The need for suggestions exists also in human relationships, where a mentor with more expertise or knowledge in a particular area enlightens a person that lacks that knowledge. The ultimate goal for both the computerized recommender system and the mentor is to provide help tailored to a particular user or person.

While some recommender systems provide the same recommendations to all users, this article focuses on systems that rely on user modeling in order to provide *personalization*. These recommender systems are designed and built to address the issues described above: providing the options, alternatives, and choices that are most suitable given the selections a particular user has made. In fact, recommender systems can be regarded as information agents who serve their users, trying to provide them with the information that best serves their needs. Personalization has been identified as a primary objective of many online systems [3].

2.2 Existing Techniques

There are a number of techniques used for the design of a recommender system. We can categorize these techniques as Content-Based, Social or Collaborative, and Hybrid.

Content-Based Techniques

In content-based techniques, the user model includes information about the content of items of interest-- whether these are web pages, movies, music, or anything else. Using these items as a basis, the technique identifies similar items that are returned as recommendations. These techniques might prove highly suitable for users who have specific interests and who are looking for related recommendations. Many machine

learning techniques have been applied to this problem. Some researchers working on these, such as Chen and Norcio [4] and Jennings and Higuchi [5] have modeled users with the application of neural network methodology. NewsWeeder [6], NewsDude [7] also belong to the category of content-based recommender systems.

One of the limitations when using content-based techniques is that no new topics are explored; only those that are similar to topics already in the user's profile. This leads to over-specialization: one is restricted to seeing items similar to those that have already been rated highly. This has been addressed in some cases with the injection of randomness [8]. Content-based techniques, moreover, are difficult to apply to situations where the desirability of an item, for example a web page, is determined in part by multi-media content or aesthetic qualities. These types of materials are generally incompatible with the type of content analysis that these techniques require in order to make further recommendations [9].

Additionally, recommender systems of this kind frequently require feedback about the relevance of their suggestions. Users often find generating this feedback a tedious task and try to avoid it. The user model in such systems consists entirely of user ratings of items of interest. Recommendations are solely based on these, making them the main factor influencing performance: the fewer the ratings, the more limited the set of possible recommendations. Several systems that make use of machine learning techniques have been developed and presented in the literature (e.g. [10]). Feedback is required in machine learning techniques that need it for their "learning" process. These techniques often require lengthy computation to learn the user's preferences. Once computed, however, the user's preferences will not remain static. Therefore, this process will need to be repeated with a frequency that depends on how quickly the user model changes.

Collaborative Filtering Techniques

In social or collaborative filtering, the system constructs rating profiles of its users, locates other users with similar rating profiles and returns items that the similar users rated highly. As in content-based techniques, these systems depend on their users providing ratings or feedback. The scarcity of ratings and scalability are both issues in social filtering. User profiles in this case are usually sparse vectors of ratings. A partial solution to this might be to use implicit feedback [11], or methods to increase the density of the dataset. Scalability is a problem because computation grows linearly with the number of users and items.

Finding the "nearest neighbors" to the active user in order to retrieve recommendations is a task that requires the definition of the term "similarity" for a particular system. This is one of the main points where collaborative systems differ. Specifying which users are to be considered similar determines the performance of the system in terms of accuracy of recommendations. Keeping this in mind, a user that is considered unusual based on his profile (interests) will probably not be similar to any of the other users, which will lead to poor recommendations. Moreover, since no information about the content of items is kept, even users with similar (but not identical) interests will not be considered similar themselves.

The first collaborative filtering system was Tapestry [12] and since then there has been significant research in the field. Several algorithms have been used for collaborative filtering, and specifically for computing the aforementioned similarity

between two users. Examples include using Pearson's correlation coefficients between the users preference functions [13], vector similarity [14], probabilistic distance measure [15], nearest neighbors [7], Bayes' rule [16] and mean squared differences [17].

The advantage of social filtering, compared to content-based techniques, is that the pool from which recommendations originate is not restricted to items for which the active user has demonstrated interest. The pool will also include items that other users, users that are in some respect similar, have rated highly. This can prove to be instrumental in enhancing the user's model: social filtering systems give the user the opportunity to explore new topics and items.

Breese et al. divide the collaborative filtering algorithms into memory-based and model-based techniques. Memory-based collaborative filtering algorithms predict a user rating for a particular item by using a similarity-weighted sum of the other user ratings. The method used to calculate weights is a distinguishing characteristic of algorithms in this category. Model-based algorithms depend on a model, such as a Bayesian network, built to represent the user data. This model will subsequently be queried to get the recommendations. The construction of the model is a learning process that is often time consuming. Indicatively, the time required can run up to 8 hours, which is very significant, especially when the user models are dynamic. However, the advantage here is that after the model is determined, recommendations can be returned with great speed.

In collaborative filtering, recommendations are often based on the comparison between the models of the active user and the population of other users, where the user models are sets of votes. A common shortcoming of collaborative filtering algorithms recommendations will only come from the users with which the active user shares votes. For example Pearson's correlation coefficient [13] finds the similarity between user α and ι:

$$sim(\alpha,\iota) = \frac{\sum_j (v_{a,j} - \overline{v_a})(v_{\iota,j} - \overline{v_\iota})}{\sqrt{\sum_j (v_{a,j} - \overline{v_a})^2 \sum_j (v_{\iota,j} - \overline{v_\iota})^2}} \quad (1)$$

where v_{ij} is user ι vote on item j, j is the number of items for which both users have recorded votes and $\overline{v_\iota}$ is the mean vote for user ι. Consequently, only items that exist in both user models will contribute to their similarity. This can lead to problems when relatively few votes are available for either the active user or the users being used for comparisons. *Default Voting* has been used to overcome this problem. With this method, default votes are assigned to unobserved items, so that there is more overlap between the two users to be compared - or even to items that neither user has observed - in order to increase the total number of votes [14]. Appropriate default votes might be a user's average vote or the average vote, over all users, for a particular item.

Additional extensions to memory-based algorithms include Inverse User Frequency and Case Amplification [14]. Inverse User Frequency borrows its main principle from the Inverse Document Frequency measure used in Information Retrieval. With this measure, words that occur frequently across all documents are

considered less representative of the document, whereas words that rarely occur are more likely to be indicative of the document's topic [18]. In a collaborative filtering context, this principle has been applied to transform the similarity between two users; methods are used to minimize the weights for universally liked items compared to items that are liked by fewer users. Case Amplification is a transformation applied to the weights that strengthens higher weights and weakens lower ones.

Hybrid Techniques

Hybrid techniques seem to promise to combine the positive features of both content-based and social filtering methods, diminish their shortcomings, and thus produce a more robust system. The philosophy here is that the content of items is taken into consideration when identifying similar users for collaborative recommendation. Alspector, Kolcz and Karunanithi [19] provide evidence that collaborative filtering systems can be enhanced by adding content-based filters. Fab, developed at Stanford University and operational since 1994, uses these ideas [9]. It has both a personal and a "group" filter, which are altered based on the user's relevance feedback. Fab's users receive recommendations both from collection agents that represent a particular interest area and from a selection agent that operates exclusively on the particular user's interests.

The GroupLens project [20] that started in 1992 at the University of Minnesota, successfully dealt with the problem of applying collaborative filtering to Usenet News. This domain was greatly different from previous movie and music related domains, where new items were relatively infrequent and lifetimes were relatively long. The GroupLens team showed that collaborative filtering coupled with personalized information agents, leads to better recommendations [21]. Recently, the same team presented an approach termed "item-based collaborative filtering", which first analyzes the "user-item matrix" to identify relationships between different items, and then uses these to indirectly compute recommendations [22].

Claypool, Gokhale and Miranda [23] presented a system in which they do not "blend" content and collaborative filters, but instead leave them entirely separate. Their prediction is based on a weighted average of the content-based prediction and the collaborative one. The weights are determined on a per-user and per-item basis: if the number of users and votes increase then the corresponding prediction's weight (collaborative) will be increased as well.

Basu, Hirsh and Cohen [24] proposed an inductive learning approach to recommendation that is able to use both user votes and other forms of information about each item. They do not return a predicted vote as such; rather they return a classification {*liked*, *disliked*} for a particular item.

3 A Hybrid System Using Conceptual Graphs

The method we present in this paper is a hybrid approach that attempts to maintain the advantages of both content and social filtering methods. Specifically, we employ Conceptual Graphs [25], as means for representing both the information we have about the user and the information about rated items (Figure 1). This has significant advantages, including the ability to capture relationships between concepts. Another

important feature is the ability to maintain domain-dependent knowledge pertaining to similarity by using the *type lattice,* which accommodates the intrinsic notion of hierarchical ordering of concepts. We have introduced [26] a similarity measure for the comparison of fuzzy conceptual graphs and have described how this can be used in user modeling.

Our approach falls under the "memory-based" category according to Breese et al. [14], since we consider all library users as sources of information for prediction and do not build a model for subsequent use. We construct Conceptual Graphs to form the user models based on the data we have available. Ideally, the user model should represent the user as accurately as possible. We believe that information captured should not be restricted to one type only, such as the user's ratings of preference items, but also include other potentially useful details about the user, such as "job", "age group" and other characteristics. This leads to a more complete (and sometimes more complex) "picture" of the user. Our models also include content information about the voted items – not merely the titles of the items. The system can be considered hybrid, since it depends both on content knowledge and the ratings of other users to determine recommendations.

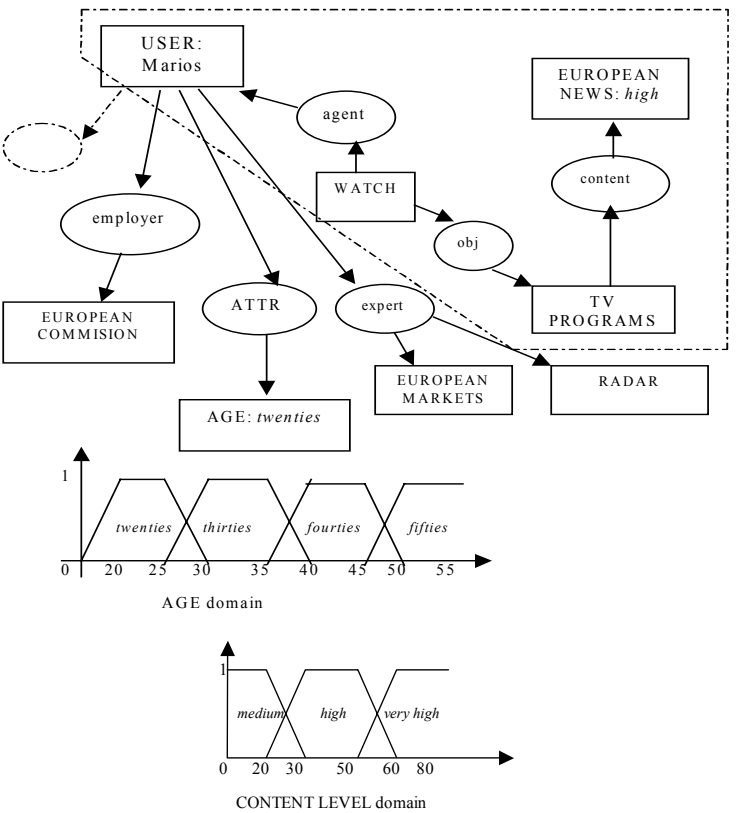


Fig. 1. Example User Model

As in other memory-based collaborative systems, predicting the interest i_p of a particular user, termed the *probe* user, is done by determining the similarity of the user to all m other users of the system (the *library users*) who have rated the item in question. The predicted rating is the weighted average of the library user's interest.

$$i_p = \frac{1}{s_t}\sum_{j=1}^{m} s_j i_j \quad \text{where} \quad s_t = \sum_{j=1}^{m} s_j \tag{2}$$

3.1 Conceptual Graph Similarity

Conceptual graphs are directed graphs with two types of nodes: *concepts* and *relations*. Each concept is associated with a *concept-type* and each relation is associated with a *relation-type*. Each concept is also associated with a particular value called the *referent*. Arcs flow from concepts to relations and from relations to concepts. Each relation, however, is limited to one incoming arc, and relations of the same type have the same number of outgoing arcs.

In our earlier work, we defined a similarity measure between two conceptual graphs, *probe* and *lib,* that was the maximum sum of the concept similarity over all legal matchings of concepts and relations of the two graphs. A matching is legal if relations of one graph are matched only with relations of the same type in the other graph. Furthermore, for any two matched relations, if their incoming arcs are from concepts c_{probe} and $c_{library}$, then the concepts c_{probe} and $c_{library}$ are paired in the matching.

The similarity of two concepts c_{probe} and $c_{library}$ with referents r_{probe} and $r_{library}$ and types t_{probe} and $t_{library}$ is

$$S_C(c_{probe}, c_{library}) = S_R(r_{probe}, r_{library}) S_T(t_{probe}, t_{library}) S_{RT}(c_{probe}, c_{library}) \tag{3}$$

where $S_R(r_{probe},r_{library})$ is the *referent similarity* and $S_T(t_{probe},t_{library})$ is the *type similarity*, which is described below. The *relation-type similarity*, $S_{RT}(c_{probe},c_{library})$, is a new term that describes the similarity between the relations to which concepts c_{probe} and $c_{library}$ have outgoing arcs.

It should be noted that the relation-type similarity measure, $S_{RT}(c_{probe},c_{library})$, is not communicative. That is, it is not necessarily true that $S_{RT}(c_{probe},c_{library}) = S_{RT}(c_{library},c_{probe})$. This is because there may be much more information available about one user than another. If everything we know about the probe user is matched by the library user, then the similarity is high - even if there is additional information known about the library user. If we know little about the library user, then even if everything we know matches information about the probe user, the similarity measure will not be as high.

3.2 Referent Similarity

Considerable expressiveness is added to the user model if *fuzzy referents* are allowed in the concepts. A fuzzy referent specifies the fuzzy set [27] associated with the concept. For instance, in the EachMovie application described below, we use fuzzy sets to represent a user's rating of a movie as ‘Bomb' or one to five stars. The similarity between two fuzzy referents is computed by means of unification of the associated fuzzy sets. Crisp referents, on the other hand, have a similarity of 1.

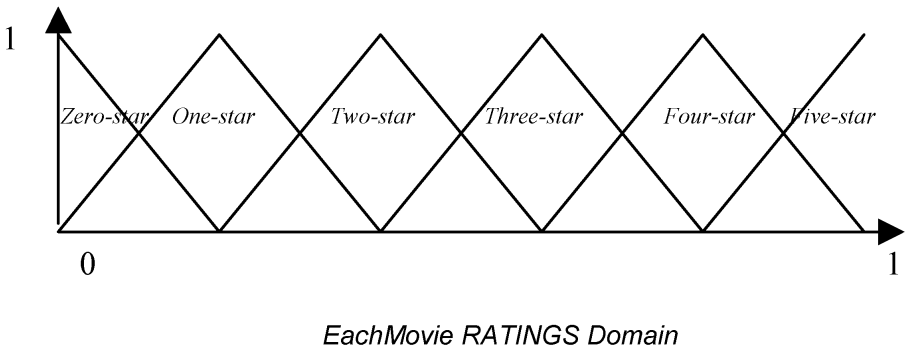


Fig. 2. Fuzzy Sets on the *RATINGS* domain

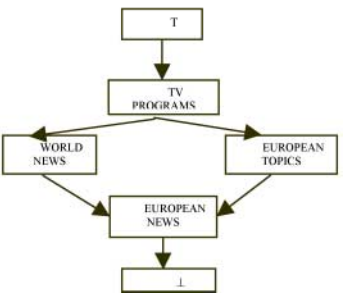


Fig. 3. Type Lattice

3.3 Type Similarity

The type lattice (Figure 3), a partial ordering defined over the set of concept types, plays a crucial role in our approach, and is a unique approach to providing knowledge about user similarity in particular domains. For type labels t and u, $t \leq u$ means that t is a subtype of u, or that u is a super-type of t, which can also be written $u \geq t$. If $t \leq u$ and $t \neq u$, then t is called a proper subtype of u, written $t < u$ and u is a proper super-type, written $u > t$. If we consider another type label v and $v \leq t$ and $v \leq u$, then v is a common subtype of t and u. If $v \geq t$ and $v \geq u$, v is a common super-type of t and u.

Two requirements for types in type lattice are:

for any pair of type labels t, u, there is:
- a minimal common super-type, written lub(t, u)
- a maximal common subtype, written glb(t, u)

two primitive types are needed:
- the universal type T
- the absurd type $\perp$

so that for any type t, $\perp \leq t \leq T$

The type similarity $S_T(t,t')$ of the types t and t' is determined by using the type lattice using the formula

$$S_T(t,t') = \frac{d(\text{lub}(t,t'))}{m} \tag{4}$$

where *d(t)* is the depth of node *t* within the type lattice.

This measure is based on the intuitive notion that the amount of similarity possible between concepts of two types, increases as more is known about the types. For example, two instances of "ballet-slippers" are known to be more similar than two instances of "foot-wear".

One of the main shortcomings of other methods is scarcity of information about users: the user models might not include enough information because users have not provided the system with enough feedback. As a result, the systems "pictures" of its users will be incomplete and any comparisons will not necessarily yield the correct results. Moreover, most of the existing systems cannot determine the similarity between users unless both users have voted on, or generally shown interest in, the exact same topics. The type lattice is one way our method addresses this problem. If two users have voted on topics that are related through the type lattice, our method will explore this similarity and can thus operate with less data. For example, one user might watch TV programs related to European News and another might watch programs related to World News. These two topics are not identical, but are definitely similar in some respect. Using the type lattice described above, "European News" would be a proper subtype of "World News" in the type lattice, and our system can use this fact to determine that there is some similarity between the two users.

3.4 Relation-Type Similarity

Perhaps the most important component of concept similarity for collaborative filtering problems is the relation-type similarity. In collaborative filtering problems, each user concept may have outgoing arcs to numerous relations that represent the user's rating of various items. It is how well two users' ratings of similar items agree that determines, to a large degree, the similarity of the users.

Let concepts c_{probe} and $c_{library}$ be the concepts that are being compared from the two graphs. Relation nodes to which these concepts have arcs are grouped according to their relation types. For example, the *user* concepts in our EachMovie experiments have links to relations with types: *age*, *gender*, *zip*, and *movie-rating*. Within each relation type, the relations to which c_{probe} has arcs, are paired with the relations to which $c_{library}$ has arcs, so that the sum of the *match-degree* of the matched relations is maximized. Again using the EachMovie application as an example, the match-degree between two movie-rating relations will depend on the content-similarity of the movies being rated.

After a pairing has been found which maximizes the match-degree within a particular relation type, the average *relation similarity* between the paired relations is taken over the number of relations of the given type that the *probe* concept has arcs to. In the EachMovie application, the relation similarity between two movie ratings depends both on the content-similarity of the movies that were rated and on the rating that each user gave their respective movie.

It is at this stage that the asymmetry of the relation-type similarity is introduced. Since the average is taken over the number of relations that the *probe* concept has arcs to, it is impossible to get a high degree of similarity unless a good deal more is known about the library user's opinions than about the probe user's opinions.

The final step in finding the relation-type similarity of the two concepts c_{probe} and $c_{library}$, is to take a weighted average, over all types of relations that c_{probe} has links to, of the average relation similarities computed above. This final weighting step provides additional flexibility. The weights to use are determined by the type of the concept c_{probe}. In the experiments described below, the weights were set so that similarity of movie ratings was the only criteria used when comparing users.

3.5 Computational Complexity of Conceptual Graph Operations

As in other memory-based collaborative filtering systems, scalability is a concern in our method, since the number of comparisons grows linearly with the number of users in the user library. From the description of the conceptual graph operations given above, it might appear that the similarity operation used on the conceptual graphs would require too much computation to be practical. However, we have found that with the conceptual graphs that arise in practice that the similarity metric is tractable. The most expensive operation is finding the correct match for relations when determining relation type similarity. This problem can be reduced to the problem of finding maximal matchings in a bipartite graph, for which there are well know polynomial time solutions. The results reported below also indicate that this method might require fewer library users than other methods in order to obtain accurate results.

4 Experimental Results

For the testing of our method we used the EachMovie[1] dataset, provided by the Compaq Systems Research Center. This data was collected over an 18-month period and consists of 72916 users (only 61263 actually voted), 1648 movies and 2811983 movie votes. User ratings range from zero to five stars. Zero stars indicate extreme dislike, whereas five stars indicate maximum delight. We feel these are linguistic values, and so treat them as fuzzy sets in the algorithm. They are to be the referents for the concept nodes that have the particular observed movie as a type. The fuzzy sets defined on the “ratings” domain are depicted in Figure 2. An example of a Conceptual Graph used in the experiments in shown in Figure 4. Some parts of the graph have not been included due to lack of space.

The experiments we performed were based on the protocols defined by Breese et al. [14]. These are the *All but 1*, *Given 2*, *Given 5*, and *Given 10* protocols. In the All-but-1 protocol, a randomly selected vote is removed from a user's profile and an attempt is made to predict its value. In the others protocols 2, 5, and 10 votes are selected, respectively, and are considered observed votes, which are used to predict the remaining votes. Generally, the All-but-1 experiment illustrates the algorithm's performance when there is plenty of data available about the user, and the other experiments test the algorithm's performance when less data is available.

[1] http://research.compaq.com/SRC/eachmovie

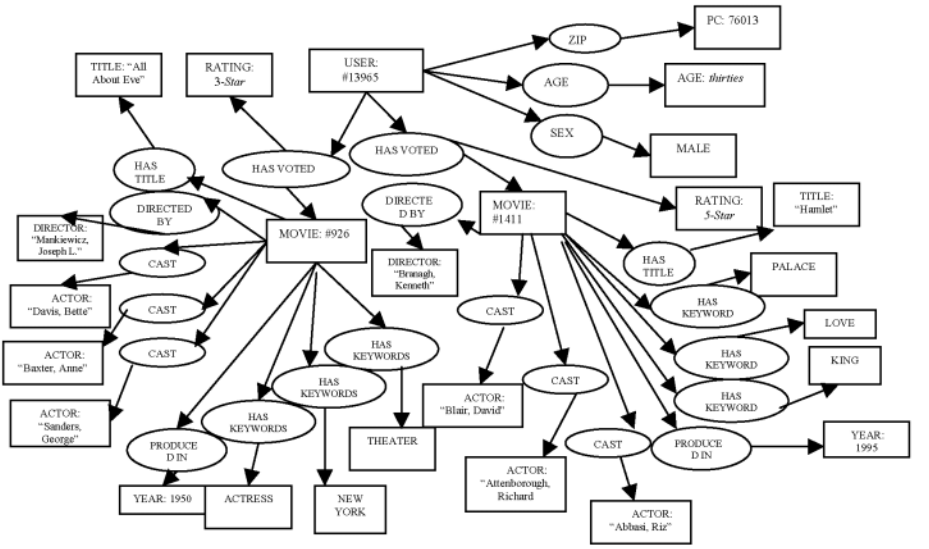


Fig. 4. Example Conceptual Graph created from the EachMovie data

Take a user and call him the active user
Select a random movie the active user has rated
and call it the test movie.
Remove the rating from the active user's record (In other words,
we're simulating the case where we don't know the user's rating of the
test movie)
For each user
call the user the library user
if the library user isn't the active user
if the library user has rated the test movie
Determine the similarity between the library user and the active user
Multiply the active user's rating of test movie by
the similarity between the users
Sum the product computed above for each library user

Fig. 5. Testing algorithm

Specifically, referring to the first protocol, we select one user as the active user. We randomly remove one observed movie's vote and try to predict it based on the remaining information in the user's model, along with the information in the library users' models. From the library users only those who have voted on the movie in question are considered at this stage. The algorithm we have employed for the particular protocol is outlined in Figure 5.

Apart from movie ratings in the user model we include information about the user himself, such as age group, residence location, etc., although this information was not used for comparison in the current experiments; rather, only the similarities of ratings of similar movies were used. On the other hand, large amounts on content information about the observed movies, including themes, actors, director, etc, were used to determine the similarity of movies.

Evaluation of our system was performed by measuring how well it predicts the vote that was removed each time. The primary evaluation metric used was the average error: the absolute value of the difference between the prediction and the actual rating is summed over all users and divided by the number of users. We also computed the standard deviation, which demonstrates whether the predictions were fairly accurate for most of the users, rather than excellent for some and bad for others.

We ran our experiments on sets of 10, 50, 100, 500 and 1000. The results are depicted in Tables 1, 2, 3 and 4. The algorithms that we compared against originate from [14], who also used the EachMovie dataset for their experiments. One algorithm uses Pearson's Correlation Coefficient with Case Amplification (labeled CORR in the tables), and another uses the same Coefficient but with Inverse User Frequency transformation as well as Case Amplification (labeled INV in the tables). Default votes, taking on the value of the user's average vote, were used with both of these systems. POP, the last algorithm, uses the mean average vote of each movie as the predicted vote for each user. Our method is labeled CG. Bold face indicates lower value that POP. The users selected for the tests had all rated at least 11 movies so that the *Given 10* protocol had enough data. We must stress that the same randomly selected users were used by all protocols and algorithms, with the same randomly selected ratings used within each protocol.

The experimental results we can see that our algorithm performs consistently better than the algorithms it was compared against. In all the tests except one, it outperformed POP, in contrast with the remaining methods, which did not perform as well in several cases.

Table 1. *All but 1* Protocol

System	*# of users*	*Avg. Error*	*Std. Dev.*
pop	10	0.173330	0.151372
corr	10	0.219617	0.159293
inv	10	0.220329	0.163195
cg	10	0.263031	**0.134602**
pop	50	0.258427	0.257511
corr	50	0.280593	**0.254570**
inv	50	0.311771	0.261328
cg	50	**0.185113**	**0.123475**
pop	100	0.253722	0.195443
corr	100	**0.228490**	0.204029
inv	100	**0.233660**	0.212370
cg	100	**0.210823**	**0.145001**
pop	500	0.189398	0.146623
corr	500	**0.180765**	0.148826
inv	500	**0.181693**	0.151226
cg	500	**0.179819**	**0.116594**
pop	1000	0.190915	0.148955
corr	1000	**0.185433**	**0.146645**
inv	1000	**0.184648**	**0.148120**
cg	1000	**0.187433**	**0.125086**

Table 2. *Given 2* Protocol

System	*# of users*	*# of tests*	*Avg. Error*	*Std. Dev.*
pop	10	419	0.354650	0.297839
corr	10	419	0.482907	0.321835
inv	10	419	0.480290	0.324581
cg	10	419	**0.205733**	**0.123095**
pop	50	2425	0.279268	0.241292
corr	50	2425	0.357011	0.282830
inv	50	2425	0.358891	0.281333
cg	50	2424	**0.196481**	**0.129172**
pop	100	5546	0.229814	0.191118
corr	100	5546	0.270598	0.232863
inv	100	5546	0.273823	0.234577
cg	100	5546	**0.192420**	**0.128366**
pop	500	26923	0.189061	0.149610
corr	500	26923	0.224164	0.186868
inv	500	26923	0.225191	0.186838
cg	500	26923	**0.184171**	**0.122474**
pop	1000	51605	0.186911	0.146867
corr	1000	51605	0.230380	0.190492
inv	1000	51605	0.231339	0.188396
cg	1000	51605	**0.183171**	**0.123059**

Table 3. *Given 5* Protocol

System	*# of users*	*# of tests*	*Avg. Error*	*Std. Dev.*
pop	10	389	0.350411	0.301303
corr	10	389	0.361243	**0.295485**
inv	10	389	0.359852	**0.295478**
cg	10	389	**0.206767**	**0.122810**
pop	50	2275	0.280428	0.242582
corr	50	2275	0.295387	0.250872
inv	50	2275	0.303836	0.254580
cg	50	2274	**0.195840**	**0.130014**
pop	100	5246	0.231069	0.190734
corr	100	5246	0.236680	0.198817
inv	100	5246	0.242162	0.201131
cg	100	5246	**0.193429**	**0.128795**
pop	500	25423	0.188907	0.149537
corr	500	25423	0.202935	0.162492
inv	500	25423	0.206663	0.165127
cg	500	25423	**0.183958**	**0.122378**
pop	1000	48605	0.186269	0.146590
corr	1000	48605	0.202997	0.162496
inv	1000	48605	0.205655	0.163611
cg	1000	48605	**0.182690**	**0.122946**

Table 4. *Given 10* Protocol

System	*# of users*	*# of tests*	*Avg. Error*	*Std. Dev.*
pop	10	339	0.359066	0.301303
corr	10	339	0.354271	**0.295485**
inv	10	339	0.356595	**0.295478**
cg	10	339	**0.208976**	**0.122810**
pop	50	2025	0.287755	0.242582
corr	50	2025	**0.285731**	0.250872
inv	50	2025	0.291674	0.254580
cg	50	2024	**0.197926**	**0.130014**
pop	100	4746	0.231239	0.190734
corr	100	4746	**0.222935**	0.198817
inv	100	4746	**0.226447**	0.201131
cg	100	4746	**0.192749**	**0.128795**
pop	500	22923	0.188700	0.149537
corr	500	22923	0.193922	0.162492
inv	500	22923	0.197892	0.165127
cg	500	22923	**0.183832**	**0.122353**
pop	1000	43605	0.186683	0.146590
corr	1000	43605	0.191942	0.162496
inv	1000	43605	0.194410	0.163611
cg	1000	43605	**0.183107**	**0.123191**

Table 5. Summary of All-But-1 Protocol Test Results Using 100 Datasets of 10 Randomly Selected Users

Method	*Avg. Error*	*Avg. Std. Dev.*	*Better Avg.*	*Better Std. Dev.*	*Best Avg.*	*Best Std. Dev.*
CG	0.190629	0.0153904	95	98	93	96
CORR	0.318865	0.0777568	56	54	3	3
INV	0.32759	0.0782967	48	52	4	1
POP	0.32671	0.0809687	-	-	3	2

Specifically, in the *All But 1* protocol, our algorithm's performance (both in terms of average error and standard deviation) is most of the times significantly better than that of the other methods when the number of users on the system is small - a difference that diminishes as the group of users grows larger. The same characteristic is even more apparent when observing the results obtained using the remaining protocols, in which the methods are tested for their behavior when less data is available.

Since the algorithm did not match the performance of POP on the small data set in the All-but-1 protocol, we did further experiments to determine if this was an anomaly. The *All But 1* protocol was used to run experiments on 100 additional sets of 10 users apiece which were formed by randomly selecting users that had at least three observed movies in their profile. The results are summarized in Table 5 and include the number of datasets in which each algorithm showed better performance (lower average error and lower standard deviation) in comparison to POP, as well as

the best performance overall. Our algorithm outperformed POP 95% of the time, whereas the CORR and INV methods only managed to beat it 56% and 48% of the time respectively. Finally, the CG algorithm scored the best average error and standard deviation 93% of the time.

4.1 Analysis of Results

Prediction using Conceptual Graphs as means for representing users and inference is not only competitive to other filtering methods, but also demonstrates superiority in most experiments conducted using the EachMovie dataset. Users in our case have a profile that consists of details about the user, the movies they have rated and the ratings themselves. When capturing details about the observed movies we do not only include the titles – we include vital information about the movie such as genre, director, actors, etc. By taking this approach, we are able to identify similarities between movies and as a result use this as a source from which to extract vital information. For example, if we match one user who has observed two movies, against another who has observed two different but similar movies, our method will in fact detect this similarity and will consider the two users somewhat similar. The other methods that compute similarity based on the common movies in user models, would find the two users of our example dissimilar. This shortcoming, which is even more serious when there is limited data available, is dealt with by adding votes to the models with techniques such as default voting. Our method does not require such extensions since its performance remains unaffected under these circumstances.

This advantage is further supported by the use of fuzzy sets as means for representing and dealing with votes. There is no way one could strictly define the boundaries between e.g. a 3-star vote and a 4-star vote: where does the first end and where does the second begin, what exactly is their definition? By assuming they are crisp numbers we automatically lead ourselves to consider two ratings such as a 3-star and a 4-star entirely different, when in fact they are not. Fuzzy 3-star and 4-star votes can be matched against each other to compute their similarity, which can be further exploited and interpreted as a supporting factor when matching two movies. The similarity between two users will increase the more similar movies they have observed and the more similar their ratings for those movies are.

5 Conclusion - Future Work

This article has presented a new algorithm that can be used in recommender systems that support personalization. This algorithm uses Conceptual Graphs to represent users and to reason. Conceptual Graphs have been proven suitable means for representation and have been used successfully in several application areas. We have shown that they present numerous advantages in the field of User Modeling as well. Information can be captured in a unique way, relating concepts and at the same time allowing its manipulation.

The algorithm lies under the collaborative filtering category of methods, due to the fact that is employs the information available about the library users in order to make predictions, but also under the content-based filtering category because it also uses

significant information about the items a user has selected. The combination of the two sources provides us with a more complete user model, a model that apparently leads to better recommendations.

Our algorithm was tested using the widely used for these purposes EachMovie dataset, yielding results that undoubtedly indicate that for a wide range of conditions it outperforms competing methods. The results show that the method is promising and should be further explored.

The algorithm uses fuzzy values in only a limited number of fields, but we believe this could be broadened to include additional fuzzy concepts. A great number of concepts that participate in the formation of the user's model can be seen as vague and unclear, lending themselves to computation with fuzzy sets. Vagueness is an obvious characteristic of our communication today: we rarely speak in a precise manner - we nearly always use imprecise words and phrases. Bearing this in mind, it is only natural that we try to do the same when describing the user. Descriptions are rarely crisp and so representation and reasoning with fuzzy sets is suitable in this case. By taking this route, our answers can also be fuzzy. For example, it seems more meaningful and useful to return an answer of something being "very similar" to something else, rather than a point value "0.8" that shows similarity.

The algorithm presented in this article was shown to significantly improve the process of recommendation. However, this is definitely not the only possible area of application. It can theoretically be applied to any area that involves personalization because it is in fact a technique for capturing user-related information and manipulating it to assist the user. Assistance can take many forms, from adapting content on web pages to proposing a solution to a problem or difficulty.

References

[1] Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM. **40**(3) (1997) 56-58

[2] Wahlster, W., Kobsa, A.: User Models in Dialog Systems. In: A. Kobsa and W. Wahlster (eds.): User Models in Dialog Systems. Springer-Verlag, Heidelberg-Berlin (1989)

[3] Kobsa, A., Koenemann, J., Pohl, W.: Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships. The Knowledge Engineering Review. **16**(2) (2001) 111-155

[4] Chen, Q., Norcio, A.F.: Modeling a user's domain knowledge with neural networks. International Journal of Human-Computer Interaction. **9**(1) (1997) 25-40

[5] Jennings, A., Higuchi, H.: A user model neural network for a personal news service. User Modeling and User Adapted Interaction. **3** (1993) 1-25

[6] Lang, K.: NewsWeeder: Learning to filter Netnews. In: Proceedings of the 12th International Conference on Machine Learning, Tahoe City, CA (1995)

[7] Billsus, D., Pazzani, M.J.: A Hybrid User Model for News Story Classification. In: J. Kay (ed.) User Modeling '99 - Proceedings of the 7th International Conference. Springer-Verlag, Wien-New York (1999) 99-108

[8] Sheth, B., Maes, P.: Evolving agents for personalized information filtering. In: Proceedings of the Ninth Conference on AI for Applications. IEEE Computer Society Press (1993)
[9] Balabanovic, M., Shoham Y.: Fab: Content-based collaborative recommendation. Communications of the ACM. **40**(3) (1997) 66-72
[10] Pazzani, M., Billsus, D.: Learning and Revising User Profiles: The identification of interesting web sites. Machine Learning. **27** (1997) 313-331
[11] Oard, D.W., Kim, J.: Implicit feedback for Recommender Systems. In: Proceedings of the AAAI Workshop on Recommender Systems. Madison, WI (1998) 81-83
[12] Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM. **35**(12) (1992) 61-70
[13] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An open architecture for collaborative filtering of Netnews. In: Proceedings of ACM Conference on Computer-Supported Cooperative Work. (1994) 175-186
[14] Breese, J.S., Heckerman, D., Kadie C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. Madison, WI (1998) 43-52
[15] Ha, V., Haddawy, P.: Toward Case-Based Preference Elicitation: Similarity Measures on Preference Structures. In: Proceedings of 14th Conference on Uncertainty in Artificial Intelligence. Madison, WI (1998) 193-201
[16] Pennock, D.M., Horvitz, E., Lawrence, S., Lee Giles, C.: Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach. In: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence. (2000) 473-480
[17] Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating "word of mouth". In: Proceedings of ACM Conference on Human Factors in Computing Systems. (1995) 210-217
[18] Wu, H., Salton, G.: A Comparison of Search Term Weighting: Term Relevance vs. Inverse Document Frequency. In: Proceedings of the 4th International Conference on Information Storage and Retrieval: Theoretical Issues in Information Retrieval. (1981)
[19] Alspector, J., Kolcz, A., Karunanithi, N.: Comparing Feature-Based and Clique-Based User Models for Movie Selection. In: Proceedings of the 3rd ACM Conference on Digital Libraries. Pittsburgh, PA (1998)
[20] Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.: Grouplens: Applying collaborative filtering to Usenet news. Communications of the ACM. **40**(3) (1997) 77-87
[21] Good, N., Schafer, J.B., Konstan, J.A., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J.: Combining Collaborative Filtering with Personal Agents for Better Recommendations. In: Proceedings of the 16th National Conference on Artificial Intelligence. (1999)
[22] Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl J.: Item-based Collaborative Filtering Recommendation Algorithms. In: Proceedings of the 10th International World Wide Web Conference. Hong Kong (2001)

[23] Claypool, M., Gokhale, A., Miranda, T.: Combining Content-Based and Collaborative Filters in an Online Newspaper. In: Proceedings of the ACM SIGIR Workshop on Recommender Systems. (1999)
[24] Basu, C., Hirsh, H., Cohen, W.: Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In: Proceedings of the 15th National Conference on Artificial Intelligence. (1998)
[25] Sowa, J.F.: Conceptual Structures. Addison-Wesley, Reading, MA (1984)
[26] Tzanavari, A., Paulson, P.: Intelligent Information Processing for User Modeling. In: Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems. Annecy, France (2002) 1763-1770
[27] Zadeh, L.A.: Fuzzy Sets. Information and Control. **8** (1965) 338-353

Random Sets and Appropriateness Degrees for Modelling with Labels

Jonathan Lawry

Department of Engineering Mathematics, University of Bristol, Bristol, UK
j.lawry@bris.ac.uk

Abstract. A random set semantics for imprecise concepts is introduced. It is then demonstrated how label prototypes describing data sets can be learnt in this framework. These prototypes take the form of vectors of mass assignments showing the distribution of appropriate labels across the database for various attributes. The potential of this approach for fusion, classification and query evaluation is then investigated.

1 Introduction

There is an increasing need in science and engineering to learn models from data that can be understood at the level of natural language. In other words, for a large number of problem areas it is a requirement that the learnt models should be transparent and provide information regarding underlying trends and relationships. Expressing such models within a knowledge representation framework at least close to natural language has obvious advantages in terms of transparency, and also allows for the possibility of fusing the information provided by the model with expert/background knowledge in natural language form. We would certainly want any proposed framework to incorporate uncertainty and vagueness since these are inherent to almost all human communication. However, unlike natural language we would also want our representation framework to be formal and have a clear underlying interpretation so that we can have sufficient confidence in the validity of our models. This combination of transparency, formality and interpretability lies at the heart of 'modelling with words'.

In this paper we propose a framework for modelling imprecise labels in terms of random sets and the related concepts of appropriateness degree and mass assignment. Appropriateness degrees are functions measuring the appropriateness of a linguistic expression as a description of a particular value. A calculus for these measures is introduced which is functional, and agrees with the fuzzy logic *min* and *max* operators at the simple label level while still satisfying the laws of excluded middle and non contradiction. We then define linguistic models in this framework to be vectors of mass assignments on labels and show how these can be inferred from data. Such vectors are label descriptions of subsets of the database and hence are referred to as label prototypes since they represent an amalgam of elements. It is shown that such prototypes form flexible models providing information both at the attribute and label levels while maintaining good prediction accuracy.

J. Lawry, J. Shanahan, A. Ralescu (Eds.): Modelling with Words, LNAI 2873, pp. 186–208, 2003.
© Springer-Verlag Berlin Heidelberg 2003

In recent years the representation of imprecise or vague labels has mainly been studied in connection with fuzzy logic [13]. However, while fuzzy methods have been very successful in areas such has fuzzy control they have a number of unresolved semantic problems. Principal amongst these is the difficulty in providing a clear interpretation of membership functions. Although a number of different interpretations have been suggested (see [1] and [2]), it is often difficult to see how these can be reconciled with a fully truth-functional calculus such as that of fuzzy logic. On the other hand a clear semantics is necessary for any high-level framework aiming to provide understandable models. Transparency cannot be achieved if the validity of the underlying concepts and inference processes are either obscured or in doubt. Instead we develop a random set calculus for modelling imprecise concepts based on an explicit semantics as outlined in the following section and apply this to data analysis.

2 Label Semantics

For an attribute (or variable) x into a domain of discourse Ω we identify a finite set of words LA with which to label the values of x. Then for a specific value $a \in \Omega$ an individual I identifies a subset of LA, denoted $\mathcal{D}_a^I$ to stand for the description of a given by I, as the set of words with which it is appropriate to label a. Within this framework then, an expression such as 'the diastolic blood pressure is *high*', as asserted by I, is interpreted to mean $high \in \mathcal{D}_{bp}^I$ where bp denotes the value of the variable blood pressure. If we allow I to vary across a population of individuals V then we naturally obtain a random set $\mathcal{D}_x$ from V into the power set of LA where $\mathcal{D}_x(I) = \mathcal{D}_x^I$. A probability distribution (or mass assignment) associated with this random set can be defined and is dependent on the prior distribution over the population V. We can view the random set $\mathcal{D}_x$ as a description of the variable x in terms of the labels in LA.

Definition 1. *(Label Description) For $x \in \Omega$ the label description of x is a random set from V into the power set of LA, denoted $\mathcal{D}_x$, with associated distribution m_x, given by*

$$\forall S \subseteq LA \; m_x(S) = Pr(\{I \in V : \mathcal{D}_x^I = S\})$$

Another high level measure associated with m_x is the following quantification of the degree of appropriateness of a particular word $L \in LA$ as a label of x.

Definition 2. *(Appropriateness Degrees)*

$$\forall x \in \Omega, \; \forall L \in LA \; \mu_L(x) = \sum_{S \subseteq LA : L \in S} m_x(S)$$

Clearly, this framework is related to the random set interpretation for fuzzy sets as proposed by Goodman [6], [5]. However, while the latter defined random sets on the power set of the attribute universe, label semantics defines random

sets on the power set of labels. In our view focusing on the labels in this way gives an interesting new perspective as well as providing a coherent functional calculus.

Example 1. Suppose three professors are asked to categorise students as being weak, average or good on the basis of their project mark graded between 1 and 5. In this case $LA = \{weak, average, good\}$, $\Omega = \{1, \ldots, 5\}$ and $V = \{I_1, I_2, I_3\}$. A possible assignment of sets of appropriate labels is as follows:

$$\mathcal{D}_1^{I_1} = \mathcal{D}_1^{I_2} = \mathcal{D}_1^{I_3} = \{weak\}$$

$$\mathcal{D}_2^{I_1} = \{weak\},\ \mathcal{D}_2^{I_2} = \{weak\},\ \mathcal{D}_2^{I_3} = \{weak, average\}$$

$$\mathcal{D}_3^{I_1} = \{average\},\ \mathcal{D}_3^{I_2} = \{weak, average, good\},\ \mathcal{D}_3^{I_3} = \{average\}$$

$$\mathcal{D}_4^{I_1} = \{average, good\},\ \mathcal{D}_4^{I_2} = \{good\},\ \mathcal{D}_4^{I_3} = \{good\}$$

$$\mathcal{D}_5^{I_1} = \mathcal{D}_5^{I_2} = \mathcal{D}_5^{I_3} = \{good\}$$

Assuming a uniform distribution on V this generates the following mass assignments:

$$m_1 = \{weak\} : 1$$

$$m_2 = \{weak\} : \frac{2}{3},\ \{weak, average\} : \frac{1}{3}$$

$$m_3 = \{average\} : \frac{2}{3},\ \{weak, average, good\} : \frac{1}{3}$$

$$m_4 = \{good\} : \frac{2}{3},\ \{average, good\} : \frac{1}{3}$$

$$m_5 = \{good\} : 1$$

From these we can evaluate the appropriateness degrees for weak, poor and average as follows:

$$\mu_{weak}(1) = 1,\ \mu_{weak}(2) = 1, \mu_{weak}(3) = \frac{1}{3}$$

$$\mu_{average}(2) = \frac{1}{3},\ \mu_{average}(3) = 1,\ \mu_{average}(4) = \frac{1}{3}$$

$$\mu_{good}(5) = 1,\ \mu_{good}(4) = 1, \mu_{good}(3) = \frac{1}{3}$$

We now make the additional assumption that value descriptions are consonant random sets. This simply means that individuals in V differ regarding what labels are appropriate for a value only in terms of generality or specificity. The consonance restriction could be justified by the idea that all individuals share a common ordering on the appropriateness of labels for a value and that the composition of $\mathcal{D}_x^I$ is consistent with this ordering for each I. The consonance assumption also means that m_x can be completely determined from the values

of $\mu_L(x)$ for $L \in LA$ as follows: If $\{\mu_L(x) : L \in LA\} = \{y_1, \ldots, y_n\}$ ordered such that $y_i > y_{i+1}$ for $i = 1, \ldots, n-1$ then for $S_i = \{L \in LA : \mu_L(x) \geq y_i\}$,

$$m_x(S_i) = y_i - y_{i+1} \; for \; i = 1, \ldots, n-1$$

$$m_x(S_n) = y_n \; and \; m_x(\emptyset) = 1 - y_1$$

This has considerable practical advantages since we no longer need to have any knowledge of the underlying population of individuals V in order to determine m_x. Rather, for reasoning with label semantics in practice we need only define appropriateness degrees μ_L for $L \in LA$ corresponding to the imprecise definition of each label.

Example 2. Let x be a variable into $[0, 30]$ with associated label set $LA = \{small(s),\ medium(m),\ large(l)\}$. Suppose that the appropriateness degree for the three labels are defined by the following piecewise linear trapezoidal functions (see figure 1): $\forall x \in [0, 30]$

$$\mu_{small}(x) = \begin{cases} 1 : x \in [0, 10] \\ \frac{15-x}{5} : x \in (10, 15] \\ 0 : otherwise \end{cases}$$

$$\mu_{medium}(x) = \begin{cases} \frac{x-5}{5} : x \in [5, 10] \\ 1 : x \in (10, 20] \\ \frac{25-x}{5} : x \in (20, 25] \\ 0 : otherwise \end{cases}$$

$$\mu_{large}(x) = \begin{cases} \frac{x-15}{5} : x \in [15, 20] \\ 1 : x \in (20, 30] \\ 0 : otherwise \end{cases}$$

Now for $x = 8$ we have that $\mu_{small}(8) = 1$, $\mu_{medium}(8) = 0.6$, $\mu_{large}(x) = 0$. From this we obtain the following consonant mass assignment for $\mathcal{D}_8$:

$$m_8 = \{small,\ medium\} : 0.6,\ \{small\} : 0.4$$

Allowing x to vary across $[0, 30]$ we obtain the following definition of m_x as a function of x where all sets of labels not listed have zero mass (see figure 2):

$$m_x(\{small\}) = \begin{cases} 1 : x \in [0, 5] \\ \frac{10-x}{5} : x \in (5, 10] \\ 0 : otherwise \end{cases}$$

$$m_x(\{small,\ medium\}) = \begin{cases} \frac{x-5}{5} : x \in [5, 10] \\ \frac{15-x}{5} : x \in (10, 15] \\ 0 : otherwise \end{cases}$$

$$m_x(\{medium\}) = \begin{cases} \frac{x-10}{5} : x \in [10, 15] \\ \frac{20-x}{5} : x \in (15, 20] \\ 0 : otherwise \end{cases}$$

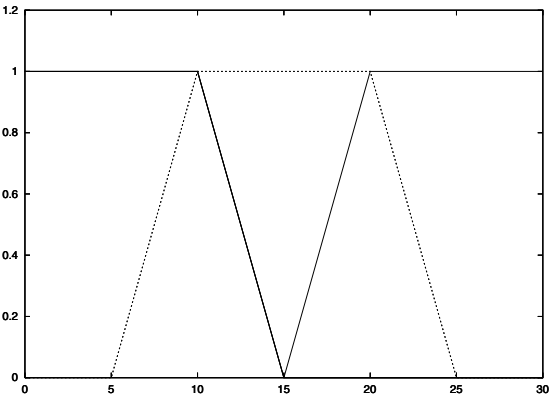


Fig. 1. Appropriateness degrees for, from left to right, *small*, *medium* and *large*

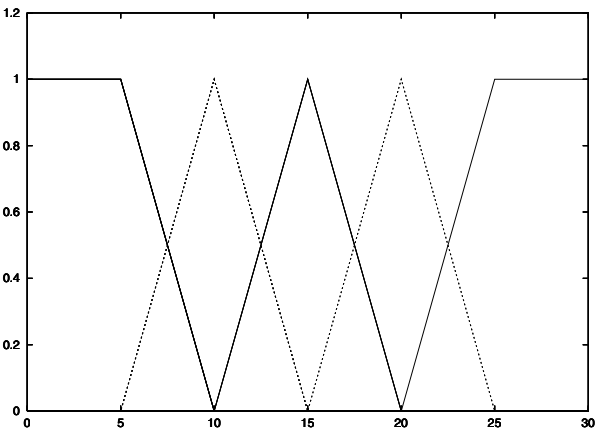


Fig. 2. Mass assignments for varying x; shown from left to right, $m_x(\{small\})$, $m_x(\{small, medium\})$, $m_x(\{medium\})$, $m_x(\{medium, large\})$ and $m_x(\{large\})$

$$m_x(\{medium,\ large\}) = \begin{cases} \frac{x-15}{5} : x \in [15, 20] \\ \frac{25-x}{5} : x \in (20, 25] \\ 0 : otherwise \end{cases}$$

$$m_x(\{large\}) = \begin{cases} \frac{x-20}{5} : x \in [20, 25] \\ 1 : x \in (25, 30] \\ 0 : otherwise \end{cases}$$

From the above example, we can see that for LA defined according to a given set of appropriateness degrees it is often the case that only a strict subset of 2^{LA} can actually occur as sets of appropriate labels. For instance, in example 2 since

large and *small* do not overlap the set $\{small, large\}$ cannot occur. The possible label sets are referred to as focal elements and are formally defined as follows:

Definition 3. *Set of Focal Elements*
Given labels LA together with associated appropriateness degrees then the set of focal elements for LA is given by:

$$\mathcal{F} = \{S \subseteq LA | \exists x \in \Omega,\ m_x(S) > 0\}$$

For more general linguistic reasoning a mechanism is required for evaluating compound label expressions. For example, we may wish to know whether or not expressions such as $medium \wedge low$, $medium \vee low$ and $\neg high$ are appropriate to describe a value $x \in \Omega$.

Definition 4. *Label Expressions*
The set of label expressions of LA, LE, is defined recursively as follows:

(i) $L_i \in LE$ *for* $i = 1, \ldots, n$
(ii) If $\theta, \varphi \in LE$ *then* $\neg\theta, \theta \wedge \varphi, \theta \vee \varphi, \theta \rightarrow \varphi \in LE$

In the context of this assertion-based framework we interpret the main logical connectives in the following manner: $L_1 \wedge L_2$ means that both L_1 and L_2 are appropriate labels, $L_1 \vee L_2$ means that either L_1 or L_2 are appropriate labels and $\neg L$ means that L is not an appropriate label. More generally, if we consider label expressions formed from LA by recursive application of the connectives then an expression θ identifies a set of possible label sets $\lambda(\theta)$ as follows:

Definition 5. *For $L \in LA$ $\lambda(L) = \{S \subseteq LA : L \in S\}$ and for label expressions θ and φ*

(i) $\forall L_i \in LA\ \lambda(L_i) = \{S \subseteq LA | L_i \in S\}$
(ii) $\lambda(\theta \wedge \varphi) = \lambda(\theta) \cap \lambda(\varphi)$
(iii) $\lambda(\theta \vee \varphi) = \lambda(\theta) \cup \lambda(\varphi)$
(iv) $\lambda(\neg\theta) = \overline{\lambda(\theta)}$
(v) $\lambda(\theta \rightarrow \varphi) = \lambda(\neg\theta) \cup \lambda(\varphi)$

Intuitively, $\lambda(\theta)$ corresponds to those subsets of LA identified as being possible values of $\mathcal{D}_x$ by expression θ. In this sense the imprecise linguistic restriction 'x is θ' on x corresponds to the strict constraint $\mathcal{D}_x \in \lambda(\theta)$ on $\mathcal{D}_x$. Hence, we can view label descriptions as an alternative to linguistic variables [12] as a means of encoding linguistic constraints.

The notion of appropriateness measure given above can now be extended so that it applies to compound label expressions. The idea here is that $\mu_\theta(x)$ quantifies the degree to which expression θ is appropriate to describe x.

$$\mu_\theta(x) = \sum_{S \in \lambda(\theta)} m_x(S)$$

Example 3. Let $LA = \{L_1, L_2, L_3\}$ then

$$\lambda(\neg L_1) = \{\{L_2\}, \{L_3\}, \{L_2, L_3\}, \emptyset\}$$

$$\lambda(L_1 \wedge L_2) = \{\{L_1, L_2\}, \{L_1, L_2, L_3\}\}$$

$$\lambda(L_1 \vee L_2) = \{\{L_1\}, \{L_2\}, \{L_1, L_2\}, \{L_1, L_3\}, \{L_2, L_3\}, \{L_1, L_2, L_3\}\}$$

$$\lambda(L_2 \wedge \neg L_3) = \{\{L_1, L_2\}, \{L_2\}\}$$

In the case where $L_1 = small,\ L_2 = medium$ and $L_3 = large$ with appropriateness degrees as defined in example 2 then

$$\mu_{medium \wedge \neg large}(x) = m_x(\{small,\ medium\}) + m_x(\{medium\})$$

so that

$$\mu_{medium \wedge \neg large}(x) = \begin{cases} \frac{10-x}{5} : x \in [5, 10] \\ 1 : x \in (10, 15] \\ \frac{20-x}{5} : x \in (15, 20] \\ 0 : otherwise \end{cases}$$

Given this basic framework the following properties have been proven to hold. (see [9])

Proposition 1. *For expression θ and φ generated recursively from LA the following hold:*

(i) If $\theta \models \varphi$ then $\forall x \in \Omega\ \mu_\theta(x) \leq \mu_\varphi(x)$
(ii) If $\theta \equiv \varphi$ then $\forall x \in \Omega\ \mu_\theta(x) = \mu_\varphi(x)$
(iii) If θ is a tautology then $\forall x \in \Omega\ \mu_\theta(x) = 1$
(iv) If θ is a contradiction then $\forall x \in \Omega$
$\mu_\theta(x) = 0$
(v) $\forall x \in \Omega\ \mu_{\neg\theta}(x) = 1 - \mu_\theta(x)$
(vi) $\forall L_i, L_j \in LA,\ \forall x \in \Omega$
$\mu_{L_i \wedge L_j}(x) = \min(\mu_{L_i}(x), \mu_{L_j}(x))$
(vii) $\forall L_i, L_j \in LA,\ \forall x \in \Omega$
$\mu_{L_i \vee L_j}(x) = \max(\mu_{L_i}(x), \mu_{L_j}(x))$

In the following sections we investigate how this framework can be applied to data analysis. Initially, however, we extend it to the multi-dimensional case.

3 Multi-dimensional Appropriateness Degrees

Most modelling problems involve multiple attributes or variables. Therefore, if label semantics is to provide an effective knowledge representation framework for linguistic modelling it must be generalised to the multi-dimensional case. In other words, we need to provide a means of interpreting and evaluating linguistic expressions involving more than one variable.

Specifically, consider a modelling problem with n variables (or attributes) $x_1, \ldots, x_n$ with associated universes $\Omega_1, \ldots, \Omega_n$. For each variable we defined a set of labels $LA_j = \{L_{1,j}, \ldots, L_{k_j,j}\}$ for $j = 1, \ldots, n$. In this case we ask individuals from V to provide a set of appropriate labels for each attribute value. Hence, an individual I will provide a vector of label descriptions $\langle \mathcal{D}^I_{x_1}, \ldots, \mathcal{D}^I_{x_n} \rangle$ for the attribute vector $\langle x_1, \ldots x_n \rangle$. In this context we can extend the definitions of mass assignment and appropriateness degree given in section 2 to the multi-dimensional case. Initially, however, we formally define n-dimensional linguistic expressions.

Let LE_j be the set of label expression for variable x_j generated by recursive application of the connectives $\wedge, \vee, \rightarrow, \neg$. We can now define the set of multi-dimensional label expression for describing linguistic relationships between variables as follows:

Definition 6. *(Multi-dimensional Label Expressions)*
$MLE^{(n)}$ is the set of all multi-dimensional label expressions that can be generated from the label expression $LE_j : j = 1, \ldots, n$ and is defined recursively by:

- *(i) If $\theta \in LE_j$ for $j = 1, \ldots, n$ then $\theta \in MLE^{(n)}$*
- *(ii) If $\theta, \varphi \in MLE^{(n)}$ then $\neg\theta,\ \theta \wedge \varphi,\ \theta \vee \varphi,\ \theta \rightarrow \varphi \in MLE^{(n)}$*

Any $n-$dimensional label expression θ identifies a subset of $2^{LA_1} \times \ldots \times 2^{LA_n}$, denoted $\lambda^{(n)}(\theta)$, constraining the cross product of label descriptions $\mathcal{D}_{x_1} \times \ldots \times \mathcal{D}_{x_n}$. In this way the imprecise constraint θ on $x_1 \times \ldots \times x_n$ is interpreted as the precise constraint $\mathcal{D}_{x_1} \times \ldots \times \mathcal{D}_{x_n} \in \lambda^{(n)}(\theta)$.

Definition 7. *(Multi-dimensional Appropriate Label Sets)*
$\forall \theta \in MLE^{(n)}\ \lambda^{(n)}(\theta) \subseteq 2^{LA_1} \times \ldots \times 2^{LA_n}$ *such that*

- $\forall \theta \in LE_j\ \lambda^{(n)}(\theta) = {}^{1}\lambda(\theta) \times \bigtimes_{i \neq j} 2^{LA_i}$
- $\forall \theta, \varphi \in MLE^{(n)}\ \lambda^{(n)}(\theta \wedge \varphi) = \lambda^{(n)}(\theta) \cap \lambda^{(n)}(\varphi)$
- $\lambda^{(n)}(\theta \vee \varphi) = \lambda^{(n)}(\theta) \cup \lambda^{(n)}(\varphi)$
- $\lambda^{(n)}(\theta \rightarrow \varphi) = \overline{\lambda^{(n)}(\theta)} \cup \lambda^{(n)}(\varphi)$
- $\lambda^{(n)}(\neg\theta) = \overline{\lambda^{(n)}(\theta)}$

Note that in practice it may be more convenient to evaluate $\lambda^{(n)}(\theta) \cap \bigtimes_{j=1}^{n} \mathcal{F}_j$ where $\mathcal{F}_j$ is the set of focal elements for LA_j (see definition 3).

Example 4. Consider a modelling problem with two variables x_1 and x_2 for which $LA_1 = \{small, medium, large\}$ and $LA_2 = \{low, moderate, high\}$. Also suppose the focal elements for LA_1 and LA_2 are, respectively:

$$\mathcal{F}_1 = \{\{small\}, \{small, medium\}, \{medium\}, \{medium, large\}, \{large\}\}$$

$$\mathcal{F}_2 = \{\{low\}, \{low, moderate\}, \{moderate\}, \{moderate, high\}, \{high\}\}$$

[1] $\lambda(\theta) \subseteq LA_j$ refers to the one dimensional appropriate label set as given in definition 5.

Now according to definition 7 we have that:

$$\lambda^{(2)}((medium \wedge \neg small) \wedge \neg low) = \lambda^{(2)}(medium \wedge \neg small) \cap \lambda^{(2)}(\neg low)$$
$$= \lambda(medium \wedge \neg small) \times \lambda(\neg low)$$

Now

$$\lambda(medium \wedge \neg small) \cap \mathcal{F}_1 = \{\{medium\}, \{medium, large\}\}$$

and

$$\lambda(\neg low) \cap \mathcal{F}_2 = \{\{moderate\}, \{moderate, high\}, \{high\}\}$$

Hence,

$$\lambda^{(2)}((medium \wedge \neg small) \wedge \neg low) \cap (\mathcal{F}_1 \times \mathcal{F}_2) =$$
$$\{\langle\{medium\}, \{moderate\}\rangle,$$
$$\langle\{medium\}, \{moderate, high\}\rangle, \langle\{medium\}, \{high\}\rangle,$$
$$\langle\{medium, large\}, \{moderate\}\rangle, \langle\{medium, large\}, \{moderate, high\}\rangle,$$
$$\langle\{medium, large\}, \{high\}\rangle\}$$

We can now extend the concepts of mass assignment and appropriateness degree to multi-dimensions as follows:

Definition 8. *(Joint Mass Assignment)*

$$\forall x_j \in \Omega_j \ \forall S_j \subseteq LA_j : j = 1, \ldots, n \ m_{\langle x_1, \ldots, x_n\rangle}(S_1, \ldots, S_n) = \prod_{j=1}^{n} m_{x_j}(S_j)$$

Now

$$m_{\langle x_1, \ldots, x_n\rangle}(S_1, \ldots, S_n) = Pr(\{I \in V : \mathcal{D}^I_{x_1} = S_1, \ldots, \mathcal{D}^I_{x_n} = S_n\})$$

provided we make the following conditional independence assumption. It is assumed that for each individual I the choice of appropriate labels for variable x_j is dependent only on the value of x_j, once this is known, and is independent of the value of any other variables. This is actually quite a weak assumption and does not *a prior* imply independence between the variables.

Definition 9. *(Multi-dimensional Appropriateness Degrees)*

$$\forall \theta \in MLE^{(n)}, \ \forall x_j \in \Omega_j : j = 1, \ldots, n$$
$$\mu^{(n)}_\theta(x_1, \ldots, x_n) = \sum_{\langle S_1, \ldots, S_n\rangle \in \lambda^{(n)}(\theta)} m_{\langle x_1, \ldots, x_n\rangle}(S_1, \ldots, S_n)$$
$$= \sum_{\langle S_1, \ldots, S_n\rangle \in \lambda^{(n)}(\theta)} \prod_{j=1}^{n} m_{x_j}(S_j)$$

4 Label Prototypes

In this section we formalize the notion of label prototype as a multi-dimensional linguistic description of an amalgam of elements. Suppose we have a database DB of N elements, associated with each of which are n measurements $x_1, \ldots, x_n$ so that $DB = \{\langle x_1(i), \ldots, x_n(i)\rangle : i = 1, \ldots, N\}$ where $x_j(i)$ denotes the value of x_j for object i. Further, suppose that we select a set of labels LA_j for each attribute x_j for $j = 1, \ldots, n$ where each label is defined by an appropriateness measure. The label prototype for DB is now defined to be a vector of mass assignments as follows:

Definition 10. *(Label Prototype for DB) The label prototype for DB is a vector $\mathcal{L}(DB) = \langle m_1, \ldots, m_n\rangle$ where*

$$\forall S \subseteq LA_j \; m_j(S) = \frac{1}{N}\sum_{i=1}^{N} m_{x_j(i)}(S)$$

Given a label prototype for DB we now evaluate a joint mass assignment on $2^{LA_1} \times \ldots \times 2^{LA_n}$ so that $\forall S_j \in 2^{LA_j}, j = 1, \ldots, n$

$$m_{DB}(S_1, \ldots, S_n) = \prod_{j=1}^{n} m_j(S_j)$$

Clearly, we are making an independence assumption here and in some cases this may not be appropriate. In order to overcome this problem one approach is to partition DB into a number of disjoint sets $P_1, \ldots, P_c$, perhaps according to some standard clustering algorithm, where the elements contained in each partition set are assumed to be sufficiently similar to allow an independence assumption. We can then learn label prototypes $\mathcal{L}(P_k)$ for $k = 1, \ldots, c$ and combine them to form an overall mass assignment for DB as follows: Let $\mathcal{L}(P_k) = \langle m_{1,k}, \ldots, m_{n,k}\rangle$ then $\forall S_j \in 2^{LA_j}, j = 1, \ldots, n$

$$m_{DB}(S_1, \ldots, S_n) = \sum_{k=1}^{c} \frac{|P_k|}{N} \prod_{j=1}^{n} m_{j,k}(S_j)$$

Such label prototypes can be thought of as imprecise definitions of the points in $P_k : k = 1, \ldots, c$. Now given a joint mass assignment on DB and a multi-dimensional label expression $\theta \in MLE^{(n)}$, we can now use labels semantics to evaluate the appropriateness of θ for describing DB in the following way:

$$\mu_\theta(DB) = \sum_{\langle S_1, \ldots S_n\rangle \in \lambda^{(n)}(\theta)} m_{DB}(S_1, \ldots, S_n)$$

5 Mapping from Mass Assignments to Densities

For many types of data analysis it is useful to be able to estimate the distribution on underlying variables given the information contained in DB. In

the current context our knowledge of DB is represented by the mass assignment m_{DB} and hence we need to be able to evaluate a distribution on the base variables $x_1, \ldots, x_n$ conditional on m_{DB} [8]. For simplicity, we now assume that all variables are continuous with domains of discourse comprising of bounded closed intervals of the real line. Furthermore, we assume a prior joint distribution $p(x_1, \ldots, x_n)$ for the base variables. In the case that p is unknown we will assume it to be the uniform distribution. Furthermore, to simplify the following definition we will at least assume that $x_1, \ldots, x_n$ are *a priori* independent so that $p(x_1, \ldots, x_n) = \prod_{j=1}^{n} p_j(x_j)$, p_j being the marginal prior on x_j.

Definition 11. *(Conditional Density given a Mass Assignment)*
Let x be a variable into Ω with prior distribution $p(x)$, LA be a set of labels for x and m be a posterior mass assignment for the set of appropriate labels of x (i.e. $\mathcal{D}_x$) inferred from some database DB. Then the posterior distribution of x conditional on m is given by:

$$\forall x \in \Omega \; p(x|m) = p(x) \sum_{S \subseteq LA} \frac{m(S)}{pm(S)} m_x(S)$$

where pm is the prior mass assignment generated by the prior distribution p according to

$$pm(S) = \int_{\Omega} m_x(S) p(x) dx$$

This definition is motivated by the following argument based on the theorem of total probability: $\forall a \in \Omega$

$$p(a|m) = \sum_{S \subseteq LA} p(a|\mathcal{D}_x = S) Pr(\mathcal{D}_x = S) = \sum_{S \subseteq LA} p(a|\mathcal{D}_x = S) m(S)$$

Also

$$p(a|\mathcal{D}_x = S) = \frac{Pr(\mathcal{D}_x = S|x = a)p(a)}{Pr(\mathcal{D}_x = S)} = \frac{m_a(S)p(a)}{pm(S)}$$

Making the relevant substitutions and then simplifying gives the expression from definition 11. In effect this conditioning process provides a mapping from mass assignments on label sets to probability density functions on the underlying attribute space.

Example 5. Let x be variable into $\Omega = [0, 122]$ with a uniform prior distribution so that $\forall x \in [0, 122]$ $p(x) = \frac{1}{122}$. For $LA = \{very\ small,\ small,\ medium,\ large,\ very\ large\}$ the appropriateness degrees are defined such that the prior mass assignment is given by:

$$pm = \{very\ small\} : 0.368, \{very\ small, small\} : 0.143, \{small\} : 0.041,$$
$$\{small, medium\} : 0.031, \{medium\} : 0.021, \{medium, large\} : 0.025,$$
$$\{large\} : 0.029, \{large, very\ large\} : 0.096, \{very\ large\} : 0.246$$

Also suppose we are given the following posterior mass assignment:

$$m = \{very\ small\} : 0.08, \{very\ small, small\} : 0.111, \{small\} : 0.102,$$
$$\{small, medium\} : 0.118, \{medium\} : 0.075, \{medium, large\} : 0.084,$$
$$\{large\} : 0.107, \{large, very\ large\} : 0.223, \{very\ large\} : 0.1$$

Now if for $x = 68$

$$m_{68} = \{small\} : 0.4, \{small, medium\} : 0.6$$

then

$$\begin{aligned} p(68|m) &= \frac{1}{122}[\frac{m(\{small\})}{pm(\{small\})} m_{68}(\{small\}) \\ &+ \frac{m(\{small, medium\})}{pm(\{small, medium\})} m_{68}(\{small, medium\})] \\ &= \frac{1}{122}[\frac{0.102}{0.041}(0.4) + \frac{0.118}{0.031}(0.6)] = 0.027 \end{aligned}$$

This definition is then extended to the case where your posterior knowledge consists of a set of prototype descriptions.

Definition 12. *(Conditional Densities from Prototype Descriptions) For m_{DB} generated from prototype descriptions $\mathcal{L}(P_k), k = 1, \ldots, c$*

$$\forall \boldsymbol{x} \in \Omega_1 \times \ldots \times \Omega_n \ p(\boldsymbol{x}|m_{DB}) = \sum_{k=1}^{c} \frac{|P_k|}{N} \prod_{j=1}^{n} p(x_j|m_{j,k})$$

6 Fusion of Data Models and Background Knowledge

In this section we consider the case where in addition to the database DB we also have background knowledge provided by one or more experts. The object then is to combine or fuse the background knowledge with the knowledge obtained from DB. For simplicity, we consider only the one-dimensional case in this context (i.e where there is only one attribute x). However, the scaling to multi-dimensions is straightforward.

It is assumed that expert knowledge takes the form of a label expression θ. Hence, to be able to combine such knowledge with that inferred from DB we must first consider how to evaluate conditional mass assignments and densities given θ.

Definition 13. *(Conditional Densities from Label Expressions) Assuming a prior distribution p on Ω then the posterior distribution resulting from conditioning on θ is defined by:*

$$\forall x \in \Omega \ p(x|\theta) = \frac{Pr(\theta|x)p(x)}{\int_\Omega Pr(\theta|x)p(x)dx} = \frac{\mu_\theta(x)p(x)}{\int_\Omega \mu_\theta(x)p(x)dx}$$

Similarly, we can also condition on θ at the mass assignment level.

Definition 14. *(Conditional Mass Assignments given Label Expressions) Given a prior mass assignment pm on 2^{LA} then the posterior mass assignment resulting from conditioning on θ is defined by:*

$$\forall S \subseteq LA \; m_\theta(S) = \frac{pm(S)}{\sum_{S \in \lambda(\theta)} pm(S)} \; if \; S \in \lambda(\theta)$$
$$= 0 \; otherwise$$

Suppose that, in addition to the initial prior distribution p on x, we also have background knowledge in the form of label expression θ and database DB. We now propose two possible approaches by which the expert knowledge and the data can be combined to obtain an estimate for the distribution of x.

Approach 1: Update prior followed by data-analysis
We form an updated prior $p(\bullet|\theta)$ and then evaluate the mass assignment m_{DB} as described above. The updated prior is now incorporated into the estimation process so that,

$$\forall x \in \Omega \; p(x|\theta, DB) = p(x|\theta) \sum_{S \subseteq LA} \frac{m_{DB}(S)}{pm(S|\theta)} m_x(S)$$

where

$$pm(S|\theta) = \int_\Omega p(x|\theta) m_x(S) dx$$

Approach 2: Data-analysis followed by Conditioning
In this case we first evaluate m_{DB} and then condition on θ to give the mass assignment

$$\forall S \subseteq LA \; m(S|DB, \theta) = \frac{m_{DB}(S)}{\sum_{S \in \lambda(\theta)} m_{DB}(S)} \; if \; S \in \lambda(\theta)$$
$$= 0 \; otherwise$$

We now evaluate the distribution from this mass assignment based on the original prior p so that,

$$p(x|DB, \theta) = p(x) \sum_{S \subseteq LA} \frac{m(S|DB, \theta)}{pm(S)} m_x(S)$$

Above we have assumed that the expert knowledge was certain, however, in practice this may not be the case. We now examine two distinct cases. In the first case we have a number of experts each of which provides some prior knowledge. The uncertainty, here, comes from the fact that we have a different level of confidence in each expert. In the second case, we may have one expert who gives us a number of pieces of background information but with each of which

they associate a belief value (i.e. subjective probability). Clearly this is epistemologically distinct from the first case. In both of these cases we examine how we might fuse such information with information from data.

Case 1

Suppose we have c experts and expert j provides us with background information θ_j for $j = 1, \ldots, c$. Further suppose that our confidence level in expert j is represented by a positive real number w_j such that $\sum_{j=1}^{c} w_j = 1$. If, overall, this expert knowledge is represented by K then we have two possibilities:

Approach 1

We determine an updated prior

$$\forall x \in \Omega \; p(x|K) = \sum_{j=1}^{c} w_j p(x|\theta_i)$$

and then integrate this into the estimate from DB to obtain $p(x|K, DB)$

Approach 2

First evaluate m_{DB} and then condition on K as follows:

$$\forall S \subseteq LA \; m(S|DB, K) = \sum_{j=1}^{c} w_j m(S|DB, \theta_j)$$

and then take $p(x|DB, K) = p(x|m(\bullet|DB, K))$.

Case 2

In this case our expert provides us with a set of expressions with associated probability, $\theta_j : \alpha_j, j = 1, \ldots, t$. This situation is more complex than case 1 since here we have partial knowledge regarding a probability distributions across all label expressions LE. Initially, consider the simple case when $t = 1$ where we simply have the information $\theta : \alpha$. In this case we can either evaluate

$$p(x|K) = \alpha p(x|\theta) + (1 - \alpha) p(x|\neg\theta)$$

for approach 1 and for approach 2

$$m(S|DB, K) = \alpha m(S|DB, \theta) + (1 - \alpha) m(S|DB, \neg\theta)$$

The case involving more than one expression, however, is more complex since we do not have information regarding the logical structure of the expressions $\theta_j, j = 1, \ldots, t$ and how they are related to each other. Essentially, we need to determine a distribution on all expressions of the form $\varphi \equiv \bigwedge_{j=1}^{t} \mp\theta_j$ where $+\theta_i$ denotes θ_i and $-\theta_i$ denotes $\neg\theta$, consistent with K. One possibility, is to assume that the probability α_j was evaluated independently of the other expressions $\theta_i, i \neq j$. In this case we would expect:

$$\acute{P}r(\varphi) = \prod_{j=1}^{t} Pr(\mp\theta_j)$$

where $Pr(\theta_j) = \alpha_j$ and $Pr(\neg\theta_j) = 1 - \alpha_j$. However, we must also take into account that the resulting distribution must be consistent with the definition of the labels in LA as given by their appropriateness degrees. Hence, we must define

$$Pr(\varphi) = \frac{Pr'(\varphi)}{1 - \sum_{\varphi:\lambda(\varphi)\cap\mathcal{F}=\emptyset} Pr'(\varphi)} \; if \; \lambda(\varphi) \cap \mathcal{F} \neq \emptyset$$
$$= 0 \; otherwise$$

This is basically the application of dempsters law of combination [11] in this new label semantics framework.
We can now evaluate, for approach 1:

$$p(x|K) = \sum_{\varphi:\lambda(\varphi)\cap\mathcal{F}\neq\emptyset} Pr(\varphi)p(x|\varphi)$$

or for approach 2

$$m(S|DB, K) = \sum_{\varphi:\lambda(\varphi)\cap\mathcal{F}\neq\emptyset} Pr(\varphi)m(S|DB, \varphi)$$

Example 6. Let $LA = \{small, medium, large\}$ with appropriateness degrees defined on $\Omega = [0, 10]$ as follows (see figure 3: $\forall x \in [0, 10]$

$$\mu_{small}(x) = \begin{cases} 1 : x \in [0, 2] \\ 2 - \frac{x}{2} : x \in (2, 4] \\ 0 : otherwise \end{cases}$$

$$\mu_{medium}(x) = \begin{cases} \frac{x}{2} - 1 : x \in [2, 4] \\ 1 : x \in (4, 6] \\ 4 - \frac{x}{2} : x \in (6, 8] \\ 0 : otherwise \end{cases}$$

$$\mu_{large}(x) = \begin{cases} \frac{x}{2} - 3 : x \in [6, 8] \\ 1 : x \in (8, 10] \\ 0 : otherwise \end{cases}$$

Suppose our knowledge K consists of the statements

$$small \wedge medium : 0.7, \; \neg large : 0.4$$

We can now generate the following tableau:

labels	$\neg l : 0.4$	$l : 0.6$
$s \wedge m : 0.7$	$s \wedge m \wedge \neg l : 0.28$	$s \wedge m \wedge l : 0.42$
$\neg(s \wedge m) : 0.3$	$\neg(s \wedge m) \wedge \neg l : 0.12$	$\neg(s \wedge m) \wedge l : 0.18$

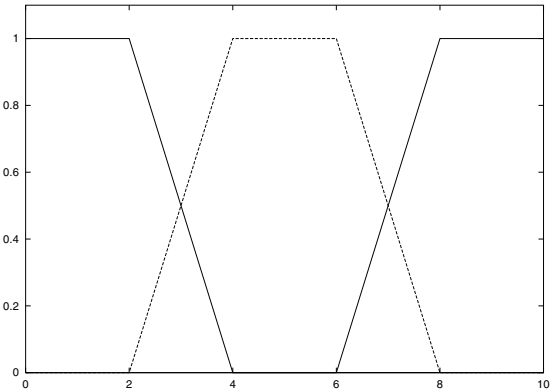


Fig. 3. Appropriateness degrees for, from left to right, *small*, *medium* and *large*

Now recall

$$\mathcal{F} = \{\emptyset, \{small\}, \{small, medium\}, \{medium\}, \{medium, large\}, \{large\}\}$$

From this we see that $\lambda(small \wedge medium \wedge large) \cap \mathcal{F} = \emptyset$. Hence, assuming a uniform prior distribution

$$p(x|K) = \frac{0.28}{1-0.42}p(x|s \wedge m \wedge \neg l) + \frac{0.12}{1-0.42}p(x|\neg(s \wedge m) \wedge \neg l)$$

$$+\frac{0.18}{1-0.42}p(x|\neg(s \wedge m) \wedge l)$$

$$= 0.4828p(x|s \wedge m \wedge \neg l) + 0.2069p(x|\neg(s \wedge m) \wedge \neg l) + 0.3104p(x|\neg(s \wedge m) \wedge l)$$

$$= 0.4828\frac{\mu_{s\wedge m\wedge\neg l}(x)}{\int_0^{10}\mu_{s\wedge m\wedge\neg l}(x)dx} + 0.2069\frac{\mu_{\neg(s\wedge m)\wedge\neg l}(x)}{\int_0^{10}\mu_{\neg(s\wedge m)\wedge\neg l}(x)dx}$$

$$+0.3104\frac{\mu_{\neg(s\wedge m)\wedge l}(x)}{\int_0^{10}\mu_{\neg(s\wedge m)\wedge l}(x)dx}$$

$$= 0.4828\frac{\mu_{s\wedge m\wedge\neg l}(x)}{0.5} + 0.2069\frac{\mu_{\neg(s\wedge m)\wedge\neg l}(x)}{6.5} + 0.3104\frac{\mu_{\neg(s\wedge m)\wedge l}(x)}{3}$$

$$= 0.9656\mu_{s\wedge m\wedge\neg l}(x) + 0.0318307\mu_{\neg(s\wedge m)\wedge\neg l}(x) + 0.1035\mu_{\neg(s\wedge m)\wedge l}(x)$$

Figure 6 shows the resulting density $p(x|K)$.

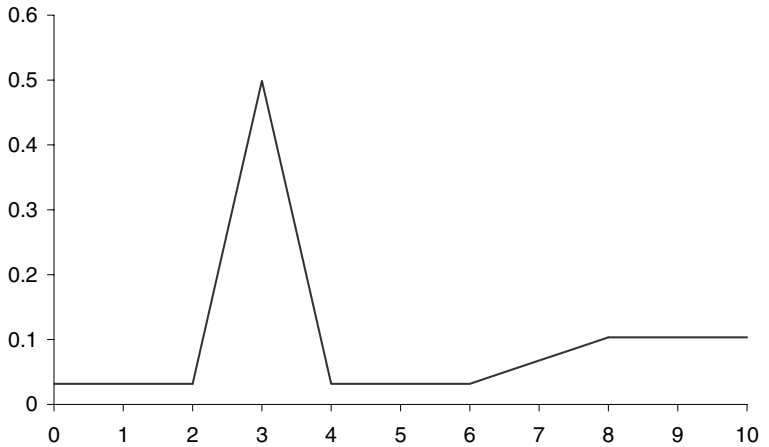


Fig. 4. $p(x|K)$: the density conditional of prior knowledge K

7 Label Models and Linguistic Queries

To illustrate the potential of this framework we shall briefly describe how it can be applied to classification problems. In principle, however, the approach can also be applied to prediction and cluster analysis. Suppose then that the objects of DB can be categorised as belonging to one of the classes $C_1, \ldots, C_t$ and let DB_j denote the subset of DB containing only the elements with class C_j. We can now determine m_{DB_j} on the basis of some partition and evaluate $p(\boldsymbol{x}|m_{DB_j})$ as described above. If we now take $p(\boldsymbol{x}|m_{DB_j})$ as an approximation for $p(\boldsymbol{x}|C_j)$ then from Bayes theorem we have $Pr(C_j|\boldsymbol{x}) \propto p(\boldsymbol{x}|m_{DB_j})|DB_j|$. Given this estimate for each class probability, classification can be then carried out in the normal way. In the limit case when the partition of DB_j has only one set (i.e. DB_j) then this method corresponds to a version of the well known Naive Bayes algorithm [10].

In the context of classification problems we extend the notation for queries as follows: For $\theta \in MLE^{(n)}$

Type I Query: $\theta : C_j$
This represents the question:
Do elements of class C_j satisfy θ?
The support for this query is given by $Pr(\theta|C_j) = \mu_\theta(DB_j)$.

Type II Query: θ
This represents the question:
Do elements of DB satisfy θ?
The support for this query is given by
$Pr(\theta) = \mu_\theta(DB) = \sum_{k=1}^{t} Pr(DB_k)\mu_\theta(DB_k)$.

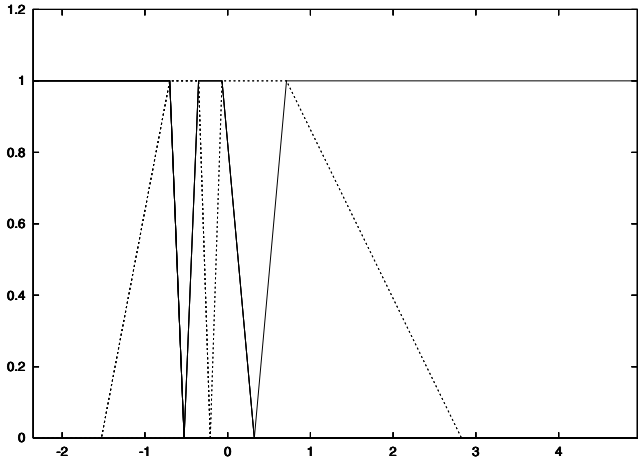


Fig. 5. Non uniform appropriateness degrees for, from left to right, *very small*, *small*, *medium*, *large* and *very large* for attribute 1 generated using a percentile algorithm

Type III Query: $C_j : \theta$
This represents the question:
Do elements satisfying θ belong to class C_j?
The support for this query is given by

$$Pr(C_j|\theta) = \frac{\mu_\theta(DB_j)Pr(DB_j)}{\mu_\theta(DB)}$$

Example 7. The Naive Bayes version of the above algorithm was applied to the UCI repository problem on glass categorisation. The problem has 6 classes and 9 continuous attributes. 5 labels were define for attributes 1-2 and 4-7. Attribute 3 was allocated 4 labels and attributes 8 and 9 were not used as their variance across DB was too low for effective labelling. For all attributes the labels were defined by trapezoidal appropriateness degrees positioned according to a simple percentile method (figure 5). The database of 214 elements was randomly split into a test and training set of 107 elements each and a classification accuracy of 78.5% on the training set and 72.9% on the test set was obtained. This is comparable with other approaches; for example a feed-forward Neural Network with architecture 9-6-6 gives 72% on a smaller test set where the network was trained on 50% of the data, validated on 25% and tested on 25%. The density function for attribute 1 generated from the label prototype for class 1 according to definition 11 is shown in figure 6. Now suppose for the attributes with five labels that these correspond to domain specific versions of *very low*, *low*, *medium*, *high* and *very high* . In the following, the attribute number to which a label refers is shown as a subscript. Consider the following queries:

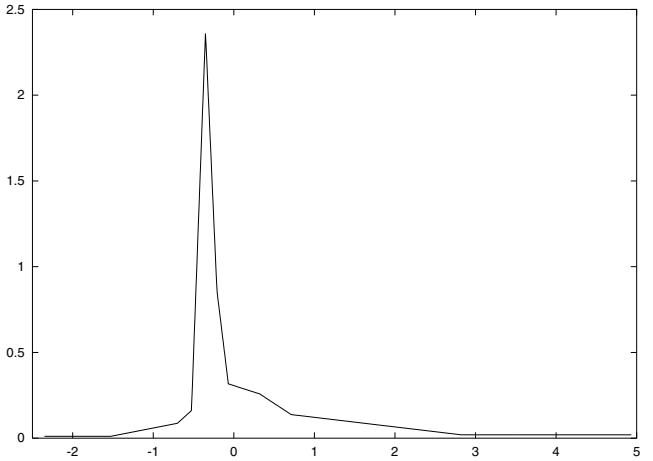


Fig. 6. Density function for attribute 1 conditional on class 1 generated from the mass assignment for attribute 1 in the label description of class 1

Type I Query

What is the probability that float processed building window glass (class 1) has a medium to low or high refractive index (att. 1) and a very low or low sodium concentration (att. 2)?

To answer this query we note that $\mathcal{L}(DB_1) = \langle m_{1,1}, \ldots, m_{n,1} \rangle$ where
$m_{1,1} = \{very\ low_1\} : 0.01373, \{low_1, very\ low_1\} : 0.04342,$
$\{low_1\} : 0.02804, \{low_1, medium_1\} : 0.37391, \{medium_1\} : 0.12208,$
$\{medium_1, high_1\} : 0.08424, \{high_1\} : 0.10059,$
$\{high_1, very\ high_1\} : 0.17233, \{very\ high_1\} : 0.06167.$

In this case the query is given by
$\theta_1 \wedge \theta_2 : C_1$
where $\theta_1 \equiv (medium_1 \wedge low_1) \vee high_1$ and $\theta_2 \equiv very\ low_2 \vee low_2$.
For this query we have that

$$\mu_{\theta_1 \wedge \theta_2}(DB_1) = \sum_{\langle S_1, \ldots S_n \rangle \in \lambda^{(n)}(\theta_1 \wedge \theta_2)} \prod_{i=1}^{n} m_{i,1}(S_i)$$

$$= \left(\sum_{S_1 \in \lambda(\theta_1)} m_{1,1}(S_1) \right) \times \left(\sum_{S_2 \in \lambda(\theta_2)} m_{2,1}(S_2) \right)$$

Now from the training database we have

$$\sum_{S \in \lambda(\theta_1)} m_{1,1}(S_1) = m_{1,1}(\{large, medium\}) + m_{1,1}(\{medium, high\})$$

$$+ m_{1,1}(\{high\}) + m_{1,1}(\{high, very\ high\}) = 0.73107$$

Similarly,

$$\sum_{S \in \lambda(\theta_2)} m_{2,1}(S) = 0.92923$$

so that

$$\mu_\theta(DB_1) = 0.73107 \times 0.92923 = 0.67933$$

this being the required probability.

Type II Query

What is the probability that a glass fragment has a medium to low or high refractive index (att. 1) and a very low or low sodium concentration (att. 2)? This query has the form

$$\theta_1 \wedge \theta_2$$

and the required probability is given by $\mu_{\theta_1 \wedge \theta_2}(DB)$. To evaluate this we note that for classes $C_1, \ldots, C_6$ the probabilities of each class satisfying θ are given by $\mu_{\theta_1 \wedge \theta_2}(DB_1) = 0.67933$, $\mu_{\theta_1 \wedge \theta_2}(DB_2) = 0.18429$, $\mu_{\theta_1 \wedge \theta_2}(DB_3) = 0.09036$, $\mu_{\theta_1 \wedge \theta_2}(DB_4) = 0.42108$, $\mu_{\theta_1 \wedge \theta_2}(DB_5) = 0$, $\mu_{\theta_1 \wedge \theta_2}(DB_6) = 0.02173$.
Also the number of data elements in $DB_1, \ldots, DB_6$ are respectively $35, 38, 8, 7, 4$ and 15. From this we can evaluate: $\mu_{\theta_1 \wedge \theta_2}(DB) = \frac{1}{107}(0.67933(35) + 0.18429(38) + 0.09036(8) + 0.42108(7) + 0.02173(15)) = \frac{34.7760}{107} = 0.32501$.

Type III Query

What is the probability that a glass fragment with a medium to low or high refractive index (att. 1) and a very low or low sodium concentration (att. 2) is a fragment of float processed building window glass (class 1)?
For this query has the form

$$C_1 : \theta_1 \wedge \theta_2$$

and the required probability $Pr(C_1|\theta_1 \wedge \theta_2)$ is given by:

$$Pr(C_1|\theta_1 \wedge \theta_2) = \frac{\mu_{\theta_1 \wedge \theta_2}(DB_1) Pr(DB_1)}{\mu_{\theta_1 \wedge \theta_2}(DB)}$$

$$= \frac{0.67933 \frac{35}{107}}{0.32501} = 0.6837$$

Probabilistic approaches to classification often suffer from the so-called 'curse of dimensionality'. In simple terms this is due to the fact that the amount of data required to estimate a joint probability distribution is exponential in the dimension of that distribution. The current approach based on label prototypes can overcome this problem in a number of ways. Firstly models are based on a number of prototypes for each of which we assume independence between attributes. Clearly, this means that only one dimensional marginal distributions need to be inferred from different subsets of the database. Secondly, the inherent interpolation obtain from the use of labels defined in terms of graded appropriateness degrees allow us to estimate distributions on the basis of a small number of partition granules across each attribute universe. The next example illustrates the potential of this framework for modelling at least moderately high dimensional databases.

Table 1. Table of classification accuracy for training (upper value) and test (lower value) sets for varying numbers of prototypes. The number of prototypes for cylinders are listed horizontally and the number of prototypes for rocks are listed vertically

	1	2	3	4
1	88.46%	90.38%	91.35%	94.23%
	82.69%	83.65%	81.73%	80.77%
2	81.73%	97.12%	97.12%	98.08%
	74.04%	84.62%	84.62%	83.65%
3	82.69%	97.12%	97.12%	98.08%
	72.12%	82.69%	83.65%	83.65%
4	83.65%	97.12%	98.08%	100%
	75%	90.39%	88.46%	90.39%

Example 8. This database is also taken from the UCI repository and was originally used by Gorman and Sejnowski [7] as part of a study into classifying sonar signals using Neural Networks. The objective is to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock. The database consists of 208 patterns, 97 of which are from rocks and 111 from metal cylinders. The training and test sets used for this problem both contain 104 patterns and are identical to those used in [7]. The data is split in such a way as to take account of aspect-angle dependencies.

Each pattern consists of the values of 60 real valued attributes for a particular object where each attribute represents the energy within a particular frequency band, integrated over a certain period of time. Label sets containing 3 words were defined for each attribute where the associated appropriateness degrees were trapezoidal functions derived from a percentile algorithm as in the previous example. The c-means [3] algorithm, was used to partition both the sub-databases of cylinder and rock patterns with varying values of c for each class. Linguistic descriptions of these partitions generated a number of prototype models for both cylinders and rocks. Unlike the glass problem this problem displays significant decomposition error as can be seen from table 1 showing the results for the training and test sets for varying numbers of prototypes.

The difference in classification accuracy between the model consisting of 1 prototype per class and that consisting of 4 prototypes per class (in this case the optimal) is 11.5% on the training set and 7.7% on the test set. However, the overall accuracy, on the test set in particular, is very sensitive to the number of prototypes for both classes. For instance, it is not simply the case that accuracy is a monotonically increasing function of the number of prototypes for each class. To see this notice that holding the number of prototypes for the cylinder class constant at 1 and increasing the number of prototypes for rocks to 4 brings about an overall decrease in classification accuracy of 7.79%. Clearly then the automated learning of optimal prototype numbers for each class remains an open

and difficult problem. In the meantime one simplistic solution could be to set an upper bound on prototype numbers and carry out an exhaustive search of the relevant tableau of classification accuracy values on a separate validation set.

Interestingly, these results are comparable with Gorman and Sejnowski [7] who experimented with a back propagation neural network with 60 inputs and up to 24 hidden nodes. In fact their optimal solution was obtained using a network with 12 hidden nodes which gave an accuracy of 90.4% on the test set. Also of interest are recent results obtain by Freiβ, Christianini and Campbell [4] using a SVM where an accuracy of 95.2% was obtained.

8 Conclusions

A framework for modelling with labels has been introduced based on random sets. Labels are defined in terms of appropriateness degrees quantifying their suitability to describe values from the underlying universe. Such appropriateness degrees can then be extended to compound expressions generated by recursive application of logical connectives to the basic labels.

A description of a database can be defined in this framework as a tuple of mass assignments on appropriate label sets. Such descriptions assume independence between attribute labels and in cases when this is not appropriate the database must be partitioned into subsets of elements sufficiently similar to allow such an assumptions. The label descriptions of these subsets can be view as linguistic prototypes each representing an amalgam of points.

Label prototypes can be fused with background knowledge in the form of label expressions. This knowledge may be uncertain, either because it is collected from a number of experts with varying reliability or because it is given in terms of differing levels of subjective belief by a single expert. In both cases, we investigate two approaches to fusion. The first involves inferring a new posterior density from the background knowledge and then integrating it into the learning process. In second approach we infer a mass assignment from the database and then condition this given the background knowledge.

The potential of this framework has been demonstrated by its application to classification and query evaluation. Both of the example problems considered were benchmark problems taken from the UCI machine learning repository. The sonar database has sixty attributes and the results suggest that this framework can scale to at least relatively high dimensional problems.

Future work will focus on the development of the linguistic querying algorithm to allow for the evaluation of more complex queries. Also to be investigated are potential heuristics for automatically learning the number of prototypes in classification problems. Finally, it is intended to consider applications of this framework to unsupervised learning and to prediction/regression problems.

Acknowledgements

This research is partially funded by EPSRC grant number GR/M89126/01.

References

[1] D. Dubois, H. Prade, 'Three Semantics of Fuzzy Sets' *Fuzzy Sets and Systems*, Vol. 90 (1997) pp141-150 187
[2] D. Dubois, F. Esteva, L. Godo, H. Prade, 'An Information-Based Discussion of Vagueness' it Proceedings of FUZZ-IEEE 2001, Vol.2, (2001) 187
[3] R. Duda, P. Hart, *Pattern Classification and Scene Analysis* Wiley, New York (1973) 206
[4] Th. Frieß, N. Cristianini, C. Campbell 'The Kernel-Adatron: A Fast and Simple Procedure for Support Vector Machines', *Proceedings of the Fifteenth International Conference on Machine Learning* ,(1998), pp188-196 207
[5] 'Fuzzy Sets as Equivalence Classes of Random Sets' in *Fuzzy Set and Possibility Theory* (ed. R. Yager), (1982) pp327-342 187
[6] I. R. Goodman, H. T. Nguyen, *Uncertainty Models for Knowledge Based Systems* North Holland (1985) 187
[7] R. P. Gorman, T. J. Sejnowski, 'Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets' *Neural Networks*, Vol.1, (1988), pp75-89 206, 207
[8] J. Lawry, 'Label Prototypes for Modelling with Words' *Proceedings of The North American Fuzzy Information Processing Society 2001 Conference*, (2001) 196
[9] J. Lawry, 'Label Semantics: A Formal Framework for Modelling with Words' *Lecture Notes in Artificial Intelligence*, **2143** (ed. S. Benferhat, P. Besnard), (2001) 374–384 192
[10] D. D. Lewis, 'Naive Bayes at Forty: The independence Assumption in Information Retrieval' *Lecture Notes in Artificial Intelligence*, **1398** (1998) 4–15 202
[11] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton (1976) 200
[12] L. A. Zadeh, 'The Concept of Linguistic Variable and its Application to Approximate Reasoning' PartI-III, *Information Sciences* Vols. 8-9, (1975-76),pp119-429, pp301-357, pp43-80 191
[13] L. A. Zadeh 'Fuzzy Logic=Computing with Words' *IEEE Transactions on Fuzzy Systems*, Vol. 2, (1996),pp103-111 187

Interpretability Issues in Fuzzy Genetics-Based Machine Learning for Linguistic Modelling

Hisao Ishibuchi and Takashi Yamamoto

Department of Industrial Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan
{hisaoi,yama}@ie.osakafu-u.ac.jp

Abstract. This chapter discusses several issues related to the design of linguistic models with high interpretability using fuzzy genetics-based machine learning (GBML) algorithms. We assume that a set of linguistic terms has been given for each variable. Thus our modelling task is to find a small number of fuzzy rules from possible combinations of the given linguistic terms. First we formulate a three-objective optimization problem, which simultaneously minimizes the total squared error, the number of fuzzy rules, and the total rule length. Next we show how fuzzy GBML algorithms can be applied to our problem in the framework of multi-objective optimization as well as single-objective optimization. Then we point out a possibility that misleading fuzzy rules can be generated when general and specific fuzzy rules are simultaneously used in a single linguistic model. Finally we show that non-standard inclusion-based fuzzy reasoning removes such an undesirable possibility.

1 Introduction

Since Takagi & Sugeno's pioneering work [33], fuzzy modelling has been extensively studied [27]. In the 1990s, many approaches were proposed for fuzzy modelling such as heuristic methods [26, 35], fuzzy-neuro methods [9, 19, 32], and genetic fuzzy methods [2, 7, 24] where emphasis was primarily placed on the improvement in the accuracy of fuzzy models. The interpretability of fuzzy models was also discussed in some studies [29, 31, 34]. Recently the existence of a tradeoff between the accuracy and the interpretability of fuzzy models was recognized [3] and taken into account in many studies on fuzzy modelling [22, 23, 28, 30]. While multiple criteria were simultaneously considered in the design of fuzzy models in those studies, fuzzy modelling was handled in the framework of single-objective optimization. That is, the final goal in those studies was to design a single fuzzy model with high accuracy and high interpretability. The handling of the design of fuzzy models in the framework of multi-objective optimization was first proposed for fuzzy rule-based classification in [11] where the goal was not to find a single fuzzy model but to find multiple non-dominated fuzzy models with respect to the classification accuracy and the number of

J. Lawry, J. Shanahan, A. Ralescu (Eds.): Modelling with Words, LNAI 2873, pp. 209-228, 2003.
© Springer-Verlag Berlin Heidelberg 2003

fuzzy rules. The two-objective formulation in [11] was extended to the case of three-objective optimization in [13, 14] where the total rule length was used as the third objective. Jimenez et al. [20, 21] discussed multi-objective optimization of Takagi-Sugeno models where the accuracy, the transparency and the compactness were considered. Since Takagi-Sugeno models have a linear function in the consequent part of each fuzzy rule, their linguistic interpretability is not high. Thus we use more descriptive fuzzy rules with linguistic terms in both the antecedent and consequent parts (i.e., Mamdani rules).

Let us assume that we have m input-output pairs $(\mathbf{x}_p, y_p)$, $p = 1,2,...,m$ for an n-input and single-output unknown nonlinear function where $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$ is an n-dimensional input vector and y_p is the corresponding output value. We also assume that a set of linguistic terms has been given by domain experts or human users for each variable. For simplicity of explanation, we use five linguistic terms in Fig. 1 for all the input and output variables. Our task is to linguistically describe the unknown nonlinear function using fuzzy rules of the following form:

$$\text{Rule } R_k : \text{If } x_1 \text{ is } A_{k1} \text{ and } ... \text{ and } x_n \text{ is } A_{kn} \text{ then } y \text{ is } B_k, \quad (1)$$

where R_k is the label of the kth fuzzy rule, x_i is the ith input variable, A_{ki} is an antecedent fuzzy set on the ith input variable x_i, y is the output variable, and B_k is a consequent fuzzy set. The consequent fuzzy set B_k is one of the five linguistic terms in Fig. 1 while the antecedent fuzzy set A_{ki} can assume *don't care* in addition to the five linguistic terms. Thus the total number of fuzzy rules of the form in (1) is $(5+1)^n \cdot 5$. We do not modify the membership function of each linguistic term because the modification usually degrades the interpretability of fuzzy rules. Thus the design of a linguistic model can be viewed as finding a subset of $(5+1)^n \cdot 5$ fuzzy rules. The size of the search space is 2^N where $N = (5+1)^n \cdot 5$.

In the next section, we formulate our modelling task as a three-objective optimization problem. The three objectives are to minimize the total squared error, the number of fuzzy rules, and the total rule length. The rule length is defined by the number of antecedent conditions. In Section 3, we show the handling of our modelling task in the framework of single-objective optimization where the weighted sum of the three objectives is used as a scalar fitness function. A Pittsburgh-style fuzzy GBML algorithm is used for finding a single linguistic model. In Section 4, we show the handling of our modelling task in the framework of multi-objective optimization. The single-objective fuzzy GBML algorithm is extended using multi-objective genetic algorithms [4, 6]. Our modelling task is tackled by genetic rule selection in Section 5 where we also explain heuristic fuzzy rule generation using rule evaluation measures in data mining [1, 8, 18]. In Section 6, we point out a possibility that misleading fuzzy rules can be extracted when linguistic models include both general and specific fuzzy rules. After explaining why misleading fuzzy rules are extracted, we show that the use of a non-standard fuzzy reasoning method [10] removes such an undesirable possibility. Section 7 concludes this chapter.

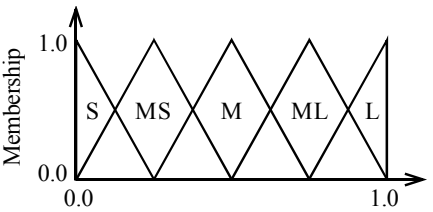


Fig. 1. Membership functions of five linguistic terms (S: *small*, MS: *medium small*, M: *medium*, ML: *medium large*, and L: *large*)

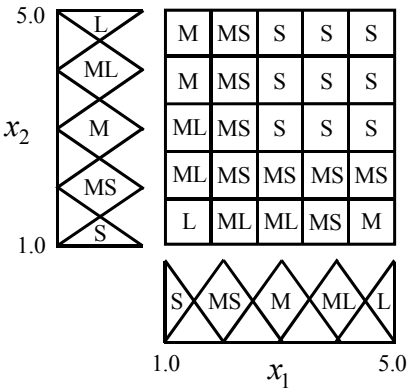


Fig. 2. A fuzzy rule table that linguistically describes the nonlinear function in (2)

2 Formulation of Linguistic Modelling

First we explain the basic idea of linguistic modelling using the following two-input and single-output nonlinear function [33]:

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \le x_i \le 5 \text{ for } i = 1,2. \tag{2}$$

Nozaki et al. [26] extracted 25 linguistic rules in Fig. 2 where the five linguistic terms in Fig. 1 are used as consequent fuzzy sets. While it is not easy to intuitively understand the shape of the nonlinear function from the mathematical description in (2), we can easily grasp a rough three-dimensional shape of the nonlinear function from the linguistic description shown in Fig. 2. Thus the fuzzy rule table in Fig. 2 is an interpretable linguistic model of the nonlinear function in (2). As shown in this example, two-input and single-output nonlinear functions can be linguistically described by fuzzy rule tables in a human understandable manner.

The main difficulty in the application of fuzzy rule tables to high-dimensional nonlinear functions is the exponential increase in the number of fuzzy rules, which is often referred to as the curse of dimensionality. Let K be the number of linguistic terms for each input variable (e.g., $K = 5$ in Fig. 2), the number of fuzzy rules in an n-dimensional fuzzy rule table is K^n. Thus the interpretability of fuzzy rule tables is

severely deteriorated by the increase in the number of input variables. For example, let us consider the following nonlinear function [12]:

$$y = \frac{1}{2\left(1+\exp\{\sum_{i=1}^{3}(-60x_i+55)\}\right)}, \quad 0 \le x_i \le 1 \text{ for } i = 1,2,3. \tag{3}$$

When we have the five linguistic terms in Fig. 1 for all the three input and single output variables, we can easily generate 125 fuzzy rules using a heuristic method (e.g., [26, 35]). It is, however, not easy for human users to understand the nonlinear function from the generated 125 fuzzy rules. This is because the number of the generated fuzzy rules is too large. It should be noted that the understanding of the nonlinear function from the mathematical description in (3) is also difficult.

Even when the number of fuzzy rules is small, linguistic models are not always interpretable. Another difficulty in the handling of high-dimensional problems is the increase in the rule length. It is not easy for human users to intuitively understand long fuzzy rules with many antecedent conditions. Thus the length of each fuzzy rule should be small when we design linguistic models with high interpretability. In this chapter, the number of antecedent conditions of each fuzzy rule is referred to as the rule length. For generating short fuzzy rules for high-dimensional problems, we use "*don't care*" as an additional antecedent fuzzy set. Since *don't care* is fully compatible with any input values, its membership function is defined as

$$\mu_{don't\ care}(x) = 1 \text{ for } \forall x. \tag{4}$$

Since *don't care* conditions are usually omitted from the antecedent part, fuzzy rules with many *don't care* conditions are short and interpretable. As an example, let us consider the following fuzzy rule:

If x_1 is *don't care* and x_2 is *don't care* and x_3 is *large*
then y is *medium large*. (5)

We omit the two *don't care* conditions as

If x_3 is *large* then y is *medium large*. (6)

Short and long fuzzy rules are referred to as general and specific rules, respectively.

The use of *don't care* is also supported from the viewpoint of the number of fuzzy rules required for covering the whole input space. As we can see from Fig. 1, each linguistic term covers the following fraction of the domain interval [0, 1] of each input variable:

small: 1/4 $(0 \le x_i < 0.25)$, *medium small*: 1/2 $(0 < x_i < 0.5)$,
medium: 1/2 $(0.25 < x_i < 0.75)$, *medium large*: 1/2 $(0.5 < x_i < 1)$,
large: 1/4 $(0.75 < x_i \le 1)$.

Thus we can see that each linguistic term covers on average 2/5 of the domain interval [0, 1] where

$$\frac{2}{5}=\frac{1}{5}\times\left(\frac{1}{4}\times 2+\frac{1}{2}\times 3\right). \tag{7}$$

Since each fuzzy rule has n antecedent conditions, it covers on average $(2/5)^n$ of the n-dimensional input space $[0, 1]^n$ if no *don't care* conditions are included. That is, the fraction covered by each fuzzy rule is exponentially decreased by the increase in the dimensionality of the input space. The minimum number of fuzzy rules required for covering the whole input space is roughly estimated as $(5/2)^n$. This becomes huge in the case of high-dimensional problems. For example, $(5/2)^n$ is 9537 for $n = 10$ and about 91 million for $n = 20$. This discussion clearly shows the necessity of *don't care* conditions when we try to linguistically describe high-dimensional nonlinear functions. General fuzzy rules with many *don't care* conditions can cover a large portion of the input space. Thus the whole input space can be covered by a small number of general fuzzy rules. For example, the following two fuzzy rules were generated in [12] for the nonlinear function in (3).

y is *small*, (8)

If x_1 is *large* and x_2 is *large* and x_3 is *large* then y is *medium*. (9)

The first fuzzy rule has no antecedent conditions (i.e., it has *don't care* conditions on all the three input variables). The whole input space $[0, 1]^3$ is covered by these two fuzzy rules (Actually it is covered by the first fuzzy rule). We can easily grasp a rough shape of the nonlinear function in (3) from the two fuzzy rules in (8)-(9).

A linguistic model with only a small number of general fuzzy rules has high interpretability. If the approximation accuracy is also high, we may be able to correctly understand the nonlinear function from the linguistic model. On the other hand, the linguistic model is unreliable if its approximation accuracy is very low. Thus not only the interpretability but also the approximation accuracy should be high when we design a linguistic model for linguistically describing a nonlinear function.

When we use K linguistic terms and *don't care* in the antecedent part and K linguistic terms in the consequent part, the total number of possible fuzzy rules is $(K+1)^n \cdot K$. Let S be a subset of those fuzzy rules. Our linguistic modelling task is formulated as a three-objective combinatorial optimization problem where the following objectives are to be minimized:

$f_1(S)$: The total squared error by the rule set S.
$f_2(S)$: The number of fuzzy rules in the rule set S.
$f_3(S)$: The total rule length of fuzzy rules in the rule set S.

A similar three-objective problem was formulated for fuzzy rule-based classification in [13, 14]. It should be noted that the third objective is not the average rule length but the total rule length. This is because the average rule length does not appropriately measure the complexity of linguistic models. For example, let us consider a linguistic model with three fuzzy rules of the average length 3. If we

include an additional fuzzy rule of the length 1, the average rule length is decreased from 3 to 2.5 while the actual complexity of the linguistic model is increased.

The first objective is calculated from the difference between the actual output value y_p and the estimated output value $\hat{y}(\mathbf{x}_p)$. The latter is calculated as

$$\hat{y}(\mathbf{x}_p) = \frac{\sum_{R_k \in S} \mu_{\mathbf{A}_k}(\mathbf{x}_p) \cdot b_k}{\sum_{R_k \in S} \mu_{\mathbf{A}_k}(\mathbf{x}_p)}, \tag{10}$$

where $\mu_{\mathbf{A}_k}(\mathbf{x}_p)$ is the compatibility grade of the antecedent part $\mathbf{A}_k = (A_{k1}, ..., A_{kn})$ of the fuzzy rule R_k with the input vector $\mathbf{x}_p$, and b_k is a representative real number of the consequent fuzzy set B_k. As the representative real number of each linguistic term in Fig. 1, we use the center of its triangular membership function (i.e., 0.0 for *small*, 0.25 for *medium small*, 0.5 for *medium*, 0.75 for *medium large*, and 1.0 for *large*). The compatibility grade $\mu_{\mathbf{A}_k}(\mathbf{x}_p)$ is calculated by the product operation as

$$\mu_{\mathbf{A}_k}(\mathbf{x}_p) = \mu_{A_{k1}}(x_{p1}) \cdot \mu_{A_{k2}}(x_{p2}) \cdot \ ... \ \cdot \mu_{A_{kn}}(x_{pn}), \tag{11}$$

where $\mu_{A_{ki}}(\cdot)$ is the membership function of the antecedent fuzzy set A_{ki}.

The total squared error over the m input-output pairs $(\mathbf{x}_p, y_p)$, $p = 1,2,...,m$ is calculated from the actual output value y_p and the estimated output value $\hat{y}(\mathbf{x}_p)$ as

$$f_1(S) = \frac{1}{2} \sum_{p=1}^{m} | \hat{y}(\mathbf{x}_p) - y_p |^2. \tag{12}$$

If there is no compatible fuzzy rule with the input vector $\mathbf{x}_p$, the estimated output value $\hat{y}(\mathbf{x}_p)$ cannot be calculated by (10). In this case, we use a pre-specified large penalty value as the corresponding squared error. In our computer simulations, we specify the penalty value as $| \hat{y}(\mathbf{x}_p) - y_p |^2 = 1$ because the range of the output variable is the unit interval [0, 1] in numerical examples of this chapter.

3 Single-Objective Fuzzy GBML Algorithm

3.1 Problem Specification

When our three-objective linguistic modelling problem is handled in the framework of single-objective optimization, a scalar fitness function is defined from the three objectives. We use the following weighted sum of the three objectives as the scalar fitness function to be maximized in fuzzy GBML algorithms:

$$f(S) = -w_1 \cdot f_1(S) - w_2 \cdot f_2(S) - w_3 \cdot f_3(S), \tag{13}$$

where w_i is a user-definable positive weight for the ith objective $f_i(S)$, $i = 1,2,3$. Our three-objective linguistic modelling problem is reduced to the task of finding the optimal rule set that maximizes the scalar fitness function in (13). In this case, the obtained optimal rule set totally depends on the specification of the three weights.

3.2 Pittsburgh-Style Fuzzy GBML Algorithm

Many fuzzy GBML algorithms can be classified into two categories: Michigan Approach and Pittsburgh Approach (see [5] for various fuzzy GBML algorithms). Each fuzzy rule is handled as an individual in Michigan-style algorithms while a set of fuzzy rules is handled as an individual in Pittsburgh-style algorithms. In general, Michigan-style algorithms need much less computational load than Pittsburgh-style algorithms. The optimization of rule sets is indirectly executed through the evolution of fuzzy rules in Michigan-style algorithms while rule sets are directly optimized in Pittsburgh-style algorithms through the evolution of rule sets.

Since the scalar fitness function in (13) involves the minimization of the number of fuzzy rules, the application of Michigan-style algorithms is difficult. This is because the minimization of the number of fuzzy rules means the minimization of the population size in Michigan-style algorithms. Thus we use a Pittsburgh-style algorithm. The outline of our Pittsburgh-style algorithm is written as follows:

[Outline of Pittsburgh-Style Fuzzy GBML Algorithm]

Step 1: Randomly generate a number of rule sets as an initial population.
Step 2: Repeat the following procedures for generating new rule sets.
  (a) Select a pair of parent rule sets from the current population.
  (b) Generate a new rule set from the selected pair by a crossover operation.
  (c) Apply mutation operations to the generated rule set.
Step 3: Update the current population using the newly generated rule sets.
Step 4: If a pre-specified stopping condition is not satisfied, return to Step 2.

In our fuzzy GBML algorithm, the fuzzy rule R_k in (1) is coded by its n antecedent and single consequent fuzzy sets as $R_k = A_{k1}A_{k2}\cdots A_{kn}B_k$. A rule set S is represented by a concatenated string where each substring of the length $(n+1)$ corresponds to a single fuzzy rule. Initial rule sets are generated by randomly assigning a linguistic term or *don't care* to A_{ki} and a linguistic term to B_k.

From the current population, two parent rule sets are selected according to their fitness values. We use the binary tournament selection where two rule sets are randomly drawn with replacement from the current population and the better one with the higher fitness value is chosen as a parent. The binary tournament selection is iterated for selecting a pair of parent rule sets.

Since the number of fuzzy rules is minimized in our fuzzy GBML algorithm, the string length is not fixed. The number of fuzzy rules in each rule set is modified by a crossover operation, which generates a new string whose length is different from its parent strings. We use a kind of one-point crossover with different cutoff points illustrated in Fig. 3 where R_k denotes a substring of the length $(n+1)$. One of the two children in Fig. 3 is randomly selected as a new rule set while it is also possible to

use both children. The crossover operation is applied to each pair of selected parents with a pre-specified crossover probability. When the crossover operation is not applied, one of the two parents is handled as a new rule set. The crossover operation in Fig. 3 can be viewed as a special form of the cut and splice crossover used in messy genetic algorithms (see [5] for details of the cut and splice crossover). For efficiently searching for compact rule sets, we use a heuristic procedure after the crossover operation. The heuristic procedure imposes an upper bound on the number of fuzzy rules in each rule set. In our computer simulations, only the first 20 fuzzy rules from the left of each string are used and the other rules are removed from the string when the number of fuzzy rules exceeds 20.

A mutation operation is applied with a pre-specified mutation probability after the crossover operation. Our mutation operation randomly replaces each antecedent (and consequent) fuzzy set with another one. It should be noted that *don't care* is used only in the antecedent part. We also use a different kind of mutation, which randomly removes each fuzzy rule from the rule set with a pre-specified probability. We can also use heuristic-based mutation operations for improving the search ability of our fuzzy GBML algorithm. For example, the consequent fuzzy set of each fuzzy rule is probabilistically replaced with more appropriate one using compatible input-output pairs with its antecedent part. Moreover a new fuzzy rule can be directly generated from an input-output pair with the largest error in a heuristic manner and added to a rule set. See [16] for details of these heuristic-based mutation operations.

Let N_{pop} be the population size. The selection, crossover and mutation are iterated for generating $(N_{\text{pop}} - 1)$ rule sets as a new population. The best rule set with the largest fitness value in the current population is added to the generated new population with no modifications as an elite rule set.

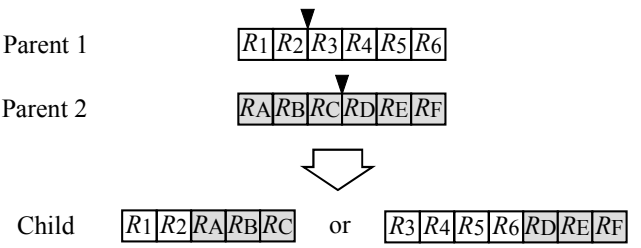


Fig. 3. A kind of one-point crossover with different cutoff points

4 Three-Objective Fuzzy GBML Algorithm

4.1 Problem Specification

Our task is to find all non-dominated rule sets (i.e., Pareto-optimal solutions) with respect to the three objectives when linguistic modelling is handled in the framework of multi-objective optimization. First we briefly describe the concept of Pareto-optimality. A rule set S_A is said to dominate another rule set S_B (i.e., S_A is better than S_B : $S_A \prec S_B$) if all the following three inequalities hold:

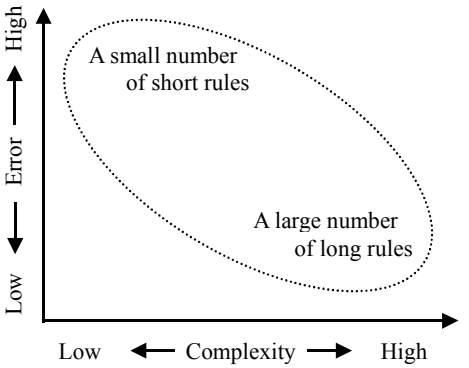


Fig. 4. Illustration of the tradeoff between the error and the complexity of rule sets

$$f_1(S_A) \le f_1(S_B), \quad f_2(S_A) \le f_2(S_B), \quad f_3(S_A) \le f_3(S_B), \tag{14}$$

and at least one of the following three inequalities holds:

$$f_1(S_A) < f_1(S_B), \quad f_2(S_A) < f_2(S_B), \quad f_3(S_A) < f_3(S_B). \tag{15}$$

The first condition (i.e., all the three inequalities in (14)) means that no objective of S_A is worse than S_B. The second condition (i.e., one of the three inequalities in (15)) means that at least one objective of S_A is better than S_B. When a rule set S is not dominated by any other rule sets, S is said to be a Pareto-optimal solution with respect to the three objectives. Our three-objective linguistic modelling problem is to find all Pareto-optimal solutions. Since there exists a tradeoff between the accuracy and the complexity of linguistic models [3], our linguistic modelling problem has many Pareto-optimal solutions with different accuracy and different complexity. The tradeoff between the error and the complexity of rule sets is illustrated in Fig. 4.

4.2 Multi-objective Fuzzy GBML Algorithm

Our Pittsburgh-style fuzzy GBML algorithm in the previous section can be extended to the case of three-objective optimization as in [13, 14] for fuzzy rule-based pattern classification. Recently many multi-objective genetic algorithms (MOGAs) have been proposed [4, 6] together with various performance measures [25]. Since most MOGAs are general-purpose search algorithms, they can be used for finding Pareto-optimal solutions of our three-objective linguistic modelling problem.

The main difference between single-objective and multi-objective fuzzy GBML algorithms is the fitness calculation for each rule set. The fitness calculation was very simple in the previous section because the three objectives were integrated into the scalar fitness function using the user-definable weight values. On the other hand, we do not assume any *a priori* knowledge about the relative importance of the three objectives in this section. Thus the fitness value of each rule set is calculated based on the Pareto-dominance relation defined by (14)-(15). Larger fitness values are usually assigned to non-dominated rule sets than dominated ones. For maintaining the

diversity of solutions (i.e., finding a variety of Pareto-optimal solutions), the concept of fitness sharing or crowding is also used in the fitness calculation. A larger fitness value is usually assigned to a rule set that is less similar to other rule sets. The concept of elite solutions should be modified in multi-objective optimization. While the best solution with the largest fitness value was used as an elite solution in the previous section, each non-dominated solution in the current population can be viewed as an elite solution. For various implementations of MOGAs, see [4, 6].

Some MOGAs have a secondary population where tentative non-dominated solutions are stored separately from the current population. The secondary population is updated by comparing it with the current population in every generation. When we use a MOGA with a secondary population, the outline of our single-objective fuzzy GBML algorithm in the previous section is extended to the case of multi-objective optimization as follows:

[Outline of Pittsburgh-Style Multi-Objective Fuzzy GBML Algorithm]

Step 1: Randomly generate a number of rule sets as an initial population. A copy of each non-dominated rule set in the initial population is included in the secondary population.
Step 2: Repeat the following procedures for generating new rule sets.
(a) Select a pair of parent rule sets from the current population.
(b) Generate a new rule set from the selected pair by a crossover operation.
(c) Apply mutation operations to the generated rule set.
Step 3: Update the secondary population using the newly generated rule sets in Step 2. Generate a new population using the current population, the newly generated rule sets, and the secondary population.
Step 4: If a pre-specified stopping condition is not satisfied, return to Step 2.

When the execution is terminated, non-dominated rule sets stored in the secondary population are presented to human users as solutions of the three-objective linguistic modelling problem. Those rule sets are used for examining the tradeoff between the accuracy and the interpretability of linguistic models. When a single linguistic model should be chosen, the choice depends on the preference of human users. In general, the choice of a single linguistic model from multiple non-dominated ones is much easier than the pre-specification of the weight value to each objective.

5 Genetic Rule Selection

5.1 Basic Idea of Genetic Rule Selection

The design of a linguistic model for an n-input and single-output nonlinear function can be viewed as finding a subset of $(K+1)^n \cdot K$ fuzzy rules where K is the number of linguistic terms given for each variable. When n is small, we can handle linguistic modelling as a rule selection problem where a small number of fuzzy rules are selected from $(K+1)^n \cdot K$ candidate rules. Single-objective and multi-objective genetic algorithms are directly applicable to such a rule selection problem because

each rule set is naturally represented by a binary string of the length $(K+1)^n \cdot K$. The size of the search space is 2^N where $N=(K+1)^n \cdot K$.

Genetic rule selection was originally proposed for fuzzy rule-based classification by Ishibuchi et al. [15] where the weighted sum of the classification accuracy and the number of fuzzy rules was used as a fitness function. Their study was extended to two-objective rule selection in [11] and three-objective rule selection in [13, 14]. Since the number of candidate rules exponentially increases with the number of input variables, the computational load and the memory storage for genetic rule selection also exponentially increase. As a result, genetic rule selection is much slower than fuzzy GBML algorithms as shown in [16] except for the case of low-dimensional problems. Moreover, it is impractical to use all the $(K+1)^n \cdot K$ fuzzy rules as candidate rules when the number of input variables is large (i.e., when *n* is large).

5.2 Heuristic Rule Generation Using Data Mining Criteria

When the number of input variables is small, we can use all the $(K+1)^n \cdot K$ fuzzy rules as candidate rules in genetic rule selection. On the other hand, we need some prescreening procedure of candidate rules in the application of genetic rule selection to high-dimensional problems. We proposed the use of heuristic rule evaluation criteria for candidate rule prescreening in [17] for fuzzy rule-based classification. More specifically, two rule evaluation measures (i.e., *support* and *confidence*) were employed for evaluating fuzzy rules. The proposed idea can be also used for linguistic modelling. The two rule evaluation measures, which were originally used for evaluating association rules in the area of data mining [1], were extended to the case of fuzzy rules in [8, 18].

The confidence $c(R_k)$ of the fuzzy rule R_k in (1) is defined using the given *m* input-output pairs $(\mathbf{x}_p, y_p)$, $p=1,2,...,m$ as

$$c(R_k)=\frac{\sum_{p=1}^{m}\mu_{\mathbf{A}_k}(\mathbf{x}_p)\cdot\mu_{B_k}(y_p)}{\sum_{p=1}^{m}\mu_{\mathbf{A}_k}(\mathbf{x}_p)}, \tag{16}$$

where $\mu_{\mathbf{A}_k}(\mathbf{x}_p)$ is the compatibility grade of the input vector $\mathbf{x}_p$ with the antecedent part $\mathbf{A}_k=(A_{k1},...,A_{kn})$ of the fuzzy rule R_k, and $\mu_{B_k}(y_p)$ is the compatibility grade of the output value y_p with the consequent part B_k of R_k. The denominator of (16) corresponds to the number of input-output pairs that are compatible with the antecedent part $\mathbf{A}_k$ of the fuzzy rule R_k. The numerator corresponds to the number of input-output pairs that are compatible with both the antecedent and consequent parts of R_k.

The support $s(R_k)$ of the fuzzy rule R_k is defined as

$$s(R_k) = \frac{\sum_{p=1}^{m} \mu_{\mathbf{A}_k}(\mathbf{x}_p) \cdot \mu_{B_k}(y_p)}{m}. \tag{17}$$

When both the antecedent and consequent parts of the fuzzy rule R_k are specified by non-fuzzy concepts, these two definitions in (16) and (17) are exactly the same as those used for non-fuzzy association rules in data mining [1].

The two rule evaluation measures are employed for extracting a pre-specified number of candidate rules in various manners. For example, we can use one of the following rule extraction criteria:

(1) Choose candidate rules using the confidence measure.
(2) Choose candidate rules using the support measure.
(3) Choose candidate rules using the confidence measure from fuzzy rules whose support values are not less than a pre-specified minimum support level.
(4) Choose candidate rules using the support measure from fuzzy rules whose confidence values are not less than a pre-specified minimum confidence level.
(5) Choose candidate rules using a composite criterion of the confidence and support measures. A simple example of such a composite criterion is their product.

The length of fuzzy rules can be used as a constraint condition on candidate rules. That is, candidate rules are chosen using a rule extraction criterion from fuzzy rules that are shorter than or equal to a pre-specified maximum length. The use of the upper bound on the length of candidate rules is consistent with the third objective of our linguistic modelling problem (i.e., minimization of the total rule length).

5.3 Genetic Algorithms for Rule Selection

Let us assume that we have *N* candidate rules for genetic rule selection. Any subset *S* of those candidate rules is denoted by a binary string of the length *N* as

$$S = s_1 s_2 \cdots s_N, \tag{18}$$

where $s_j = 1$ and $s_j = 0$ mean that the *j*th candidate rule is included in *S* and excluded from *S*, respectively.

When the weight values for the three objectives are given from domain experts or human users, we can use the weighted sum in (13) as a scalar fitness function. In this case, we can use standard genetic algorithms for finding the optimal rule set that maximizes the scalar fitness function. On the other hand, genetic rule selection is performed using multi-objective genetic algorithms [4, 6] when no *a priori* knowledge is given for the relative importance of the three objectives.

As shown in (18), the length of the binary string *S* is *N* (i.e., the number of candidate rules). Thus the size of the search space is 2^N. This means that long computation time and large memory storage are needed for executing genetic rule selection when the number of candidate rules is large. Two heuristic procedures were used for improving the efficiency of genetic rule selection for fuzzy rule-based classification [11, 13, 14, 15, 17]. One is the use of biased mutation where a larger

mutation probability is assigned to the mutation from 1 to 0 than that from 0 to 1. The biased mutation is for efficiently decreasing the number of fuzzy rules in each rule set. The other is the removal of unnecessary fuzzy rules. If the antecedent part of a fuzzy rule is not compatible with any input-output pair, we can remove the fuzzy rule without deteriorating the approximation accuracy. At the same time, the removal of such an unnecessary rule improves the second and third objectives of our linguistic modelling problem. While the efficiency of genetic rule selection mainly depends on the choice of candidate rules, the biased mutation and the removal of unnecessary rules also improve the search ability to efficiently find good rule sets.

6 Modification of Fuzzy Reasoning

6.1 Computer Simulations on Simple Numerical Examples

As a test problem, we generated 9261 input-output pairs $(x_{p1}, x_{p2}, x_{p3}, y_p)$, $p = 1$, 2, ..., 9261, from the three-input and single-output nonlinear function in (3) using the $21 \times 21 \times 21$ uniform grid of the input space $[0, 1]^3$: $x_{pi} = 0.00, 0.05, 0.10, ..., 1.00$ for $i = 1,2,3$. The five linguistic terms in Fig. 1 were used for all the three input and single output variables. We also used *don't care* as an additional antecedent fuzzy set.

We assumed that the following scalar fitness function was given:

$$f(S) = -100 f_1(S) - f_2(S) - f_3(S). \tag{19}$$

We used the fuzzy GBML algorithm in Section 3 for finding the optimal rule set with respect to this scalar fitness function. As explained in Section 3, the heuristic procedure with the upper bound on the number of fuzzy rules (i.e., 20 rules) was used. The other heuristic procedures were not utilized in computer simulations. Our fuzzy GBML algorithm was executed under the following parameter specifications:

Population size: 200,
The number of fuzzy rules in each initial rule set: 10,
Crossover probability: 0.8,
Mutation probability for replacing each fuzzy set with another one: 0.1,
Mutation probability for removing each fuzzy rule: 0.1,
Stopping condition: 5000 generations.

We applied our fuzzy GBML algorithm to the generated 9261 input-output pairs 10 times using different initial populations. A rule set with the following two rules was obtained from 4 out of 10 runs:

$$R_A : y \text{ is } small, \tag{20}$$

$$R_B : \text{If } x_1 \text{ is } large \text{ and } x_2 \text{ is } large \text{ and } x_3 \text{ is } large \text{ then } y \text{ is } large. \tag{21}$$

The total squared error over the 9261 input-output pairs was 1.045. Thus we can see that the accuracy of the rule set with R_A and R_B is high. In all the other six runs, these two rules were obtained as a part of larger rule sets with additional fuzzy rules. The total squared error of those larger rule sets was slightly better than the case of the

rule set with the above two fuzzy rules. Actually it was between 0.738 and 0.968. These simulation results were much more sensitive to the specification of the weight values in (19) than the above parameter values in our fuzzy GBML algorithm.

We can easily understand a rough shape of the nonlinear function from the above two fuzzy rules. That is, one may think that the output value is *small* except for the region with *large* x_1, *large* x_2 and *large* x_3 where the output value is *large*.

Using the confidence measure, we evaluated the two fuzzy rules. The confidence value of each fuzzy rule was calculated as follows:

$$c(R_A) = 0.994 \text{ and } c(R_B) = 0.000. \tag{22}$$

We can see that the confidence value of R_B is very small while that of R_A is large. For comparison, we also calculated the confidence values of the following fuzzy rules that have the same antecedent part as R_B but different consequent fuzzy sets:

R_C : If x_1 is *large* and x_2 is *large* and x_3 is *large* then y is *small*,
R_D : If x_1 is *large* and x_2 is *large* and x_3 is *large* then y is *medium small*,
R_E : If x_1 is *large* and x_2 is *large* and x_3 is *large* then y is *medium*,
R_F : If x_1 is *large* and x_2 is *large* and x_3 is *large* then y is *medium large*.

The confidence value of each fuzzy rule was calculated as follows:

$$c(R_C) = 0.660,\ c(R_D) = 0.148,\ c(R_E) = 0.192, \text{ and } c(R_F) = 0.000. \tag{23}$$

We can see that the fuzzy rules R_C, R_D and R_E are more compatible with the given input-output pairs than the obtained fuzzy rule R_B. That is, the output value is not *large* but *small*, *medium small* or *medium* for input vectors with *large* x_1, *large* x_2 and *large* x_3. In this sense, the obtained fuzzy rule R_B in (21) is misleading.

For visually examining why such a misleading fuzzy rule was obtained, we applied our fuzzy GBML algorithm to a two-input and single-output nonlinear function in Fig. 5 in the same manner as the previous computer simulation. We first generated 441 input-output pairs (x_{p1}, x_{p2}, y_p), $p = 1,2, ..., 441$, from the nonlinear function using the 21×21 uniform grid of the input space $[0, 1]^2$. Then we applied our fuzzy GBML algorithm to the generated 441 input-output pairs 20 times using different initial populations. A rule set with the following three rules was obtained from 17 out of 20 runs:

$$R_I : y \text{ is } small, \tag{24}$$

$$R_{II} : \text{If } x_1 \text{ is } small \text{ then } y \text{ is } medium, \tag{25}$$

$$R_{III} : \text{If } x_1 \text{ is } small \text{ and } x_2 \text{ is } small \text{ then } y \text{ is } large. \tag{26}$$

The total squared error was 0. Actually we depicted the nonlinear function in Fig. 5 by applying the fuzzy reasoning method in (10) to the three fuzzy rules in (24)-(26).

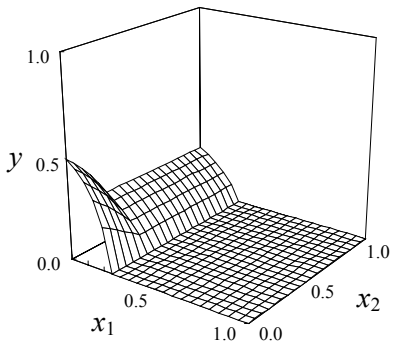


Fig. 5. Nonlinear function to be linguistically described

Let us try to imagine a three-dimensional shape of a nonlinear function from the three fuzzy rules in (24)-(26). From the fuzzy rule R_{III}, we think that the output is *large* for *small* x_1 and *small* x_2. From the fuzzy rule R_{II}, we think that the output is *medium* for *small* x_1. For other input vectors, the output values seem to be *small*. As a result, we may have a three-dimensional shape that is similar to Fig. 6. It should be noted that the intuitively imagined shape is different from Fig. 5 from which the three fuzzy rules in (24)-(26) were derived by our fuzzy GBML algorithm. We further examined the three fuzzy rules using the confidence measure. The confidence value of each fuzzy rule was calculated from the 441 input-output pairs as follows:

$$c(R_{\mathrm{I}}) = 0.819, \quad c(R_{\mathrm{II}}) = 0.078 \text{ and } \quad c(R_{\mathrm{III}}) = 0.000. \tag{27}$$

While the first fuzzy rule R_{I} has a large confidence value, the confidence values of the other fuzzy rules are very small. This means that the fuzzy rules R_{II} and R_{III} are not consistent with the given input-output pairs.

Let us explain why the misleading fuzzy rules were obtained for the nonlinear functions in (3) and Fig. 5. In our approach to linguistic modelling, we use *don't care* as an additional antecedent fuzzy set for generating short fuzzy rules and covering the whole input space by a small number of fuzzy rules. As we have already explained in Section 2, the use of *don't care* is necessary for linguistically explaining a nonlinear function using a small number of fuzzy rules. Thus our linguistic model is a mixture of general and specific fuzzy rules. When we intuitively estimate an output value from general and specific rules, specific rules have usually higher priority than general rules. For example, we may mainly use the most specific fuzzy rule R_{III} in (26) when we estimate an output value for *small* x_1 and *small* x_2 using the three fuzzy rules in (24)-(26). In this case, the output is intuitively estimated as *large* (see Fig. 6). On the other hand, most fuzzy reasoning methods are based on the interpolation of compatible fuzzy rules. Thus the estimated output for *small* x_1 and *small* x_2 is usually calculated as *medium* (see Fig. 5). This difference between our intuition and fuzzy reasoning leads to linguistic models with misleading fuzzy rules.

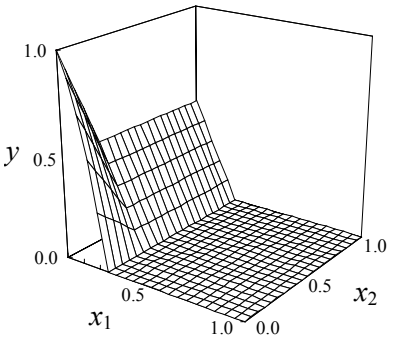


Fig. 6. Nonlinear function that is intuitively depicted from R_{I}, R_{II} and R_{III} in (24)-(26)

6.2 Non-standard Inclusion-Based Fuzzy Reasoning

A non-standard fuzzy reasoning method based on an inclusion relation among fuzzy rules was proposed for obtaining intuitively acceptable fuzzy reasoning results [10]. The proposal of such an inclusion-based fuzzy reasoning method was motivated by the above-mentioned difficulty of standard interpolation-based fuzzy reasoning.

First we explain an inclusion relation using the following two fuzzy rules:

$$R_k: \text{If } x_1 \text{ is } A_{k1} \text{ and } ... \text{ and } x_n \text{ is } A_{kn} \text{ then } y \text{ is } B_k, \tag{28}$$

$$R_q: \text{If } x_1 \text{ is } A_{q1} \text{ and } ... \text{ and } x_n \text{ is } A_{qn} \text{ then } y \text{ is } B_q. \tag{29}$$

When the inclusion relation $A_{qi} \subseteq A_{ki}$ holds between the antecedent fuzzy sets for $i = 1,2,...,n$, we say that the fuzzy rule R_q is included in the fuzzy rule R_k (i.e., $R_q \subseteq R_k$). In order to implement the preference for more specific fuzzy rules, the standard fuzzy reasoning method in (10) is modified as

$$\hat{y}(\mathbf{x}_p) = \frac{\sum_{R_k \in S} \phi(R_k, \mathbf{x}_p) \cdot b_k \cdot \mu_{\mathbf{A}_k}(\mathbf{x}_p)}{\sum_{R_k \in S} \phi(R_k, \mathbf{x}_p) \cdot \mu_{\mathbf{A}_k}(\mathbf{x}_p)}, \tag{30}$$

where $\phi(R_k, \mathbf{x}_p)$ is a weight determined by the inclusion relation between R_k and the other fuzzy rules in the rule set S. The value of $\phi(R_k, \mathbf{x}_p)$ is small when R_k includes more specific rules compatible with the input vector $\mathbf{x}_p$. In this case, the weight of R_k is discounted in fuzzy reasoning. Actually $\phi(R_k, \mathbf{x}_p)$ is defined using a user-definable non-negative parameter β as

$$\phi(R_k, \mathbf{x}_p) = \prod_{\substack{R_q \subseteq R_k \\ q \neq k}} (1 - \mu_{\mathbf{A}_q}(\mathbf{x}_p))^{\beta}. \tag{31}$$

When no fuzzy rule is included in R_k, $\phi(R_k, \mathbf{x}_p)$ is specified as $\phi(R_k, \mathbf{x}_p) = 1$ because the weight of R_k should not be discounted in this case.

The nonlinear function in Fig. 6 was the fuzzy reasoning result where the inclusion-based fuzzy reasoning method with $\beta = 1$ in (30) was applied to the three fuzzy rules R_{I}, R_{II} and R_{III} in (24)-(26). From the comparison between the three fuzzy rules in (24)-(26) and the three-dimensional shape of the nonlinear function in Fig. 6, we can see that an intuitively acceptable fuzzy reasoning result was obtained by the non-standard inclusion-based fuzzy reasoning method.

6.3 Computer Simulations Using Non-standard Fuzzy Reasoning

In the same manner as Subsection 6.1, we applied our fuzzy GBML algorithm to the nonlinear function in Fig. 5. When the total squared error was calculated, we used the non-standard inclusion-based fuzzy reasoning method with $\beta = 2$ instead of the standard interpolation-based fuzzy reasoning method in (10). The computer simulation was iterated 20 times using different initial populations. A rule set with the following three fuzzy rules was obtained from 15 out of 20 runs:

y is *small*, (32)

If x_1 is *small* then y is *medium small*, (33)

If x_1 is *small* and x_2 is *small* then y is *medium*. (34)

From the comparison between the obtained three fuzzy rules and Fig. 5, we can see that the nonlinear function in Fig. 5 is linguistically described in an intuitively acceptable manner. That is, the obtained fuzzy rules in (32)-(34) are consistent with the three-dimensional shape of the nonlinear function in Fig. 5.

In the same manner as Subsection 6.1, we also applied our fuzzy GBML algorithm to the nonlinear function in (3) using the non-standard fuzzy reasoning method with $\beta = 5$. The computer simulation was iterated 10 times using different initial populations. A rule set with the following two fuzzy rules was obtained from all the 10 runs.

R_{A} : y is *small*, (35)

R_{E} : If x_1 is *large* and x_2 is *large* and x_3 is *large* then y is *medium*. (36)

As we have already explained in Subsection 6.1, these two fuzzy rules have large confidence values (i.e., $c(R_{\mathrm{A}}) = 0.994$ and $c(R_{\mathrm{E}}) = 0.192$). While the misleading fuzzy rule R_{B} with a very small confidence value (i.e., $c(R_{\mathrm{B}}) = 0.000$) was obtained in Subsection 6.1 using the standard interpolation-based fuzzy reasoning method, the fuzzy rule R_{E} with a larger confidence value was obtained using the non-standard inclusion-based fuzzy reasoning method.

7 Concluding Remarks

In this chapter, we discussed linguistic modelling for linguistically describing nonlinear functions in a human understandable manner. That is, we discussed linguistic modelling for obtaining linguistic models with high interpretability as well as high accuracy. We assumed that a set of linguistic terms was given for each variable from domain experts or human users. Thus we did not discuss the interpretability of fuzzy partitions. The interpretability of linguistic models was defined by the number of fuzzy rules and the total rule length. In this context, we explained the validity of the use of the total rule length instead of the average rule length as a complexity measure of linguistic models. We also explained the necessity of the use of *don't care* as an additional antecedent fuzzy set for linguistically describing high-dimensional nonlinear functions using a small number of fuzzy rules with high interpretability.

Linguistic modelling was formulated as a three-objective optimization problem where the total squared error, the number of fuzzy rules and the total rule length were minimized. We explained how the formulated linguistic modelling problem can be handled by single-objective and multi-objective genetic algorithms. We showed two approaches to our linguistic modelling problem: fuzzy genetics-based machine learning and genetic rule selection. Then we pointed out a possibility that misleading fuzzy rules can be obtained from our linguistic modelling problem. Finally we demonstrated that the use of the non-standard inclusion-based fuzzy reasoning method removed such an undesirable possibility.

References

[1] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A. I.: Fast Discovery of Association Rules, in Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (eds.) *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Metro Park (1996) 307-328.

[2] Carse, B., Fogarty, T. C., and Munro, A.: Evolving Fuzzy Rule Based Controllers Using Genetic Algorithms, *Fuzzy Sets and Systems* 80 (1996) 273-293.

[3] Casillas, J., Cordon, O., Herrera, F., and Magdalena, L. (Eds.): *Trade-off between Accuracy and Interpretability in Fuzzy Rule-Based Modelling*, Physica-Verlag (2002).

[4] Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, Boston (2002).

[5] Cordon, O., Herrera, F., Hoffmann, F., and Magdalena, L.: *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, World Scientific (2001).

[6] Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Chichester (2001).

[7] Homaifar, A., and McCormick, E.: Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms, *IEEE Trans. on Fuzzy Systems* 3 (1995) 129-139.

[8] Hong, T. -P., Kuo, C. -S., and Chi, S. -C.: Trade-off between Computation Time and Number of Rules for Fuzzy Mining from Quantitative Data, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 9 (2001) 587-604.

[9] Horikawa, S., Furuhashi, T., and Uchikawa, Y.: On Fuzzy Modelling using Fuzzy Neural Networks with the Back-Propagation Algorithm, *IEEE Trans. on Neural Networks* 3 (1992) 801-806.

[10] Ishibuchi, H.: Fuzzy Reasoning Method in Fuzzy Rule-based Systems with General and Specific Rules for Function Approximation, *Proc. of 8th IEEE International Conference on Fuzzy Systems* (1999) 198-203.

[11] Ishibuchi, H., Murata, T., and Turksen, I. B.: Single-Objective and Two-Objective Genetic Algorithms for Selecting Linguistic Rules for Pattern Classification Problems, *Fuzzy Sets and Systems* 89 (1997) 135-149.

[12] Ishibuchi, H., Nakashima, T.: Genetic-Algorithm-Based Approach to Linguistic Approximation of Nonlinear Functions with Many Input Variables, *Proc. of 8th IEEE International Conference on Fuzzy Systems* (1999) 779-784.

[13] Ishibuchi, H., Nakashima, T., and Murata, T.: Three-Objective Genetics-Based Machine Learning for Linguistic Rule Extraction, *Information Sciences* 136 (2001) 109-133.

[14] Ishibuchi, H., Nakashima, T., and Murata, T.: Multiobjective Optimization in Linguistic Rule Extraction from Numerical Data, in Zitzler, E., Deb, K., Thiele, L., Coello Coello, C. A., and Corne, D. (eds.) *Evolutionary Multi-Criterion Optimization: Lecture Notes in Computer Science 1993*, Springer-Verlag, Berlin (2001) 588-602.

[15] Ishibuchi, H., Nozaki, K., Yamamoto, N., and Tanaka, H.: Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms, *IEEE Trans. on Fuzzy Systems* 3 (1995) 260-270.

[16] Ishibuchi, H., Takeuchi, D., and Nakashima, T.: GA-Based Approaches to Linguistic Modelling of Nonlinear Functions, *Proc. of 9th IFSA World Congress and 20th NAFIPS International Conference* (2001) 1229-1234.

[17] Ishibuchi, H., and Yamamoto, T.: Fuzzy Rule Selection by Data Mining Criteria and Genetic Algorithms, *Proc. of Genetic and Evolutionary Computation Conference* (2002) 399-406.

[18] Ishibuchi, H., Yamamoto, T., and Nakashima, T.: Fuzzy Data Mining: Effect of Fuzzy Discretization, *Proc. of 1st IEEE International Conference on Data Mining* (2001) 241-248.

[19] Jang, J. -S. R.: ANFIS: Adaptive-Network-Based Fuzzy Inference System, *IEEE Trans. on Systems, Man, and Cybernetics* 23 (1993) 665-685.

[20] Jimenez, F., Gomez-Skarmeta, A. F., Roubos, H., and Babuska, R.: Accurate, Transparent and Compact Fuzzy Models for Function Approximation and Dynamic Modelling Through Multi-Objective Evolutionary Optimization, in Zitzler, E., Deb, K., Thiele, L., Coello Coello, C. A., and Corne, D. (eds.) *Evolutionary Multi-Criterion Optimization: Lecture Notes in Computer Science 1993*, Springer-Verlag, Berlin (2001) 653-667.

[21] Jimenez, F., Gomez-Skarmeta, A. F., Roubos, H., and Babuska, R.: A Multi-Objective Evolutionary Algorithm for Fuzzy Modelling, *Proc. of 9th IFSA World Congress and 20th NAFIPS International Conference* (2001) 1222-1228.
[22] Jin, Y.: Fuzzy Modelling of High-dimensional Systems: Complexity Reduction and Interpretability Improvement, *IEEE Trans. on Fuzzy Systems* 8 (2000) 212-221.
[23] Jin, Y., Von Seelen, W., and Sendhoff, B.: On Generating FC^3 Fuzzy Rule Systems from Data Using Evolution Strategies, *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybernetics* 29 (1999) 829-845.
[24] Karr, C. L., and Gentry, E. J.: Fuzzy Control of pH Using Genetic Algorithms, *IEEE Trans. on Fuzzy Systems* 1 (1993) 46-53.
[25] Knowles, J. D., and Corne, D. W.: On Metrics for Comparing Non-Dominated Sets, *Proc. of 2002 Congress on Evolutionary Computation* (2002) 711-716.
[26] Nozaki, K., Ishibuchi, H., and Tanaka, H.: A Simple but Powerful Heuristic Method for Generating Fuzzy Rules from Numerical Data, *Fuzzy Sets and Systems* 86 (1997) 251-270.
[27] Pedrycz, W.: *Fuzzy Modelling: Paradigms and Practice*, Kluwer Academic Publishers, Boston (1996).
[28] Roubos, H., and Setnes, M.: Compact and Transparent Fuzzy Models and Classifiers Through Iterative Complexity Reduction, *IEEE Trans. on Fuzzy Systems* 9 (2001) 516-524.
[29] Setnes, M., Babuska, R., Kaymak, U., and Van Nauta Lemke, H. R.: Similarity Measures in Fuzzy Rule Base Simplification, *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics* 28 (1998) 376-386.
[30] Setnes, M., Babuska, R., and Verbruggen, B.: Rule-based Modelling: Precision and Transparency, *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews* 28 (1998) 165-169.
[31] Sugeno, M., and Yasukawa, T.: A Fuzzy-Logic-Based Approach to Qualitative Modelling, *IEEE Trans. on Fuzzy Systems* 1 (1993) 7-31.
[32] Sun, C. -T.: Rule-Base Structure Identification in an Adaptive-Network-Based Fuzzy Inference System, *IEEE Trans. on Fuzzy Systems* 2 (1994) 64-73.
[33] Takagi, T., and Sugeno, M.: Fuzzy Identification of Systems and Its Applications to Modelling and Control, *IEEE Trans. on Systems, Man, and Cybernetics* 15 (1985) 116-132.
[34] Valente de Oliveira, J.: Semantic Constraints for Membership Function Optimization, *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans* 29 (1999) 128-138.
[35] Wang, L. X., and Mendel, J. M.: Generating Fuzzy Rules by Learning from Examples, *IEEE Trans. on Systems, Man, and Cybernetics* 22 (1992) 1414-1427.

Author Index

Alcalá, Rafael 44

Barro, Senén 1
Bouchon-Meunier, Bernadette . 102
Bugarín, Alberto 1

Cao, Tru H. 80
Cariñena, Purificación 1
Chen, Yi-Ping Phoebe 153
Cordón, Oscar 44

Díaz-Hermida, Felix 1

Hernández, Enric 26
Herrera, Francisco 44

Ishibuchi, Hisao 209

Laurent, Anne 102
Lawry, Jonathan 186

Marsala, Christophe 102

Paulson, Patrick 168

Recasens, Jordi 26
Rossiter, Jonathan 124

Shen, Qiang 64

Tzanavari, Aimilia 168

Yamamoto, Takashi 209