Aijun An Jerzy Stefanowski
Sheela Ramanna Cory J. Butz
Witold Pedrycz Guoyin Wang (Eds.)

LNAI 4482

# Rough Sets, Fuzzy Sets, Data Mining and Granular Computing

11th International Conference, RSFDGrC 2007
Toronto, Canada, May 2007
Proceedings

Springer

Lecture Notes in Artificial Intelligence 4482

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Aijun An Jerzy Stefanowski
Sheela Ramanna Cory J. Butz
Witold Pedrycz Guoyin Wang (Eds.)

# Rough Sets, Fuzzy Sets, Data Mining and Granular Computing

11th International Conference, RSFDGrC 2007
Toronto, Canada, May 14-16, 2007
Proceedings

Springer

Volume Editors

Aijun An
York University, Toronto,Canada
E-mail: aan@cse.yorku.ca

Jerzy Stefanowski
Poznań University of Technology, Poland
E-mail: Jerzy.Stefanowski@cs.put.poznan.pl

Sheela Ramanna
University of Winnipeg, Canada
E-mail: s.ramanna@uwinnipeg.ca

Cory J. Butz
University of Regina, Canada
E-mail: butz@cs.uregina.ca

Witold Pedrycz
University of Alberta, Canada
E-mail: pedrycz@ee.ualberta.ca

Guoyin Wang
Chongqing University of Posts and Telecommunications, P.R. China
E-mail: wanggy@ieee.org

The paper "A New Cluster Validity Index for Fuzzy Clustering Based on Similarity Measure" starting on p. 127 has been retracted as a large proportion of its contents were copied from the following paper: "A Cluster Validation Index for GK Cluster Analysis Based on Relative Degree of Sharing" by Young-Il Kim et al.

Library of Congress Control Number: 2007926026

CR Subject Classification (1998): I.2, H.2.4, H.3, F.4.1, F.1, I.5, H.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-72529-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-72529-9 Springer Berlin Heidelberg New York



Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12063925 06/3180 5 4 3 2 1 0

# Preface

This volume contains the papers selected for presentation at the 11th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC 2007), a part of the Joint Rough Set Symposium (JRS 2007) organized by Infobright Inc. and York University. JRS 2007 was held for the first time during May 14–16, 2007 in MaRS Discovery District, Toronto, Canada. It consisted of two conferences: RSFDGrC 2007 and the Second International Conference on Rough Sets and Knowledge Technology (RSKT 2007).

The two conferences that constituted JRS 2007 investigated rough sets as an emerging methodology established more than 25 years ago by Zdzisław Pawlak. Rough set theory has become an integral part of diverse hybrid research streams. In keeping with this trend, JRS 2007 encompassed rough and fuzzy sets, knowledge technology and discovery, soft and granular computing, data processing and mining, while maintaining an emphasis on foundations and applications.

RSFDGrC 2007 followed in the footsteps of well-established international initiatives devoted to the dissemination of rough sets research, held so far in Canada, China, Japan, Poland, Sweden, and the USA. RSFDGrC was first organized as the 7th International Workshop on Rough Sets, Data Mining and Granular Computing held in Yamaguchi, Japan in 1999. Its key feature was to stress the role of integrating intelligent information methods to solve real-world, large, complex problems concerned with uncertainty and fuzziness. RSFDGrC achieved the status of a bi-annual international conference, starting from 2003 in Chongqing, China.

In RSFDGrC 2007, a special effort was made to include research spanning a broad range of theory and applications. This was achieved by including in the conference program a number of special sessions, invited talks, and tutorials.

Overall, we received 319 submissions to the Joint Rough Set Symposium. Every paper was examined by at least two reviewers. The submission and review processes were performed jointly for both conferences that together constituted JRS 2007, i.e., RSFDGrC 2007 and RSKT 2007.

Out of the papers initially selected, some were approved subject to revision and then additionally evaluated. Finally, 139 papers were accepted for JRS 2007. This gives an acceptance ratio slightly over 43% for the joint conferences.

Accepted papers were distributed between the two conferences on the basis of their relevance to the conference themes.

The JRS 2007 conference papers are split into two volumes (LNAI 4481 for RSKT 2007 and LNAI 4482 for RSFDGrC 2007). The regular, invited, and special session papers selected for presentation at RSFDGrC 2007 are included within 12 chapters and grouped under specific conference topics.

This volume contains 69 papers, including 4 invited papers presented in Chap. 1. The remaining 65 papers are presented in 11 chapters related to

fuzzy-rough hybridization, fuzzy sets, soft computing in medical image processing, soft computing in information retrieval, clustering, text and Web mining, learning, data mining and rough classifiers, granular computing, soft computing in multimedia processing, soft computing applications, and rough and complex concepts.

We wish to thank all of the authors who contributed to this volume. We are very grateful to the chairs, advisory board members, Program Committee members, and other reviewers not listed in the conference committee for their help in the acceptance process.

We are grateful to our Honorary Chairs, Setsuo Ohsuga and Lotfi Zadeh, for their support and visionary leadership. We also acknowledge the scientists who kindly agreed to give the keynote, plenary, and tutorial lectures: Andrzej Bargiela, Mihir K. Chakraborty, Bernhard Ganter, Sushmita Mitra, Sadaaki Miyamoto, James F. Peters, Andrzej Skowron, Domenico Talia, Xindong Wu, Yiyu Yao, Chengqi Zhang, and Wojciech Ziarko. We also wish to express our deep appreciation to all special session organizers.

We greatly appreciate the co-operation, support, and sponsorship of various companies, institutions and organizations, including: Infobright Inc., MaRS Discovery District, Springer, York University, International Rough Set Society, International Fuzzy Systems Association, Rough Sets and Soft Computation Society of the Chinese Association for Artificial Intelligence, and National Research Council of Canada.

We wish to thank several people whose hard work made the organization of JRS 2007 possible. In particular, we acknowledge the generous help received from: Tokuyo Mizuhara, Clara Masaro, Christopher Henry, Julio V. Valdes, April Dunford, Sandy Hsu, Lora Zuech, Bonnie Barbayanis, and Allen Gelberg.

Last but not least, we are thankful to Alfred Hofmann of Springer for support and co-operation during preparation of this volume.

May 2007

Aijun An
Jerzy Stefanowski
Sheela Ramanna
Cory Butz
Witold Pedrycz
Guoyin Wang

# RSFDGrC 2007 Conference Committee

| | |
|---|---|
| JRS Honorary Chairs | Setsuo Ohsuga, Lotfi A. Zadeh |
| JRS Conference Chairs | Dominik Ślęzak, Guoyin Wang |
| JRS Program Chairs | Nick Cercone, Witold Pedrycz |
| RSFDGrC 2007 Chairs | Aijun An, Jerzy Stefanowski, Sheela Ramanna, Cory Butz |
| JRS Organizing Chairs | Jimmy Huang, Miriam G. Tuerk |
| JRS Publicity Chairs | Aboul E. Hassanien, Shoji Hirano, Daniel Howard, Igor Jurisica, Tai-hoon Kim, Duoqian Miao, Bhanu Prasad, Mark S. Windrim |

## RSFDGrC 2007 Steering Committee

James F. Peters (Chair)
Hans-Dieter Burkhard
Gianpiero Cattaneo
Mihir K. Chakraborty
Juan-Carlos Cubero
Didier Dubois
Ivo Düntsch
Aboul E. Hassanien
Masahiro Inuiguchi
Tsau Young Lin
Qing Liu
Sadaaki Miyamoto
Masoud Nikravesh
Witold Pedrycz
Lech Polkowski
Władysław Skarbek
Dominik Ślęzak
Roman Słowiński
Hui Wang
Wen-Xiu Zhang
Wojciech Ziarko

## RSFDGrC 2007 Program Committee

Rakesh Agrawal
Rajen Bhatt
Chien-Chung Chan
Chris Cornelis
Andrzej Czyżewski
Jitender Deogun
M.-C. Fernandez-Baizan
Ryszard Janicki
Jouni Järvinen
Richard Jensen
Jianmin Jiang
Licheng Jiao
Janusz Kacprzyk
Haeng-Kon Kim
Jacek Koronacki
Krzysztof Krawiec
Vladik Kreinovich
Marzena Kryszkiewicz
Mineichi Kudo
Jungwoo Lee
Churn-Jung Liau
Chunnian Liu
Lawrence Mazlack
Wojtek Michałowski
Mikhail Moshkov
Tetsuya Murai
Michinori Nakata
Hung Son Nguyen
Piero Pagliani
Mirek Pawlak
Leonid Perlovsky
Georg Peters
Fred Petry
Bhanu Prasad
Ingrid Rewitzky
Leszek Rutkowski
Hiroshi Sakai
B. Uma Shankar
Arul Siromoney
Jarosław Stepaniuk
Andrzej Szałas
Ruppa Thulasiram
I. Burhan Turksen
Gwo-Hshiung Tzeng
Dimiter Vakarelov
Lipo Wang
Paul P. Wang
Patrick S.P. Wang
Piotr Wasilewski
Richard Weber
Jakub Wróblewski

Dan Wu
Xindong Wu
Justin Zhan
Chengqi Zhang
Qingfu Zhang
Qiangfu Zhao
Xueyuan Zhou
Zhi-Hua Zhou
Constantin Zopounidis

## Non-committee Reviewers

Haider Banka
Klaas Bosteels
Yaohua Chen
Piotr Dalka
Alicja Grużdż
Liting Han
You Sik Hong
Andrzej Kaczmarek
Hyung Jun Kim
Pavani Kuntala
Tianrui Li
Gabriela Lindemann
Hailin Liu
Mohamed Mostafa
Kia Ng
Xiaoping Qiu
Claudius Schnoerr
Raj Singh
Ying Weng
Sebastian Widz
Yang Xu

# Table of Contents

## Invited Papers



## Fuzzy-Rough Hybridization



## Fuzzy Sets

## Soft Computing in Medical Image Processing



## Soft Computing in Information Retrieval

## Clustering



## Text and Web Mining



## Learning, Data Mining and Rough Classifiers

## Granular Computing

## Soft Computing in Multimedia Processing



## Soft Computing Applications

## Rough and Complex Concepts

# Toward Rough-Granular Computing
## Extended Abstract

Andrzej Jankowski[1] and Andrzej Skowron[2]

[1] Institute of Decision Processes Support
and
AdgaM Solutions Sp. z o.o.
Wąwozowa 9 lok. 64, 02-796 Warsaw, Poland
andrzejj@adgam.com.pl
[2] Institute of Mathematics,
Warsaw University
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl

Developing methods for approximation of compound concepts expressing the result of perception belongs to the main challenges of Perception Based Computing (PBC) [70]. The perceived concepts are expressed in natural language. We discuss the rough-granular approach to approximation of such concepts from sensory data and domain knowledge. This additional knowledge, represented by ontology of concepts, is used to make it feasible searching for features (condition attributes) relevant for the approximation of concepts on different levels of the concept hierarchy defined by a given ontology. We report several experiments of the proposed methodology for approximation of compound concepts from sensory data and domain knowledge. The approach is illustrated by examples relative to interactions of agents, ontology approximation, adaptive hierarchical learning of compound concepts and skills, behavioral pattern identification, planning, conflict analysis and negotiations, and perception-based reasoning. The presented results seem to justify the following claim of Lotfi A. Zadeh: "In coming years, granular computing is likely to play an increasingly important role in scientific theories-especially in human-centric theories in which human judgement, perception and emotions are of pivotal importance". The question of how ontologies of concepts can be discovered from sensory data remains as one of the greatest challenges for many interdisciplinary projects on learning of concepts.

The concept approximation problem is the basic problem investigated in machine learning, pattern recognition and data mining [24]. It is necessary to induce approximations of concepts (models of concepts) consistent (or almost consistent) with some constraints. In the most typical case, constraints are defined by a training sample. For more compound concepts, we consider constraints defined by domain ontology consisting of vague concepts and dependencies between them. Information about the classified objects and concepts is partial. In the most general case, the adaptive approximation of concepts is performed under interaction with dynamically changing environment. In all these cases, searching for sub-optimal models relative to the minimal length principle (MLP) is

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 1–12, 2007.

performed. Notice that in adaptive concept approximation one of the components of the model should be the adaptation strategy. Components involved in construction of concept approximation which are tuned in searching for suboptimal models relative to MLP are called information granules. In rough granular computing (RGC), information granule calculi are used for construction of components of classifiers and classifiers themselves (see, e.g., [60]) satisfying given constraints. An important mechanism in RGC is related to generalization schemes making it possible to construct more compound patterns from less compound patters. Generalization degrees of schemes are tuned using, e.g., some evolutionary strategies.

Rough set theory due to Zdzisław Pawlak [43,44,45,46,17] is a mathematical approach to imperfect knowledge. The problem of imperfect knowledge has been tackled for a long time by philosophers, logicians and mathematicians. Recently it became also a crucial issue for computer scientists, particularly in the area of artificial intelligence. There are many approaches to the problem of how to understand and manipulate imperfect knowledge. The most successful one is, no doubt, the fuzzy set theory proposed by Lotfi A. Zadeh [69]. Rough set theory presents still another attempt to solve this problem. It is based on an assumption that objects and concepts are perceived by partial information about them. Due to this some objects can be indiscernible. From this fact it follows that some sets can not be exactly described by available information about objects; they are rough not crisp. Any rough set is characterized by its (lower and upper) approximations. The difference between the upper and lower approximation of a given set is called its boundary. Rough set theory expresses vagueness relative to the boundary region of a set. If the boundary region of a set is empty, it means that the set is crisp; otherwise, the set is rough (inexact). A nonempty boundary region of a set indicates that our knowledge about the set is not sufficient to define the set precisely. One can recognize that rough set theory is, in a sense, a formalization of the idea presented by Gotlob Frege [23].

One of the consequences of perceiving objects using only available information about them is that for some objects one cannot decide if they belong to a given set or not. However, one can estimate the degree to which objects belong to sets. This is another crucial observation in building the foundations for approximate reasoning. In dealing with imperfect knowledge, one can only characterize satisfiability of relations between objects to a degree, not precisely. Among relations on objects, the rough inclusion relation plays a special role in describing to what degree objects are parts of other objects. A rough mereological approach (see, e.g., [52,59,42]) is an extension of the Leśniewski mereology [31] and is based on the relation *to be a part to a degree*. It will be interesting to note here that Jan Łukasiewicz was the first who started to investigate the inclusion to a degree of concepts in his discussion on relationships between probability and logical calculi [35].

The very successful technique for rough set methods has been Boolean reasoning [12]. The idea of Boolean reasoning is based on construction for a given problem $P$ a corresponding Boolean function $f_P$ with the following property:

the solutions for the problem $P$ can be decoded from prime implicants of the Boolean function $f_P$ . It is worth while to mention that to solve real-life problems, it is necessary to deal with Boolean functions having a large number of variables.

A successful methodology based on the discernibility of objects and Boolean reasoning has been developed in rough set theory for computing of many key constructs like reducts and their approximations, decision rules, association rules, discretization of real valued attributes, symbolic value grouping, searching for new features defined by oblique hyperplanes or higher order surfaces, pattern extraction from data as well as conflict resolution or negotiation [55,38,46]. Most of the problems involving the computation of these entities are NP-complete or NP-hard. However, we have been successful in developing efficient heuristics yielding sub-optimal solutions for these problems. The results of experiments on many data sets are very promising. They show very good quality solutions generated by the heuristics in comparison with other methods reported in literature (e.g., with respect to the classification quality of unseen objects). Moreover, they are very time-efficient. It is important to note that the methodology makes it possible to construct heuristics having a very important approximation property. Namely, *expressions generated by heuristics (i.e., implicants) close to prime implicants define approximate solutions for the problem* (see, e.g., [1]).

The rough set approach offers tools for approximate reasoning in multiagent systems (MAS). The typical example is the approximation by one agent of concepts of another agent. The approximation of a concept is based on a decision table representing information about objects perceived by both agents.

The strategies for inducing data models developed so far are often not satisfactory for approximation of compound concepts that occur in the perception process. Researchers from the different areas have recognized the necessity to work on new methods for concept approximation (see, e.g., [11,68]). The main reason for this is that these compound concepts are, in a sense, too far from measurements which makes the searching for relevant features infeasible in a very huge space. There are several research directions aiming at overcoming this difficulty. One of them is based on the interdisciplinary research where the knowledge pertaining to perception in psychology or neuroscience is used to help to deal with compound concepts (see, e.g., [37,22,21]). There is a great effort in neuroscience towards understanding the hierarchical structures of neural networks in living organisms [20,51,37]. Also mathematicians are recognizing problems of learning as the main problem of the current century [51]. These problems are closely related to complex system modeling as well. In such systems again the problem of concept approximation and its role in reasoning about perceptions is one of the challenges nowadays. One should take into account that modeling complex phenomena entails the use of local models (captured by local agents, if one would like to use the multi-agent terminology [34,65,19]) that should be fused afterwards. This process involves negotiations between agents [34,65,19] to resolve contradictions and conflicts in local modeling. This kind of modeling is becoming more and more important in dealing with complex real-life phenomena

which we are unable to model using traditional analytical approaches. The latter approaches lead to exact models. However, the necessary assumptions used to develop them result in solutions that are too far from reality to be accepted. New methods or even a new science therefore should be developed for such modeling [25].

One of the possible approaches in developing methods for compound concept approximations can be based on the layered (hierarchical) learning [62,9]. Inducing concept approximation should be developed hierarchically starting from concepts that can be directly approximated using sensor measurements toward compound target concepts related to perception. This general idea can be realized using additional domain knowledge represented in natural language. For example, one can use some rules of behavior on the roads, expressed in natural language, to assess from recordings (made, e.g., by camera and other sensors) of actual traffic situations, if a particular situation is safe or not (see, e.g., [39,8,7,17]). Hierarchical learning has been also used for identification of risk patterns in medical data and extended for therapy planning (see, e.g. [5,4]). Another application of hierarchical learning for sunspot classification is reported in [40]. To deal with such problems, one should develop methods for concept approximations together with methods aiming at approximation of reasoning schemes (over such concepts) expressed in natural language. The foundations of such an approach, creating a core of perception logic, are based on rough set theory [43,44,45,46,17] and its extension called rough mereology [52,59,42]. Approximate Boolean reasoning methods can be scaled to the case of compound concept approximation.

Let us consider more examples.

The prediction of behavioral patterns of a compound object evaluated over time is usually based on some historical knowledge representation used to store information about changes in relevant features or parameters. This information is usually represented as a data set and has to be collected during long-term observation of a complex dynamic system. For example, in case of road traffic, we associate the object-vehicle parameters with the readouts of different measuring devices or technical equipment placed inside the vehicle or in the outside environment (e.g., alongside the road, in a helicopter observing the situation on the road, in a traffic patrol vehicle). Many monitoring devices serve as informative sensors such as GPS, laser scanners, thermometers, range finders, digital cameras, radar, image and sound converters (see, e.g. [66]). Hence, many vehicle features serve as models of physical sensors. Here are some exemplary sensors: location, speed, current acceleration or deceleration, visibility, humidity (slipperiness) of the road. By analogy to this example, many features of compound objects are often dubbed sensors. In the lecture, we discuss (see also [7]) some rough set tools for perception modelling that make it possible to recognize behavioral patterns of objects and their parts changing over time. More complex behavior of compound objects or groups of compound objects can be presented in the form of *behavioral graphs*. Any behavioral graph can be interpreted as a *behavioral pattern* and can be used as a complex classifier for recognition of

complex behaviours. The complete approach to the perception of behavioral patterns, that is based on behavioral graphs and the dynamic elimination of behavioral patterns, is presented in [7]. The tools for dynamic elimination of behavioral patterns are used for switching-off in the *system attention* procedures searching for identification of some behavioral patterns. The developed rough set tools for perception modeling are used to model networks of classifiers. Such networks make it possible to recognize behavioral patterns of objects changing over time. They are constructed using an ontology of concepts provided by experts that engage in approximate reasoning on concepts embedded in such an ontology. Experiments on data from a vehicular traffic simulator [3] show that the developed methods are useful in the identification of behavioral patterns.

The following example concerns human computer-interfaces that allow for a dialog with experts to transfer to the system their knowledge about structurally compound objects. For pattern recognition systems [18], e.g., for Optical Character Recognition (OCR) systems it will be helpful to transfer to the system a certain knowledge about the expert view on border line cases. The central issue in such pattern recognition systems is the construction of classifiers within vast and poorly understood search spaces, which is a very difficult task. Nonetheless, this process can be greatly enhanced with knowledge about the investigated objects provided by an human expert. We developed a framework for the transfer of such knowledge from the expert and for incorporating it into the learning process of a recognition system using methods based on rough mereology (see, e.g., [41]). Is is also demonstrated how this knowledge acquisition can be conducted in an interactive manner, with a large dataset of handwritten digits as an example.

The next two examples are related to approximation of compound concepts in reinforcement learning and planning.

In temporal difference reinforcement learning [63,16,36,28,60,47,48,50,49,71,72], the main task is to learn the approximation of the function $Q(s,a)$, where $s,a$ denotes a global state of the system and an action performed by an agent $ag$ and, respectively and the real value of $Q(s,a)$ describes the reward for executing the action $a$ in the state $s$. In approximation of the function $Q(s,a)$, probabilistic methods are used. However, for compound real-life problems it may be hard to build such models for such a compound concept as $Q(s,a)$ [68]. We propose another approach to the approximation of $Q(s,a)$ based on ontology approximation. The approach is based on the assumption that in a dialog with experts an additional knowledge can be acquired making it possible to create a ranking of values $Q(s,a)$ for different actions $a$ in a given state $s$. In the explanation given by expert about possible values of $Q(s,a)$ concepts from a special ontology are used. Then, using this ontology one can follow hierarchical learning methods to learn approximations of concepts from ontology. Such concepts can have a temporal character too. This means that the ranking of actions may depend not only on the actual action and the state but also on actions performed in the past and changes caused by these actions.

In [5,4] a computer tool based on rough sets for supporting automated planning of the medical treatment (see, e.g., [26,67]) is discussed. In this approach, a given patient is treated as an investigated complex dynamical system, whilst diseases of this patient (RDS, PDA, sepsis, Ureaplasma and respiratory failure) are treated as compound objects changing and interacting over time. As a measure of planning success (or failure) in experiments, we use a special hierarchical classifier that can predict the similarity between two plans as a number between 0.0 and 1.0. This classifier has been constructed on the basis of the special ontology specified by human experts and data sets. It is important to mention that besides the ontology, experts provided the exemplary data (values of attributes) for the purpose of concepts approximation from the ontology. The methods of construction such classifiers are based on approximate reasoning schemes (AR schemes, for short) and were described, e.g., in [8,39,8,7]. We applied this method for approximation of similarity between plans generated in automated planning and plans proposed be human experts during the realistic clinical treatment.

Further radical changes in the design of intelligent systems depend on the advancement of technology to acquire, represent, store, process, discover, communicate and learn wisdom. We call this technology *wisdom technology* (or **wistech**, for short) [27]. The term *wisdom* commonly means "judging rightly". This common notion can be refined. By *wisdom*, we understand an adaptive ability to make judgements correctly to a satisfactory degree (in particular, correct decisions) having in mind real-life constraints.

One of the basic objectives is to indicate the methods for potential directions for the design and implementation of wistech computation models. An important aspect of wistech is that the complexity and uncertainty of real-life constraints mean that in practise we must reconcile ourselves to the fact that our judgements are based on non-crisp concepts and which do not take into account all the knowledge accumulated and available to us. This is also why consequences of our judgements are usually imperfect. But as a consolation, we also learn to improve the quality of our judgements via observation and analysis of our experience during interaction with the environment. Satisfactory decision-making levels can be achieved as a result of improved judgements.

The intuitive nature of wisdom understood in this way can be expressed metaphorically as shown in *wisdom equation* (1)

$$wisdom = KSN + AJ + IP, \tag{1}$$

where *KSN, AJ, IP* denote *knowledge sources network, adaptive judgement*, and *interactive processes*, respectively. The combination of the technologies represented in (1) offers an intuitive starting point for a variety of approaches to designing and implementing computational models for wistech. We focus in the research on an adaptive RGC approach.

The issues we discuss on wistech are relevant for the other reported current research directions (see, e.g., [14,13,21,22,30,54,64] and the literature cited in these articles).

Wistech can be perceived as the integration of three technologies (corresponding to three components in the wisdom equation (1)). At the current stage the

following two of them seem to be conceptually relatively clear: (i) *knowledge sources network* – by knowledge we traditionally understand every organized set of information along with the inference rules; (ii) *interactive processes* – interaction is understood here as a sequence of stimuli and reactions over time. Far more difficult conceptually seems to be the concept of (iii) *adaptive judgement* distinguishing wisdom from the general concept of problem solving. Adaptive judgement is understood here as mechanisms in a metalanguage (meta-reasoning) which on the basis of selection of available sources of knowledge and on the basis of understanding of history of interactive processes and their current status are enable to perform the following activities under real life constraints: (i) identification and judgement of importance (for future judgement) of phenomena available for observation in the surrounding environment; (ii) planning current priorities for actions to be taken (in particular, on the basis of understanding of history of interactive processes and their current status) toward making optimal judgements; (iii) selection of fragments of ordered knowledge (hierarchies of information and judgement strategies) satisfactory for making decision at the planned time (a decision here is understood as a commencing interaction with the environment or as selecting the future course to make judgements); (iv) prediction of important consequences of the planned interaction of processes; (v) learning and, in particular, reaching conclusions from experience leading to adaptive improvement in the adaptive judgement process.

One of the main barriers hindering an acceleration in the development of wistech applications lies in developing satisfactory computation models implementing the functioning of "adaptive judgement". This difficulty primarily consists of overcoming the complexity of the process of integrating the local assimilation and processing of changing non-crisp and incomplete concepts necessary to make correct judgements. In other words, we are only able to model tested phenomena using local (subjective) models and interactions between them. In practical applications, usually, we are not able to give global models of analyzed phenomena (give quotes from MAS and complex adaptive systems (CAS); see, e.g., [65,32,33,19,15]). However, we one can approximate global models by integrating the various incomplete perspectives of problem perception. One of the potential computation models for "adaptive judgement" might be the RGC approach.

The research on the foundations on wistech is based on a continuation of approaches to computational models of approximate reasoning developed by Rasiowa (see [53]), Pawlak (see [43]) and their students. In some sense, it is a continuation of ideas initiated by Leibniz, Boole and currently continued in a variety of forms. Of course, the Rasiowa - Pawlak school is also some kind continuation of the Polish School of Mathematics and Logics which led to the development of the modern understanding of the basic computational aspects of logic, epistemology, ontology, foundations of mathematics and natural deduction. The two fundamental tools of the Rasiowa - Pawlak school are the following: (i) *Computation models of logical concept (especially such concepts as deduction or algebraic many-valued models for classical, modal, and constructive mathematics)* - based on the method of treating the sets of logically equivalent statements

(or formulas) as abstract algebras known as Lindebaum - Tarski algebras; (ii) *Computation models of vague concept*- originally Łukasiewicz has proposed to treat uncertainty (or vague concepts) as concepts of many valued logic. The rough set concept, due to Pawlak [43], developed in the Rasiowa-Pawlak school is based on classical two valued logic. The rough set approach has been developed to deal with uncertainty and vagueness. The approach makes it possible to reason precisely about approximations of vague concepts. These approximations are temporary, subjective, and are adaptively changing with changes in environments [6,57,60].

Solving complex problems by multi-agent systems requires new approximate reasoning methods based on new computing paradigms. One such recently emerging computing paradigm is RGC. Computations in RGC are performed on information granules representing often vague, partially specified, and compound concepts delivered by agents engaged in tasks such as knowledge representation, communication with other agents, and reasoning.

One of the RGC challenges is to develop approximate reasoning techniques for reasoning about dynamics of distributed systems of judges, i.e., agents judging rightly. These techniques should be based on systems of evolving local perception logics rather than on a global logic [56,58]. The approximate reasoning about global behavior of judge's system is infeasible without methods for approximation of compound vague concepts and approximate reasoning about them. One can observe here an analogy to phenomena related to the emergent patters in complex adaptive systems [15]. Let us observe that judges can be organized into a hierarchical structure, i.e., one judge can represent a coalition of judges in interaction with other agents existing in the environment [2,29,32]. Such judges representing coalitions play an important role in hierarchical reasoning about behavior of judges populations. Strategies for coalition formation and cooperation [2,32,33] are of critical importance in designing systems of judges with dynamics satisfying to a satisfactory degree the given specification. Developing strategies for discovery of information granules representing relevant coalitions and cooperation protocols is another challenge for RGC.

All these problems can be treated as problems of searching for information granules satisfying vague requirements. The strategies for construction of information granules should be adaptive. It means that the adaptive strategies should make it possible to construct information granules satisfying constraints under dynamically changing environment. This requires reconstruction or tuning of already constructed information granules which are used as components of data models, e.g., classifiers. In the adaptive process, the construction of information granules generalizing some constructed so far information granules plays a special role. The mechanism for relevant generalization here is crucial. One can imagine for this task many different strategies, e.g., based on adaptive feedback control for tuning the generalization. Cooperation with specialists from different areas such as neuroscience (see, e.g., [37] for visual objects recognition), psychology (see, e.g., [51] for discovery of mechanisms for hierarchical perception), biology (see, e.g., [10] for cooperation based on swarm intelligence), adaptive learning

based on ethology and approximation spaces [48,50] or social science (see, e.g., [32] for modeling of agents behavior) can help to discover such adaptive strategies for extracting sub-optimal (relative to the minimal length principle) data models satisfying soft constraints. This research may also help us to develop strategies for discovery of ontologies relevant for compound concept approximation.

In the current projects, we are developing rough set based methods in combination with other soft computing and statistical methods for RGC on which wistech can be based. The developed methods are used to construct wisdom engines. By wisdom engine we understand a system which implements the concept of wisdom. We plan to design specific systems for some tasks such as (1) Intelligent Document Manager; (2) Job Market Search; (3) Brand Monitoring; (4) Decision Support for global management systems (e.g., World Forex, Stock Market, World Tourist); (5) Intelligent Assistant (e.g., Physician, Lawyer); (6) Discovery of Processes from Data (e.g., Gene Expression Networks); (7) Rescue System (for more details see [27,19]).

**Acknowledgement.** The research has been supported by the grant 3 T11C 002 26 from Ministry of Scientific Research and Information Technology of the Republic of Poland.

Many thanks to Anna Gomolińska, and Tuan Trung Nguyen, James Peters, for their incisive comments and for suggesting many helpful ways to improve this article.

## References

1. *Rough Set Exploration System (RSES).* Available at: `logic.mimuw.edu.pl/~rses`.
2. R. M. Axelrod. *The Complexity of Cooperation.* Princeton University Press, Princeton, NJ, 1997.
3. J. Bazan. *The Road simulator*
   Available at: `logic.mimuw.edu.pl/~bazan/simulator`.
4. J. Bazan, P. Kruczek, S. Bazan-Socha, A. Skowron, and J. J. Pietrzyk. Automatic planning of treatment of infants with respiratory failure through rough set modeling. In *Proceedings of RSCTC'2006*, LNAI. Springer, Heidelberg, 2006. to be published.
5. J. Bazan, P. Kruczek, S. Bazan-Socha, A. Skowron, and J. J. Pietrzyk. Risk pattern identification in the treatment of infants with respiratory failure through rough set modeling. In *Proceedings of IPMU'2006, Paris, France, July 2-7, 2006*, pages 2650–2657. Éditions E.D.K., Paris, 2006.
6. J. Bazan, A. Skowron, and R. Swiniarski. Rough sets and vague concept approximation: From sample approximation to adaptive learning. *Transactions on Rough Sets V: LNCS Journal Subline, Springer, Heidleberg*, LNCS 4100:39–62, 2006.
7. J. G. Bazan, J. F. Peters, and A. Skowron. Behavioral pattern identification through rough set modelling. In Ślęzak et al. [61], pages 688–697.
8. J. G. Bazan and A. Skowron. Classifiers based on approximate reasoning schemes. In Dunin-Kęplicz et al. [19], pages 191–202.
9. S. Behnke. *Hierarchical Neural Networks for Image Interpretation*, volume 2766 of *LNCS*. Springer, Heidelberg, 2003.

10. E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence. From Natural to Artificial Systems*. Oxford University Press, Oxford, UK, 1999.
11. L. Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231, 2001.
12. F. Brown. *Boolean Reasoning*. Kluwer Academic Publishers, Dordrecht, 1990.
13. N. L. Cassimatis. A cognitive substrate for achievinbg human-level intelligence. *AI Magazine*, 27:45–56, 2006.
14. N. L. Cassimatis, E. T. Mueller, and P. H. Winston. Achieving human-level intelligence through integrated systems and research. *AI Magazine*, 27:12–14, 2006.
15. A. Desai. Adaptive complex enterprices. *Comm. ACM*, 48:32–35, 2005.
16. T. G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Artificial Intelligence*, 13(5):227–303, 2000.
17. P. Doherty, W. Łukaszewicz, A. Skowron, and A. Szałas. *Knowledge Representation Techniques : A Rough Set Approach*, volume 202 of *Studies in Fuzziness and Soft Computing*. Springer, Heidelberg, Germany, 2006.
18. R. Duda, P. Hart, and R. Stork. *Pattern Classification*. John Wiley & Sons, New York, NY, 2002.
19. B. Dunin-Kęplicz, A. Jankowski, A. Skowron, and M. Szczuka, editors. *Monitoring, Security, and Rescue Tasks in Multiagent Systems (MSRAS'2004)*. Advances in Soft Computing. Springer, Heidelberg, 2005.
20. M. Fahle and T. Poggio. *Perceptual Learning*. The MIT Press, Cambridge, MA, 2002.
21. K. D. Forbus and T. R. Hinrisch. Companion congnitive systems: A step toward human-level ai. *AI Magazine*, 27:83–95, 2006.
22. K. D. Forbus and T. R. Hinrisch. Engines of the brain: The computational instruction set of human cognition. *AI Magazine*, 27:15–31, 2006.
23. G. Frege. *Grundgesetzen der Arithmetik, 2*. Verlag von Hermann Pohle, Jena, 1903.
24. J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, Heidelberg, 2001.
25. M. Gell-Mann. *The Quark and the Jaguar - Adventures in the Simple and the Complex*. Brown and Co., London, 1994.
26. M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Elsevier, Morgan Kaufmann, CA, 2004.
27. A. Jankowski and A. Skowron. A wistech paradigm for intelligent systems. *Transactions on Rough Sets VI: LNCS Journal Subline, Springer, Heidleberg*, LNCS 4374:94-132, 2007.
28. L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:227–303, 1996.
29. S. Kraus. *Strategic Negotiations in Multiagent Environments*. The MIT Press, Massachusetts, 2001.
30. P. Langley. Cognitive architectures and general intelligent systems. *AI Magazine*, 27:33–44, 2006.
31. S. Leśniewski. Grungzüge eines neuen Systems der Grundlagen der Mathematik. *Fundamenta Mathematicae*, 14:1–81, 1929.
32. J. Liu. *Autonomous Agents and Multi-Agent Systems: Explorations in Learning, Self-Organization and Adaptive Computation*. World Scientific Publishing, Singapore, 2001.
33. J. Liu, X. Jin, and K. C. Tsui. *Autonomy Oriented Computing: From Problem Solving to Complex Systems Modeling*. Kluwer/Springer, Heidelberg, 2005.
34. M. Luck, P. McBurney, and C. Preist. *Agent Technology. Enabling Next Generation Computing: A Roadmap for Agent Based Computing*. AgentLink, 2003.

35. J. Łukasiewicz. Die logischen Grundlagen der Wahrscheinlichkeitsrechnung, Kraków1913. In L. Borkowski, editor, *Jan Łukasiewicz - Selected Works*, pages 16–63. North Holland & Polish Scientific Publishers, Amsterdam, London, Warsaw, 1970.
36. A. McGovern. *Autonomous Discovery of Temporal Abstractions from Interaction with an Environment*. PhD thesis, University of Massachusetts, Amherst, 2002.
37. R. Miikkulainen, J. A. Bednar, Y. Choe, and J. Sirosh. *Computational Maps in the Visual Cortex*. Springer, Hiedelberg, 2005.
38. H. S. Nguyen. Approximate boolean reasoning: Foundations and applications in data mining. In J. F. Peters and A. Skowron, editors, *Transactions on Rough Sets V: Journal Subline, Springer, Heidelberg*, LNCS 4100:344–523, 2006.
39. H. S. Nguyen, J. Bazan, A. Skowron, and S. H. Nguyen. Layered learning for concept synthesis. *Transactions on Rough Sets I: LNCS Journal Subline, Springer, Heidleberg*, LNCS 3100:187–208, 2004.
40. S. H. Nguyen, T. T. Nguyen, and H. S. Nguyen. Rough set approach to sunspot classification. In Ślęzak et al. [61], pages 263–272.
41. T. T. Nguyen and A. Skowron. Rough set approach to domain knowledge approximation. In G. Wang, Q. Liu, Y. Yao, and A. Skowron, editors, *Proceedings of the 9-th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2003), Chongqing, China, Oct 19-22, 2003*, volume 2639 of *LNCS*, pages 221–228. Springer, Heidelberg.
42. S. K. Pal, L. Polkowski, and A. Skowron, editors. *Rough-Neural Computing: Techniques for Computing with Words*. Cognitive Technologies. Springer, Heidelberg, 2004.
43. Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*, volume 9 of *System Theory, Knowledge Engineering and Problem Solving*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.
44. Z. Pawlak and A. Skowron. Rudiments of rough sets. *Information Sciences* 177(1) (2007) 3-27.
45. Z. Pawlak and A. Skowron. Rough sets: Some extensions. *Information Sciences* 177(1) (2007) 28-40.
46. Z. Pawlak and A. Skowron. Rough sets and boolean reasoning. *Information Sciences* 177(1) (2007) 41-73.
47. J. F. Peters. Approximation spaces for hierarchical intelligent behavioural system models, In: B.D.-Kepliçz, A. Jankowski, A. Skowron, M. Szczuka (Eds.), *Monitoring, Security and Rescue Techniques in Multiagent Systems*, Advances in Soft Computing, pages 13–30, Physica-Verlag, Heidelberg, 2004.
48. J. F. Peters. Rough ethology: Towards a biologically-inspired study of collective behaviour in intelligent systems with approximation spaces. *Transactions on Rough Sets III: LNCS Journal Subline, Springer, Heidleberg*, LNCS 3400:153–174, 2005.
49. J.F. Peters, C. Henry, S. Ramanna. Rough ethograms: A study of intelligent system behavior. In: Mieczyslaw A. Kłopotek, Sławomir Wierzchoń, Krzystof Trojanowski(Eds.), Intelligent Information Systems. *Advances in Soft Computing*. Springer-Verlag, Heidelberg, (2005) 117-126
50. J. F. Peters, C. Henry. Reinforcement learning with approximation spaces. *Fundamenta Informaticae* **71** (2-3) (2006) 323-349.
51. T. Poggio and S. Smale. The mathematics of learning: Dealing with data. *Notices of the AMS*, 50(5):537–544, 2003.
52. L. Polkowski and A. Skowron. Rough mereology: A new paradigm for approximate reasoning. *International Journal of Approximate Reasoning*, 15(4):333–365, 1996.

53. H. Rasiowa. *Algebraic models of logics*. Warsaw University, Warsaw, 2001.
54. C. Schlenoff, J. Albus, E. Messina, A. J. Barbera, R. Madhavan, and S. Balakirsky. Using 4d/rcs to address ai knowledge integration. *AI Magazine*, 27:71–81, 2006.
55. A. Skowron. Rough sets in KDD (plenary talk). In Z. Shi, B. Faltings, and M. Musen, editors, *16-th World Computer Congress (IFIP'2000): Proceedings of Conference on Intelligent Information Processing (IIP'2000)*, pages 1–14. Publishing House of Electronic Industry, Beijing, 2000.
56. A. Skowron. Perception logic in intelligent systems. In S. Blair et al, editor, *Proceedings of the 8th Joint Conference on Information Sciences (JCIS 2005), July 21-26, 2005, Salt Lake City, Utah, USA*, pages 1–5. X-CD Technologies: A Conference & Management Company, ISBN 0-9707890-3-3, Toronto, Ontario, Canada, 2005.
57. A. Skowron. Rough sets and vague concepts. *Fundamenta Informaticae*, 64(1-4):417–431, 2005.
58. A. Skowron. Rough sets in perception-based computing (keynote talk). In S. K. Pal, S. Bandoyopadhay, and S. Biswas, editors, *First International Conference on Pattern Recognition and Machine Intelligence (PReMI'05) December 18-22, 2005, Indian Statistical Institute, Kolkata*, volume 3776 of *LNCS*, pages 21–29, Heidelberg, 2005. Springer.
59. A. Skowron and J. Stepaniuk. Information granules and rough-neural computing. In Pal et al. [42], pages 43–84.
60. A. Skowron, J. Stepaniuk, J. F. Peters, and R. Swiniarski. Calculi of approximation spaces. *Fundamenta Informaticae*, 72(1-3):363–378, 2006.
61. D. Ślęzak, J. T. Yao, J. F. Peters, W. Ziarko, and X. Hu, editors. *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'2005), Regina, Canada, August 31-September 3, 2005, Part II*, volume 3642 of *LNAI*. Springer, Heidelberg, 2005.
62. P. Stone. *Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer*. The MIT Press, Cambridge, MA, 2000.
63. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.
64. W. Swartout, J. Gratch, R. W. Hill, E. Hovy, S. Marsella, J. Rickel, and D. Traum. Towards virtual humans. *AI Magazine*, 27:96–108, 2006.
65. K. Sycara. Multiagent systems. *AI Magazine*, pages 79–92, Summer 1998.
66. C. Urmson, J. Anhalt, M. Clark, T. Galatali, J. P. Gonzalez, J. Gowdy, A. Gutierrez, S. Harbaugh, M. Johnson-Roberson, H. Kato, P. L. Koon, K. Peterson, B. K. Smith, S. Spiker, E. Tryzelaar, and W. R. L. Whittaker. High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004. Technical Report CMU-RI-TR-04-37, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, June 2004.
67. W. Van Wezel, R. Jorna, and A. Meystel. *Planning in Intelligent Systems: Aspects, Motivations, and Methods*. John Wiley & Sons, Hoboken, New Jersey, 2006.
68. V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, NY, 1998.
69. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
70. L. A. Zadeh, A new direction in AI: Toward a computational theory of perceptions, AI Magazine 22 (1) (2001) 73-84.
71. C.J.C.H. Watkins. *Learning from Delayed Rewards*, Ph.D. Thesis, supervisor: Richard Young, King's College, University of Cambridge, UK, May, 1989.
72. P. Wawrzyński. *Intensive Reinforcement Learning*, Ph.D. dissertation, supervisor: Andrzej Pacut, Institute of Control and Computational Engineering, Warsaw University of Technology, May 2005.

# Data Clustering Algorithms for Information Systems

Sadaaki Miyamoto

Department of Risk Engineering
Faculty of Systems and Information Engineering
University of Tsukuba, Ibaraki 305-8573, Japan
miyamoto@risk.tsukuba.ac.jp

**Abstract.** Although the approaches are fundamentally different, the derivation of decision rules from information systems in the form of tables can be compared to supervised classification in pattern recognition; in the latter case classification rules should be derived from the classes of given points in a feature space. We also notice that methods of unsupervised classification (in other words, data clustering) in pattern recognition are closely related to supervised classification techniques. This observation leads us to the discussion of clustering for information systems by investigating relations between the two methods in the pattern classification. We thus discuss a number of methods of data clustering of information tables without decision attributes on the basis of rough set approach in this paper. Current clustering algorithms using rough sets as well as new algorithms motivated from pattern classification techniques are considered. Agglomerative clustering are generalized into a method of poset-valued clustering for discussing structures of information systems using new notations in relational databases. On the other hand $K$-means algorithms are developed using the kernel function approach. Illustrative examples are given.

**Keywords:** Information system; agglomerative clustering; $K$-means algorithms; kernel function.

## 1 Introduction

Although rough sets [20,21,22] have been studied as a new methodology to investigate uncertainties with applications to classification rules for information systems, there is an important feature that is left unnoticed by many researchers, that is, clustering of information systems and related techniques. To understand a motivation for this subject, consider methods [4] of pattern classification for the moment, where many algorithms of automatic classification have been developed. Although many methods therein are for supervised classification, there is another class of unsupervised classification that is also called clustering. We moreover observe a loose coupling between a method of supervised classification and an algorithm of clustering. For example, the nearest neighbor classification

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 13–24, 2007.

is related to the single link clustering [5] and a nearest prototype classification can be associated with the $K$-means clustering [1,4].

A problem in rough set studies is thus to investigate whether and/or how methods in pattern clustering can be applied to unsupervised classification in rough sets or information systems, or to obtain new algorithms in rough sets by observing features in methods of pattern classification.

In this paper we will see a number of existing and new methods of clustering for information systems. Agglomerative clustering are generalized into a method of poset-valued clustering for discussing structures of information systems. Moreover different $K$-means algorithms are developed; one of them uses the kernel function approach in support vector machines [25,26,3].

## 2 Existing Studies of Clustering in Rough Sets

We assume the set of objects for clustering is denoted by $X = \{x_1, \ldots, x_n\}$ and a generic object in $X$ is also denoted by $x \in X$.

We notice two types of foregoing studies of clustering in the presence of rough sets, that is, a series of studies in rough $K$-means [8] and rough $K$-medoids [23], and algorithms for rough clustering [6].

To describe the methods below, we first show a generalized $K$-means algorithm in which the distance between two objects is denoted by $d(x, y)$.

**GKM: A generalized algorithm of $K$-means clustering.**

**GKM1.** Give initial cluster centers $v_1, \ldots, v_K$. Let the cluster represented by $v_i$ be $G_i$ or $G(v_i)$.

**GKM2 (nearest prototype classification).** Reallocate each object $x$ to the nearest center $v_i$:

$$i = \arg \min_{1 \leq j \leq K} d(x, v_j).$$

**GKM3.** After all objects are reallocated, update the cluster center:

$$v_i = \arg \min_{v} \sum_{x_k \in G_i} d(x_k, v). \tag{1}$$

**GKM4.** Check the convergence criterion. If not convergent, go to **GKM2**.
**End of GKM.**

The criterion for the convergence is omitted here. They are given in standard literature [1,9,12].

Notice that in the case of the Eulidean distance $d(x, y) = \|x-y\|^2$, equation (1) is reduced to the centroid

$$v_i = \frac{1}{|G_i|} \sum_{x_k \in G_i} x_k, \tag{2}$$

where $|G_i|$ is the number of elements in $G_i$.

The method of rough $K$-means [8] is an adaptation of the ordinary $K$-means to rough approximations. It has the next two features:

1. The upper bound and the lower bound of a cluster represented by a center are defined using a *threshold*: if $v$ is the nearest center to object $x$ and if there is another center $v'$ such that $d(x,v') - d(x,v) \leq threshold$ then $x$ belong to the upper approximations of the two clusters represented by $v$ and $v'$ (i.e. $x \in \overline{G}(v)$ and $x \in \overline{G}(v')$); if there is no such $v'$, then $x$ belongs to the lower approximation of $v$ (i.e., $x \in \underline{G}(v)$).
2. The calculation of $v$ takes two weight parameters $w_{\text{lower}}$ and $w_{\text{upper}}$ for the objects in the lower and upper approximations.

The rough medoids [23] is another adaptation of the ordinary $K$-medoids algorithm [7] to rough sets. Note that a $K$-medoids technique is similar to the $K$-means in the sense that the nearest prototype classification in **GKM2** is used, but instead of the centroid (2), an object $\hat{x}_i$ represents a cluster $G_i$, i.e.,

$$\hat{x}_i = \arg\min_{x \in X} \sum_{x' \in G_i} d(x', x). \tag{3}$$

An advantage of the $K$-medoids is that the representative object summarizes information for the cluster and hence useful in many applications; on the other hand a drawback is that computation of the medoids is much more complex than the $K$-means. Notice also that the Euclidean distance is assumed in the rough $K$-means.

The method of rough medoids [23] is thus the combination of the idea of the $K$-medoids and the rough $K$-means. It uses the nearest medoid allocation with the threshold that determines the upper approximation $\overline{G}(\hat{x})$ and the lower approximation $\underline{G}(\hat{x})$, and moreover the weights $w_{\text{lower}}$ and $w_{\text{upper}}$ are introduced. We omit the details of the algorithm [23], as the idea is now clear.

The rough clustering [19,6] begins with a number of initial equivalence relations $R_i$ $(1 \leq i \leq n)$ that consists of two classes $P_i$ and $X - P_i$ where $P_i$ consists of those objects of which the distance to $x_i$ is less than a given threshold. The intersection $\bigcap_{i=1}^{n} R_i$ of the initial relations defines initial clusters and then a merging procedure to coarser classes starts using a indiscernibility degree $\gamma(x,x')$ defined between an arbitrary pair of objects of which the definition is omitted here (see [6]). Then a new class $P_i'$ is defined: $P_i'$ consists of those objects of which the indiscernibility degree to $x_i$ is above a given threshold. This method is different from $K$-means or $K$-medoids; it is more similar to, but still different from, agglomerative hierarchical clustering. A drawback is that the computation is more complicated than $K$-means and $K$-medoids.

The above two types of the algorithms have been applied to a set of objects in a Euclidean space, and do not assume the information system, while the methods below are designed for analyzing an information table in which a value may be numerical or non-numerical.

## 3 A Method of Poset-Valued Clustering

A generalization of agglomerative hierarchical clustering that is adapted to information systems has been proposed by the author [17]. In this paper we describe this method with an extension and new notations. We begin with a notation in relational databases [24], as an information system can be regarded as a relation.

Let $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$ be a relational schema in which $a_1, a_2, \ldots, a_m$ are attributes. For each attribute $a_i$, we have the corresponding domain $D_i$. An information system or in other words, information table $\mathcal{T}$ is a finite subset of the product $D_1 \times \cdots \times D_m$, or in other words, $\mathcal{T}$ is a relation [24]. An element $t \in \mathcal{T}$ is called a tuple using the term in relational database. Let us assume

$$\mathcal{T} = \{t_1, \ldots, t_n\}$$

and an attribute value of $t$ with respect to $a_i$ is denoted by $t(a_i)$. Thus,

$$t = (t(a_1), \ldots, t(a_m)).$$

### 3.1 A Generalization of Agglomerative Clustering

Let us briefly review a generalization of hierarchical classification that the author has proposed [17]. Note that a family of clusters $\mathcal{G} = \{G_1, \ldots, G_K\}$ is a partition of $X$:

$$\bigcup_{i=1}^{K} G_i = X, \qquad G_i \cap G_j = \emptyset \quad (i \neq j).$$

Assume that an inter-cluster distance is denoted by $d(G_i, G_j)$. Furthermore, we consider a family of clusters that depends on a parameter $\alpha$,

$$\mathcal{G}(\alpha) = \{G_1(\alpha), \ldots, G_K(\alpha)\}.$$

Accordingly the inter-cluster distance is $d(G_i(\alpha), G_j(\alpha))$.

A general procedure of agglomerative clustering is as follows.

**1.** Let the initial clusters be individual objects. Define inter-cluster distances as the distance between the corresponding objects. Let the number of clusters be $K = n$.
**2.** Merge two clusters of the minimum distance. Reduce the number of clusters by 1: $K \leftarrow K - 1$. The minimum value is stored as the level of the merge $m_K$.
**3.** If $K = 1$, stop, else update the distances between the merged cluster and other clusters. Go to step **2**.

There are different ways to update the distances in step **3**, and accordingly we have various methods of agglomerative clustering such as the single link, the complete link, *etc.* [5], which we omit here. We notice, however, that the level $m_K$ is monotone increasing for most well-known methods except the centroid method [11]:

$$m_{n-1} \leq m_{n-2} \leq \cdots \leq m_2 \leq m_1.$$

Assume $\alpha = m_K$. Then the next property is valid.

**Proposition 1.** *For every $\alpha \preceq \alpha'$ and for each $G_i(\alpha') \in \mathcal{G}(\alpha')$ there exists $G_j(\alpha) \in \mathcal{G}(\alpha)$ such that $G_j(\alpha) \subseteq G_i(\alpha')$.*

The proof is easy and omitted. The above property states that the parameter-dependent clusters forms a hierarchical classification. More generally we define a poset-valued hierarchical cluster as follows.

**Definition 1.** *Let $P$ is a poset [10] of which the preorder is defined by $\preceq$. We say $\mathcal{G}(\alpha) = \{G_1(\alpha), \ldots, G_K(\alpha)\}$ $(\alpha \in P)$ is a poset-valued hierarchical classification if for every $\alpha \preceq \alpha'$ and for each $G_i(\alpha') \in \mathcal{G}(\alpha')$ there exists $G_j(\alpha) \in \mathcal{G}(\alpha)$ such that*

$$G_j(\alpha) \subseteq G_i(\alpha').$$

*We write $\mathcal{G}(\alpha) \triangleright \mathcal{G}(\alpha')$ if this property holds.*

*Moreover if the poset has the structure of a lattice [2,10], we say $\mathcal{G}(\alpha)$ is a lattice-valued hierarchical classification.*

Such a poset-valued classification is closely related to information systems. Let us return to the consideration of the information table.

For a given subset $A = (a_{i_1}, \ldots, a_{i_r})$ of the attribute set $\mathcal{A}$, define

$$t(A) = (t(a_{i_1}), \ldots, t(a_{i_r})).$$

For a given set $T(\subset \mathcal{T})$ of tuples, we define

$$T(A) = \{\, t(A) : \ t \in T \,\}.$$

We hence have

$$t \in T \ \Rightarrow \ t(A) \in T(A),$$

while the converse $\Leftarrow$ is not true in general.

For a given subset $A(\subseteq \mathcal{A})$ of attributes, we define a relation $R_A$:

$$tR_At' \iff t(A) = t'(A).$$

It is easy to see that $R_A$ is an equivalence relation.

Note that $\mathcal{A}$ is a lattice in which the natural inclusion of subsets is the pre-ordering of the poset and the union and the intersection are respectively sup and inf operation of the lattice [2,10]: $\sup(A, A') = A \cup A'$ and $\inf(A, A') = A \cap A'$.

We have the quotient set, in other words, a classification

$$\mathcal{G}(A) = \mathcal{T}/R_A = \{\, [t]_{R_A} : \ t \in T \,\} \tag{4}$$

where

$$[t]_{R_A} = \{t' \in \mathcal{T} : tR_At'\} = \{t' \in \mathcal{T} : t(A) = t'(A)\}.$$

We have the next proposition.

**Proposition 2.** *The above defined equivalence relation $R_A$ generates a hierarchical classification. That is, for every pair of subsets $B \supseteq A$, we have $\mathcal{G}(B) \triangleright \mathcal{G}(A)$.*

**Table 1.** An example of an information table

| $\mathcal{T}$ | $D$ | $E$ | $F$ |
|---|---|---|---|
| $t_1$ | $a_1$ | $b_1$ | $c_1$ |
| $t_2$ | $a_1$ | $b_1$ | $c_2$ |
| $t_3$ | $a_1$ | $b_2$ | $c_1$ |
| $t_4$ | $a_1$ | $b_2$ | $c_2$ |
| $t_5$ | $a_2$ | $b_1$ | $c_1$ |
| $t_6$ | $a_2$ | $b_1$ | $c_2$ |
| $t_7$ | $a_2$ | $b_2$ | $c_1$ |

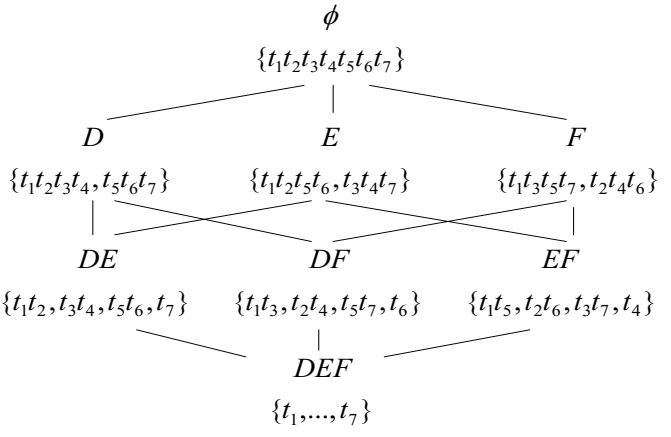



**Fig. 1.** An example of the lattice-valued clustering

Let us consider a simple example.

*Example 1.* Consider seven tuples shown in Table 1 with the schema $\mathcal{A} = (D, E, F)$. Here these three letters are attributes. The lattice is $\Lambda = 2^{\mathcal{A}} = \{\emptyset, D, E, F, DE, DF, FE, DEF\}$ where the abbreviated symbol $DE$ implies $\{D, E\}$, and so on. We have

$$\mathcal{G}(DEF) = \mathcal{T}/R_{DEF} = \{t_1, \ldots, t_7\},$$
$$\mathcal{G}(DE) = \mathcal{T}/R_{DE} = \{t_1t_2, t_3t_4, t_5t_6, t_7\}$$

*etc.* where $t_it_j$ is an abbreviated symbol for $\{t_i, t_j\}$.

Figure 1 shows the Hasse diagram of $\Lambda = 2^{\mathcal{A}}$ together with the partitions attached to each element of the lattice.

Proposition 2 and Example 1 show a relation between the generalized hierarchical clustering and an information system. A cluster in a lattice diagram shows a class of indistinguishable objects (tuples) given the corresponding subset of attributes. Although the diagram with the clusters are without a decision attribute, we can extend the diagram to the case of a decision table.

*Example 2.* Consider the same information table except that a decision attribute $d$ is added. The lattice-valued clusters are shown in Fig. 2 in which the underline shows those objects with the positive mark. We easily observe the exact decision rule: $[D = a_1 \Rightarrow d = 1]$.

**Table 2.** An information table with the decision attribute $d$

| $\mathcal{T}$ | $D$ | $E$ | $F$ | $d$ |
|---|---|---|---|---|
| $t_1$ | $a_1$ | $b_1$ | $c_1$ | 1 |
| $t_2$ | $a_1$ | $b_1$ | $c_2$ | 1 |
| $t_3$ | $a_1$ | $b_2$ | $c_1$ | 1 |
| $t_4$ | $a_1$ | $b_2$ | $c_2$ | 1 |
| $t_5$ | $a_2$ | $b_1$ | $c_1$ | 0 |
| $t_6$ | $a_2$ | $b_1$ | $c_2$ | 0 |
| $t_7$ | $a_2$ | $b_2$ | $c_1$ | 0 |

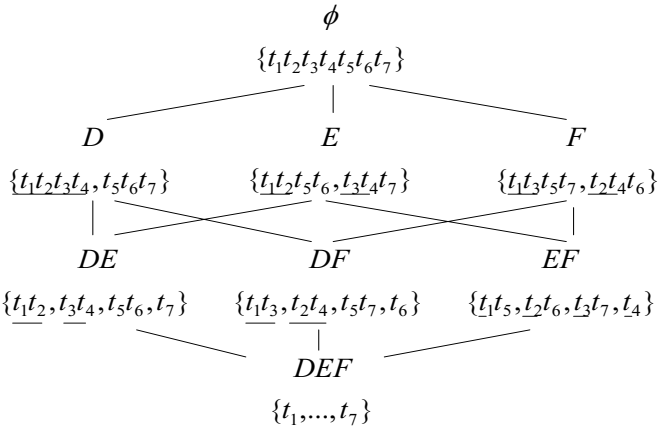



**Fig. 2.** Lattice-valued clustering with underlines to the positive class

In real applications, a Hasse diagram is huge and a figure like Figures 1 and 2 cannot be written. In such cases, a part of the Hasse diagram should be observed of which the detail are given in [17].

### 3.2 Distance and the Poset-Valued Clustering

As most clustering algorithms are based on a distance or similarity measure between two objects, we introduce a distance into attribute values in an information table. We assume a distance measure $d_i(x, y)$ for $x, y \in D_i$, and accordingly the distance between two objects as tuples is defined for a given $B \in \mathcal{A}$:

$$d(t, t'; B) = \sum_{A_i \in B} d_i(t(A_i), t'(A_i)). \tag{5}$$

A simple method based on a distance is to use connected components generated from the network of the vertices $\mathcal{T}$ and with the weight $d(t,t';B)$ on the edge $\{t,t'\}$, and the threshold $\epsilon > 0$, which is defined as follows.

1. Consider first the complete graph whose vertices are all tuples of $\mathcal{T}$. Put the value $d(t,t';B)$ on the edge $\{t,t'\}$.
2. Delete all those edges $\{t,t'\}$ which satisfy $d(t,t';B) > \epsilon$ .
3. Let the obtained connected components be $G_1^\epsilon(B),\ldots,G_K^\epsilon(B)$ of which the set of vertices are $V(G_1^\epsilon(B)),\ldots,V(G_K^\epsilon(B))$, respectively.

We define the equivalence relation:

$$tR_B^\epsilon t' \iff t,t' \in V(G_j^\epsilon).$$

It is obvious that equivalence classes are generated from this definition [11], in other words, we are considering

$$\mathcal{G}(B) = \{V(G_1^\epsilon(B)),\ldots,V(G_K^\epsilon(B))\}.$$

We have the following properties of which the proofs are omitted here [17].

**Proposition 3.** *The equivalence relation $R_B^\epsilon$ generates a hierarchical classification. That is, we have*

$$B \supseteq A \;\Rightarrow\; \mathcal{G}(B) \triangleright \mathcal{G}(A).$$

**Proposition 4**
*Let*

$$d_i(t(a_i),t'(a_i)) = \begin{cases} 0 & (t(a_i) = t'(a_i)), \\ 1 & (t(a_i) \neq t'(a_i)). \end{cases} \tag{6}$$

*and assume $0 < \epsilon < 1$. Then the generated hierarchical classification is the same as (4).*

## 4 *K*-Means Algorithms for Information Systems

The method in the last section is comparable to the rough clustering in that a cluster center is not used therein, while we consider a family of $K$-means type methods for an information system as a table.

The basic algorithm **GKM** of $K$-means works with adequate modifications. Note first that the $K$-medoids algorithm can be used without a modification, since equation (3) is applicable for any distance.

We proceed to consider $K$-means algorithm. Since a distance is not Euclidean in general, the centroid solution (2) cannot be used for (1). We assume that the values $t(a)$ are non-numerical, i.e., $D_i$ consists of finite symbols. We have two methods for a general distance $d(x,y)$. First is a $K$-mode calculation instead of the $K$-means; the second is the use of a kernel function [25].

### 4.1 *K*-Mode Algorithm

Assume that the distance is given by (6). Then the idea of the $K$-means is reduced to a $K$-mode algorithm. Let $\#(z, B)$ be the number of the symbol $z$ in $B$, in other words,

$$\#(z, B) = |\{x \in \bigcup_{i=1}^{n} D_i : x = z\} \cap B|,$$

where $|A|$ is the number of elements in $A$. We moreover put

$$md(B; D_i) = \max_{z \in D_i} \#(z, B).$$

We have the next proposition of which the proof is easy and omitted.

**Proposition 5** *Let us apply the algorithm* **GKM** *to the information table where the distance is defined by (6). Then* $v_i = (v_i^1, \ldots, v_i^m)$ *in* **GKM3** *is given by*

$$v_i^j = md(G_i, D_j).$$

### 4.2 Kernel-Based Algorithm

A reason why the Euclidean space is useful is that it is an inner product space. Although the distance $d(x, y)$ for tuples is not a Euclidean distance, there is a way to define an inner product space by using kernel functions [25], in other words, an implicit mapping $x \mapsto \Phi(x)$ into a high-dimensional feature space which has the inner product represented by a kernel

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle.$$

There are many studies on the use of kernel functions for data analysis [3,25,26]. Note that the mapping $\Phi(x)$ itself is unknown but the form of the kernel $K(x, y)$ is assumed to be given.

We consider if the method of a kernel function can be used for the information table with non-numerical values.

To this end we define a distance-preserving embedding of the attribute values into a Euclidean space. Assume $D_i = \{z_i^1, \ldots, z_i^q\}$ and note that the distance is given by $d_i(z, z')$ for $z, z' \in D_i$.

**Definition 2.** *We say the space* $(D_i, d_i)$ *has an exact Euclidean embedding if there exists a Euclidean space* $R^L$ *and a mapping* $\Psi \colon D_i \to R^L$ *such that*

$$\|\Psi(z) - \Psi(z')\| = d_i(z, z')$$

*for all* $z, z' \in D_i$.

$D_i$ with the distance (6) has an exact Euclidean embedding with $L = q$. To see this, let

$$\Psi(z_i^1) = \frac{1}{\sqrt{2}}(1, 0, \ldots, 0), \quad \Psi(z_i^2) = \frac{1}{\sqrt{2}}(0, 1, 0, \ldots, 0), \quad \cdots$$

When the space $(D_i, d_i)$ $(1 \leq i \leq n)$ has an exact Euclidean embedding, then the kernel-based methods are applicable. To see this, note first that

$$K(x, y) = \exp(-\lambda \|x - y\|^{\beta})$$

is a kernel for $\lambda > 0$ and $0 \leq \beta \leq 2$ and that if $K_i(x, y)$ $(1 \leq i \leq n)$ is a kernel, then $K(x, y) = \prod_{i=1}^{n} K_i(x, y)$ is also a kernel [25].

If $(D_i, d_i)$ $(1 \leq i \leq n)$ has an exact Euclidean embedding, we can assume that all $z \in D_i$ are on a Euclidean space, and hence $K_i(z, z') = \exp(-\lambda d_i(z, z')^{\beta})$ is a kernel. Hence

$$K(t, t') = \prod_{i=1}^{n} K_i(x, y) = \exp(-\lambda \sum_{i=1}^{n} d_i(z, z')^{\beta}) \tag{7}$$

is a kernel function.

When a kernel function (7) is used for the $K$-means, we have the following algorithm [14].

**Algorithm KKM: Kernel-based $K$-means clustering.**

**KKM1.** Choose $K$ different objects $y_1, \ldots, y_K \in D_i$ randomly. Let the initial cluster centers be $v_1 = y_1, \ldots, v_K = y_K$ and

$$d(x_k, v_i) = K(x_k, x_k) + K(v_i, v_i) - 2K(x_k, v_i).$$

**KKM2.** Reallocate each object $x$ to the nearest center $v_i$:

$$i = \arg \min_{1 \leq j \leq K} d(x, v_j).$$

**KKM3.** Update the distance:

$$d(x_k, v_i) = K(x_k, x_k) - \frac{2}{|G_i|} \sum_{y \in G_i} K(x_k, y) + \frac{1}{|G_i|^2} \sum_{y, y' \in G_i} K(y, y').$$

for all $x_k$ and $v_i$.

**KKM4.** Check the convergence criterion. If not convergent, go to **KKM2**.

**End of KKM.**

Kernel-based fuzzy $c$-means clustering algorithms can moreover be derived without difficulty. We omit the detail (see [13,15]).

## 5 Conclusion

We have overviewed current methods of clustering related to rough sets and proposed new clustering algorithms to analyze information systems in the form of a table. The fundamental idea is how the basic methods of agglomerative clustering and the $K$-means algorithms are adapted to the information systems.

We thus have the poset-valued clustering, the $K$-mode clustering, and kernel-based algorithms. The last approach of the kernel function is especially interesting in the sense that it connects three different areas of rough sets, clustering, and support vector machines.

These methods are basic and there are many problems to be studied. For example the method of lattice-valued clustering requires simplification of an output display; we should check if the condition of the exact embedding property is satisfied before a kernel-based method of $K$-means is used. We thus have many rooms for further consideration and development.

Applications to a variety of real problems should also be studied. A promising application area is a model of document retrieval, where the present approach is applicable [16,18].

## Acknowledgment

This study has partly been supported by the Grant-in-Aid for Scientific Research, Japan Society for the Promotion of Science, No.16300065.

## References

1. Bezdek, J,C,: *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum (1981).
2. Birkhoff, G: *Lattice Theory*, Amer. Math. Soc. (1967).
3. Cristianini, N., Shawe-Taylor, J.: *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press (2000).
4. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification, 2nd Ed.*, Wiley (2001).
5. Everitt, B.S.: *Cluster Analysis, 3rd Ed.*, Arnold, London (1993).
6. Hirano, S., Tsumoto, S.: A framework for unsupervised selection of indiscernibility threshold in rough clustering, In: Greco S. et al., eds.: *Rough Sets and Current Trends in Computing, (RSCTC2006)*, LNAI 4259, Springer (2006) 872–881.
7. Kaufman, L., Rousseeuw, P.J.: *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley (1990).
8. Lingras, P., West, C.: Interval set clustering of web users with rough $K$-means, *J. of Intel. Informat. Sci.*, **23**(1) (2004) 5–16.
9. Liu, Z.Q., Miyamoto, S., eds.: *Soft Computing and Human-Centered Machines*, Springer, Tokyo (2000).
10. MacLane, S., Birkhoff, G.: *Algebra, 2nd Ed.*, Macmillan (1979).
11. Miyamoto, S.: *Fuzzy Sets in Information Retrieval and Cluster Analysis*, Kluwer, Dordrecht (1990).
12. Miyamoto, S.: *Introduction to Cluster Analysis: Theory and Applications of Fuzzy Clustering*, Morikita-Shuppan, Tokyo (1990) (in Japanese).
13. Miyamoto, S., Suizu, D.: Fuzzy $c$-means clustering using transformations into high-dimensional spaces, In: Proc. of FSKD'02: 1st International Conference on Fuzzy Systems and Knowledge Discovery, Nov. 18-22, 2002, Singapore, Vol.2, (2002) 656–660.
14. Miyamoto, S., Nakayama, Y.: Algorithms of hard $c$-means clustering using kernel functions in support vector machines, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **7**(1) (2003) 19–24.

15. Miyamoto, S., Suizu, D.: Fuzzy $c$-means clustering using kernel functions in support vector machines, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, **7**(1) (2003) 25–30.
16. Miyamoto, S., Hayakawa, S.: A fuzzy neighborhood model for clustering, classification, and approximations, In: Greco S. et al., eds.: *Rough Sets and Current Trends in Computing, (RSCTC2006)*, LNAI 4259, Springer (2006) 882–890.
17. Miyamoto, S.: Lattice-valued hierarchical clustering for analyzing information systems, In: Greco S. et al., eds.: *Rough Sets and Current Trends in Computing, (RSCTC2006)*, LNAI 4259, Springer (2006) 909–917.
18. Miyamoto, S., Mizutani, K.: Fuzzy multiset space and c-means clustering using kernels with application to information retrieval, In: Bilgic T. et al., eds.: *Proc. of IFSA2003, Istanbul, Turkey June 29- July 3, 2003*, LNAI 2715 (2003) 387–395.
19. Okuzaki, T., Hirano, S., Kobashi, S., Hata, Y., Takahashi, Y.: A rough set based clustering method by knowledge combination, *IEICE Trans. on Informat. Syst.*, **E85-D**(12) (2002) 1898–1908.
20. Pawlak, Z.: Rough sets, *International Journal of Computer and Information Sciences*, **11** (1982) 341–356.
21. Z. Pawlak, *Rough Sets*, Kluwer Academic Publishers, Dordrecht (1991).
22. Pawlak, Z., Skowron, A.: Rudiments of rough sets, *Information Sciences*, **177** (2007) 3–27.
23. Peters, G., Lampert, M.: A partitive rough clustering algorithm, In: Greco S., et al., eds.: *Rough Sets and Current Trends in Computing, (RSCTC2006)*, LNAI 4259, Springer (2006) 657–666.
24. Ullman, J.D.: *Database and Knowledge-base Systems: Volume I*, Computer Science Press, Rockville, Maryland (1988).
25. Vapnik, V.N.: *Statistical Learning Theory*, Wiley (1998).
26. Vapnik V.N.: *The Nature of the Statistical Learning Theory, 2nd Ed.*, Springer (2000).

# From Parallel Data Mining to Grid-Enabled Distributed Knowledge Discovery

Eugenio Cesario[1] and Domenico Talia[1,2]

[1] ICAR-CNR, Italy
[2] DEIS-University of Calabria, Italy
cesario@icar.cnr.it
talia@deis.unical.it

**Abstract.** Data mining often is a compute intensive and time requiring process. For this reason, several data mining systems have been implemented on parallel computing platforms to achieve high performance in the analysis of large data sets. Moreover, when large data repositories are coupled with geographical distribution of data, users and systems, more sophisticated technologies are needed to implement high-performance distributed KDD systems. Recently computational Grids emerged as privileged platforms for distributed computing and a growing number of Grid-based KDD systems have been designed. In this paper we first outline different ways to exploit parallelism in the main data mining techniques and algorithms, then we discuss Grid-based KDD systems.

**Keywords:** Rough Set, Parallel Data Mining, Distributed Data Mining, Grid.

## 1 Introduction

In our daily activities we often deal with flows of data much larger than we can understand and use. Thus we need a way to sift those data for extracting what is interesting and relevant for our activities. Knowledge discovery in large data repositories can find what is interesting in them representing it in an understandable way. Data mining is the automated analysis of large volumes of data looking for relationships and knowledge that are implicit in data and are *interesting* in the sense of impacting an organization's practice.

Mining large data sets requires powerful computational resources. A major issue in data mining is scalability with respect to the very large size of current-generation and next-generation databases, given the excessively long processing time taken by (sequential) data mining algorithms on realistic volumes of data. In fact, data mining algorithms working on very large data sets take a very long time on conventional computers to get results. It is not uncommon to have sequential data mining applications that require several days or weeks to complete their task. To mention just two examples, [1] estimates that *C4.5* with rule pruning would take 79 years on a 150-MHz processor in order to mine a database with 500,000 tuples. [2] reports that a sequential version of the RL algorithm is impractical (i.e. takes too long to run) on databases of more than 70,000 tuples.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 25–36, 2007.

A first approach to reduce response time is *sampling*, that is the reduction of the original data set in a "less large data set" composed only of a portion of data considered representative of he whole data set. Nevertheless, in some cases reducing data might result in inaccurate models, in some other cases it is not useful (e.g., outliers identification). For such reasons, sampling techniques can not be considered an effective way to tackle a long computation time.

A second approach is the *parallel computing*, that is the choice to process and analyze data sets by parallel algorithms. Under a data mining perspective, such a field is known as *Parallel Data Mining*. High performance computers and parallel data mining algorithms can offer a very efficient way to mine very large data sets [3], [4] by analyzing them in parallel. *Parallel computing* systems can bring significant benefits in the implementation of data mining and knowledge discovery applications by means of the exploitation of inherent parallelism of data mining algorithms. Benefits consist in both performance improvement and in quality of data selection. When data mining tools are implemented on high-performance parallel computers, they can analyze massive databases in a reasonable time. Faster processing also means that users can experiment with more models to understand complex data.

Beyond the development of knowledge discovery systems based on parallel computing platforms to achieve high performance in the analysis of large data sets stored in a single site, a lot of work has been devoted to design systems able to handle and analyze multi-site data repositories. Data mining in large settings like virtual organization networks, the Internet, corporate intranets, sensor networks, and the emerging world of ubiquitous computing questions the suitability of centralized architectures for large-scale knowledge discovery in a networked environment. Under this perspective, the field of *Distributed Data Mining* offers an alternative approach. It works by analyzing data in a distributed fashion and pays particular attention to the trade-off between centralized collection and distributed analysis of data. Knowledge discovery is speeded up by executing in a distributed way a number of data mining processes on different data subsets and then combining the results through meta-learning. This technology is particularly suitable for applications that typically deal with very large amount of data (e.g., transaction data, scientific simulation and telecommunication data), which cannot be analyzed in a single site on traditional machines in acceptable times. Moreover, parallel data mining algorithms can be a component of distributed data mining applications, that can exploit both parallelism and data distribution.

*Grid* technology integrates both distributed and parallel computing, thus it represents a critical infrastructure for high-performance distributed knowledge discovery. Grid computing is receiving an increasing attention both from the research community and from industry and governments, watching at this new computing infrastructure as a key technology for solving complex problems and implementing distributed high-performance applications [5]. The term *Grid* defines a global distributed computing platform through which - like in a power grid - users gain ubiquitous access to a range of services, computing and data

resources. The driving Grid applications are traditional high-performance applications, such as high-energy particle physics, and astronomy and environmental modeling, in which experimental devices create large quantities of data that require scientific analysis. Grid computing differs from conventional distributed computing because it focuses on large-scale resource sharing, offers innovative applications, and, in some cases, it is geared toward high-performance systems. Although originally intended for advanced science and engineering applications, Grid computing has emerged as a paradigm for coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations in industry and business. For these reasons, Grids can offer an effective support to the implementation and use of knowledge discovery systems.

The rest of the paper is organized as follows. Section 2 introduces Parallel and Distributed data mining, and shows by a trivial example how parallel strategies can be applied in data mining techniques based on rough set theory. Section 3 analyzes the Grid-based data mining approach. Section 4 introduces the *Knowledge Grid*, a reference software architecture for geographically distributed PDKD systems. The Section 5 gives concluding remarks.

## 2 Parallel and Distributed Data Mining

In this section Parallel Data Mining and Distributed Data Mining approaches are discussed.

### 2.1 Parallel Data Mining

*Parallel Data Mining* is the field concerning the study and application of data mining analysis by parallel algorithms. The key idea underlying such a field is that parallel computing can give significant benefits in the implementation of data mining and knowledge discovery applications, by means of the exploitation of inherent parallelism of data mining algorithms. Main goals of the use of parallel computing technologies in the data mining field are: *(i)* performance improvements of existing techniques, *(ii)* implementation of new (parallel) techniques and algorithms, and *(iii)* concurrent analysis using different data mining techniques in parallel and result integration to get a better model (i.e., more accurate results).

As observed in [6], three main strategies can be identified in the exploitation of parallelism in data mining algorithms: *Independent Parallelism*, *Task Parallelism* and *Single Program Multiple Data (SPMD) Parallelism*. A brief description of the underlying idea of such strategies follows.

**Independent Parallelism.** It is exploited when processes are executed in parallel in an independent way. Generally, each process has access to the whole data set and does not communicate or synchronize with other processes. Such a strategy, for example, is applied when $p$ different instances of the same algorithm are executed on the whole data set, but each one with a different setting of input parameters. In this way, the computation finds out $p$ different models, each one

determined by a different setting of input parameters. A validation step should learn which one of the $p$ predictive models is the most reliable for the topic under investigation. This strategy often requires commutations among the parallel activities.

**Task Parallelism.** In some scientific communities it is known also as *Control Parallelism*. It supposes that each process executes different operations on (a different partition of) the data set. The application of such a strategy in decision tree learning, for example, leads to have $p$ different processes running, each one associated to a particular subtree of the decision tree to be built. The search goes parallely on in each subtree and, as soon as all the $p$ processes finish their executions, the whole final decision tree is composed by joining the various subtrees obtained by the processes.

**SPMD Parallelism.** It is exploited when a set of processes execute in parallel the same algorithm on different partitions of a data set, and processes cooperate to exchange partial results. According to this strategy, the dataset is initially partitioned in $p$ parts, if $p$ is the apriori-fixed parallelism degree (i.e., the number of processes running in parallel). Then, the $p$ processes search in parallel a predictive model for the subset associated to it. Finally, the global result is obtained by exchanging all the local models information.

These three strategies for parallelizing data mining algorithms are not necessarily alternative. In fact, they can be combined to improve both performance and accuracy of results. For completeness, we say also that in combination with strategies for parallelization, different data partition strategies may be used : *(i)* sequential partitioning (separate partitions are defined without overlapping among them), *(ii)* cover-based partitioning (some data can be replicated on different partitions) and *(iii)* range-based query partitioning (partitions are defined on the basis of some queries that select data according to attribute values).

Now, we have to notice that architectural issues are a fundamental aspect for the goodness of a parallel data mining algorithm. In fact, interconnection topology of processors, communication strategies, memory usage, I/O impact on algorithm performance, load balancing of the processors are strongly related to the efficiency and effectiveness of the parallel algorithm. For lack of space, we can just cite those. The mentioned issues (and others) must be taken into account in the parallel implementation of data mining techniques. The architectural issues are strongly related to the parallelization strategies and there is a mutual influence between knowledge extraction strategies and architectural features. For instance, increasing the parallelism degree in some cases corresponds to an increment of the communication overhead among the processors. However, communication costs can be also balanced by the improved knowledge that a data mining algorithm can get from parallelization. At each iteration the processors share the approximated models produced by each of them. Thus each processor executes a next iteration using its own previous work and also the knowledge produced by the other processors. This approach can improve the rate at which a data mining algorithm finds a model for data (knowledge) and make

up for lost time in communication. Parallel execution of different data mining algorithms and techniques can be integrated not just to get high performance but also high accuracy.

**Parallel Rough Set Computation.** In this section we discuss, as an example, how parallel strategies can be applied in data mining techniques based on rough set theory. Rough Set data analysis (RSDA), first developed by Z. Pawlak and his co-workers in [7], has become a promising research topic for the scientific community. Main thrust in current applications of rough set theory are attribute reduction, rule generation, prediction.

RSDA offers purely structural methods to discovery data dependencies and to reduce the number of attributes of an information system $\mathcal{I}$. Let us suppose a dataset $U$ of objects, described by a set $\Omega$ of attributes. Two basic tasks concerning the rough set theory are the computation of the *reduct* and the *core* of $\mathcal{I}$. A set $P \subseteq Q \subseteq \Omega$ is a *reduct* of $Q$, i.e. $\texttt{reduct}(Q)$, if $P$ is minimal among all subsets of $Q$ which generate the same classification as $Q$. In other words, all the attributes in $P$ are indispensable and none of them can be omitted for distinguishing objects as they are distinguished by the set $Q$. Obviously, it is not hard to see that each $Q \subseteq \Omega$ has a reduct, though this is usually not unique. The intersection of all reducts of $Q$ is called the *core of* $Q$, i.e. $\texttt{core}(Q)$, and the elements of the core of $Q$ are called *indispensable* for $Q$. As pointed out in [8], the problem of finding a reduct of minimal cardinality is NP-hard, and finding all the reducts has exponential complexity. An interesting method to find the core and the reducts of an information system $\mathcal{I}$ is given in [9], where authors propose to compute the $|U| \times |U|$ *discernibility matrix* $D$, where the generic element $D(x, y)$ is constituted by the set of attributes in $\Omega$ for which $x$ and $y$ assume different values. Authors demonstrate that the core of $\mathcal{I}$ can be computed by the union of singleton elements in $D$.

Let us describe the application of such a method by a trivial example. Let us suppose the set $U = \{o_1, o_2, o_3, o_4\}$ of objects, described on the set $\Omega = \{q_1, q_2, q_3, q_4\}$ of attributes. Figure 1(a) shows such a dataset. Figure 1(b) shows the corresponding *discernibility matrix*, obtained as a natural result of the application of a discernibility function (for all the attributes in $\Omega$) to each pair of objects. In this example, the *discernibility matrix* points out that the objects $o_2, o_3$ can be distinguished by attributes $q_2, q_4$ (because they assume different values w.r.t. such attributes), while the objects $o_3, o_4$ can be distinguished by the only attribute $q_1$. The entries $< o_1, o_2 >, < o_1, o_3 >, < o_1, o_4 >$ and $< o_3, o_4 >$ show that the core of the system $\mathcal{I}$ is $\texttt{core}(\mathcal{I}) = \{q_1, q_2, q_4\}$. As it is really evident, the time and space complexity of the core computation by such a method could be very expensive, if we consider a large number of objects described by a large number of attributes (high-dimensional data) are involved in. Let us observe that having a $O(|\Omega| \cdot |U|^2)$ space and time complexity, this method is not suitable for large data sets (e.g. with over 10,000 objects) even on powerful workstations. A parallel and distributed approach to such a topic could be very useful. For example, let us describe how parallelism can be applied and which benefits it could give in a such scenario.

| objects \ attributes | $q_1$ | $q_2$ | $q_3$ | $q_4$ |
|---|---|---|---|---|
| $o_1$ | $v_1$ | $v_2$ | $v_4$ | $v_5$ |
| $o_2$ | $v_1$ | $v_3$ | $v_4$ | $v_5$ |
| $o_3$ | $v_1$ | $v_2$ | $v_4$ | $v_6$ |
| $o_4$ | $v_7$ | $v_2$ | $v_4$ | $v_5$ |

(a) Data Set.

| | $o_1$ | $o_2$ | $o_3$ | $o_4$ |
|---|---|---|---|---|
| $o_1$ | -- | $q_2$ | $q_4$ | $q_1$ |
| $o_2$ | -- | -- | $q_2$,$q_4$ | $q_1$,$q_2$ |
| $o_3$ | -- | -- | -- | $q_1$ |
| $o_4$ | -- | -- | -- | -- |

(b) Discernibility matrix.

**Fig. 1.** Original Data Set and Discernibility Matrix

Let us first describe the scenario when an independent parallelism is applied. Let us suppose to have $N$ different and independent processes, with $N \leq |U|$, and that each one is able to access to the whole data set (i.e., by replicating it for each process, or providing a shared repository). In this way, each process takes in charge to compute only the rows of the discernibility matrix corresponding to some objects in the dataset. As a limit case, if $N = |U|$, each process could compute only one raw, corresponding to a particular objects. As a final task, the whole result is naturally obtained by unifying partial results obtained by all the processes.

A further parallelism method consists in applying the SPMD strategy, managed by $N$ processors. In this case, the whole data set $U$ can be splitted in $N$ partitions, each one composed of $|U|/N$ objects[1]. Now, each process computes the rows of the discernibility matrix corresponding to the objects within the partition assigned to it. It is clear that, in this way, each process is not able to compute an entire row of the matrix, but just the columns corresponding to objects it contains. For this reason, the final result can be obtained by aggregating partial results obtained by the processes. Indeed, this final task can be managed in different ways. A first way is that, as soon as a processor finishes its execution, it sends its results to all the other processors: such a solution guarantees that each node, when all the computations are finished, holds the final and complete result. A second way is that as soon as a processor finishes its computation, it sends local results to an aggregator node being in charge to join the various results: in this case, the only aggregator node manages the join step and holds the final result.

As a final consideration, it is clear that the parallelism for the core computation scenario is very useful, and the largest the dataset size and its dimensionality are, the most appreciable are their benefits. Moreover, we point out that the network load due to the transmission of the partial results could be very low because processors need to communicate not the entire matrix, but just the attributes contained in singleton elements.

[1] As a variant, we notice that the dataset could be partitioned with respect attributes, and in similar way the computation goes on.

### 2.2 Distributed Data Mining

Traditional warehouse-based architectures for data mining suppose to have centralized data repository. Such a centralized approach is fundamentally inappropriate for most of the distributed and ubiquitous data mining applications. In fact, the long response time, lack of proper use of distributed resource, and the fundamental characteristic of centralized data mining algorithms do not work well in distributed environments. A scalable solution for distributed applications calls for distributed processing of data, controlled by the available resources and human factors. For example, let us suppose an ad hoc wireless sensor network where the different sensor nodes are monitoring some time-critical events. Central collection of data from every sensor node may create traffic over the limited bandwidth wireless channels and this may also drain a lot of power from the devices. A distributed architecture for data mining is likely aimed to reduce the communication load and also to reduce the battery power more evenly across the different nodes in the sensor network. One can easily imagine similar needs for distributed computation of data mining primitives in ad hoc wireless networks of mobile devices like PDAs, cellphones, and wearable computers [10]. The wireless domain is not the only example. In fact, most of the applications that deal with time-critical distributed data are likely to benefit by paying careful attention to the distributed resources for computation, storage, and the cost of communication. As an other example, let us consider the World Wide Web: it contains distributed data and computing resources. An increasing number of databases (e.g., weather databases, oceanographic data, etc.) and data streams (e.g., financial data, emerging disease information, etc.) are currently made on-line, and many of them change frequently. It is easy to think of many applications that require regular monitoring of these diverse and distributed sources of data. A distributed approach to analyze this data is likely to be more scalable and practical particularly when the application involves a large number of data sites. Hence, in this case we need data mining architectures that pay careful attention to the distribution of data, computing and communication, in order to access and use them in a near optimal fashion. *Distributed Data Mining* (sometimes referred by the acronym *DDM*) considers data mining in this broader context.

DDM may also be useful in environments with multiple compute nodes connected over high speed networks. Even if the data can be quickly centralized using the relatively fast network, proper balancing of computational load among a cluster of nodes may require a distributed approach. The privacy issue is playing an increasingly important role in the emerging data mining applications. For example, let us suppose a consortium of different banks collaborating for detecting frauds. If a centralized solution was adopted, all the data from every bank should be collected in a single location, to be processed by a data mining system. Nevertheless, in such a case a Distributed Data Mining system should be the natural technological choice: both it is able to learn models from distributed data without exchanging the raw data between different repository, and it allows detection of fraud by preserving the privacy of every bank's customer transaction data.

For what concerns techniques and architecture, it is worth noticing that many several other fields influence Distributed Data Mining systems concepts. First, many DDM systems adopt the Multi-Agent System (MAS) architecture, which finds its root in the Distributed Artificial Intelligence (DAI). Second, altough Parallel Data Mining often assumes the presence of high speed metwork connections among the computing nodes, the development of DDM has also been influenced by the PDM literature. Most DDM algorithms are designed upon the potential parallelism they can apply over the given distributed data. Typically, the same algorithm operates on each distributed data site concurrently, producing one local model per site. Subsequently, all local models are aggregated to produce the final model. In figure 2 a general Distributed Data Mining framework is presented. In essence, the success of DDM algorithms lies in the aggregation. Each local model represents locally coherent patterns, but lacks details that may be required to induce globally meaningful knowledge. For this reason, many DDM algorithms require a centralization of a subset of local data to compensate it. The ensemble approach has been applied in various domains to increase the accuracy of the predictive model to be learnt. It produces multiple models and combines them to enhance accuracy. Typically, voting (weighted or un-weighted) schema are employed to aggregate base model for obtaining a global model. As we have discussed above, minimum data transfer is another key attribute of the successful DDM algorithm.

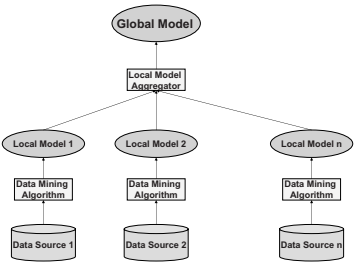



**Fig. 2.** General Distributed Data Mining Framework

## 3 Grid-Based Data Mining

In the last years, *Grid computing* is receiving an increasing attention both from the research community and from industry and governments, watching at this new computing infrastructure as a key technology for solving complex problems and implementing distributed high-performance applications. *Grid* technology integrates both distributed and parallel computing, thus it represents a critical infrastructure for high-performance distributed knowledge discovery. *Grid computing* differs from conventional distributed computing because it focuses on large-scale dynamic resource sharing, offers innovative applications, and, in some cases, it is geared toward high-performance systems. The *Grid* emerged as a privileged computing infrastructure to develop applications over geographically

distributed sites, providing for protocols and services enabling the integrated and seamless use of remote computing power, storage, software, and data, managed and shared by different organizations.

Basic Grid protocols and services are provided by toolkits such as *Globus Toolkit* (`www.globus.org/toolkit`), *Condor* (`www.cs.wisc.edu/condor`), *Legion* (`legion.virginia.edu`), and *Unicore* (`www.unicore.org`). In particular, the *Globus Toolkit* is the most widely used middleware in scientific and data-intensive Grid applications, and is becoming a de facto standard for implementing Grid systems. The toolkit addresses security, information discovery, resource and data management, communication, fault-detection, and portability issues. It does so through mechanisms, composed as bags of services, that execute operations in Grid applications. A wide set of applications is being developed for the exploitation of Grid platforms. Since application areas range from scientific computing to industry and business, specialized services are required to meet needs in different application contexts. In particular, *data Grids* have been designed to easily store, move, and manage large data sets in distributed data-intensive applications. Besides core data management services, *knowledge-based Grids*, built on top of computational and data Grid environments, are needed to offer higher-level services for data analysis, inference, and discovery in scientific and business areas [11]. In many recent papers [12],[13], [14] is claimed that the creation of *knowledge Grids* is the enabling condition for developing high-performance knowledge discovery processes and meeting the challenges posed by the increasing demand of power and abstractness coming from complex problem solving environments.

## 4 The Knowledge Grid

The *Knowledge Grid* [15] is an environment providing knowledge discovery services for a wide range of high performance distributed applications. Data sets and analysis tools used in such applications are increasingly becoming available as stand-alone packages and as remote services on the Internet. Examples include gene and DNA databases, network access and intrusion data, drug features and effects data repositories, astronomy data files, and data about web usage, content, and structure. Knowledge discovery procedures in all these applications typically require the creation and management of complex, dynamic, multi-step workflows. At each step, data from various sources can be moved, filtered, and integrated and fed into a data mining tool. Based on the output results, the developer chooses which other data sets and mining components can be integrated in the workflow, or how to iterate the process to get a knowledge model. Workflows are mapped on a Grid by assigning nodes to the Grid hosts and using interconnections for implementing communication among the workflow nodes.

The *Knowledge Grid* supports such activities by providing mechanisms and higher level services for searching resources, representing, creating, and managing knowledge discovery processes, and for composing existing data services and data mining services in a structured manner, allowing designers to plan, store,

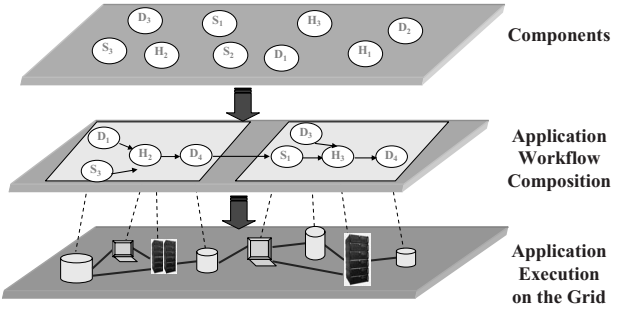



**Fig. 3.** Main steps of application composition and execution in the Knowledge Grid

document, verify, share and re-execute their workflows as well as manage their output results. The *Knowledge Grid* architecture is composed of a set of services divided in two layers: the *Core K-Grid layer* and the *High-level K-Grid layer*. The first interfaces the basic and generic Grid middleware services, while the second interfaces the user by offering a set of services for the design and execution of knowledge discovery applications. Both layers make use of repositories that provide information about resource metadata, execution plans, and knowledge obtained as result of knowledge discovery applications.

In the Knowledge Grid environment, discovery processes are represented as workflows that a user may compose using both concrete and abstract Grid resources. Knowledge discovery workflows are defined using a visual interface that shows resources (data, tools, and hosts) to the user and offers mechanisms for integrating them in a workflow. Information about single resources and workflows are stored using an XML-based notation that represents a workflow (called execution plan in the Knowledge Grid terminology) as a data-flow graph of nodes, each one representing either a data mining service or a data transfer service. The XML representation allows the workflows for discovery processes to be easily validated, shared, translated in executable scripts, and stored for future executions. Figure 3 shows the main steps of the composition and execution processes of a knowledge discovery application on the Knowledge Grid.

As an application scenario, in [6] a simple meta-learning process over the Knowledge Grid is presented. Meta-learning is aimed to generate a number of independent classifiers by applying learning programs to a collection of distributed data sets in parallel. The classifiers computed by learning programs are then collected and combined to obtain a global classifier. Figure 4 shows a distributed meta-learning scenario, in which a global classifier $GC$ is obtained on $Node_Z$ starting from the original data set $DS$ stored on $Node_A$. This process can be described through the following steps:

1. On $Node_A$, training sets $TR_1, \ldots, TR_n$, testing set $TS$ and validation set $VS$ are extracted from $DS$ by the partitioner $P$. Then $TR_1, \ldots, TR_n$, $TS$ and $VS$ are respectively moved from $Node_A$ to $Node_1, \ldots, Node_n$, and to $Node_Z$.

2. On each $Node_i (i = 1, ..., n)$ the classifier $C_i$ is trained from $TR_i$ by the learner $L_i$. Then each $C_i$ is moved from $Node_i$ to $Node_Z$.
3. On $Node_Z$, the $C_1, \ldots, C_n$ classifiers are combined and tested on $TS$ and validated on $VS$ by the combiner/tester $CT$ to produce the global classifier $GC$.

Being the Knowledge Grid an oriented service architecture, a Knowledge Grid user interacts with some services to design and execute such an application. More in detail, she/he can interacts with the *DAS* (*Data Access Service*) and *TAAS* (*Tools and Algorithms Access Service*) services to find data and mining software and with the *EPMS* (*Execution Plan Management Service*) service to compose a workflow (execution plane) describing at a high level the needed activities involved in the overall data mining computation. Through the execution plan, computing, software and data resources are specified along with a set of requirements on them. The execution plan is then processed by the *RAEMS* (*Resource Allocation and Execution Management Service*), which takes care of its allocation. In particular, it first finds appropriate resources matching user requirements (i.e., a set of concrete hosts $Node_1, \ldots, Node_n$, offering the software $L$, and a node $Node_Z$ providing the $CT$ software), then manages the execution of overall application, enforcing dependencies among data extraction, transfer, and mining steps. Finally, the *RAEMS* manages results retrieving, and visualize them by the *RPS* (*Results Presentation Service*) service.

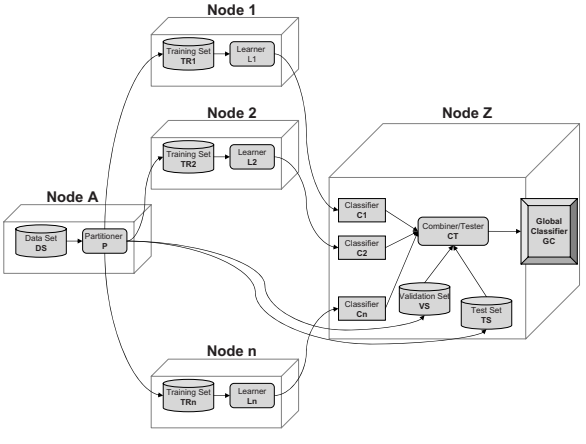



**Fig. 4.** A distributed meta-learning scenario

## 5 Conclusion

This paper discussed main issues and approaches in parallel and distributed data mining. Both this research areas are critical for the development of efficient, scalable and accurate knowledge discovery applications that deal with large data sources and distributed data repositories. In the last decade Grid computing

systems have been developed as parallel and distributed platforms for complex distributed applications. In this paper we discussed as Grids can be exploited in parallel and distributed data mining and outlined the main features of the Knowledge Grid as an example of Grid-aware environment for distributed knowledge discovery applications. This is a very promising research area that should be further investigated looking for efficient solutions for high-performance KDD.

## References

1. Cohen, W.W.: Fast Effective Rule Induction. Proc. of the 12th Int. Conf. Machine Learning (ICML'95), Tahoe City, California, USA (1995) 115-123.
2. Provost, F.J., Aronis J.M.: Scaling up inductive learning with massive parallelism. International Journal of Machine Learning, vol. 23 n. 1 (1996) 33-46.
3. Skillicorn, D.: Strategies for Parallel Data Mining. IEEE Concurrency, vol. 7, n. 4 (1999) 26-35.
4. Talia, D.: Parallelism in Knowledge Discovery Techniques. Proc. of 6th Int. Conf. on Applied Parallel Computing (PARA02), Helsinki, Finland (2002) 127-136.
5. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Globus Project (2002), www.globus.org/alliance/publications/papers/ogsa.pdf.
6. Congiusta, A., Talia, D., Trunfio, P.: Parallel and Grid-Based Data Mining, in Data Mining and Knowledge Discovery Handbook. Springer Publishing, (2005) 1017-1041.
7. Pawlak Z.: Rough Sets. International Journal of Computer and Information Science, vol. 11 (1982) 341-356.
8. Düntsch I., Günther G.: Roughian: Rough information analysis. International Journal of Intelligent Systems, vol. 16 n. 1 (2001) 121-147.
9. Skowron, A., Rauszer, C.: The Discernibility Matrices and Functions in Information Systems, in Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory. Kluwer Academic Publisher (1992).
10. Park, B., Kargupta, H.: Distributed Data Mining: Algorithms, Systems, and Applications. Data Mining Handbook. IEA Publisher (2002) 341-358.
11. Moore, R.. Knowledge-based Grids. Proc. of the 18th IEEE Symposium on Mass Storage Systems and 9th Goddard Conference on Mass Storage Systems and Technologies, San Diego, USA (2001).
12. Berman, F.: From TeraGrid to Knowledge Grid. Communications of the ACM, vol. 44 n. 11 (2001) 27-28.
13. Johnston, W.E.: Computational and Data Grids in Large Scale Science and Engineering. Future Generation Computer Systems vol. 18 n. 8 (2002) 1085-1100.
14. Cannataro, M., Talia, D., Trunfio, P.: Knowledge Grid: High Performance Knowledge Discovery Services on the Grid. Proc. of the 2nd International Workshop GRID (2001).
15. Cannataro, M., Talia, D.: The Knowledge Grid. Communications of the ACM, vol. 46 n. 1 (2003) 89-93.

# A New Algorithm for Attribute Reduction in Decision Tables

Xuegang Hu [1], Junhua Shi [1], and Xindong Wu [1, 2]

[1] School of Computer Science and Information Engineering,
Hefei University of Technology, Anhui 230009, China
[2] Department of Computer Science, University of Vermont,
Burlington, VT 50405, USA
xueghu@mail.hf.ah.cn, junhuash@163.com, xwu@cs.uvm.edu

**Abstract.** This paper presents a new attribute reduction algorithm, ARIMC, for both consistent and inconsistent decision tables. ARIMC eliminates all redundant and inconsistent objects in a decision table, extracts the core attributes when they exist in the decision table in an efficient way, and utilizes the core attributes and their absorptivity as the optimization condition to construct items of the discernibility matrix. Compared with Skowron et al's reduction algorithm [2], ARIMC shows its advantages in simplicity, practicability and time efficiency.

**Keywords:** Rough Set; Attribute Reduction; Decision Table.

## 1 Introduction

Rough set theory was proposed by Z. Pawlak in 1982. It is a mathematical tool for vague and uncertain problems, and has been widely used in artificial intelligence, data mining, pattern recognition, failure detection and other related fields. Compared with statistics, evidence theory and other mathematical tools that can also solve vague and uncertain problems, rough set theory does not need background knowledge of the given data, and it defines knowledge as a family of indiscernibility relations so that it can give knowledge clear mathematical meanings. Rough set theory applies definite mathematical methods to solve uncertain problems, and provides an effective way for further data analysis.

A decision table is an important knowledge representation system that consists of both condition and decision attribute sets. Reduction is an important technique in rough set theory. The key part of attribute reduction in a decision table is to keep the indispensable attributes of the decision table by eliminating the redundant attributes in order to improve the efficiency and effectiveness of data analysis in the decision table. Attribute reduction plays an important role in Machine Learning and Data Mining. Because calculating all possible reductions of a decision table is an NP-hard problem, seeking efficient and effective algorithms has become an active research topic in the rough set community.

There have been several attribute reduction methods [1,2,4,8,10,5,11]. According to the classical algorithm proposed by Skowron et al [2], a discernibility matrix

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 37–46, 2007.

should be constructed at first and then a discernibility function can be built on the discernibility matrix. Absorptivity can be applied to simplify the discernibility function so that every conjunctive sub-expression in a disjunctive normal form (DNF) is a reduction. The classical method can find all reductions; however, it works only in small datasets, because the simplification is a time-consuming process. In an algorithm based on a frequency function, Hu [8] proposed that the importance of an attribute can be computed by the appearing times in the discernibility matrix. Dai and Li [9] also developed a method based on attribute frequency. Both [8] and [9] can only be used in consistent decision systems. In [6], Wang pointed out the inconsistency of the decision table in [3], which affects reduction results.

This paper presents an algorithm called ARIMC (Attribute Reduction based on an Improved Matrix and the Core), based on an improved discernibility matrix (IDM) and the optimization of matrix construction using core attributes and absorptivity. During the IDM construction, redundant and inconsistent data in a decision table are dealt with at the same time, and therefore, the construction method can be applied to both consistent and inconsistent systems. Absorptivity controls the entries of the IDM and the core attributes speed up the IDM construction. Attribute reduction is conducted efficiently by a frequency function.

The rest of the paper is organized as follows. Section 2 introduces relevant concepts and algorithms in related work. Section 3 presents a detailed description of the ARIMC algorithm with an analysis, an example and a comparison with the classical method by Skowron et al [2]. The paper is concluded with a summary in Section 4.

## 2 Related Work

**Definition 1:** An approximation space is a pair AS = (U, R), where U is a non-empty and finite set, called the universe, and R is an equivalence relation family.

**Definition 2:** Given a pair S = (U, A), where U is the universe, and A is a set of attributes, with every attribute $a \in A$, there is a set V of associated values, called the domain of a. Any subset B of A determines a binary relation IND (B) on U, which is called an indiscernibility relation, and defined as follows: $(x, y) \in IND(B)$ if and only if a (x)= a (y), for every $a \in A$, where a (x) denotes the value of attribute a of an given object x.

**Definition 3:** Let C and D be subsets of A, such that $D \cap C = \varnothing$ and D U C=A. D depends on C in a degree k ($0 \le k \le 1$), denoted $C \Rightarrow_k D$, if

$$k = \gamma(C, D) = \sum_{x \in U/D} \frac{card(C_*(X))}{card(U)}. \tag{1}$$

where card (X) is the cardinality of X.

**Definition 4:** Let C and $D \subseteq A$ be the sets of condition and decision attributes respectively. $C' \subseteq C$ is a D-reduct of C, if C' is a minimal subset of C such that

$$\gamma(C,D) = \gamma(C',D). \tag{2}$$

**Definition 5:** Let **R** be an equivalence relation family, R∈**R, and** IND (B) denote an indiscernibility relation**.** If I (**R**)=I (**R**-{R}), then R can be omitted in **R;** otherwise R can not be omitted in **R.** If none of the relations in **R** can be omitted, **R** is independent; otherwise **R** is dependent.

**Definition 6:** Let C⊆A be the set of condition attributes. The set of all the relations that cannot be omitted in C is called the core of C, denoted as CORE (C).

Obviously, there may be many possible reductions of C. If RED (C) is used to express all the reductions of C, below is a theorem.

**Theorem 1:** The core of an equivalence relation family C is equal to the intersection of all the reductions of C, that is $\mathrm{CORE}(C) = \cap \mathrm{RED}(C)$.

**Definition 7:** Tables with distinguished condition and decision attributes are referred to as decision tables.

**Definition 8:** Every dependency, $C \Rightarrow_k D$, can be described by a set of decision rules in the form "If . . . then". Given any $y \neq x$, if dx|C=dy|C implicates that dx|D=dy|D, the decision rule is consistent; otherwise the rule is inconsistent. If all the decision rules in a decision table are consistent, this table is consistent; otherwise the table is inconsistent.

### 2.1 Traditional Discernibility Matrix and Attribute Reduction

The classical discernibility matrix of a decision table proposed by Skowron et al. [2] is an n-rank symmetrical matrix. The elements are defined as below (where 'a' is a condition attribute, and xi, xj are two objects) :

$$\mathrm{Cij} = \begin{cases} \{a \mid a \in \mathrm{C} \wedge a(\mathrm{xi}) \neq a(\mathrm{xj})\}, & \mathrm{D(xi)} \neq \mathrm{D(xj)} \\ 0, & \mathrm{D(xi)=D(xj)} \\ -1, & a(\mathrm{xi}) = a(\mathrm{xj}), \mathrm{D(xi)} \neq \mathrm{D(xj)} \end{cases}. \tag{3}$$

This definition will be revised in our IDM in Section 3.

The core set of the whole discernibility matrix is the union of every single attribute in the matrix.

$$\mathrm{CORE_D(C)}=\{\mathrm{a} \in \mathrm{C} \mid \mathrm{cij}=\{\mathrm{a}\}\ 1 \le \mathrm{i,j} \le \mathrm{n}\}. \tag{4}$$

The discernibility function is defined as follows:

$$\rho = \wedge\{\vee cij\}. \tag{5}$$

Based on the above definitions, the steps of the classical algorithm are given below:

1. Construct the discernibility matrix according to Equation (3);
2. Generate the discernibility function according to Equation (5);
3. Use absorptivity to simplify the discernibility function;
4. Perform attribute reduction. Every conjunctive sub-expression in the disjunctive normal form (DNF) is a reduction.

The above traditional attribute reduction algorithm constructs the matrix at first, and then applies absorptivity to generate the discernibility function. This means that this algorithm allows many unnecessary data items to enter into the matrix and so it takes a lot of time to calculate the discernibility function and perform the reduction. Therefore, it is only suitable for small datasets.

### 2.2 Other Improved Matrix and Attribute Reduction Algorithms

Another kind of matrix, named improved difference matrix, proposed by Hu and other scholars [3] is as follows:

$$Cij = \begin{cases} \{a|\ a \in C \wedge a(xi) \neq a(xj)\} & D(xi) \neq D(xj) \\ \varnothing & Otherwise \end{cases}. \tag{6}$$

The algorithm in [9] adopts this matrix. It first constructs the matrix according to Equation (6), computes each attribute's appearing times as a way to judge the importance of the attribute, sorts the attributes by their importance, and finally adds the attributes in turns until a reduction is completed. This method made a progress in time complexity, but still takes a significant time to construct the difference matrix and can not deal with inconsistent data either.

## 3 The ARIMC Algorithm

ARIMC first constructs an improved discernibility matrix (IDM) to deal with redundant and inconsistent objects in the given decision table and get the core attributes, and then takes the core attribute set as the initial reduction set to calculate the appearing times of the condition attributes in the IDM. This process is repeated until all elements of the IDM are processed.

### 3.1 The Improved Discernibility Matrix (IDM)

The j>i items, Cij, of the IDM can be divided into two parts, $Cij_1$ and $Cij_2$, and they are constructed by Equations (7) and (8) respectively (where 'b' is a decision attribute).

$$Cij_1 = \begin{cases} \{a|\ \exists a \in C \wedge f(xi,a) \neq f(xj,a)\} \\ 0 \quad \forall a \in C \wedge f(xi,a)=f(xj,a) \end{cases}. \tag{7}$$

$$Cij_2 = \begin{cases} 1 \quad \exists b \in D \wedge f(xi,b) \neq f(xj,b) \\ 0 \quad \forall b \in D \wedge f(xi,b)=f(xj,b) \end{cases}. \tag{8}$$

Through the above definitions of the matrix, we can obtain the following properties:

1. If $Cij_1$=0 and $Cij_2$=0, then xi and xj are duplicate objects, and so row j can be deleted from the original data table;
2. If $Cij_1$=0 and $Cij_2$=1, then xi and xj are inconsistent objects, and so rows i and j should be deleted to maintain the consistency in the decision table.
3. If $Cij_1$ is a single attribute and $Cij_2$=0, then this single attribute is a reductive attribute and can be eliminated;
4. If $Cij_1$ is a single attribute and $Cij_2$=1, then this single attribute is a core attribute and must be added to the core attribute set.

Properties 1, 2, and 3 are obvious. We prove Property 4 as follows.

**Proof of Property 4:** Let C and D be the condition and decision attribute sets of a decision table respectively, and xi and xj be two random objects from the table. The classical construction of the discernibility matrix elements is given in Equation (3).

According to Equation (4), if objects xi and xj are different only in one condition attribute, this attribute must be a core attribute, in order to make them distinguishable in D.

In the IDM, $Cij_1$ takes charge of the condition attributes, and $Cij_2$ takes charge of decision attributes.

If $Cij_1$ is a single attribute and $Cij_2$=1

$\Leftrightarrow$ Objects xi and xj have only one different condition attribute value so that (xi, xj) $\notin$ IND (D)

Therefore, this attribute is also a core attribute.

### 3.2 Description of the ARIMC Algorithm

In the following algorithm, Step 1 adopts the IDM and eliminates redundant and inconsistent data in the decision table. This step can compress the decision table in a significant way, and the compression is not available in the classical algorithms by Skowron et al [2] or by any other authors.

Also, the ARIMC algorithm uses core attributes as the optimization information, and accelerates the construction of the IDM. It scans the existing items in the IDM immediately when a core attribute appears, and deletes those of them that include the core attributes, so that the remaining construction of the matrix is optimized. The reason is that if two objects have different values on the same core attribute, then this attribute is recorded in $Cij_1$ and all the items of $Cij_1$ including the core attributes will be deleted (according to absorptivity), and therefore these items cannot be added onto the IDM. Compared with the algorithm in [9], ARIMC can deal with redundant and inconsistent data in the input decision tables, and is more efficient in both time and space because there are fewer items in the IDM. Step 2 evaluates the importance of each attribute by the frequency information. Because of the decrease of the IDM items, the calculation of the frequency function is faster and simpler. Step 3 keeps selecting attributes for the reduction set until all the elements in the IDM have been processed, which guarantees a correct completion of the attribute reduction.

**Algorithm 1:** Algorithm for ARIMC

**Input:** a decision table S=(U,C ∪D,V,f).
**Output:** the attribute reduction table S'=( U,R ∪D,V',f') of S, R is a D-reduct of C.

**Step 1:**
CORE←∅ ; R←∅ ; Temp ←∅ ;
for i:=1 to card(U) do
{for j:=1 to card(U) do
{while (CORE≠∅ )
{for (every core attribute ∈ CORE)
{if (∃c∈CORE and (c(xi)≠c(xj)))
{skip xij (denoted by '@'); j←j+1}
else construct the IDM of S and take proper action on redundant or inconsistent data}};
If (Cij1 is a core attribute)
{CORE←CORE ∪Cij1;
Go back to delete all the foregoing items in the IDM which contain this new core Cij1 (denoted by '@')}}}
**Step 2:**
P←{all the items (Cij1) which are not '00', '01', '@' or core attributes in the IDM};
M←{a1,a2,…,an, n ≤card(C), ai is a single attribute};
//M is a set of single attributes which appear in P;
Calculate w(ai) by the number of items which contain ai in P; // w(ai) is the weight of ai;
Sort attributes by w(ai);
**Step 3:**
while (! all the items in P are '@') do
{R←CORE;
R←R ∪{a1}, M←M-{a1};// select a1 which has the largest weight;
Temp←{all the other attributes which appear in the items that contain a1 in P };
All the weights of attributes that appear in 'Temp' are decreased by 1;
Delete all the items which contain a1 in P (denoted by '@')
};
R ∪D is a D-reduct of C.

### 3.3 An Example

Table 1 provides a decision table S= (U, A), where U={x1,x2,x3,x4,x5,x6,x7} is the set of objects, A=C⋃D is the set of attributes, C={a,b,c,d} is the set of condition attributes, and D={e} is the decision attribute.

We run both Skowron et al's algorithm [2] and our ARIMC algorithm in Section 3.2 on Table 1 in this subsection, to demonstrate ARIMC's simplicity, practicability and time efficiency. With Skowron et al's classical algorithm,

**Table 1.** A decision table

| R / U | a | b | c | d | e |
|---|---|---|---|---|---|
| X1 | 1 | 0 | 2 | 1 | 1 |
| X2 | 1 | 0 | 2 | 0 | 1 |
| X3 | 1 | 2 | 0 | 0 | 2 |
| X4 | 1 | 2 | 2 | 1 | 0 |
| X5 | 2 | 1 | 0 | 0 | 2 |
| X6 | 2 | 1 | 1 | 0 | 2 |
| X7 | 2 | 1 | 2 | 1 | 1 |

**Step 1:** Construct the discernibility matrix in Table 2.

**Table 2.** The Discernibility Matrix of Table 1 by [2]

| U | X1 | X2 | X3 | X4 | X5 | X6 | X7 |
|---|---|---|---|---|---|---|---|
| X1 | | | | | | | |
| X2 | 0 | | | | | | |
| X3 | bcd | bc | | | | | |
| X4 | b | bd | cd | | | | |
| X5 | abcd | abc | 0 | abcd | | | |
| X6 | abcd | abc | 0 | abcd | 0 | | |
| X7 | 0 | 0 | abcd | ab | cd | cd | |

**Steps 2 and 3:** Construct the discernibility function and simplify it.

$$\rho = (b \vee c \vee d)b(a \vee b \vee c \vee d)(a \vee b \vee c \vee d)(b \vee c)(b \vee d)(a \vee b \vee c)(a \vee b \vee c)$$
$$(c \vee d)(a \vee b \vee c \vee d)(a \vee b \vee c \vee d)(a \vee b \vee c \vee d)(a \vee b)(c \vee d)(c \vee d)$$
$$= b \wedge (c \vee d)$$
$$= bc \vee bd$$

**Step 4:** Get the D-reduct of C: {b, c} or {b, d}.

With our proposed ARIMC algorithm,

**Step 1:** Construct the IDM in Table 3:

**Table 3.** The IDM of Table 1 by ARIMC

| U | X2 | X3 | X4 | X5 | X6 | X7 |
|---|---|---|---|---|---|---|
| X1 | d0 | bcd1 (@) | b1 (@) | abcd1 (@) | abcd1 (@) | ab0 (@) |
| X2 | | bc1 (@) | bd1 (@) | abc1 (@) | abc1 (@) | abd0 (@) |
| X3 | | | cd1 | ab0 (@) | abc0 (@) | abcd1 (@) |
| X4 | | | | abcd1 (@) | abcd1 (@) | ab1 (@) |
| X5 | | | | | c0 | cd1 |
| X6 | | | | | | cd1 |

The items without the '(@)' suffix are the results according to Equations (7) and (8), and the '@' for each of them is the actual result from the ARIMC algorithm in Section 3.2. This is how ARIMC has improved the time efficiency. Obviously, the core attribute here is 'b'.

**Step 2:** P = {cd} and M = {c,d}. Then the weights: w(a) = w(b) = 0, w (c) = w(d) = 3;

**Step 3:** First, initialize R: R={b}. Then, select attribute c into R and R={b, c}, M={d}, Temp={d}, w (d)=2; P={@}. Because all the items in P are '@', meaning that all these items in the matrix have been processed, the looping can now be stopped.

The attribute reduction of Table 1 after the above 2 steps is R={b,c}. If attribute 'd' is first selected into R, not 'c', another reduction of the table, R={b, d} can be obtained following the same steps. As there is no redundant or inconsistent data in this decision table, there are no items like '00' or '01' in the IDM.

### 3.4 Experimental Results and Analysis

Our experiments are conducted on a P4 2.4G, 512M RAM computer, with Java 1.4.2 and the Windows 2000 operating system. The initial data sets are from the UCI Machine Learning Database Repository. The experimental results between our ARIMC and the classical algorithm in [2] are given in Table 4. From this table, ARIMC has demonstrated its advantage over the classical algorithm in time efficiency. The efficiency gain comes from the fact that ARIMC does not include core-embodied items when constructing the IDM, hence simplifies the calculation of the frequency function and saves a considerable amount of time. On the contrary, the classical algorithm spends much time in the construction and reduction of its discernibility matrix. If the set of the core attributes is empty or comes at the very end of the reduction process, the performance difference of the two algorithms may not be very significant. Assuming the universe of a decision table is U, the number of objects is |U|, and the set of condition attributes is C, the time complexity of ARIMC is $O(|C|\times|U|^2)$. In practice, the actual running time is usually much less than this

**Table 4.** A time comparison between ARIMC and the classical algorithm

| Data set | Number of initial attributes | Number of attributes after reduction | Number of records | Time consumption of classical algorithm (seconds) | Time consumption of ARIMC (seconds) |
|---|---|---|---|---|---|
| Balloons | 5 | 4 | 20 | 0. 094 | 0. 063 |
| Zoo | 17 | 15 | 101 | 17. 592 | 0. 641 |
| Mushrooms | 22 | 8 | 100 | 8. 973 | 0. 766 |
| Iris | 5 | 4 | 150 | 1. 797 | 1. 203 |
| Breast-cancer | 10 | 9 | 286 | 3. 977 | 2. 625 |
| Liver-disorders | 7 | 4 | 345 | 6. 114 | 6. 328 |
| Letter | 17 | 14 | 800 | 902. 936 | 64. 703 |
| Vehicle | 19 | 4 | 1046 | 3669. 157 | 168. 594 |

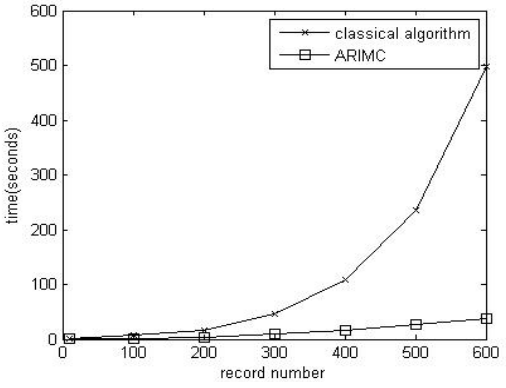



**Fig. 1.** Another time comparison with the increase of data records

time complexity. We have also taken some data from the 'Letter' data set from the UCI Database Repository and processed them using the two algorithms. Their comparative results are shown in Figure 1.

## 4 Conclusion

The ARIMC algorithm proposed in this paper can find the core attributes efficiently and take the core attributes as the optimization condition for the construction of the improved discernibility matrix. It improves the time performance over the classical attribute reduction algorithm. Furthermore, ARIMC can eliminate redundant and inconsistent data in the given decision table during the construction of the improved discernibility matrix, and a frequency function is employed as the evaluation measure of attribute importance. The example and experimental results in this paper have verified the simplicity, practicability and time efficiency of ARIMC. Due to the limitations of the discernibility matrix, the space performance is still an open research issue in existing attribute reduction algorithms. For the future work, we plan to apply the ARIMC algorithm to large-scale database systems to further explore its efficiency in both time and space.

## References

1. Pawlak, Z.: Rough sets, *International Journal of Computer and Information Sciences*, **11** (1982) 341-356.
2. Skowron, A. Rauszer, C.: The discernibility matrices and functions in information systems, *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory* (1992) 331-362.
3. Cercone, N.: Learning in relational database: a rough set approach, *Computational Intelligence*, **11** (2) (1995) 323-337 .
4. Hu, X.: Knowledge discovery in databases: an attribute-oriented rough set approach, *Doctoral Dissertation*, University of Regina, Canada (1995).

5. Wang, G.: Algebra view and information view of rough sets theory, *Proc. SPIE, Data Mining and knowledge Discovery: Theory, Tools, and Technology III*, **4384** (2001) 200-207 .
6. Wang, G: Calculation methods for core attributes of decision tables, *Chinese Journal of Computers* **26** (5) ( 2003) 1086-1088 .
7. Liu, Q.: *Rough set and its illation*, Science press (2001).
8. Hu, K.: A data mining research based on the concept lattice and rough set, *Doctoral Dissertation*, Tsinghua University, China (2001).
9. Dai, J., Li, Y.: An algorithm for reduction of attributes in decision system based on rough set, *Journal of Mini-Micro Systems* **24**(3) (2003).
10. Miao, D., Hu, G.: A heuristic algorithm of knowledge reduction, *Journal of Computer Research and Development* **36** (6) (1999) 681-684 .
11. Wang, G., Yu, H., Yang, D.: Algorithms of reduction in decision tables based on conditional information entropy, *Chinese Journal of Computers* **25**(7) (2002) 759-766 .
12. Ye, D.: A new discernibility matrix and the computation of a core, *Acta Electronica Sinica* **30**(7) (2002).

# Algebraic Properties of Adjunction-Based Fuzzy Rough Sets

Tingquan Deng[1], Yanmei Chen[2], and Guanghong Gao[2]

[1] College of Science, Harbin Engineering University, Harbin 150001 P.R. China
[2] Department of Mathematics, Harbin Institute of Technology, Harbin 150001 P.R. China
Deng.Tq@hrbeu.edu.cn, Chen.Yanmei@163.com

**Abstract.** A fuzzy rough set is a fuzzy generalization of rough set. There are already several definitions for it, and most of them are given with reference to a t-norm $*$, a fuzzy ($*$-)similarity relation and the duality principle. In this paper, a generalization of fuzzy rough sets is investigated regarding a general fuzzy relation and a lower semi-continuous fuzzy conjunction logical operator in its second argument. The generalized fuzzy rough approximation operators are established by using the adjunction between the fuzzy conjunction operator and a fuzzy implication operator. Algebraic properties of the generalized fuzzy rough approximation operators are discussed. It has been shown that information with much more necessity measure and with less probability measure for a fuzzy set can be mined in comparison with existing methods of fuzzy rough sets.

**Keywords:** Fuzzy logic, adjunction, fuzzy relation, fuzzy rough sets, necessity measure, probability measure.

## 1 Introduction

The theory of rough sets, initiated by Pawlak [13,14], is an excellent tool to handle granularity of data by revealing knowledge hidden in information systems with two sets called rough lower approximation and rough upper approximation. It has become a very active research theme in information and computer sciences, and has attracted wide attention of many researchers.

Since the values of attributes of data in a knowledge representation system usually involve vagueness and uncertainty, rough sets have been combined with fuzzy sets to deal data with fuzzy natures. One of the first work to fuzzify rough sets was contributed by Dubois and Prade [6]. Dubois and Prade modeled the concept of fuzzy rough sets by using a pair of fuzzy sets called fuzzy rough lower approximation and fuzzy rough upper approximation. The fuzzy rough approximations are generated by replacing the equivalence relation in the Pawlak rough universe with a fuzzy similarity relation. Meanwhile, the Zadeh *min* and

[1] This work was supported by the postdoctoral science-research developmental foundation of Heilongjiang province (LBH-Q05047) and the fundamental research foundation of Harbin Engineering University (HEUFT05087).

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 47–54, 2007.

*max* operators are used for characterizing the intersection and union of fuzzy sets. Radzikowska and Kerre [16] extended the Zadeh operators to a t-norm and a fuzzy implication, and studied three classes of fuzzy rough sets by taking into account three classes of particular implications. Morsi and Yakout [11] also developed a generalized definition of fuzzy rough sets within the framework of a lower semi-continuous t-norm $*$ and a fuzzy $*$-similarity relation. Other fuzzifications of rough sets and comparative studies on them [1,2,7,10,15,18,20] have been investigated. Most of classical fuzzy rough sets are generated based on the notions of fuzzy similarity relations and t-norms.

This paper presents an approach to fuzzy rough sets regarding a general fuzzy relation on an arbitrary universe, a lower semi-continuous fuzzy conjunction operator and its adjunctional implication operator. It will be shown that the presented fuzzy rough sets can be considered as a fuzzy generalization of Pawlak rough sets and as an extension of classical fuzzy rough sets.

The rest of this paper is organized as follows. In Section 2, some elementary concepts and operations from fuzzy logic are summarized. A generalized definition of fuzzy rough sets is introduced and basic algebraic properties of the generalized fuzzy rough approximation operators are comparatively investigated in Section 3. Section 4 studies the generalized fuzzy rough sets on special fuzzy rough universes and conclusions are given in Section 5.

## 2 Preliminaries

Let $\mathcal{I}$ denotes the unit interval $[0\,,1]$ and $\mathcal{I}^2 = \mathcal{I} \times \mathcal{I}$.

**Definition 1.** *A binary operator $* : \mathcal{I}^2 \to \mathcal{I}$ is called a fuzzy logical conjunction (conjunction, in short) if it is non-decreasing in both arguments satisfying $0 * 1 = 1 * 0 = 0$ and $1 * 1 = 1$. Let a binary operator $\to$: $\mathcal{I}^2 \to \mathcal{I}$ be non-increasing in its first argument, non-decreasing in its second, and satisfy the conditions $0 \to 0 = 1 \to 1 = 1$ and $1 \to 0 = 0$, then $\to$ is called a fuzzy logical implication (implication, in short). A t-norm $*$ is a commutative and associative conjunction satisfying $1 * s = s$ for all $s \in \mathcal{I}$.*

If $*$ is a conjunction and $\to$ an implication, then $s * 0 = 0 * s = s$ and $0 \to s = s \to 1 = 1$ for all $s \in \mathcal{I}$. When $\mathcal{I}$ reduces to $\{0\,,1\}$, $*$ and $\to$ will be replaced by the corresponding two-valued logical operators characterized by their boundary conditions.

**Definition 2.** *An implication $\to$ and a conjunction $*$ on $\mathcal{I}$ are said to be an adjunction if*

$$s * t \le r \iff t \le s \to r \tag{1}$$

*for all $s\,, t\,, r \in \mathcal{I}$. In which case, the pair $(\to\,, *)$ is called an adjunction.*

From Definition 2, if $(\to\,, *)$ is an adjunction, then the equivalence $1 * s = s \iff 1 \to s = s$ holds for all $s \in \mathcal{I}$. Furthermore, substituting $s * t$ for $r$ in (1) leads to $t \le s \to (s * t)$ for all $s\,, t \in \mathcal{I}$. If $*$ is a t-norm, then the implication $\to$ is called the adjunctional implication or R-implication of $*$.

**Proposition 1.** *If $(\rightarrow, *)$ is an adjunction on $\mathcal{I}$, then for each $s \in \mathcal{I}$, the unary operator $s * \bullet$ is lower semi-continuous on $\mathcal{I}$, whereas $s \rightarrow \bullet$ is upper semi-continuous on $\mathcal{I}$. Therefore, for any collection of points $\{t_i\} \subseteq \mathcal{I}$,*

$$s * \vee_i t_i = \vee_i (s * t_i), s \rightarrow \wedge_i t_i = \wedge_i (s \rightarrow t_i), \tag{2}$$

*where $\wedge$ and $\vee$ denote the infimum (minimum) and supremum (maximum).*

## 3 Generalized Fuzzy Rough Sets

Let $E$ be an arbitrary nonempty universe and $R$ be a fuzzy relation on $E$, the pair $(\mathcal{F}(E), R)$ is called a fuzzy rough universe, where $\mathcal{F}(E) = \{F \mid F : E \rightarrow \mathcal{I}\}$.

**Definition 3.** *Let $(\rightarrow, *)$ be an adjunction on $\mathcal{I}$ and $(\mathcal{F}(E), R)$ be a fuzzy rough universe. For a fuzzy set $F \in \mathcal{F}(E)$, its generalized fuzzy rough approximations are defined by the pair $\mathcal{F}R(F) = (\mathcal{L}_R(F), \mathcal{U}_R(F))$. $\mathcal{L}_R(F)$ and $\mathcal{U}_R(F)$ are called the generalized fuzzy rough lower approximation and the upper one of $F$ in $(\mathcal{F}(E), R)$, respectively, defined by, $x \in E$,*

$$\begin{aligned} \mathcal{L}_R(F)(x) &= \vee_{y \in E}(R(x, y) * \wedge_{z \in E}(R(z, y) \rightarrow F(z))), \\ \mathcal{U}_R(F)(x) &= \wedge_{y \in E}(R(y, x) \rightarrow \vee_{z \in E}(R(y, z) * F(z))). \end{aligned} \tag{3}$$

*If $\mathcal{L}_R(F) = \mathcal{U}_R(F)$, then $F$ is called definable or exact.*

If a fuzzy set $F$ is interpreted to be the probability distribution of a fuzzy object (or a fuzzy case), or a probability distribution can be derived from $F$ for a fuzzy case, then $\mathcal{L}_R(F)$ and $\mathcal{U}_R(F)$ are referred to as the necessity measure and the probability measure, respectively, of the fuzzy data $F$ in the theory of random sets [6,8,9,12,19].

**Proposition 2.** *Let $(\rightarrow, *)$ be an adjunction on $\mathcal{I}$, and let $(\mathcal{F}(E), R)$ be an arbitrary fuzzy rough universe, then for arbitrary $F, G \in \mathcal{F}(E)$,*

(1) *$\mathcal{L}_R(F) \subseteq \mathcal{L}_R(G)$ and $\mathcal{U}_R(F) \subseteq \mathcal{U}_R(G)$ if $F \subseteq G$;*
(2) *$\mathcal{L}_R(F) \subseteq F \subseteq \mathcal{U}_R(F)$;*
(3) *$\mathcal{L}_R(1_\emptyset) = 1_\emptyset, \mathcal{U}_R(1_E) = 1_E$;*
(4) *$\mathcal{L}_R\mathcal{L}_R(F) = \mathcal{L}_R(F)$, $\mathcal{U}_R\mathcal{U}_R(F) = \mathcal{U}_R(F)$;*
(5) *$\mathcal{L}_R(F) \subseteq \mathcal{U}_R\mathcal{L}_R(F) \subseteq \mathcal{U}_R(F)$, $\mathcal{L}_R(F) \subseteq \mathcal{L}_R\mathcal{U}_R(F) \subseteq \mathcal{U}_R(F)$.*

*Proof.* (1) Evidently.

(2) Let $F \in \mathcal{F}(E)$ and $x \in E$, then

$$\begin{aligned} \mathcal{L}_R(F)(x) &= \vee_{y \in E}(R(x, y) * \wedge_{z \in E}(R(z, y) \rightarrow F(z))) \\ &\leq \vee_{y \in E} \wedge_{z \in E} (R(x, y) * \vee\{r \in \mathcal{I} \mid R(z, y) * r \leq F(z)\}) \\ &= \vee_{y \in E} \wedge_{z \in E} \vee_{r \in \mathcal{I}} \{R(x, y) * r \mid R(z, y) * r \leq F(z)\} \\ &\leq \vee_{y \in E} \vee_{r \in \mathcal{I}} \{R(x, y) * r \mid R(x, y) * r \leq F(x)\} \leq F(x), \end{aligned}$$

and

$$\mathcal{U}_R(F)(x) = \wedge_{y \in E}(R(y, x) \rightarrow \vee_{z \in E}(R(y, z) * F(z)))$$

$$\geq \wedge_{y\in E} \vee_{z\in E} \vee\{r \in \mathcal{I} \mid R(y\,,x) * r \leq R(y\,,z) * F(z)\}$$
$$\geq \wedge_{y\in E} \vee \{r \in \mathcal{I} \mid R(y\,,x) * r \leq R(y\,,x) * F(x)\} \geq F(x)\,.$$

(3) They are straightforward from (2).

(4) Let $\underline{\mathcal{R}}(F)(x) = \wedge_{y\in E}(R(y\,,x) \to F(y))$ and $\overline{\mathcal{R}}(F)(x) = \vee_{y\in E}(R(x\,,y) * F(y))$, then $\mathcal{L}_R = \overline{\mathcal{R}}\,\underline{\mathcal{R}}$ and $\mathcal{U}_R = \underline{\mathcal{R}}\,\overline{\mathcal{R}}$. Thus $\mathcal{L}_R \subseteq \mathcal{L}_R\mathcal{L}_R$ and $\mathcal{U}_R\mathcal{U}_R \subseteq \mathcal{U}_R$.

On the other hand, the inclusions $\mathcal{L}_R\mathcal{L}_R(F) \subseteq \mathcal{L}_R(F)$ and $\mathcal{U}_R(F) \subseteq \mathcal{U}_R\mathcal{U}_R(F)$ are obvious. Therefore, $\mathcal{L}_R(F) = \mathcal{L}_R\mathcal{L}_R(F)$ and $\mathcal{U}_R\mathcal{U}_R(F) = \mathcal{U}_R(F)$.

(5) Obviously.

## 4 Generalized Fuzzy Rough Approximation Operators in Special Fuzzy Rough Universes

From Definition 3, if $E$ is a translation-invariant additive group and $G \in \mathcal{F}(E)$ is an arbitrary non-null fuzzy set, let $R(x\,,y) = G_y(x)\,(x\,,y \in E)$, then $(\mathcal{F}(E)\,,R)$ is a fuzzy rough universe, and $\mathcal{L}_R(F) = (F \ominus_{\to} G) \oplus_* G$ and $\mathcal{U}_R(F) = (F \oplus_* G) \ominus_{\to} G$, where $(H \oplus_* G)(x) = \vee_{y\in E}(G(x-y) * H(y))$ and $(H \ominus_{\to} G)(x) = \wedge_{y\in E}(G(y-x) \to H(y))$. Note that a lower semi-continuous conjunction in its second argument is necessary to ensure that the generalized fuzzy rough approximation operators share attractive fundamental properties and to keep link of the generalized fuzzy rough sets with fundamental morphological operators [5].

In terms of the relationship between adjunctional implication and $\nu$-implication $\to_\nu$ ($s \to_\nu t = \nu(s * \nu(t))$, $\nu(s) = 1 - s$) of a conjunction $*$ [4], we have the following consequence.

**Proposition 3.** *Let $(\to\,,*)$ be an adjunction satisfying $\to = \to_\nu$ on $\mathcal{I}$. If $R$ is a symmetric fuzzy relation on $E$, then $\mathcal{L}_R$ and $\mathcal{U}_R$ are dual.*

*Proof.* It is evident that $s * \nu(t) = \nu(s \to t)$ and $s \to \nu(t) = \nu(s * t)$ for all $s\,,t \in \mathcal{I}$. From which, the conclusion is implied.

**Proposition 4.** *Let $(\to\,,*)$ be an adjunction on $\mathcal{I}$, if $R$ is a reflexive fuzzy relation on $E$, then*

(1) $\mathcal{L}_R(1_E) = 1_E$, $\mathcal{U}_R(1_\emptyset) = 1_\emptyset$.

*Furthermore, if $*$ satisfies the boundary condition $1 * s = s\,(\forall\, s \in \mathcal{I})$, then*

(2) $\underline{\mathcal{R}}(F) \subseteq \mathcal{L}_R(F) \subseteq F \subseteq \mathcal{U}_R(F) \subseteq \overline{\mathcal{R}}(F)$ *for all $F \in \mathcal{F}(E)$;*
(3) *For arbitrary $r \in \mathcal{I}$, $\bar{r}$ is definable, where $\bar{r} \in \mathcal{F}(E)$ is a constant fuzzy set, defined by $\bar{r}(x) \equiv r$ for all $x \in E$.*

*Proof.* (1) It suffices to prove that $\mathcal{L}_R(1_E)(x) \geq 1$ and $\mathcal{U}_R(1_\emptyset)(x) \leq 0$ $(\forall\, x \in E)$.

Let $x \in E$, $\mathcal{L}_R(1_E)(x) = \vee_{y\in E}(R(x\,,y) * \wedge_{z\in E}(R(z\,,y) \to 1)) = \vee_{y\in E}(R(x\,,y) * 1) \geq R(x\,,x) * 1 = 1$. And, $\mathcal{U}_R(1_\emptyset)(x) = \wedge_{y\in E}(R(y\,,x) \to \vee_{z\in E}(R(y\,,z) * 0)) = \wedge_{y\in E}(R(y\,,x) \to 0) \leq R(x\,,x) \to 0 = 0$.

(2) It is clear that $\underline{\mathcal{R}}(F) \subseteq F \subseteq \overline{\mathcal{R}}(F)$. The results $\underline{\mathcal{R}}(F) \subseteq \mathcal{L}_R(F)$ and $\mathcal{U}_R(F) \subseteq \overline{\mathcal{R}}(F)$ follow from the monotone of $\underline{\mathcal{R}}$ and $\overline{\mathcal{R}}$.

(3) We need to prove the inclusions $\mathcal{U}_R(\bar{r}) \subseteq \bar{r} \subseteq \mathcal{L}_R(\bar{r})$.

Let $x \in E$, then $\mathcal{U}_R(\bar{r})(x) = \wedge_{y\in E}(R(y\,,x) \to \vee_{z\in E}(R(y\,,z) * r)) \leq R(x\,,x) \to \vee_{z\in E}(R(x\,,z) * r) = \vee_{z\in E}(R(x\,,z) * r) \leq \vee_{z\in E}(1 * r) = r$.

$\mathcal{L}_R(\bar{r})(x) = \vee_{y\in E}(R(x\,,y) * \wedge_{z\in E}(R(z\,,y) \to r)) \geq R(x\,,x) * \wedge_{z\in E}(R(z\,,x) \to r) = \wedge_{z\in E}(R(z\,,x) \to r) \geq \wedge_{z\in E}(1 \to r) = r$.

This proposition indicates that the fuzzy rough boundary $\mathcal{U}_R - \mathcal{L}_R$, which means $\mathcal{U}_R(F)(x) - \mathcal{L}_R(F)(x)$ for all $x \in E$, is 'smaller' than $\overline{\mathcal{R}} - \underline{\mathcal{R}}$. Therefore, the necessity measure of a given fuzzy data increases, whereas the probability measure decreases. Alternatively, the concept of proximity of degree [3,17,21,22] of two fuzzy sets may be used for characterizing the close-degree between the fuzzy rough approximations. Thus, $D(\underline{\mathcal{R}}(F)\,,\overline{\mathcal{R}}(F)) \leq D(\mathcal{L}_R(F)\,,\mathcal{U}_R(F))$ whatever the definition of the degree of proximity $D(\cdot\,,\cdot)$ is. The performance of data analysis is therefore enhanced.

**Proposition 5.** *Let $*$ be an associative conjunction on $\mathcal{I}$ satisfying $1 * s = s\,(\forall\, s \in \mathcal{I})$, and $(\to\,,*)$ be an adjunction. If $R$ is a reflexive and $*$-transitive fuzzy relation on $E$, then the generalized fuzzy rough approximations of an arbitrary fuzzy set are definable.*

*Proof.* From Proposition 4, it suffices to verify the inclusions $\mathcal{L}_R(F) \subseteq \underline{\mathcal{R}}(F)$ and $\overline{\mathcal{R}}(F) \subseteq \mathcal{U}_R(F)$ for all $F \in \mathcal{F}(E)$. Let $x \in E$, then

$$\begin{aligned}
\mathcal{L}_R(F)(x) &= \vee_{y\in E}(R(x\,,y) * \wedge_{z\in E}(R(z\,,y) \to F(z))) \\
&\leq \vee_{y\in E} \wedge_{z\in E} (R(x\,,y) * (\vee_{u\in E}(R(z\,,u) * R(u\,,y)) \to F(z))) \\
&\leq \vee_{y\in E} \wedge_{z\in E} \wedge_{u\in E}(R(x\,,y) * ((R(z\,,u) * R(u\,,y)) \to F(z))) \\
&\leq \vee_{y\in E} \wedge_{z\in E} (R(x\,,y) * ((R(z\,,x) * R(x\,,y)) \to F(z))) \\
&\leq \vee_{y\in E} \wedge_{z\in E} (R(z\,,x) \to F(z)) = \underline{\mathcal{R}}(F)(x), \\
\mathcal{U}_R(F)(x) &= \wedge_{y\in E}(R(y\,,x) \to \vee_{z\in E}(R(y\,,z) * F(z))) \\
&\geq \wedge_{y\in E} \vee_{z\in E} (R(y\,,x) \to (R(y\,,x) * R(x\,,z) * F(z))) \\
&\geq \wedge_{y\in E} \vee_{z\in E} (R(x\,,z) * F(z)) = \overline{\mathcal{R}}(F)(x)\,.
\end{aligned}$$

Therefore, $\mathcal{L}_R = \underline{\mathcal{R}}$, $\mathcal{U}_R = \overline{\mathcal{R}}$, and so $\mathcal{U}_R\mathcal{L}_R = \mathcal{L}_R$ and $\mathcal{L}_R\mathcal{U}_R = \mathcal{U}_R$.

Proposition 5 tells us that the presented fuzzy rough sets preserve the definability of fuzzy rough approximations. In which case, they reduce to the existing fuzzy rough sets in the literature [6,10,11,15,16,18,20]. The generalized fuzzy rough approximation operators can also be interpreted as the generalized opening and closure operators with respect to arbitrary fuzzy relations [1,2].

The definability of granularity is also preserved from the following proposition.

**Proposition 6.** *Let $*$ be a commutative conjunction on $\mathcal{I}$ satisfying $1 * s = s\,(\forall\, s \in \mathcal{I})$, and $(\to\,,*)$ be an adjunction, if $R$ is a fuzzy $*$-similarity relation on $E$, then for any $x \in E$, $[x]_R$ is definable, where $[x]_R(y) = R(x\,,y)$.*

*Proof.* By the $*$-similarity of $R$, it follows that $R(x\,,y) = \vee_{z\in E}(R(x\,,z) * R(z\,,y))$ and $R(x\,,y) = \wedge_{z\in E}(R(z\,,x) \to R(z\,,y))$ for all $x\,,y \in E$.

Let $u \in E$, then $\mathcal{L}_R([x]_R)(u) = \vee_{y\in E}(R(u\,,y) * \wedge_{z\in E}(R(z\,,y) \to R(x\,,z))) = \vee_{y\in E}(R(u\,,y) * R(y\,,x)) = R(u\,,x) = [x]_R(u)$. And, $\mathcal{U}_R([x]_R)(u) = \wedge_{y\in E}(R(y\,,u) \to \vee_{z\in E}(R(y\,,z) * R(x\,,z))) = \wedge_{y\in E}(R(y\,,u) \to R(y\,,x)) = R(u\,,x) = [x]_R(u)$.

**Proposition 7.** *Let $*$ be a conjunction on $\mathcal{I}$ satisfying $s*1=1*s=s\,(\forall\, s\in\mathcal{I})$, and $(\rightarrow,*)$ be an adjunction, if $R$ is a reflexive and $*$-transitive fuzzy relation on $E$, then $\mathcal{U}_R(1_y)(x)=R(x,y)$, $x,y\in E$.*

*Proof.* Let $x,y\in E$, then $\mathcal{U}_R(1_y)(x)=\wedge_{u\in E}(R(u,x)\rightarrow\vee_{v\in E}(R(u,v)*1_y(v)))=\wedge_{u\in E}(R(u,x)\rightarrow(R(u,y)*1))=\wedge_{u\in E}(R(u,x)\rightarrow R(u,y))=R(x,y)$.

**Proposition 8.** *Let $*$ be a commutative conjunction on $\mathcal{I}$ satisfying $1*s=s\,(\forall\, s\in\mathcal{I})$, and $(\rightarrow,*)$ be an adjunction such that $\rightarrow=\rightarrow_\nu$. If $R$ is a fuzzy $*$-similarity relation on $E$, then $\mathcal{L}_R(1_{\{x\}^c})(y)=\mathcal{L}_R(1_{\{y\}^c})(x)=\nu(R(x,y))$ for all $x,y\in E$.*

*Proof.* Clearly, $\mathcal{L}_R$ and $\mathcal{U}_R$ are dual. Thus $\mathcal{L}_R(1_{\{x\}^c})(y)=\nu(\mathcal{U}_R(\nu(1_{\{x\}^c}))(y))=\nu(\mathcal{U}_R(1_{\{x\}})(y))=\nu(\mathcal{U}_R(1_x)(y))=\nu(R(y,x))=\nu(R(x,y))$.

The equality $\mathcal{L}_R(1_{\{y\}^c})(x)=\nu(R(x,y))$ can be proved in the same way.

**Proposition 9.** *Let $*$ be a conjunction on $\mathcal{I}$ satisfying $1*s=s\,(\forall\, s\in\mathcal{I})$, and $(\rightarrow,*)$ be an adjunction, if $R$ is the identity relation on $E$, then $\mathcal{L}_R(F)=\mathcal{U}_R(F)=F$ for all $F\in\mathcal{F}(E)$.*

*Proof.* Let $F\in\mathcal{F}(E)$ and $x\in E$,

$\mathcal{L}_R(F)(x)=\vee_{y\in E}(R(x,y)*\wedge_{z\in E}(R(z,y)\rightarrow F(z)))=\vee_{y\in E}(R(x,y)*(1\rightarrow F(y)))=\vee_{y\in E}(R(x,y)*F(y))=1*F(x)=F(x)$.

The equality $\mathcal{U}_R(F)=F$ can be proved in the same manner.

Proposition 7–Proposition 9 lay down algebraic foundations of axiomatic characterization of generalized fuzzy rough sets.

**Proposition 10.** *Let $*$ be a conjunction on $\mathcal{I}$ satisfying $1*s=s\,(\forall\, s\in\mathcal{I})$, and $(\rightarrow,*)$ be an adjunction, if $R$ is a crisp symmetric relation on $E$, then*

$$\mathcal{L}_R(F)(x)=\vee_{y\in C(x)}\wedge_{z\in C(y)}F(z)\,,\mathcal{U}_R(F)(x)=\wedge_{y\in C(x)}\vee_{z\in C(y)}F(z)\,. \quad (4)$$

*for any $F\in\mathcal{F}(E)$ and $x\in E$, where $C(x)=\{y\mid(x,y)\in R\}$.*

*Proof.* Let $x\in E$, then $y\in C(x)\iff(x,y)\in R\iff 1_R(x,y)=1$. Thus

$$\begin{aligned}\mathcal{L}_R(F)(x)&=\vee_{y\in E}(1_R(x,y)*\wedge_{z\in E}(1_R(z,y)\rightarrow F(z)))\\&=\vee_{y\in C(x)}(1_R(x,y)*\wedge_{z\in C(y)}(1_R(z,y)\rightarrow F(z)))\\&=\vee_{y\in C(x)}\wedge_{z\in C(y)}(1_R(z,y)\rightarrow F(z))=\vee_{y\in C(x)}\wedge_{z\in C(y)}F(z)\,.\end{aligned}$$

The second can be proved in the same way.

In particular, if $R\in\mathcal{P}(E\times E)$ is an equivalence relation, then $\mathcal{L}_R(F)(x)=\wedge_{y\in[x]_R}F(y)$ and $\mathcal{U}_R(F)(x)=\vee_{y\in[x]_R}F(y)$.

In the case that both the relation $R$ and the set $F$ are crisp, the following proposition indicates that the generalized fuzzy rough sets reduce to the generalized rough sets.

**Proposition 11.** *Let $R \in \mathcal{P}(E \times E)$ be a symmetry relation, then for any conjunction $*$ and implication $\rightarrow$ on $\mathcal{I}$, $\mathcal{L}_R(1_X) = 1_{L_R(X)}$ and $\mathcal{U}_R(1_X) = 1_{U_R(X)}$, where $L_R(X) = \cup\{C(u) \mid C(u) \subseteq X\,, u \in E\}$ and $U_R(X) = (L_R(X^c))^c$ are called the generalized rough approximations of $X \in \mathcal{P}(E)$.*

*Proof.* Evidently, $\mathcal{L}_R$ and $\mathcal{U}_R$ are dual. It suffices to prove the equality $\mathcal{L}_R(1_X) = 1_{L_R(X)}$ for any $X \in \mathcal{P}(E)$.

Let $x \in E$ and $\mathcal{L}_R(1_X)(x) = 1$, then there exists $u \in E$ such that $1_R(x\,, u) = 1$ and $1_R(v\,, u) \rightarrow 1_X(v) = 1$ for all $v \in E$. So $C(u) \neq \emptyset$ since $x \in C(u)$. For every $z \in C(u)$, we have that $1_R(u\,, z) = 1_R(z\,, u) = 1$. If $z \in X^c$, then $1_X(z) = 0$. A contradiction arises. Thus $z \in X$, and so $C(u) \subseteq X$. Therefore $x \in L_R(X) = \cup\{C(u) \mid C(u) \subseteq X\,, u \in E\}$.

On the other hand, let $x \in L_R(X)$, then there exists $y \in E$ such that $x \in C(y)$ and $C(y) \subseteq X$. So $1_R(x\,, y) = 1_R(y\,, x) = 1$, and $1_X(v) = 1$ for all $v \in C(y)$.

$$\begin{aligned}\mathcal{L}_R(1_X)(x) &= \vee_{u\in E}(1_R(x\,, u) * \wedge_{v\in E}(1_R(v\,, u) \rightarrow 1_X(v)))\\ &\geq 1_R(x\,, y) * \wedge_{v\in E}(1_R(v\,, y) \rightarrow 1_X(v))\\ &= \wedge_{v\in E}(1_R(v\,, y) \rightarrow 1_X(v))\\ &= \wedge_{v\in C(y)}(1_R(v\,, y) \rightarrow 1) \wedge \wedge_{v\notin C(y)}(0 \rightarrow 1_X(v)) = 1\,.\end{aligned}$$

Thus $\mathcal{L}_R(1_X)(x) = 1$, and therefore $\mathcal{L}_R(1_X) = 1_{L_R(X)}$.

From Proposition 11, if $R$ is an equivalence relation, then $C(x)$ is exactly the $[x]_R$. In which case, the generalized fuzzy rough sets are in agreement with the Pawlak rough sets.

## 5 Conclusions

Necessity measure and probability measure are two important concepts in the theory of random sets and probability theory. Both of which can be used for characterizing the degree of beliefs of knowledge in data analysis and processing. Through replacing a t-norm and its t-conorm with a lower semi-continuous conjunction and its adjunctional implication, the presented approach to fuzzy rough sets is a fuzzy generalization of Pawlak rough sets in an arbitrary fuzzy rough universe (i.e., an arbitrary nonempty universe with a general fuzzy relation on it). A 'smaller' rough boundary and higher proximity degree between the generalized fuzzy rough approximations, as well as more necessity measures and less probability measures of fuzzy data are implied. Particularly, it has been proved that the main properties of the Pawlak rough sets have been preserved for the generalized fuzzy rough sets.

If the fuzzy rough universe for the study of generalized fuzzy rough sets is with a translation-invariant additive group structure, which is an appropriate one for image and signal analysis, the generalized fuzzy rough approximation operators have their special expressions that are closely linked with fundamental morphological operators. Much characterization of fuzzy rough sets with mathematical morphology theory is under consideration.

## References


1. Bodenhofer, U.: Generalized opening and closure operators of fuzzy relations. In Proceedings of the second International ICSC Congress on Computational Intelligence: Methods and Applications, Bangor, UK (2001) 677–683
2. Bodenhofer, U.: A unified framework of opening and closure operators with respect to arbitrary fuzzy relations. Soft Computing **7** (2003) 220–227
3. Chen, Y., Mo, Z.-W.: The close-degree of fuzzy rough sets and rough fuzzy sets and the application. Chinese Quarterly Journal of Mathematics **17** (2002) 70–77
4. Deng, T.-Q., Chen, Y.-M., Wu, C.-X.: Adjunction and duality of morphological operators. In Proceedings of International Conference on Fuzzy Information Processing Theories and Applications, Beijing (2003) 13–18
5. Deng, T.-Q., Heijmans, H.: Grey-scale morphology based on fuzzy logic. Journal of Mathematical Imaging Vision **16** (2002) 155-171
6. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. International Journal of General Systems **17**(1990) 191–209
7. Fernández Salido, J.M., Murakami, S.: Rough set analysis of a general type of fuzzy data using transitive aggregations of fuzzy similarity relation. Fuzzy Sets and Systems **139** (2003) 635–660
8. Inuiguchi, M., Tanino, T.: Necessity measures and fuzzy rough sets defined by certainty qualifications. In Proceedings of the Annual Conference of the North American Fuzzy Information Processing Society (Joint 9th IFSA World Congress and 20th NAFIPS International Conference), Vancouver (2001) 1940–1945
9. Jamisona, K.D., Lodwicka, W.A.: The construction of consistent possibility and necessity measures. Fuzzy Sets and Systems **132** (2002) 1–10
10. Mi J.-S., Zhang, W.-X.: An axiomatic characterization of a fuzzy generalization of rough sets. Information Sciences **160** (2004) 235–249
11. Morsi, N.N., Yakout, M.M.: Axiomatics for fuzzy rough sets. Fuzzy Sets and Systems **100** (1998) 27–342
12. Nguyen, H.T., Walker, E.A.: A First Course in Fuzzy Logic. 2nd edn. Chapman & Hall/CRC, Boca Raton (1999)
13. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences **11** (1982) 341–356
14. Pawlak, Z.: Rough Sets, Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
15. Radzikowska, A.M.: Fuzzy modal-like approximaton operators based on double residuated lattices. Journal of Applied Non-Classical Logics **16** (2006) 485–506
16. Radzikowska, A.M., Kerre, E.E.: A comparative study of fuzzy rough sets. Fuzzy Sets and Systems **126** (2002) 137–155
17. Sadaaki, M.: Proximity measures for terms based on fuzzy neighborhoods in document sets. International Journal of Approximate Reasoning **34** (2003) 181–199
18. Wu, W.-Z., Zhang, W.-X.: Constructive and axiomatic approaches of fuzzy approximation operators. Information Sciences **159** (2004) 233–254
19. Yao, Y.Y.: Two views of the theory of rough sets in finite universes. International Journal of Approximate Reasoning **15** (1996) 291–317
20. Yeung, D.S., Chen, D.-G., Tsang, C.C., Lee, W.T., Wang, X.-Z.: On the generalization of fuzzy rough sets. IEEE Transactions on Fuzzy Systems **13** (2005) 343–361
21. Zadeh, L.A.: Fuzzy sets. Information and Control **8** (1965) 338–353
22. Zimmerman, H.J.: Fuzzy Set Theory and its Applications. Academic Press, Boston (1991)

# Fuzzy Approximation Operators Based on Coverings

Tongjun Li[1,2] and Jianmin Ma[2]

[1] Mathematics, Physics and Information College,
Zhejiang Ocean University, Zhoushan, Zhejiang, 316004, P.R. China
litj@zjou.net.cn
[2] Institute for Information and System Sciences, Faculty of Science,
Xi'an Jiaotong University, Xi'an, Shaan'xi, 710049, P.R. China
majm@mail.xjtu.edu.cn

**Abstract.** This paper presents a general framework for the study of covering-based fuzzy approximation operators in which a fuzzy set can be approximated by some elements in a crisp or a fuzzy covering of the universe of discourse. Two types of approximation operators, crisp-covering-based rough fuzzy approximation operators and fuzzy-covering-based fuzzy rough approximation operators, are defined, their properties are examined in detail. Finally, the comparison of these new approximation operators is done, a sufficient and necessary condition is given under which some operators are equivalent, and approximation operator characterization of fuzzy partitions of the universe is obtained.

**Keywords:** Crisp coverings, fuzzy coverings, fuzzy-covering-based fuzzy rough approximation operators, fuzzy partitions, fuzzy sets.

## 1 Introduction

The theory of rough sets is proposed by Pawlak in 1982 [9], it is a new mathematical approach to deal with intelligent systems characterized by insufficient and incomplete information, and has been found very successful in many domains.

The establishment of Pawlak rough set model is based on a partition or an equivalence relation of the universe. However, in practical applications the knowledge of the domains can't be always described by partitions, so Pawlak model limited the applications of rough set theory. To address this issue, many researchers proposed several interesting and meaningful extensions of equivalence relation in the literature such as general binary relation [12,15], neighborhood system [13,16], covering and its generalization [1,3,6,10,18]. Particularly, Based on covering of the universe Pomykala [10,11] put forward a suggestion, and obtained two pairs of dual approximation operators. In addition, Yao [16,17] discussed this kind of extension by the notion of neighborhood and with granulated view respectively. On the other hand, another research topic to which many researchers payed attention is the fuzzy generalization of rough sets. Dubois and Prade [4], Chakrabarty et al. [2] introduced lower and upper approximations in fuzzy set theory to obtain an extended notions called rough fuzzy set and fuzzy rough set. By an axiomatic approach Morsi and Yakout [8] studied the fuzzy rough sets based on a fuzzy partition of the universe. In [7,14] Wu, Zhang and Mi defined generalized fuzzy rough approximation operators based on general binary fuzzy relations, which can be

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 55–62, 2007.

viewed as a fuzzy generalization of the approximation operator defined in [16]. Feng et al. [5] proposed the notion of covering-based generalized rough fuzzy sets in which fuzzy sets can be approximated by some elements in a covering of the universe.

This paper extends Pawlak rough set model on the basis of a crisp or a fuzzy covering of the universe. In the next section, we review two types of crisp-covering-based rough approximation operators and their interrelationships. In Section 3, four pairs of approximation operators are proposed, the two pairs defined in crisp-covering approximation space are called crisp-covering-based rough fuzzy approximation operators and the other two pairs defined in fuzzy-covering approximation space are called fuzzy-covering-based fuzzy rough approximation operators. Some basic properties of these approximation operators are examined. In Section 4, we compare the new approximation operators, and find that some conclusions w.r.t. the approximation operators in crisp-covering approximation space do not hold in the fuzzy-covering approximation space. To address this issue, we investigate the sufficient and necessary condition for the equivalence of two pairs of fuzzy-covering-based fuzzy rough approximation operators and the approximation operator characterization of fuzzy partition of the universe. We then conclude the paper with a summary in Section 5.

## 2 Preliminaries

Let $U$ be a finite and nonempty universe. By a *covering* of $U$, denoted by $C$, we mean a finite family of nonempty subsets of $U$ such that the union of all elements of it is $U$. Then $(U, C)$ is called a *crisp-covering approximation space*. Specially, if $C$ consists of pairwise disjoint subsets then it is called a *crisp partition* of $U$, and $(U, C)$ change into Pawlak approximation space. For $X \subseteq U$, the lower and upper rough approximations of $X$, $\underline{C}(X)$ and $\overline{C}(X)$, are defined by Pawlak as follows:

$$\underline{C}(X) = \cup \{C \in C \mid C \subseteq X\}, \quad \overline{C}(X) = \cup \{C \in C \mid C \cap X \neq \emptyset\}.$$

Let $(U, C)$ be a crisp-covering approximation space. In order to extend Pawlak rough set model in $(U, C)$, a natural approach is to replace the equivalence classes in the definition above with the elements in $C$. However, the obtained lower and upper rough approximation operators are not dual, i.e. the following properties may not be satisfied:

$$\underline{C}(X) = \sim \overline{C}(\sim X), \quad \overline{C}(X) = \sim \underline{C}(\sim X), \quad \forall X \subseteq U.$$

Where $\sim X$ denotes the complement of $X \subseteq U$. To resolve this problem, many authors [10,16,17] propose a scheme from which two pairs of dual approximation operators are obtained: $\forall X \subseteq U$,

$$\begin{aligned} &\underline{C'}(X) = \cup \{C \in C \mid C \subseteq X\} = \{x \in U \mid \exists C \in C(x \in C, C \subseteq X)\}, \\ &\overline{C'}(X) = \sim \underline{C'}(\sim X) = \{x \in U \mid \forall C \in C(x \in C \Rightarrow C \cap X \neq \emptyset)\}; \end{aligned} \tag{1}$$

$$\begin{aligned} &\overline{C''}(X) = \cup \{C \in C \mid C \cap X \neq \emptyset\} = \{x \in U \mid \exists C \in C(x \in C, C \cap X \neq \emptyset)\}, \\ &\underline{C''}(X) = \sim \overline{C''}(\sim X) = \{x \in U \mid \forall C \in C(x \in C \Rightarrow C \subseteq X)\}. \end{aligned} \tag{2}$$

We simply call $\underline{C'}$ and $\underline{C''}$ : $\mathcal{P}(U) \to \mathcal{P}(U)$ *the lower rough approximation operators* of $(U, C)$, and $\overline{C'}$ and $\overline{C''}$ : $\mathcal{P}(U) \to \mathcal{P}(U)$ *the upper rough approximation operators* of $(U, C)$ respectively.

The following two propositions follow from [16, Theorems 7 and 8].

**Proposition 1.** *Let $(U, C)$ be a crisp-covering approximation space. For $X \subseteq U$,*

$$\underline{C''}(X) \subseteq \underline{C'}(X) \subseteq X \subseteq \overline{C'}(X) \subseteq \overline{C''}(X).$$

**Proposition 2.** *The two pairs of the lower and upper rough approximation operators defined by* (1) *and* (2) *are equivalent if and only if $C$ is a crisp partition of $U$.*

## 3 Fuzzy Approximation Operators Based on Coverings

Let $U$ be a finite and nonempty universe. By fuzzy sets in $U$ we mean Zadeh fuzzy sets, and denote the family of all fuzzy sets in $U$ by $\mathcal{F}(U)$. $1_x$ denote the fuzzy singleton with value 1 at $x$ and 0 elsewhere. The denotations $\cup$, $\cap$ and $\sim$ mean Zadeh's fuzzy union, intersection and complement respectively. $\vee$ and $\wedge$ denote max and min respectively.

A fuzzy relation $R$ on $U$ (i.e. $R \in I^{U\times U}$) is said to be a *fuzzy similarity relation* [8] if satisfies for $x, y, z \in U$ the following conditions: $R(x, x) = 1$; $R(x, y) = R(y, x)$ ; $R(x, z) \wedge R(z, y) \leq R(x, y)$.

Let $C = \{C_1, \cdots, C_k\}$ be a finite family of nonempty fuzzy sets in $U$. Denote $\{1, ..., k\}$ by $K$. If $\cup_{i=1}^{k} C_i = U$, then we call $C$ a *fuzzy covering* of $U$, and $(U, C)$ a *fuzzy-covering approximation space*. If for every $C_i \in C$ $(i \in K)$, $C_i$ is normalized (i.e $\vee_{x\in U} C_i(x) = 1$), then we call the fuzzy covering $C$ a *normal fuzzy covering* of $U$. Fuzzy covering $C$ of $U$ is said to be a *fuzzy partition* of $U$ [8] if it satisfies the following conditions: Every fuzzy set in $C$ is normalized; For any $x \in U$ there is exactly one $i \in K$ with $C_i(x) = 1$; If $i, j \in K$ such that $C_i(x) = C_j(y) = 1$, then $C_i(y) = C_j(x) = \vee_{z\in U}[C_i(z) \wedge C_j(z)]$. It can be known from [8, Proposition 2.3] that there is a canonical one to one correspondence between fuzzy similarity relations on $U$ and fuzzy partitions of $U$.

In fuzzy-covering approximation space $(U, C)$, by (1) and (2) we can define two pairs of approximation operators as follows: $\forall X \in \mathcal{F}(U), x \in U$,

$$\begin{aligned} \underline{C'_{FR}}(X)(x) &= \vee_{i=1}^{k}\{C_i(x) \wedge \wedge_{y\in U}[(1 - C_i(y)) \vee X(y)]\}, \\ \overline{C'_{FR}}(X)(x) &= \wedge_{i=1}^{k}\{(1 - C_i(x)) \vee \vee_{y\in U}[C_i(y) \wedge X(y)]\}; \end{aligned} \tag{3}$$

$$\begin{aligned} \underline{C''_{FR}}(X)(x) &= \wedge_{i=1}^{k}\{(1 - C_i(x)) \vee \wedge_{y\in U}[(1 - C_i(y)) \vee X(y)]\}, \\ \overline{C''_{FR}}(X)(x) &= \vee_{i=1}^{k}\{C_i(x) \wedge \vee_{y\in U}[C_i(y) \wedge X(y)]\}. \end{aligned} \tag{4}$$

$\underline{C'_{FR}}$ and $\underline{C''_{FR}}$ : $\mathcal{F}(U) \to \mathcal{F}(U)$ are referred to as *the fuzzy-covering-based lower fuzzy rough approximation operators* on $(U, C)$, and $\overline{C'_{FR}}$ and $\overline{C''_{FR}}$ : $\mathcal{F}(U) \to \mathcal{F}(U)$ *the fuzzy-covering-based upper fuzzy rough approximation operators* on $(U, C)$ respectively.

If fuzzy covering $C$ is degenerated to a crisp covering then the fuzzy rough approximation operators defined by (3) and (4) will degenerated to the following approximation operators: $\forall X \in \mathcal{F}(U)$,

$$\underline{C'_{RF}}(X)(x) = \vee_{x\in C_i} \wedge_{y\in C_i} X(y), \quad \overline{C'_{RF}}(X)(x) = \wedge_{x\in C_i} \vee_{y\in C_i} X(y); \tag{5}$$

$$\underline{C''_{RF}}(X)(x) = \wedge_{x\in C_i} \wedge_{y\in C_i} X(y), \quad \overline{C''_{RF}}(X)(x) = \vee_{x\in C_i} \vee_{y\in C_i} X(y). \tag{6}$$

Then $\underline{C'_{RF}}$ and $\underline{C''_{RF}} : \mathcal{F}(U) \to \mathcal{F}(U)$ are referred to as *the crisp-covering-based lower rough fuzzy approximation operators* on $(U, C)$, and $\overline{C'_{RF}}$ and $\overline{C''_{RF}} : \mathcal{F}(U) \to \mathcal{F}(U)$ *the crisp-covering-based upper rough fuzzy approximation operators* on $(U, C)$ respevtively.

In sequel, to simplify, we call both of fuzzy rough approximation operators and rough fuzzy approximation operators the fuzzy approximation operators.

It can be verified that if the fuzzy covering $C$ degenerate to a crisp covering and $X \subseteq U$, then $\underline{C'_{FR}}(X)$ and $\overline{C'_{FR}}(X)$ degenerate to $\underline{C'}(X)$ and $\overline{C'}(X)$, and $\underline{C''_{FR}}(X)$ and $\overline{C''_{FR}}(X)$ degenerate to $\underline{C''}(X)$ and $\overline{C''}(X)$ respectively. Hence the fuzzy approximation operators defined by (3)-(6) are a kind of fuzzy generalizations of the rough approximation operators defined by (1) and (2). In addition, it is also easy to verify that for any $X \in \mathcal{F}(U)$ the crisp-covering-based lower rough fuzzy approximation $\underline{C'_{RF}}(X)$ equals to $X_*$ defined in [5] and called the covering-based fuzzy lower approximation of $X$. Hence, the covering-based fuzzy lower approximation operator defined by Feng et al. [5] is a special case of the lower fuzzy rough approximation operator defined by (3).

**Theorem 1.** *Let $(U, C)$ be a crisp-covering approximation space. $\underline{C'_{RF}}$ and $\overline{C'_{RF}}$ satisfy the following properties: $\forall X, Y \in \mathcal{F}(U)$,*

(1) $\underline{C'_{RF}}(\emptyset) = \emptyset, \quad \overline{C'_{RF}}(U) = U,$

(2) $\underline{C'_{RF}}(X) = \sim \overline{C'_{RF}}(\sim X), \quad \overline{C'_{RF}}(X) = \sim \underline{C'_{RF}}(\sim X),$

(3) $\underline{C'_{RF}}(U) = U, \quad \overline{C'_{RF}}(\emptyset) = \emptyset,$

(3′) $\underline{C'_{RF}}(\emptyset) = \emptyset, \quad \overline{C'_{RF}}(U) = U,$

(4) $X \subseteq Y \Rightarrow \underline{C'_{RF}}(X) \subseteq \underline{C'_{RF}}(Y), \quad \overline{C'_{RF}}(X) \subseteq \overline{C'_{RF}}(Y),$

(5) $\underline{C'_{RF}}(X) \subseteq X, \quad X \subseteq \overline{C'_{RF}}(X),$

(6) $\underline{C'_{RF}}(\underline{C'_{RF}}(X)) \supseteq \underline{C'_{RF}}(X), \quad \overline{C'_{RF}}(X) \supseteq \overline{C'_{RF}}(\overline{C'_{RF}}(X)).$

*Generally, in fuzzy-covering approximation space $(U, C)$, except* (3′), (5) *and* (6), *$\underline{C'_{FR}}$ and $\overline{C'_{FR}}$ satisfy the other properties listed above. Specially if $C$ is a normal fuzzy covering of $U$ then the property* (3′) *w.r.t. $\underline{C'_{FR}}$ and $\overline{C'_{FR}}$ holds.*

By Theorem 1 we know that the properties satisfied by $\underline{C'_{RF}}$ and $\overline{C'_{RF}}$ are more than those satisfied by $\underline{C'_{FR}}$ and $\overline{C'_{FR}}$. In addition, we must note that neither $\underline{C'_{RF}}$ and $\overline{C'_{RF}}$ nor $\underline{C'_{FR}}$ and $\overline{C'_{FR}}$ satisfy the following property:

$$\underline{C'_{FR}}(X \cap Y) = \underline{C'_{FR}}(X) \cap \underline{C'_{FR}}(Y), \quad \overline{C'_{FR}}(X \cup Y) = \overline{C'_{FR}}(X) \cup \overline{C'_{FR}}(Y). \tag{7}$$

On account of the restriction of pages we only give an example to illustrate (5) and Eq. (7) w.r.t. $\underline{C'_{FR}}$ and $\overline{C'_{FR}}$.

**Example 1.** Let $U = \{1, 2, 3\}, C = \{\{0.1/1, 1/2, 0.7/3\}, \{1/1, 0.6/2, 0.3/3\}, \{0.5/1, 0.9/2, 1/3\}\}$. Then $(U, C)$ is a fuzzy-covering approximation space. Given $X_1 = \{1/1, 1/2, 0/3\}$

and $X_2 = \{0/1, 1/2, 1/3\}$, we can figure out $\overline{C'_{FR}}(X_1 \cup X_2) = \overline{C'_{FR}}(U) = U = \{1/1, 1/2, 1/3\}$, $\overline{C'_{FR}}(X_1) = \{0.9/1, 0.9/2, 0.9/3\}$ and $\overline{C'_{FR}}(X_2) = \{0.6/1, 0.6/2, 0.7/3\}$. Thus $X_1 \notin \overline{C'_{FR}}(X_1)$ and $\overline{C'_{FR}}(X_1) \cup \overline{C'_{FR}}(X_2) = \{0.9/1, 0.9/2, 0.9/3\} \neq \{1/1, 1/2, 1/3\} = \overline{C'_{FR}}(X_1 \cup X_2)$. By the duality we have $\underline{C'_{FR}}(\sim X_1) \not\subseteq \sim X_1$ and $\underline{C'_{FR}}(\sim X_1 \cap \sim X_2) \neq \underline{C'_{FR}}(\sim X_1) \cap \underline{C'_{FR}}(\sim X_2)$. Thus the property (5) and Eq. (7) w.r.t. $\underline{C'_{FR}}$ and $\overline{C'_{FR}}$ do not hold.

A fuzzy relation on $U$ can be induced from a fuzzy-covering approximation space $(U, C)$ which is connected with approximation operators $\underline{C''_{FR}}$ and $\overline{C''_{FR}}$ and defined by:

$$R_{com}(x, y) = \vee_{i=1}^{k}[C_i(x) \wedge C_i(y)], \quad \forall x, y \in U.$$

Obviously $R_{com}$ is reflexive and symmetric, but it may not be transitive.

**Theorem 2.** *Let $(U, C)$ be a fuzzy-covering approximation space. For $X \in \mathcal{F}(U)$,*

$$\underline{C''_{FR}}(X) = \underline{R_{com}}(X), \quad \overline{C''_{FR}}(X) = \overline{R_{com}}(X).$$

*Where for $x \in U$,*

$$\underline{R_{com}}(X)(x) = \wedge_{y \in U}[(1 - R_{com}(x, y)) \vee X(y)], \ \overline{R_{com}}(X)(x) = \vee_{y \in U}[R_{com}(x, y) \wedge X(y)].$$

**Proof.** $\forall X \in \mathcal{F}(U), x \in U, \overline{C''_{FR}}(X)(x) = \vee_{i=1}^{k}\{C_i(x) \wedge \vee_{y \in U}[C_i(y) \wedge X(y)]\} = \vee_{i=1}^{k}\{\vee_{y \in U}[C_i(x) \wedge C_i(y) \wedge X(y)]\} = \vee_{y \in U} \vee_{i=1}^{k}[C_i(x) \wedge C_i(y) \wedge X(y)] = \vee_{y \in U}\{\vee_{i=1}^{k}[C_i(x) \wedge C_i(y)] \wedge X(y)\} = \vee_{y \in U}[R_{com}(x, y) \wedge X(y)] = \overline{R_{com}}(X)(x)$, that is, $\overline{C''_{FR}}(X) = \overline{R_{com}}(X)$. The another equation can be proved similarly. □

By Theorem 2 and [14, Theorems 5 and 7] we can gain the following theorem.

**Theorem 3.** *Let $(U, C)$ be a fuzzy-covering approximation space. $\underline{C''_{FR}}$ and $\overline{C''_{FR}}$ satisfy the following properties: $\forall X, Y \in \mathcal{F}(U), x, y \in U$,*

(1) $\underline{C''_{FR}}(X) = \sim \overline{C''_{FR}}(\sim X), \quad \overline{C''_{FR}}(X) = \sim \underline{C''_{FR}}(\sim X)$,

(2) $\underline{C''_{FR}}(U) = U, \quad \overline{C''_{FR}}(\emptyset) = \emptyset$,

(3) $\underline{C''_{FR}}(X) \subseteq X \subseteq \overline{C''_{FR}}(X)$,

(4) $X \subseteq Y \Rightarrow \underline{C''_{FR}}(X) \subseteq \underline{C''_{FR}}(Y), \quad \overline{C''_{FR}}(X) \subseteq \overline{C''_{FR}}(Y)$,

(5) $\underline{C''_{FR}}(X \cap Y) = \underline{C''_{FR}}(X) \cap \underline{C''_{FR}}(Y), \ \overline{C''_{FR}}(X \cup Y) = \overline{C''_{FR}}(X) \cup \overline{C''_{FR}}(Y)$,

(6) $\underline{C''_{FR}}(1_{U-\{x\}})(y) = \underline{C''_{FR}}(1_{U-\{y\}})(x), \ \overline{C''_{FR}}(1_x)(y) = \overline{C''_{FR}}(1_y)(x)$,

(7) *if $C$ is a normal fuzzy covering of $U$, then* $\underline{C''_{FR}}(\emptyset) = \emptyset, \quad \overline{C''_{FR}}(U) = U$.

## 4 Comparison of Fuzzy Approximation Operators

**Theorem 4.** *Let $(U, C)$ be a fuzzy-covering approximation space. For $X \in \mathcal{F}(U)$,*

$$\underline{C''_{FR}}(X) \subseteq \underline{C'_{FR}}(X), \ \overline{C'_{FR}}(X) \subseteq \overline{C''_{FR}}(X).$$

*In special, if $(U, C)$ is a crisp-covering approximation space, we have*

$$\underline{C''_{RF}}(X) \subseteq \underline{C'_{RF}}(X) \subseteq X \subseteq \overline{C'_{RF}}(X) \subseteq \overline{C''_{RF}}(X), \quad X \in \mathcal{F}(U).$$

**Proof.** Since $C$ is a fuzzy covering of $U$, we have that for every $x \in U$ there exists a $C_{i_x} \in C(i_x \in K)$ such that $C_{i_x}(x) = 1$. So for $X \in \mathcal{F}(U)$, we have $\underline{C''_{FR}}(X)(x) = \wedge_{i=1}^{k}\{(1 - C_i(x)) \vee \wedge_{y\in U}[(1 - C_i(y)) \vee X(y)]\} \leq (1 - C_{i_x}(x)) \vee \wedge_{y\in U}[(1 - C_{i_x}(y)) \vee X(y)] = \wedge_{y\in U}[(1 - C_{i_x}(y)) \vee X(y)] = C_{i_x}(x) \wedge \wedge_{y\in U}[(1 - C_{i_x}(y)) \vee X(y)] \leq \vee_{i=1}^{k}\{C_i(x) \wedge \wedge_{y\in U}[(1 - C_i(y)) \vee X(y)]\} = \underline{C'_{FR}}(X)(x)$, that is, $\underline{C''_{FR}}(X) \leq \underline{C'_{FR}}(X)$. By the duality, $\overline{C'_{FR}}(X) \subseteq \overline{C''_{FR}}(X)$ for all $X \in \mathcal{F}(U)$. Furthermore, since fuzzy rough approximation operators are extension of rough fuzzy approximation operators, combining Theorem 1 (5) we can gain another formula. □

Proposition 2 shows that the two pairs of rough approximation operators defined by (1) and (2) are equivalent if and only if $C$ is a crisp partition of $U$. Now a natural question is that as their fuzzy extensions, whether the fuzzy rough approximation operators defined by (3) and (4) are equivalent when the fuzzy covering $C$ is a fuzzy partition of $U$? Consider the next example.

**Example 2.** Let $U = \{x_1, x_2\}$, and $C = \{\{1.0/x_1, 0.3/x_2\}, \{0.3/x_1, 1.0/x_2\}\} \subseteq \mathcal{F}(U)$. Then $C$ is a fuzzy covering on $U$. Clearly $C$ is a fuzzy partition of $U$. For the fuzzy singleton $1_{x_1}$ we have $\overline{C'_{FR}}(1_{x_1}) = \{0.7/x_1, 0.3/x_2\}$ and $\overline{C''_{FR}}(1_{x_1}) = \{1.0/x_1, 0.3/x_2\}$. Thus $\overline{C'_{FR}}(1_{x_1}) \neq \overline{C''_{FR}}(1_{x_1})$, which shows that $\overline{C'_{FR}} = \overline{C''_{FR}}$ does not hold.

Example 2 illuminates that in a fuzzy-covering approximation space $(U, C)$, when $C$ is a fuzzy partition, the two pairs of approximation operators $\underline{C'_{FR}}$, $\overline{C'_{FR}}$ and $\underline{C''_{FR}}$ and $\overline{C''_{FR}}$ are unnecessarily identical respectively! Thus the next question is what the conditions for their equivalence are? Furthermore, when $C$ is a fuzzy partition, what properties must be satisfied by these approximation operators? The two theorems below can answer these questions.

**Theorem 5.** *Let $(U, C)$ be a fuzzy-covering approximation space. Then $C$ is a fuzzy partition of $U$ if and only if the equation $\{\overline{C''_{FR}}(1_x) \mid x \in U\} = C$ holds.*

**Proof.** "⇒" Assume that $C$ is a fuzzy partition of $U$. Let $D_i = \{x \in U \mid C_i(x) = 1\}$ $(i \in K)$. By the definition of fuzzy partition we have that $\mathcal{D} = \{D_1, D_2, \cdots, D_k\}$ is a crisp partition of $U$. For $C_i \in C$ and $x \in U$ there exists a $j \in K$ such that $x \in D_j$ and for any $y \in D_i$, $C_i(x) = C_j(y)$ holds. Then for any $x, y \in U$ and $C_i \in C$, there exist $j, l \in K$ such that $x \in D_j$ and $y \in D_l$, Furthermore, we can gain $C_i(x) \wedge C_i(y) = \vee_{z\in D_i}[C_j(z) \wedge C_l(z)]$. Hence, by the definition of fuzzy partition we have $\overline{C''_{FR}}(1_x)(y) = \vee_{i\in K}[C_i(x) \wedge C_i(y)] = \vee_{i\in K} \vee_{z\in D_i} [C_j(z) \wedge C_l(z)] = \vee_{z\in U}[C_j(z) \wedge C_l(z)] = C_j(y)$, that is, $\overline{C''_{FR}}(1_x) = C_j$. Noticing that $C$ ia a fuzzy partition of $U$ we can conclude that $\{\overline{C''_{FR}}(1_x) \mid x \in U\} = C$.

"⇐" Suppose that $\{\overline{C''_{FR}}(1_x) \mid x \in U\} = C$. Let $R(x, y) = \overline{C''_{FR}}(1_x)(y)$ for $x, y \in U$. Then $R$ is a fuzzy relation on $U$, according to [8, Proposition 2.3], in order to prove that $C$ is a fuzzy partition, it suffices to prove that $R$ ia a fuzzy similarity relation on $U$.

The reflexivity of $R$ follows from $R(x, x) = \overline{C''_{FR}}(1_x)(x) = \vee_{i\in K} C_i(x) = 1$ for all $x \in U$. For any $x, y \in U$, $R(x, y) = \vee_{i\in K}[C_i(x) \wedge C_i(y)]$, which implies that $R$ ia symmetric. For any $x, y, z \in U$, by the symmetry of $R$ and the supposition we have $\vee_{z\in U}[R(x, z) \wedge R(z, y)] = \vee_{z\in U}[R(z, x) \wedge R(z, y)] = \vee_{z\in U}[\overline{C''_{FR}}(1_z)(x) \wedge \overline{C''_{FR}}(1_z)(y)] = \vee_{i\in K}[C_i(x) \wedge C_i(y)] = R(x, y)$, that is, $R(x, z) \wedge R(z, y) \leq R(x, y)$, which indicate that $R$ is transitive. Finally, we can conclude that $R$ is a fuzzy similarity relation on $U$. □

From [8, Proposition 2.3], Theorem 5 and its proof we can know that when $C$ is a fuzzy partition, $R_{com}$ is just a fuzzy similarity relation determined by $C$ and $\overline{C''_{FR}}$ coincides with the $T$-upper approximation operator, given in [8] for $T$=Min.

**Theorem 6.** *Let $(U, C)$ be a fuzzy-covering approximation space. Two pairs of the lower and upper fuzzy approximation operators defined by* (3) *and* (4) *are equivalent if and only if $C$ is a crisp partition of $U$.*

**Proof.** "⇒" Assume that $\underline{C'_{FR}}(X) = \underline{C''_{FR}}(X)$ and $\overline{C'_{FR}}(X) = \overline{C''_{FR}}(X)$ for all $X \in \mathcal{F}(U)$. Then $\forall x \in U, \overline{C'_{FR}}(1_x)(x) = \wedge_{i=1}^{k}[(1-C_i(x))\vee C_i(x)]$ and $\overline{C''_{RF}}(1_x)(x) = \vee_{i=1}^{k} C_i(x)$. Noting that $C$ is a fuzzy covering of $U$ we have $\wedge_{i=1}^{k}[(1 - C_i(x)) \vee C_i(x)] = \vee_{i=1}^{k} C_i(x) = 1$, i.e. $1 - C_i(x) = 1$ or $C_i(x) = 1$ for all $x \in U$, which implies that the covering $C$ is a crisp covering of $U$. Furthermore, from Proposition 2 we can deduce that $C$ is a crisp partition of $U$.

"⇐" Suppose that $C$ be a crisp partition of $U$. Then $(U, C)$ is a crisp approximation space, the fuzzy rough approximation operators defined by (3) and (4) degenerate to the rough fuzzy approximation operators defined by (5) and (6) respectively. Then it is only needed to prove that $\underline{C'_{RF}} = \underline{C''_{RF}}$ and $\overline{C'_{RF}} = \overline{C''_{RF}}$.

For every $x \in U$, there exists a $C_{i_x} \in C$ such that $x \in C_{i_x}$, and for $C_i, C_j \in C$ with $x \in C_i$ and $x \in C_j$, we have $C_i = C_j$. By (5) and (6), for any $X \in \mathcal{F}(U)$ and $x \in U$ we have $\underline{C'_{RF}}(X)(x) = \vee_{x\in C_i} \wedge_{y\in C_i} X(y) = \wedge_{y\in C_{i_x}} X(y)$ and $\underline{C''_{RF}}(X)(x) = \wedge_{x\in C_i} \wedge_{y\in C_i} X(y) = \wedge_{y\in C_{i_x}} \overline{X(y)}$. Thus $\underline{C'_{RF}}(X) = \underline{C''_{RF}}(X)$ for all $X \in \mathcal{F}(U)$, By the duality we also have $\overline{C'_{RF}}(X) = \overline{C''_{RF}}(X)$ for all $X \in \mathcal{F}(U)$. □

## 5 Conclusion

In this paper, we have developed a general framework of the study of covering-based fuzzy approximation operators. With the proposed approximation operators, fuzzy sets can be approximated by a crisp or a fuzzy covering of the universe. The crisp-covering-based rough fuzzy approximation operators and fuzzy-covering-based fuzzy rough approximation operators are all fuzzy extensions of some existing rough approximation operators based on crisp covering of the universe, and the crisp-covering-based lower rough fuzzy approximation operator coincides with the lower rough fuzzy approximation operator defined in [5]. The properties of new defined approximation operators have been studied in detail. We have compared two types of fuzzy-covering-based fuzzy rough approximation operator, and given a sufficient and necessary condition for their equivalence which need the covering must be a crisp partition and not a fuzzy partition of the universe, with which we gained an approximation operator characterization of fuzzy partitions of the universe. The fuzzy approximation operators proposed here may be used to unravel knowledge hidden in fuzzy decision systems.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 60673096) and the Scientific Research Project of the Education Department of Zhejiang Province in China (No. 20061126).

## References


1. Bonikowski, Z., Bryniariski, E., Skardowska, V.W.: Extension and intensions in the rough set theory. *Information Sciences* **107** (1998) 149–167
2. Chakrabarty, K., Biawas, R., Nanda, S.: Fuzziness in rough sets. *Fuzzy Sets and Systems* **10** (2000) 247–251
3. De Cock, M., Cornelis, C., Kerre, E. E.: Fuzzy rough rets: Beyond the obvious. Proceedings of the 2004 IEEE International Conference on Fuzzy Systems **1** (2004) 103–108
4. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. *International Journal of Gerneral Systems* **17** (1990) 191–208
5. Feng, T., Mi, J.-S., Wu, W.-Z.: Covering-based generalized rough fuzzy sets. G. Wang et al. (Eds.): RSKT 2006, LNAI **4062**, (2006) 208–215
6. Li, T.-J.: Rough approximation operators in covering approximation spaces. S. Greco et al. (Eds.): RSCTC 2006, LNAI **4259**, (2006) 174–182
7. Mi, J.-S., Zhang, W.-X.:An axiomatic characterization of a fuzzy generalization of rough sets. *Information Science* **160** (2004) 235–249
8. Morsi, N.N., Yakout, M.M.: Axiomtics of fuzzy rough sets. *Fuzzy Sets and Systems* **100** (1998) 327–342
9. Pawlak, Z.: Rough sets. *International Journal of Computer and Information Science* **11** (1982) 341–356
10. Pomykala, J.A.: Approximation operations in approximation space. Bulletin of the Polish Academy of Sciences: Mathematics **35** (1987) 653–662
11. Pomykala, J.A.: On definability in the nondeterministic information system. Bulletin of the Polish Academy of Sciences: Mathematics **36** (1988), 193–210
12. Slowinski, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Transactions on Knowledge and Data Engineering **12(2)** (2000) 331–336
13. Wu, W.-Z., Zhang, W.-X.: Neighborhood operator systems and appoximations. *Infomation Sciences* **144** (2002) 201–217
14. Wu, W.-Z., Zhang, W.-X.: Constructive and axiomatic approaches of fuzzy approximation operators. *Information Sciences* **159** (2004) 233–254
15. Yao, Y.Y.: Constructive and algebraic methods of the theory of rough sets. *Information Sciences* **109** (1998) 21–47
16. Yao, Y.Y.: Relational interpretations of neighborhood operators and rough set approximation operators. *Information Sciences* **111** (1998) 239–259
17. Yao, Y.Y.: Rough sets, neighborhood systems, and granular computing. Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering, Edmonton, Canada, Meng, (ed.), IEEE press, (1999) 1553–1558
18. Zakowski, W.: Approximations in the space $(U, \Pi)$. Demenstratio Mathematica **XVI** (1983) 761–769

# Information-Theoretic Measure of Uncertainty in Generalized Fuzzy Rough Sets

Ju-Sheng Mi[1], Xiu-Min Li[2], Hui-Yin Zhao[1,3], and Tao Feng[2]

[1] College of Mathematics and Information Science,
Hebei Normal University, Shijiazhuang, Hebei, 050016, P.R. China
mijsh@263.net, hebeiahuier@163.com
[2] College of Science, Hebei University of Science and Technology,
Shijiazhuang, Hebei, 050018, P.R. China
li_xiumin@163.com, Fengtao_new@163.com
[3] College of Science, Hebei College of Industry and Technology,
Shijiazhuang, Hebei, 050091, P.R. China

**Abstract.** Rough set theory has become well-established as a mechanism for uncertainty management in a wide variety of applications. This paper studies the measurement of uncertainty in generalized fuzzy rough sets determined by a triangular norm. Based on information theory, the entropy of a generalized fuzzy approximation space is introduced, which is similar to Shannon's entropy. To measure uncertainty in generalized fuzzy rough sets, a notion of fuzziness is introduced. Some basic properties of this measure are examined. For a special triangular norm $T = \min$, it is proved that the measure of fuzziness of a generalized fuzzy rough set is equal to zero if and only if the set is crisp and definable.

**Keywords:** Approximation operators, fuzzy sets, fuzzy rough sets, triangular norm, uncertainty.

## 1 Introduction

Rough set theory [1], proposed by Pawlak in 1982, is a generalization of the classical set theory for describing and modeling of vagueness in ill defined environment. The research has recently roused great interest in the theoretical and application fronts to deal with inexact, uncertain or vague knowledge. Many authors have extended Pawlak's concept of rough sets in various aspects [2,3,4], especially in the fuzzy environment [5,6,7,8,9,10,11,12]. As a natural need, Dubois and Prade [5] combined fuzzy sets and rough sets in a fruitful way by defining rough fuzzy sets and fuzzy rough sets. Later, various extensions of generalized fuzzy rough sets have been made by introducing some logic operators [6,7,8,9,10,11,12].

Information theory, originally developed by Shannon [13] for communication theory, has been a useful mechanism for characterizing the information content in various models and applications in many diverse fields. Attempts have been made to use Shannon's entropy to measure uncertainty in rough set theory [14,15,16,17]. Recently, Chakrabarty et al. [18] introduced a measure of fuzziness in rough sets. Their measure based on a special index of fuzziness of a

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 63–70, 2007.

fuzzy set. Wierman [16], Liang et al. [19] introduced measures of uncertainty for Pawlak's rough set theory. Mi et al. [20] proposed an uncertainty measure in partition-based fuzzy rough sets.

The purpose of this paper is to study the information-theoretic measure of uncertainty in generalized fuzzy rough sets. In the next section, we give some basic notions and properties related to generalized fuzzy rough sets defined by a triangular norm. Based on information theory, the entropy of a generalized fuzzy approximation space is introduced in Section 3, which is similar to Shannon's entropy. In Section 4, a notion of fuzziness of a generalized fuzzy rough sets is defined as the entropy of rough belongingness. Some basic properties of this uncertainty measure are examined. We then conclude the paper with a summary in Section 5.

## 2 Generalized Fuzzy Rough Sets

In this section, we recall some basic notions and properties of generalized fuzzy rough sets determined by a triangular norm.

A triangular norm, or shortly $t$-norm, is an increasing, associative and commutative mapping $T : I^2 \to I$ (where $I = [0, 1]$ is the unit interval) that satisfies the boundary condition: for all $a \in I, T(a, 1) = a$. The most popular continuous $t$-norms are:

- the standard min operator $T_{\mathrm{M}}(a, b) = \min\{a, b\}$, $\forall a, b \in I$;
- the algebraic product $T_{\mathrm{P}}(a, b) = a * b$, $\forall a, b \in I$;
- the Łukasiewicz $t$-norm $T_{\mathrm{L}}(a, b) = \max\{0, a + b - 1\}$, $\forall a, b \in I$.

A binary operation $S$ on the unit interval $I$ is said to be the dual of a triangular norm $T$, if $\forall a, b \in I$, $S(a, b) = 1 - T(1 - a, 1 - b)$. $S$ is also called a triangular conorm of $T$ (or shortly $t$-conorm) in the literature.

Let $U$ and $W$ be two nonempty sets. The Cartesian product of $U$ with $W$ is denoted by $U \times W$. The class of all crisp (fuzzy, respectively) subsets of $U$ is denoted by $\mathcal{P}(U)$ ($\mathcal{F}(U)$, respectively).

**Definition 1.** *A fuzzy subset $R \in \mathcal{F}(U \times W)$ is referred to as a fuzzy binary relation from $U$ to $W$, $R(x, y)$ is the degree of relation between $x$ and $y$, where $(x, y) \in U \times W$. In particular, if $U = W$, we call $R$ a fuzzy relation on $U$. $R$ is referred to as a reflexive fuzzy relation if $R(x, x) = 1$ for all $x \in U$; $R$ is referred to as a symmetric fuzzy relation if $R(x, y) = R(y, x)$ for all $x, y \in U$; $R$ is referred to as a $T$-transitive fuzzy relation if $R(x, z) \geq \vee_{y \in U} T(R(x, y), R(y, z))$ for all $x, z \in U$. $R$ is referred to as a $T$-similarity fuzzy relation if it is a reflexive, symmetric, and $T$-transitive fuzzy relation.*

In the sequel, $T$ will be a lower semi-continuous triangular norm, therefore its dual $S$ is upper semi-continuous.

**Definition 2.** *Let $U$ and $W$ be two finite nonempty sets called the universes, and $R$ be a fuzzy relation from $U$ to $W$. The triple $(U, W, R)$ is called a generalized*

*fuzzy approximation space. If $U = W$, then it can be written as $(U, R)$. We define two fuzzy set-theoretic operators from $\mathcal{F}(W)$ to $\mathcal{F}(U)$: $\forall A \in \mathcal{F}(W)$,*

$$\overline{R}(A)(u) = \bigvee_{y \in W} T(R(u, y), A(y)), \qquad u \in U,$$
$$\underline{R}(A)(u) = \bigwedge_{y \in W} S(1 - R(u, y), A(y)), \; u \in U.$$

*$\underline{R}$ and $\overline{R}$ are called generalized fuzzy lower and upper approximation operators. The pair $(\underline{R}(A), \overline{R}(A))$ is called the generalized fuzzy rough set of $A$. If $\underline{R}(A) = A = \overline{R}(A)$, we say that $A$ is definable, otherwise it is undefinable.*

From the definition, the following theorem can be easily proved [9].

**Proposition 1.** *Let $R$ be an arbitrary fuzzy relation from $U$ to $W$. Then the generalized fuzzy lower and upper approximation operators, $\overline{R}$ and $\underline{R}$, satisfy: $\forall A, B \in \mathcal{F}(W)$, $\forall \alpha \in I$, $x \in U, y \in W$,*

| | | | |
|---|---|---|---|
| *(FTL1)* | $\underline{R}(A) = \sim \overline{R}(\sim A)$, | *(FTU1)* | $\overline{R}(A) = \sim \underline{R}(\sim A)$; |
| *(FTL2)* | $\underline{R}(W) = U$, | *(FTU2)* | $\overline{R}(\emptyset) = \emptyset$; |
| *(FTL3)* | $A \subseteq B \Longrightarrow \underline{R}(A) \subseteq \underline{R}(B)$, | *(FTU3)* | $A \subseteq B \Longrightarrow \overline{R}(A) \subseteq \overline{R}(B)$; |
| *(FTL4)* | $S(\underline{R}(A), \widehat{\alpha}) = \underline{R}(S(A, \widehat{\alpha}))$, | *(FTU4)* | $T(\overline{R}(A), \widehat{\alpha}) = \overline{R}(T(A, \widehat{\alpha}))$; |
| *(FTL5)* | $\underline{R}(1_{W \setminus \{y\}})(x) = 1 - R(x, y)$, | *(FTU5)* | $\overline{R}(1_y)(x) = R(x, y)$. |

*Furthermore, if $R$ is reflexive on $U$, then*

$$\textit{(FTL6)}\ \underline{R}(A) \subseteq A, \forall A \in \mathcal{F}(U), \quad \textit{(FTU6)}\ \overline{R}(A) \supseteq A, \forall A \in \mathcal{F}(U).$$

*If $R$ is symmetric on $U$, then $\forall x, y \in U$,*

$$\textit{(FTL7)}\ \ \underline{R}(1_{U \setminus \{y\}})(x) = \underline{R}(1_{U \setminus \{x\}})(y), \quad \textit{(FTU7)}\ \ \overline{R}(1_y)(x) = \overline{R}(1_x)(y).$$

*If $R$ is $T$ transitive on $U$, then $\forall A \in \mathcal{F}(U)$,*

$$\textit{(FTL8)}\ \underline{R}(A) \subseteq \underline{R}(\underline{R}(A)), \quad \textit{(FTU8)}\ \overline{R}(\overline{R}(A)) \subseteq \overline{R}(A).$$

*Where $\sim A$ is the complement of $A$, $\widehat{\alpha}$ is the constant fuzzy set with its membership function $\widehat{\alpha}(x) = \alpha, \forall x \in U$. $S(A, B)(y) = S(A(y), B(y))$, $1_y$ is the characteristic function of $\{y\}$.*

## 3 Entropy of a Generalized Fuzzy Approximation Space

In Pawlak's rough set theory, uncertainty may be arisen from the indiscernibility (or equivalence) relation which is imposed on the universe, partitioning all values into a finite set of equivalence classes. If every equivalence class contains only one value, then there is no loss of information caused by the partitioning. In any coarser partitioning, however, there are fewer classes, and each class will contain

a larger number of members. Our knowledge, or information, about a particular value decreases as the granularity of the partitioning becomes coarser. Based on this idea, Wierman [16] defined the entropy of Pawlak's approximation space $(U, R)$ as follows.

Let $R$ be a crisp equivalence relation on $U$, $\{X_1, \ldots, X_k\}$ be the equivalence classes partitioned by, $R$. The entropy of $R$ is then defined by

$$E(R) = -\sum_{i=1}^{k} \frac{|X_i|}{|U|} \log_2 \frac{|X_i|}{|U|}$$

Inspired by this definition, we introduce a kind of entropy for a generalized fuzzy approximation space.

Let $U = \{x_1, x_2, \ldots, x_n\}$ be the universal set, $R$ be an arbitrary fuzzy relation on $U$. Denoted by $xR$ the fuzzy set with its membership function $xR(y) = R(x, y)$. For a fuzzy set $A \in \mathcal{F}(U)$, the cardinality of $A$ is defined by $|A| = \sum_{x \in U} A(x)$. Thus we have $|xR| = \sum_{y \in U} xR(y) = \sum_{y \in U} R(x, y)$.

**Definition 3.** *The entropy of a generalized fuzzy approximation space $(U, R)$ is defined by*

$$F(R) = -\sum_{i=1}^{n} \frac{1}{|U|} \log_2 \frac{|x_i R|}{|U|},$$

*where we assume that $\log_2 0 = 0$. $F(R)$ is the information gained by performing the experiment $R$.*

**Proposition 2.** *Let $R$ be a reflexive fuzzy relation on $U$, then*

*(1) The maximum value of $F(R)$ is $\log_2 |U|$. Furthermore, $F(R) = \log_2 |U| \Longleftrightarrow x_i R = 1_{x_i}$, $\forall i \leq n$;*

*(2) The minimum value of $F(R)$ is 0. Furthermore, $F(R) = 0 \Longleftrightarrow x_i R = U$, $\forall i \leq n$.*

*Proof.* It follows immediately from Definition 3.

**Proposition 3.** *If $R$ is a crisp equivalence relation on $U$, then $F(R) = E(R)$.*

*Proof.* Let $R$ be a crisp equal relation on $U$, $U/R = \{X_1, \ldots, X_k\}$ is the equivalence classes partitioned by $R$. Then for each $x \in U$, there exists an $X_x \in U/R$ such that $xR = 1_{X_x}$. Thus we have, $|xR| = |X_x|$. Therefore,

$$\begin{aligned} F(R) &= -\sum_{x \in U} \tfrac{1}{|U|} \log_2 \tfrac{|xR|}{|U|} = -\sum_{x \in U} \tfrac{1}{|U|} \log_2 \tfrac{|xR|}{|U|} \\ &= -\sum_{j=1}^{k} \tfrac{1}{|U|} |X_j| \log_2 \tfrac{|X_j|}{|U|} = E(R). \end{aligned}$$

Proposition 3 implies that the entropy defined in Definition 3 is a generalization of the same concept defined in [16].

**Proposition 4.** *Let $P$, $Q$ be two arbitrary fuzzy relations on $U$. If $P$ is finer than $Q$, that is $P \subseteq Q$, then $F(P) \geq F(Q)$.*

*Proof.* Because the function $f(x) = \log_2(x)$ is monotonous increasing, and

$$|x_i P| = \sum_{j=1}^{n} P(x_i, x_j) \leq \sum_{j=1}^{n} Q(x_i, x_j) = |x_i Q|,$$

we have

$$F(P) = \log_2 |U| - \frac{1}{|U|} \sum_{i=1}^{n} \log_2 |x_i P| \geq \log_2 |U| - \frac{1}{|U|} \sum_{i=1}^{n} \log_2 |x_i Q| = F(Q).$$

## 4 Uncertainty in Generalized Fuzzy Rough Sets

Let $R$ be a reflexive fuzzy relation on the universal set $U$, $T$ be a lower semi-continuous triangular norm. $\forall A \in \mathcal{F}(U)$, the lower and upper approximations of $A$ are $\underline{R}(A)$ and $\overline{R}(A)$, respectively. For an element $x \in U$, the degree of rough belongingness of $x$ in $A$ is defined by

$$b(A)(x) = \frac{\sum\limits_{u \in U} T(R(x,u), A(u))}{\sum\limits_{u \in U} R(u,x)}.$$

Clearly, $\forall x \in U$, $0 \leq b(A)(x) \leq 1$. This immediately induced a fuzzy set $b(A)$ of $U$. If $R$ is a crisp equivalence relation on $U$, $A$ is a crisp subset of $U$, and $T = \min$, then $b(R)(x)$ reduces the same concept made by [1].

Rough set theory inherently models two types of uncertainty. The first type of uncertainty arises from the approximation space. If every equivalence class contains only one object, then there is no loss of information caused by the partitioning. In any coarser partitioning, however, our knowledge about a particular object decreases as the granularity of the partitioning becomes coarser. Uncertainty is also modeled through the approximation regions of rough sets where elements of the lower approximation region have total participation in the rough set and those of the upper approximation region have uncertain participation in the rough set. Equivalently, the lower approximation is the certain region and the boundary area of the upper approximation region is the possible region.

Using the function of rough belongingness, we can define the information-theoretic measure of uncertainty in generalized fuzzy rough sets as following.

**Definition 4.** *The measure of fuzziness in a generalized fuzzy rough set ($\underline{R}A$, $\overline{R}(A)$) is denoted by $FR(A)$ and is defined by the entropy of the fuzzy set $b(A)$. That is*

$$FR(A) = -\frac{2}{|U|} \sum_{x \in U} b(A)(x) \log_2 b(A)(x)$$

It is evidence that even if $R$ is a classical relation and $A$ is a crisp subset of $U$, the rough set $(\underline{R}(A), \overline{R}(A))$ may still have some fuzziness.

**Proposition 5.** *Let $(U, R)$ be a generalized fuzzy approximation space, then for every definable crisp set $A$, $FR(A) = 0$.*

*Proof.* We first prove that: if $x \in A, y \notin A$, then $R(x, y) = R(y, x) = 0$.

In fact, as $A$ is definable, we have by the definition of definable set that $\underline{R}A = A = \overline{R}A$. Then $1 = A(x) = \underline{R}A(x) = \min_{u \in U} S(1 - R(x, u), A(u)) \leq S(1 - R(x, y), A(y)) = 1 - R(x, y)$. Which implies that $R(x, y) = 0$.

Recalling that $0 = A(y) = \overline{R}A(y) = \max_{u \in U} T(R(y, u), A(u)) \geq T(R(y, x), A(x)) = R(y, x)$, we get $R(y, x) = 0$.

If $u \in A$, then

$$b(A)(u) = \frac{\sum\limits_{x \in U} T(R(u, x), A(x))}{\sum\limits_{x \in U} R(u, x)} = \frac{\sum\limits_{x \in A} T(R(u, x), 1)}{\sum\limits_{x \in A} R(u, x)} = 1.$$

Similarly, If $u \notin A$, then $b(A)(u) = \frac{\sum\limits_{x \in U} T(R(u,x), A(x))}{\sum\limits_{x \in U} R(u,x)} = 0$.

We conclude that $FR(A) = -\frac{2}{|U|} \sum\limits_{x \in U} b(A)(x) \log_2 b(A)(x) = 0$.

**Proposition 6.** *If $R$ is reflexive, then $FR(U) = FR(\phi) = 0$.*

**Proof:** By Proposition 1 we know that $U$ and $\phi$ are all definable. Thus $FR(U) = FR(\phi) = 0$ by Proposition 5.

**Proposition 7.** *Let $R$ be a reflexive relation on $U$, $T = \min$. If $FR(A) = 0$, $A \in \mathcal{F}(U)$, then $A$ is a definable crisp set.*

*Proof.* Suppose $FR(A) = 0$, by the definition of $FR(A)$, we have for each $x \in U$ either $b(A)(x) = 0$ or $b(A)(x) = 1$.

(1) $\forall x \in U$ with $b(A)(x) = 0$, from the definition of $b(A)$ we obtain for all $u \in U$, $R(x, u) = 0$ or $A(u) = 0$. But $R$ is reflexive we have $R(x, x) = 1$, therefore, $A(x) = 0$.

(2) $\forall x \in U$ with $b(A)(x) = 1$, from the definition of $b(A)$, the inequality $R(x, u) \leq A(u)$ must hold for all $u \in U$. Especially, we have $1 = R(x, x) \leq A(x)$, then $A(x) = 1$.

Combining (1) and (2) we conclude that $A$ is a crisp set.

We are now proving that $A$ is definable, that is, $\underline{R}(A) = A = \overline{R}(A)$.

Because $A$ is crisp and $T = \min$, by Definition 2 we have for all $x \in U$,

$$\overline{R}(A)(x) = \bigvee_{y \in A} R(x, y), \quad \underline{R}(A)(x) = \bigwedge_{y \notin A} (1 - R(x, y).$$

(3) If $x \notin A$, then $\underline{R}(A)(x) \leq 1 - R(x, x) = 0 = A(x)$. Since $A(x) = 0$, we have

$$b(A)(x) = \frac{\sum\limits_{u \neq x} \min(R(x, u), A(u))}{\sum\limits_{u \neq x} R(u, x) + 1} < 1.$$

Noticing that $FR(A) = 0$, it must happen that $b(A)(x) = 0$. Then by (1) for all $u \in U$, either $R(x, u) = 0$ or $A(u) = 0$. Hence, $R(x, y) = 0$, $\forall y \in A$. Thus we obtain $\overline{R}(A)(x) = 0 = A(x)$.

(4) If $x \in A$, then $A(x) = 1$. From the definition of $b(A)$ we have $b(A)(x) \neq 0$. Noticing that $FR(A) = 0$ we have $b(A)(x) = 1$. By (2) we obtain $R(x, y) \leq A(y)$, $\forall y \in U$. Which implies that $R(x, y) = 0$, $\forall y \notin A$. Hence $\underline{R}(A)(x) = 1 = A(x)$.

It is easy to see that $\overline{R}(A)(x) = \bigvee_{y \in A} R(x, y) \geq R(x, x) = 1$. Therefore, $\overline{R}(A)(x) = 1 = A(x)$.

Combining (3) and (4) we conclude that $\underline{R}(A) = A = \overline{R}(A)$, which implies that $A$ is definable.

**Proposition 8.** *Let $R$ be a reflexive relation on $U$, $T = \min$. Then for each $A \in \mathcal{F}(U)$, $FR(A) = 0$ if and only if $A$ is a definable crisp set.*

*Proof.* It follows immediately from Propositions 5 and 7.

## 5 Conclusion

Rough set theory and fuzzy set theory are two important mathematical tools to deal with inexact, vague, uncertain information. There are closed relationships between the two notions. Every fuzzy set can be approximated by two approximation sets. Every rough set can introduce a fuzzy set automatically. Thus rough sets have some fuzziness too. In the present paper, we studied the information-theoretic measure of uncertainty in generalized fuzzy rough sets defined by a triangular norm. Based on information theory, we introduced a concept of entropy of a generalized fuzzy approximation space, some properties have been examined which are similar to Shannon's entropy. A measure of fuzziness of a generalized fuzzy rough set was also defined by the entropy of the fuzzy set of rough belongingness. This measure can be used to understand the essence of rough set data analysis.

## Acknowledgements

This paper is supported by the Natural Science Foundation of Hebei Province (A2006000129) and Science Foundation of Hebei Normal University (L2005Z01).

## References

1. Pawlak, Z.: *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Boston (1991)
2. Lin, T. Y.: Neighborhood systems - application to qualitative fuzzy and rough sets. In: Wang, P. P., Ed., *Advances in Machine Intelligence and Soft-Computing*. Department of Electrical Engineering, Duke University, Durham, NC, USA (1997) 132-155

3. Slowinski, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. *IEEE Transactions on Knowledge and Data Engineering.* **12**(2) (2000) 331-336
4. Yao, Y. Y.: Constructive and algebraic methods of the theory of rough sets. *Information Sciences.* **109** (1998) 21-47
5. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. *International Journal of General Systems.* **17** (1990) 191-208
6. Moris, N. N., Yakout, M. M.: Axiomatics for fuzzy rough sets. *Fuzzy Sets and Systems.* **100** (1998) 327-342
7. Wu, W. Z., Mi, J. S., Zhang, W. X.: Generalized fuzzy rough sets. *Information Sciences.* **151** (2003) 263-282
8. Wu, W. Z., Zhang, W. X.: Constructive and axiomatic approaches of fuzzy approximation operators. *Information Sciences.* **159** (2004) 233-254
9. Mi, J. S., Wu, W. Z., Zhang, W. X.: Constructive and axiomatic approaches of the theory of rough sets. *Pattern Recognition and Artificial Intelligence.* **15**(3) (2002) 280-284
10. Mi, J. S., Zhang, W. X., An axiomatic characterization of a fuzzy generalization of rough sets. *Information Sciences.* **160**(1-4) (2004) 235-249
11. Radzikowska, A. N., Kerre, E. E.: A comparative study of fuzzy rough sets. *Fuzzy Sets and Systems.* **126** (2002) 137-155
12. Wu, W. Z., Leung, Y., Mi, J. S.: On characterizations of $(\mathcal{I}, \mathcal{T})$-fuzzy rough approximation operators. *Fuzzy Sets and Systems.* **154**(1) (2005) 76-102
13. Shannon, C. E.: The mathematical theory of communication. *The Bell System Technical Journal.* **27**(3-4)(1948) 373-423
14. Beaubouef, T., Petry, F. E., Arora, G.: Information-theoretic measures of uncertainty for rough sets and rough relational databases. *Information Sciences.* **109** (1998) 185-195
15. Yao, Y. Y.: Probabilistic approaches to rough sets. *Expert Systems.* **20**(5)(2003) 287-297
16. Wierman, M. J.: Measuring uncertainty in rough set theory. *International Journal of General Systems.* **28**(4) (1999) 283-297
17. Duntsch, I., Gediga, G.: Roughian: Rough information analysis. *International Journal of Intelligent Systems.* **16** (2001) 121-147
18. Chakrabarty, K., Biswas, R., Nanda, S.: Fuzziness in rough sets. *Fuzzy Sets and Systems.* **110** (2000) 247-251
19. Liang, J. Y., Chin, K. S., Dang, C. Y., Yam, R. C. M.: A new method for measuring uncertainty and fuzziness in rough set theory. *International Journal of General Systems.* **31**(4) (2002) 331-342
20. Mi, J. S., Leung, Y., Wu, W. Z.: An uncertainty measure in partition-based fuzzy rough sets. *International Journal of General Systems.* **34** (2005) 77-90

# Determining Significance of Attributes in the Unified Rough Set Approach

Alicja Mieszkowicz-Rolka and Leszek Rolka

Department of Avionics and Control,
Rzeszów University of Technology,
ul. W. Pola 2, 35-959 Rzeszów, Poland
{alicjamr,leszekr}@prz.edu.pl

**Abstract.** This paper discusses the problems arising in applications of the unified rough and fuzzy rough set approach to analysis of inconsistent information systems. The unified approach constitutes a parameterized generalization of the variable precision rough set model. It bases on a single notion of parameterized $\varepsilon$-approximation. As a necessary extension, a method suitable for a correct determination of attributes' significance is proposed. In particular, the notions of positive $\varepsilon$-classification region and $\varepsilon$-approximation quality are considered. A criterion for reduction of condition attributes is given. Furthermore, a generalized definition of the fuzzy extension $\omega$ is proposed.

## 1 Introduction

Noise and errors, which are inevitably present in data obtained from real decision processes, can significantly bias results and conclusions obtained in applications of the original rough set concept. In order to avoid this kind of problem, the idea of admitting of a misclassification level was introduced by Ziarko [14], in the form of a variable precision rough set (VPRS) model.

Fuzzy-rough hybridization is another important interdisciplinary area which combines the crisp rough set theory with the theory of fuzzy sets. The idea of fuzzy rough sets, given by Dubois and Prade [2], has attracted interest of many researchers (see, e.g., [3,4,5,10,13]).

In our previous work, we recognized the need for using both the VPRS model together with the concept of fuzzy rough approximation [7,8]. Developing such a combined approach was motivated by the kind of decision system that we have investigated. The considered decision process was performed by a human operator, during control of a complex dynamic plant. Firstly, modelling control actions of a human operator, in the form of a fuzzy inference system (FIS), is a well established procedure elaborated in the framework of the fuzzy set theory. The obtained fuzzy information system can be analyzed with the help of fuzzy rough approximations. Secondly, due to a large amount of data obtained from a dynamic system, it was also necessary to admit of some misclassification level.

In [8], we proposed unified parameterized crisp rough set and fuzzy rough set models, which are based on a single notion of $\varepsilon$-approximation. In this way, we

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 71–78, 2007.

were able to avoid problems of constructing a consistent variable precision fuzzy rough set (VPFRS) approach, caused by the use of different fuzzy connectives in the lower and upper fuzzy rough approximations, respectively.

In the present paper, we propose a further development of our parameterized rough set concept, aiming at its correct application to data analysis. The main problem, we focus on, is the determination of attributes' significance. A notion of the positive $\varepsilon$-classification region is given and properties of $\varepsilon$-approximation quality are considered. It is shown that a change of the $\varepsilon$-approximation quality measure cannot be used as a reliable indicator for detecting superfluous attributes. Therefore, we introduce a new criterion for reduction of condition attributes. In the case of the unified fuzzy rough set model, we propose a general form of the fuzzy extension $\omega$, in order to decide how the fuzzy $\varepsilon$-approximation is mapped from the domain of the quotient set into the domain of the universe.

We start by recalling the basics of the unified parameterized approach to approximation of crisp and fuzzy sets.

## 2 Unified Crisp Rough Set Model

The fundamental idea of the rough set theory consists in describing crisp subsets of an universe $U$ by means of a lower and upper approximation [9].

The lower approximation $\underline{R}(A)$ and upper approximation $\overline{R}(A)$ of a crisp set $A$ by an indiscernibility relation $R$ are defined as follows

$$\underline{R}(A) = \{x \in U : [x]_R \subseteq A\}\,, \tag{1}$$

$$\overline{R}(A) = \{x \in U : [x]_R \cap A \neq \emptyset\}\,, \tag{2}$$

where $[x]_R$ denotes an indiscernibility (equivalence) class which contains an element $x \in U$.

A modified relation of set inclusion was introduced by Ziarko [14], with the aim of improving results of approximation, in the case of large information systems. It can be explained using the notion of inclusion degree $\mathrm{incl}(A, B)$, of a nonempty (crisp) set $A$ in a (crisp) set $B$, defined as follows

$$\mathrm{incl}(A, B) = \frac{\mathrm{card}(A \cap B)}{\mathrm{card}(A)}\,. \tag{3}$$

By applying a lower limit $l$ and an upper limit $u$, introduced in [6], satisfying the condition $0 \leq l < u \leq 1$, we can define the $u$-lower and the $l$-upper approximation of any subset $A \in U$ by an indiscernibility relation $R$.

The $u$-lower approximation $\underline{R}_u(A)$ and $l$-upper approximation $\overline{R}_l(A)$ of $A$ by $R$ are defined as follows

$$\underline{R}_u(A) = \{x \in U\colon\ \mathrm{incl}([x]_R, A) \geq u\}\,, \tag{4}$$

$$\overline{R}_l(A) = \{x \in U\colon\ \mathrm{incl}([x]_R, A) > l\}\,, \tag{5}$$

where $[x]_R$ denotes an indiscernibility class of $R$ containing an element $x$.

In order to obtain a new form of a parameterized rough set model, we adapt the notion of rough inclusion function $\nu$, given in [11], which is defined on the Cartesian product of the powersets $\mathbb{P}(U)$ of the universe $U$

$$\nu : \mathbb{P}(U) \times \mathbb{P}(U) \to [0,1]\,. \tag{6}$$

We assume that the first parameter represents a nonempty set, and the rough inclusion function should be monotonic with respect to the second parameter

$$\nu(X,Y) \leq \nu(X,Z) \quad \text{for any } Y \subseteq Z, \quad \text{where } X,Y,Z \subseteq U\,.$$

The inclusion degree (3), proposed by Ziarko in the framework of the VPRS model, constitutes a rough inclusion function.

We introduce a unified crisp rough set approach by proposing a parameterized single form of approximation of crisp sets. Given an indiscernibility relation $R$, the $\varepsilon$-approximation $R_\varepsilon(A)$ of a crisp set $A$ is defined as follows

$$R_\varepsilon(A) = \{x \in U : \nu([x]_R, A) \geq \varepsilon\}\,, \tag{7}$$

where $\varepsilon \in (0,1]$.

The $\varepsilon$-approximation $R_\varepsilon$ has the following properties:

$$\begin{array}{lllll}
\text{(P1)} & R_\varepsilon(A) & = & \underline{R}(A) & \text{for } \varepsilon = 1\,, \\
\text{(P2)} & R_\varepsilon(A) & = & \overline{R}(A) & \text{for } \varepsilon = 0+\,, \\
\text{(P3)} & R_\varepsilon(A) & = & \underline{R}_u(A) & \text{for } \varepsilon = u\,, \\
\text{(P4)} & R_\varepsilon(A) & = & \overline{R}_l(A) & \text{for } \varepsilon = l+\,,
\end{array}$$

where $0+$ and $l+$ denote numbers infinitesimally exceeding 0 and $l$, respectively.

Using a single form of approximation is especially important for defining a fuzzy generalization of parameterized rough set model.

## 3 Unified Fuzzy Rough Set Model

The well-known and widely used concept of fuzzy rough sets was introduced by Dubois and Prade [2]. For a given fuzzy set $A \subseteq U$ and a fuzzy partition $\Phi = \{F_1, F_2, \ldots, F_n\}$ on the universe $U$, the membership functions of the lower and upper approximations of $A$ by $\Phi$ are defined as follows

$$\mu_{\underline{\Phi}(A)}(F_i) = \inf_{x \in U} \mathrm{I}(\mu_{F_i}(x), \mu_A(x))\,, \tag{8}$$

$$\mu_{\overline{\Phi}(A)}(F_i) = \sup_{x \in U} \mathrm{T}(\mu_{F_i}(x), \mu_A(x))\,, \tag{9}$$

where T and I denote a T-norm operator and an implicator, respectively.

In order to get a parameterized fuzzy rough set model, we must consider an important problem of determining the degree of inclusion of one fuzzy set into another. Many different measures of fuzzy set inclusion were proposed, (see, e.g.,

[1,3]). The novelty of our approach consists in describing inclusion of sets, by a fuzzy set rather than a number. Therefore, we introduce [8] a notion of a fuzzy inclusion set, denoted by $\mathrm{INCL}(A,B)$, which expresses the inclusion of a fuzzy set $A$ in a fuzzy set $B$. The set $\mathrm{INCL}(A,B)$ is determined with respect to particular elements (or singletons) of the set $A$.

The notions of power, support and $\alpha$-cut, defined for a finite fuzzy set $A \in U$ with $n$ elements, will be applied in our considerations: $\mathrm{power}(A) = \sum_{i=1}^{n} \mu_A(x_i)$; $\mathrm{supp}(A) = \{x : \mu_A(x_i) > 0\}$; $A_\alpha = \{x \in U : \ \mu_A(x) \geq \alpha\}$, for $\alpha \in [0,1]$.

First, we propose a fuzzy counterpart of the rough inclusion function (6), which is defined on the Cartesian product of the families $\mathbb{F}(U)$ of all fuzzy subsets of the universe $U$

$$\nu_\alpha : \mathbb{F}(U) \times \mathbb{F}(U) \to [0,1] \,. \tag{10}$$

The fuzzy rough $\alpha$-inclusion function $\nu_\alpha(A,B)$ of any nonempty fuzzy set $A$ in a fuzzy set $B$ is defined as follows

$$\nu_\alpha(A,B) = \frac{\mathrm{power}(A \cap \mathrm{INCL}(A,B)_\alpha)}{\mathrm{power}(A)} \,. \tag{11}$$

The value $\nu_\alpha(A,B)$ is needed to expresses how many elements of the nonempty fuzzy set $A$ belong, at least to the degree $\alpha$, to the fuzzy set $B$.

Furthermore, we introduce a function called `res`, defined on the Cartesian product $\mathbb{P}(U) \times \mathbb{F}(U)$, where $\mathbb{P}(U)$ denotes the powerset of the universe $U$, and $\mathbb{F}(U)$ the family of all fuzzy subsets of the universe $U$, respectively

$$\mathrm{res} : \mathbb{P}(U) \times \mathbb{F}(U) \to [0,1] \,. \tag{12}$$

We require that

$$\begin{aligned}
&\mathrm{res}(\emptyset, Y) = 0 \,, \\
&\mathrm{res}(X,Y) \in \{0,1\}, \quad \text{if } Y \text{ is a crisp set} \,, \\
&\mathrm{res}(X,Y) \leq \mathrm{res}(X,Z) \quad \text{for any } Y \subseteq Z, \quad \text{where } X \in \mathbb{P}(U), \text{ and } Y, Z \in \mathbb{F}(U) \,.
\end{aligned}$$

The value of $\mathrm{res}(X,Y)$ represents the resulting membership degree in a given fuzzy set $Y$, determined by taking into account only the elements of a given crisp set $X$. Various definitions of `res` are possible. If we want to be in accordance with the limit-based approach of Dubois and Prade, we can assume the following form of `res`

$$\mathrm{res}(X,Y) = \inf_{x \in X} \mu_Y(x) \,. \tag{13}$$

For $\varepsilon \in (0,1]$, the $\varepsilon$-approximation $\Phi_\varepsilon(A)$ of a fuzzy set $A$, by a fuzzy partition $\Phi = \{F_1, F_2, \ldots, F_n\}$, is a fuzzy set on the domain $\Phi$ with membership function expressed by

$$\mu_{\Phi_\varepsilon(A)}(F_i) = \mathrm{res}(S_\varepsilon(F_i,A), \mathrm{INCL}(F_i,A)) \,, \tag{14}$$

where

$$\begin{aligned}
S_\varepsilon(F_i,A) &= \mathrm{supp}(F_i \cap \mathrm{INCL}(F_i,A)_{\alpha_\varepsilon}) \,, \\
\alpha_\varepsilon &= \sup\{\alpha \in [0,1] : \ \nu_\alpha(F_i,A) \geq \varepsilon\} \,.
\end{aligned}$$

The set $S_\varepsilon(F_i, A)$ contains those elements of the approximating class $F_i$ that are included in $A$, at least to the degree $\alpha_\varepsilon$. The resulting membership $\mu_{\Phi_\varepsilon(A)}(F_i)$ is determined using only elements from $S_\varepsilon(F_i, A)$ instead of the whole class $F_i$.

It can be shown that applying the definition (13) of the function `res` leads to a simple form of (14): $\mu_{\Phi_\varepsilon(A)}(F_i) = \sup\{\alpha \in [0,1] : \ \nu_\alpha(F_i, A) \geq \varepsilon\}$.

By a unified definition of fuzzy rough $\varepsilon$-approximation, we avoid the use of two different fuzzy connectives, in contrast to the approximations (8) and (9).

## 4 Analysis of Decision Tables

Let us begin our consideration with the simpler case of a crisp decision table. We have a finite universe $U$ with $N$ elements: $U = \{x_1, x_2, \ldots, x_N\}$. Each element $x$ of the universe $U$ is described by a combination of values of $n$ condition attributes $C = \{c_1, c_2, \ldots, c_n\}$ and $m$ decision attributes $D = \{d_1, d_2, \ldots, d_m\}$.

We can determine a family of classes $\widetilde{C} = \{C_1, C_2, \ldots, C_{\widetilde{n}}\}$ containing indiscernible elements with respect to condition attributes $C$, and a family of indiscernibility classes $\widetilde{D} = \{D_1, D_2, \ldots, D_{\widetilde{m}}\}$, generated with respect to decision attributes $D$.

For determining the consistence of a crisp decision table, a notion of positive $\varepsilon$-classification region of $\widetilde{D}$ by $\widetilde{C}$ can be used [7]. We take into account only those elements of the approximating classes which are in accordance with the approximated classes

$$\mathrm{Pos}_{\widetilde{C}_\varepsilon}(\widetilde{D}) = \bigcup_{i=1}^{\widetilde{m}} \widetilde{C}_\varepsilon(D_i) \cap D_i \ . \tag{15}$$

Thus, we get a measure of $\varepsilon$-approximation quality, denoted by $\gamma_{\widetilde{C}_\varepsilon}(\widetilde{D})$

$$\gamma_{\widetilde{C}_\varepsilon}(\widetilde{D}) = \frac{\mathrm{card}(\mathrm{Pos}_{\widetilde{C}_\varepsilon}(\widetilde{D}))}{\mathrm{card}(U)} = \frac{\sum_{i=1}^{\widetilde{n}} \sum_{j=1}^{\widetilde{m}} \delta_{ij} \mathrm{card}(C_i \cap D_j)}{\mathrm{card}(U)} \tag{16}$$

where

$$\delta_{ij} = \begin{cases} 1 & \text{if } \nu(C_i, D_j) \geq \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

Now, we consider a simple example, which will demonstrate the problem with applying the above measure of $\varepsilon$-approximation quality for determination of attributes' significance.

*Example 1.* Given a family $\widetilde{D} = \{D_1, D_2\}$ of indiscernibility classes obtained with respect to the decision attribute $d$:

$D_1 = \{\ x_1,\ \ x_3,\ \ x_4,\ \ x_5,\ \ x_8,\ \ x_9,\ \ x_{11},\ \ x_{13},\ \ x_{15},\ \ x_{17},\ \ x_{19}\ \}$,
$D_2 = \{\ x_2,\ \ x_6,\ \ x_7,\ \ x_{10},\ \ x_{12},\ \ x_{14},\ \ x_{16},\ \ x_{18},\ \ x_{20}\ \}$,

and a family $\widetilde{C} = \{C_1, C_2, C_3\}$ of indiscernibility classes obtained with respect to all condition attributes:

$C_1 = \{\ x_1,\ \ x_3,\ \ x_5,\ \ x_7,\ \ x_9,\ \ x_{11},\ \ x_{13},\ \ x_{15},\ \ x_{17},\ \ x_{19}\ \},$
$C_2 = \{\ x_2,\ \ x_4,\ \ x_6,\ \ x_8\ \},$
$C_3 = \{\ x_{10},\ \ x_{12},\ \ x_{14},\ \ x_{16},\ \ x_{18},\ \ x_{20}\ \}.$

The inclusion degrees are obtained as follows:

$\nu(C_1, D_1) = 0.9,\ \ \nu(C_2, D_1) = 0.5,\ \ \nu(C_3, D_1) = 0,\ \ \nu(C_1, D_2) = 0,$
$\nu(C_2, D_2) = 0.5,\ \ \nu(C_3, D_2) = 1.0.$

For $\varepsilon = 1$, we obtain $\mathrm{Pos}_{\widetilde{C}_\varepsilon}(\widetilde{D}) = C_3$, and $\gamma_{\widetilde{C}_\varepsilon}(\widetilde{D}) = \frac{\mathrm{card}(C_3)}{\mathrm{card}(U)} = 0.3$. Admitting of a misclassification level, by assuming $\varepsilon = 0.8$, leads to $\mathrm{Pos}_{\widetilde{C}_\varepsilon}(\widetilde{D})) = (C_1 \cap D_1) \cup C_3$ and $\gamma_{\widetilde{C}_\varepsilon}(\widetilde{D}) = 0.75$.

In the original rough set theory ($\varepsilon = 1$), the measure of approximation quality can be used for determining significance of particular condition attributes in the decision process. Removing a condition attribute from information system can cause merging of some indiscernibility classes. In that case, a decrease of approximation quality means that the removed attribute is indispensable. We can discard an attribute, only if the approximation quality retains its previous value. However, we show now that the $\varepsilon$-approximation quality (16) can not be used as an entirely reliable indicator for making conclusion about significance of attributes in the variable precision rough set framework (for $\varepsilon < 1$).

Assume the required inclusion degree $\varepsilon = 0.8$. First, let us remove a single condition attribute, such that indiscernibility classes $C_2$ and $C_3$ merge into one class denoted by $C_{2\&3}$. We obtain the inclusion degree $\nu(C_{2\&3}, D_2) = 0.8$. Since the condition for inclusion of the class $C_{2\&3}$ in the $\varepsilon$-approximation of $D_2$ is satisfied, $C_{2\&3}$ will be included in the positive region of $\varepsilon$-approximation. Finally, we get $\mathrm{Pos}_{\widetilde{C}_\varepsilon}(\widetilde{D}) = (C_1 \cap D_1) \cup (C_{2\&3} \cap D_2)$. The $\varepsilon$-approximation quality $\gamma_{\widetilde{C}_\varepsilon}(\widetilde{D}) = 0.85$. After removing an attribute, we paradoxically obtain an increased value of the $\varepsilon$-approximation quality, which is rather expected to be decreasing.

The results obtained in the above example can be explained by the properties of the $\varepsilon$-approximation. For $\varepsilon = 1$, taking a subset of condition attributes $C' \subset C$, and a subset $A \subseteq U$, we have

$$\widetilde{C'}_\varepsilon(A) \subseteq \widetilde{C}_\varepsilon(A)\,. \tag{17}$$

This relation does not hold in general, for $\varepsilon < 1$. Even in the case, when the value of $\varepsilon$-approximation quality does not change, after removing some condition attribute, we cannot be sure, whether the considered attribute may be discarded. It is possible that we encounter a local increase of $\varepsilon$-approximation of one approximated class and a local decrease of $\varepsilon$-approximation of one another, in such a way that the total change of cardinality of the positive $\varepsilon$-classification region remains unchanged.

Thus, it is necessary to inspect a change in the positive $\varepsilon$-classification region, instead of observing the change $\Delta\gamma_{\widetilde{C'}_\varepsilon}(\widetilde{D}) = \gamma_{\widetilde{C}_\varepsilon}(\widetilde{D}) - \gamma_{\widetilde{C'}_\varepsilon}(\widetilde{D})$, before deciding whether to remove an attribute.

**Definition 1.** *For a subset of condition attributes $C' \subset C$, the change in the positive $\varepsilon$-classification region, denoted by $\Delta\mathrm{Pos}_{\widetilde{C'}_\varepsilon}(\widetilde{D})$, is defined as follows*

$$\Delta\mathrm{Pos}_{\widetilde{C'}_\varepsilon}(\widetilde{D}) = (\mathrm{Pos}_{\widetilde{C}_\varepsilon}(\widetilde{D}) \cup \mathrm{Pos}_{\widetilde{C'}_\varepsilon}(\widetilde{D})) - (\mathrm{Pos}_{\widetilde{C}_\varepsilon}(\widetilde{D}) \cap \mathrm{Pos}_{\widetilde{C'}_\varepsilon}(\widetilde{D}))\,. \tag{18}$$

**Criterion 1.** *A condition attribute $c \in C$ is dispensable: $C' = C - \{c\}$, iff $card(\Delta\mathrm{Pos}_{\widetilde{C'}_\varepsilon}(\widetilde{D})) = 0$.*

To consider the problem of determining the consistency of fuzzy decision tables and significance of fuzzy attributes, we should apply a generalized measure of $\varepsilon$-approximation quality [7]. For the families of fuzzy similarity classes $\widetilde{D}$ and $\widetilde{C}$, the $\varepsilon$-approximation quality of $\widetilde{D}$ by $\widetilde{C}$ is defined as follows

$$\gamma_{\widetilde{C}_\varepsilon}(\widetilde{D}) = \frac{\mathrm{power}(\mathrm{Pos}_{\widetilde{C}_\varepsilon}(\widetilde{D}))}{\mathrm{card}(U)}\,, \tag{19}$$

where

$$\mathrm{Pos}_{\widetilde{C}_\varepsilon}(\widetilde{D}) = \bigcup_{D_j \in \widetilde{D}} \omega(\widetilde{C}_\varepsilon(D_j)) \cap D_j\,. \tag{20}$$

Since the fuzzy rough $\varepsilon$-approximation is expressed in the domain of $\widetilde{C}$, we use in (20) a fuzzy extension $\omega$ for mapping the $\varepsilon$-approximation into the domain of the universe $U$. Let us introduce a general form of the fuzzy extension $\omega$.

**Definition 2.** *The fuzzy extension $\omega(X)$, for any fuzzy set $X$ on the domain $\widetilde{C}$, denotes a mapping from $\widetilde{C}$ into the domain of the universe $U$*

$$\mu_{\omega(X)}(x) = \mathrm{ref}(\mu_X(C_1), \mu_X(C_2), \ldots, \mu_X(C_{\widetilde{n}}), \mu_{C_1}(x), \mu_{C_2}(x), \ldots, \mu_{C_{\widetilde{n}}}(x))\,, \tag{21}$$

*where* ref *denotes a function* $: [0,1]^{2\widetilde{n}} \to [0,1]$ .

Various definitions of the fuzzy extension $\omega$ are possible. In the fuzzy rough set approach of Dubois and Prade [2], the following form of the fuzzy extension $\omega$ is used

$$\mu_{\omega(X)}(x) = \mu_X(C_i), \quad \text{if } \mu_{C_i}(x) = 1\,. \tag{22}$$

In the definition of fuzzy $\varepsilon$-classification region, we take into account only those elements of the $\varepsilon$-approximation, for which there is no contradiction between the approximated and the approximating similarity classes.

In the case of determining the significance of condition attributes with fuzzy values, we should inspect a change in the fuzzy positive $\varepsilon$-classification region (20), according to Definition 1. A fuzzy counterpart of Criterion 1, with power instead of card should be used.

## 5 Conclusions

Reduction of attributes in an information system is an important issue in applications of the rough set theory. The problem of a correct determination of

attributes' significance in the parameterized (fuzzy) rough set approach requires a detailed inspection of changes in the positive region of classification, after removing particular condition attributes. Using only a value of the $\varepsilon$-approximation quality can be misleading because that measure preserves the monotonicity property only in the special case for $\varepsilon = 1$ (original rough sets). Therefore, we propose an additional criterion for reduction of condition attributes. In future research, the properties of the unified fuzzy rough set model should be investigated, with respect to various form of the function `res` of resulting membership and the fuzzy extension $\omega$.

## References

1. Cornelis, C., Van der Donck, C., Kerre, E.: Sinha-Dougherty Approach to the Fuzzification of Set Inclusion Revisited. Fuzzy Sets and Systems **134** (2003) 283–295
2. Dubois, D., Prade, H.: Putting Rough Sets and Fuzzy Sets Together. In: Słowiński, R., (ed.): Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory. Kluwer Academic Publishers, Boston Dordrecht London (1992) 203–232
3. Fernández Salido, J.M., Murakami, S.: Rough Set Analysis of a General Type of Fuzzy Data Using Transitive Aggregations of Fuzzy Similarity Relations. Fuzzy Sets and Systems **139** (2003) 635–660
4. Greco, S., Matarazzo, B., Słowiński, R.: Rough Set Processing of Vague Information Using Fuzzy Similarity Relations. In: Calude, C.S., Paun, G., (eds.): Finite Versus Infinite — Contributions to an Eternal Dilemma. Springer-Verlag, Berlin Heidelberg New York (2000) 149–173
5. Inuiguchi, M.: Generalizations of Rough Sets: From Crisp to Fuzzy Cases. [12] 26–37
6. Katzberg, J.D., Ziarko, W.: Variable Precision Extension of Rough Sets. Fundamenta Informaticae **27** (1996) 155–168
7. Mieszkowicz-Rolka, A., Rolka, L.: Remarks on Approximation Quality in Variable Precision Fuzzy Rough Sets Model. [12] 402–411
8. Mieszkowicz-Rolka, A., Rolka, L.: An Approach to Parameterized Approximation of Crisp and Fuzzy Sets. In: Greco, S., et al., (eds.): Rough Sets and Current Trends in Computing. Lecture Notes in Artificial Intelligence, Vol. 4259. Springer-Verlag, Berlin Heidelberg New York (2006) 127–136
9. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Boston Dordrecht London (1991)
10. Radzikowska, A.M., Kerre, E.E.: A Comparative Study of Fuzzy Rough Sets. Fuzzy Sets and Systems **126** (2002) 137–155
11. Skowron, A., Stepaniuk, J.: Tolerance Approximation Spaces. Fundamenta Informaticae **27** (1996) 245–253
12. Tsumoto, S., et al., (eds.): Rough Sets and Current Trends in Computing. Lecture Notes in Artificial Intelligence, Vol. 3066. Springer-Verlag, Berlin Heidelberg New York (2004)
13. Liu, W.N., Yao, J., Yao, Y.: Rough Approximations under Level Fuzzy Sets. [12] 78–83
14. Ziarko, W.: Variable Precision Rough Sets Model. Journal of Computer and System Sciences **46** (1993) 39–59

# A Rough-Hybrid Approach to Software Defect Classification*

Sheela Ramanna[1], Rajen Bhatt[2], and Piotr Biernot[1]

[1] Department of Applied Computer Science, University of Winnipeg,
Winnipeg, Manitoba R3B 2E9 Canada
s.ramanna@uwinnipeg.ca, pbiernot@iam.uwinnipeg.ca
[2] Samsung India Software Center,
Noida-201305, Uttar Pradesh, India
rajen.bhatt@gmail.com

**Abstract.** Knowledge discovery methods used to find relationships among software engineering data and the extraction of rules have gained increasing importance in recent years. These methods have become necessary for improvements in the quality of the software product and the process. The focus of this paper is a first attempt towards combining strengths of rough set theory and neuro-fuzzy decision trees in classifying software defect data. We compare classification results for four methods: rough sets, neuro-fuzzy decision trees, partial decision trees, rough-neuro-fuzzy decision trees. The analysis of the results include a family-wise 10 fold paired t-test for accuracy and number of rules. The contribution of this paper is the application of a hybrid rough-neuro-fuzzy decision tree method in classifying software defect data.

**Keywords:** Classification, Neuro-Fuzzy-Decision Trees, Rough Sets, Software Defects.

## 1 Introduction

This paper presents approaches to classification of software defect data using data mining methods from rough set theory [9], fuzzy decision trees [15] and neuro-fuzzy decision trees [1]. In the context of software defect classification, the term data mining refers to knowledge-discovery methods used to find relationships among defect data and the extraction of rules useful in making decisions about defective modules either during development or during post-deployment of a software system. A software defect is a product anomaly (e.g, omission of a required feature or imperfection in the software product) [11]. As a result, defects have a direct bearing on the quality of the software product and the allocation of project resources to program modules. Software metrics make it possible for software engineers to measure and predict quality of both the product and the process [11]. In this work, the defect data consists of product metrics.

* The research of Sheela Ramanna and Piotr Biernot is supported by NSERC Canada grant 194376.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 79–86, 2007.

There have been several studies in applying computational intelligence techniques such as rough sets [10], fuzzy clustering [5,16], neural networks [7] to software quality data. Statistical predictive models correlate quality metrics to number of changes to the software. The predicted value is a numeric value that gives the number of changes (or defects) to each module. However, in practice, it is more useful to have information about modules that are highly defective rather than knowing the exact number of defects for each module. This has led to the application of machine learning methods to software quality data. It should also be mentioned that the software quality data used in several of the above mentioned studies were derived from non object-oriented programs. The focus of this paper is a first attempt towards combining strengths of rough set theory and neuro-fuzzy decision trees. Neuro-fuzzy decision trees (N-FDT) include a fuzzy decision tree (FDT) structure with parameter adaptation strategy based on neural networks [1]. The contribution of this paper is the presentation of software defect classification results using several methods based on rough set theory and a hybrid method that includes rough-neuro-fuzzy decision tree.

This paper is organized as follows. In Sect. 2, we give a brief overview of neuro-fuzzy decision tree algorithm. The details of the defect data and classification methods are presented in Sect. 3. This is followed by an analysis of the classification results in Sect. 4.

## 2 Neuro-fuzzy Decision Trees

Fuzzy decision trees are powerful, top-down, hierarchical search methodology to extract easily interpretable classification rules [2]. However, they are often criticized for poor learning accuracy [13]. In [1] an N-FDT algorithm was proposed to improve the learning accuracy of fuzzy decision trees. In the forward cycle, N-FDT constructs a fuzzy decision tree using the standard FDT induction algorithm fuzzy ID3 [15]. In the feedback cycle, parameters of fuzzy decision trees (FDT) have been adapted using stochastic gradient descent algorithm by traversing back from each leaf to root nodes. During the parameter adaptation stage, N-FDT retains the hierarchical structure of fuzzy decision trees. A detailed discussion of N-FDT algorithm with computational experiments using real-world datasets and analysis of results are available in [1].

### 2.1 Brief Overview of N-FDT

Fig. 1 shows an exemplary N-FDT with two summing nodes to carry out the inference process. There are five paths starting from root node to five leaf nodes. Leaf nodes are shown by dots and indexed as $m = 1, 2, , 5$. Certainty factor corresponding to $m^{th}$ leaf node and $l^{th}$ class is indicated by $\beta_{ml}$. From all the leaf nodes, certainty corresponding to class-l are summed up to calculate $y_l$. For an arbitrary pattern (or say $i^{th}$ pattern), the firing strength of $path_m$ with respect to $l^{th}$ class as defined by (1)

$$\mu^{i}_{path_m} \times \beta_{ml}, \tag{1}$$

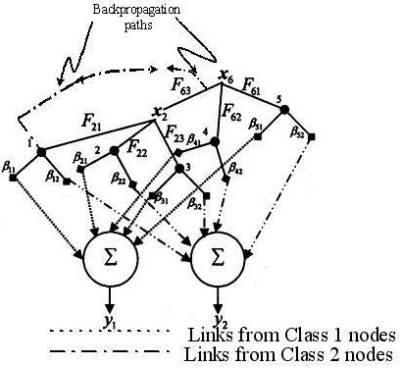



**Fig. 1.** Exemplary Neuro-Fuzzy Decision Tree

where $\mu^i_{path_m}$ is the membership degree of $i^{th}$ pattern to $path_m$ and can be calculated as shown in (2)

$$\mu^i_{path_m} = \prod_j \mu_{F^m_j}\left(x^i_j\right). \tag{2}$$

Here $x_j$ is the $j^{th}$ input attribute and $\mu_{F^m_j}\left(x^i_j\right)$is the degree of membership of $x^i_j$ into $F^m_j$ and $F^m_j$ is fuzzy membership function for $x_j$ on $path_m$. Firing strengths of all the paths for a particular class-$l$ are summed up to calculate the prediction certainty $y^i_l$ $(l = 1, ..., q)$ of $i^{th}$ pattern through fuzzy decision tree as shown in (3)

$$y^i_l = \sum_{m=1}^{M} \mu^i_{path_m} \times \beta_{ml}, \tag{3}$$

where $0 \leq y^i_l \leq 1$ and $q$ is total number of classes. When classification to a unique class is desired, the class with the highest membership degree needs to be selected, i.e., classify given pattern to class $l_0$, where

$$l_0 = \arg \max_{l=1,...,q} \left\{y^i_l\right\}. \tag{4}$$

To fuzzify input attributes, we have selected Gaussian membership functions out of many alternatives due to its differentiability property. For $i^{th}$ pattern, membership degree of $path_m$ can be calculated as shown in (5)

$$\mu^i_{path_m} = \prod_j \mu_{F^m_j}\left(x^i_j\right) = \prod_j \exp\left(\frac{\left(x^i_j - c_{jm}\right)^2}{2\sigma^2_{jm}}\right), \tag{5}$$

where $c_{jm}$ and $\sigma_{jm}$are center and standard deviation (width) of Gaussian membership function of $j^{th}$ input attribute $x_j$ on $path_m$, i.e., of $F^m_j$. We now briefly outline the strategy of N-FDT by performing an adaptation of all types of parameters (centers, widths, and certainty factors) simultaneously on the structure

shown in Fig. 1. We define as the error function of the fuzzy decision tree, the mean-square-error defined by (6)

$$E = \frac{1}{2n} \sum_{l=1}^{q} \sum_{i=1}^{n} \left( d_l^i - y_l^i \right)^2, \tag{6}$$

where $n$ is the total number of training patterns and $d_l^i$ and $y_l^i$ are the desired prediction certainty and the actual prediction certainty of $l^{th}$ class for $i^{th}$ pattern, respectively. At each epoch (iteration), the complete parameter $\mathrm{P} = \{c_{jm}, \sigma_{jm}, \beta_{ml} \mid m = 1, ..., M; l = 1, ..., q\}$ is moved by a small distance $\eta$ in the direction in which E decreases most rapidly, i.e., in the direction of the negative gradient $-\frac{\partial E}{\partial \theta}$ where $\theta$ is the parameter vector constituted from the set P. This leads to the parameter update rule shown in (7)

$$\theta^{\tau+1} = \theta^{\tau} - \eta \frac{\partial E}{\partial \theta}, \tag{7}$$

where $\tau$ is the iteration index and $\eta$ is the learning rate. The update equations for centers, widths, and certainty factors can be found in [1]. Parameter adaptation continues until error goes below certain small positive error goal $\epsilon$ or the specified number of training epochs has been completed.

## 3 Defect Data Classification

The PROMISE[1] Software Engineering Repository data set was used for our experiments. The data set includes a set of static software metrics about the *product* as a predictor of defects in the software. There are a total of 94 attributes and one decision attribute (indicator of defect level). The defect level attribute value is TRUE if the class contains one or more defects and FALSE otherwise. The metrics at the *method level* are primarily drawn from Halstead's Software Science metrics [6] and McCabe's Complexity metrics [8]. The metrics at the *class level*, include such standard measurements as Weighted Methods per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Response For a Class (RFC), Coupling Between Object Classes (CBO), and Lack of Cohesion of Methods (LCOM) [4]. The data includes measurements for 145 modules (objects).

### 3.1 Neuro-fuzzy Decision Tree Method

All the attributes have been fuzzified using fuzzy c-means algorithm [3] into three fuzzy clusters. From the clustered row data, Gaussian membership functions have been approximated by introducing the width control parameter $\lambda$. The center of each gaussian membership function has been initialized by fuzzy cluster centers generated by the fuzzy c-means algorithm. To initialize standard deviations,

[1] http://promise.site.uottawa.ca/SERepository

we have used a value proportional to the minimum distance between centers of fuzzy clusters. For each numerical attribute $x_j$ and for each gaussian membership function, the Euclidean distance between the center of $F_{jk}$ and the center of any other membership function $F_{jh}$ is given by $dc\,(c_{jk}, c_{jh})$, where $h \neq k$. For each $k^{th}$ membership function, after calculating $dc_{\min}\,(c_{jk}, c_{jh})$, the standard deviation $\sigma_{jk}$ has been obtained by (8)

$$\sigma_{jk} = \lambda \times dc_{\min}\,(c_{jk}, c_{jh})\,; 0 < \lambda \leq 1, \tag{8}$$

where $\lambda$ is the width control parameter. For the computational experiments reported here, we have selected various values of $\lambda \in (0, 1]$ to introduce variations in the standard deviations of initial fuzzy partitions. After attribute fuzzification, we run the fuzzy ID3 algorithm with cut $\alpha = 0$ and leaf selection threshold $\beta_{th} = 0.75$. These fuzzy decision trees have been tuned using the N-FDT algorithm for 500 epochs with the target MSE value 0.001.

### 3.2 Rough-Hybrid Methods

Experiments reported were performed with RSES[2] using rule-based and tree-based methods. The RSES tool is based on rough set methods. Only the rule-based method which uses genetic algorithms in rule derivation [14] is reported in this paper. Experiments with non-rough set based methods were performed with WEKA[3] using a partial decision tree-based method (DecisionTree) which is a variant of the well-known C4.5 revision 8 algorithm [12]. The experiments were conducted using 10-fold cross-validation technique. The accuracy results with ROSE (another rough-set based tool) using a basic minimal covering algorithm was 79%. However, since ROSE[4] uses an internal 10-fold cross-validation technique, we have not included the experimental results in our pair-wise t-statistic test. The attributes were discretized in the case of rough set methods. The Rough-Neuro-FDT method included i) generating reducts from rough set methods ii) using the data from the reduced set of attributes to run the NFDT algorithm.

## 4 Analysis of Classification Results

Tables 1 and 2 give a summary of computational experiments using four different classification methods. Percentage classification accuracy has been calculated by $\frac{n_c}{n} \times 100\%$, where $n$ is the total number of test patterns, and $n_c$ is the number of test patterns classified correctly. A family wise t-test was performed for the six pairs to compare whether significant differences exist between the four classification algorithms in terms of accuracy and the number of rules. The t-statistic has a student's t-distribution with $n - 1$ degrees of freedom[5].

[2] http://logic.mimuw.edu.pl/∼rses

[3] http://www.cs.waikato.ac.nz/ml/weka

[4] http://idss.cs.put.poznan.pl/site/rose.html

[5] R.V. Hogg and E.A. Tanis, E.A: Probability and Statistical Inference. Macmillan Publishing Co., Inc., New York, 1977.

**Table 1.** Defect Data Classification I

| 10CV Accuracy% Results | | | | |
|---|---|---|---|---|
| *Run* | Neuro-FDT | Rough-Neuro-FDT | Rough Methods | DecisionTree |
| 1 | 85.71 | 71.42 | 92.9 | 71.43 |
| 2 | 85.71 | 92.85 | 78.6 | 85.71 |
| 3 | 64.28 | 67.58 | 57.1 | 64.29 |
| 4 | 71.42 | 71.42 | 57.1 | 71.43 |
| 5 | 64.28 | 57.14 | 50 | 71.43 |
| 6 | 78.57 | 78.57 | 78.6 | 64.29 |
| 7 | 85.71 | 71.42 | 71.4 | 85.71 |
| 8 | 71.42 | 78.57 | 71.4 | 57.14 |
| 9 | 92.85 | 100 | 78.6 | 78.57 |
| 10 | 89.47 | 89.47 | 84.2 | 84.21 |
| *Avg.Acc* | 78.94 | 77.84 | 71.99 | 73.42 |

**Table 2.** Defect Data Classification II

| 10CV Results - Number of Rules | | | | |
|---|---|---|---|---|
| *Run* | Neuro-FDT | Rough-Neuro-FDT | Rough Methods | DecisionTree |
| 1 | 3 | 2 | 231 | 9 |
| 2 | 14 | 18 | 399 | 12 |
| 3 | 4 | 3 | 330 | 8 |
| 4 | 2 | 5 | 282 | 8 |
| 5 | 6 | 1 | 308 | 8 |
| 6 | 6 | 4 | 249 | 7 |
| 7 | 6 | 7 | 124 | 12 |
| 8 | 2 | 6 | 292 | 10 |
| 9 | 10 | 7 | 351 | 14 |
| 10 | 4 | 4 | 235 | 9 |
| $Avg.\#ofrules$ | 5.7 | 5.7 | 280 | 9.7 |

Probability distribution (Pr) values for $t_{n-1,\alpha/2*6}$ were obtained from a standard t-distribution table. In what follows, $\alpha = 0.05$ and $n - 1 = 9$ relative to 10 different training-testing runs. With 9 degrees of freedom, and significance level of 0.995, we find that $Pr = 3.25$. The paired t-test results for all combinations (6 pairs) in Table 3 shows that there is no significant difference between any of the methods in terms of accuracy.

In terms of the t-test for number of rules shown in Table 3, the results show that there is *no significant* difference between i) Neuro-FDT and Rough-Neuro-FDT and ii) Rough-Neuro-FDT and DecisionTree. The reason being that the average number of rules used by these classifiers are similar and few. The other interesting observation is that there is a slight difference in the performance of the Neuro-FDT and DecisionTree algorithms. However, the genetic algorithm-based classifier in RSES induces a large set of rules. As a result, there is a significant

**Table 3.** T-test Results

| | Accuracy | | | Number of Rules | | |
|---|---|---|---|---|---|---|
| *Pairs* | Avg | Std. Deviation | t-stat | Avg | Std. Deviation | t-stat |
| $R-NFDT/NFDT$ | −1.10 | 8.24 | −0.42 | 0.00 | 3.02 | 0.00 |
| $R-NFDT/Rough$ | 5.85 | 11.67 | 1.59 | −274.00 | 74.36 | −11.67 |
| $R-NFDT/DT$ | 4.42 | 12.58 | 1.11 | −4.00 | 3.83 | −3.30 |
| $NFDT/Rough$ | 6.95 | 7.57 | 2.91 | −274.40 | 74.42 | −11.66 |
| $NFDT/DT$ | 5.52 | 8.09 | 2.16 | −4.00 | 2.94 | −4.30 |
| $Rough/DT$ | −1.43 | 14.22 | −0.32 | 270 | 75.96 | 11.26 |

difference when classifiers are compared with the rough classifier on the basis of number of rules. The other important observation is the role that reducts play in defect data classification. On an average, only 10 attributes (out of 95) were used by the rough set method with no significant reduction in classification accuracy. In fact, the Rough-Neuro-FDT (hybrid) method results in a minimal number of rules with comparable accuracy. The average number of attributes (over 10 runs) is about 4. The metrics that are most significant on the class-level include: DIT, RFC, CBO and LCOM. At the method level, the metrics that are most significant include: i) Halstead's metric of *content* where the complexity of a given algorithm independent of the language used to express the algorithm ii) Halstead's metric of *level* which is level at which the program can be understood iii) Halstead's metric of *number of unique operands* which includes variables and identifiers, constants (numeric literal or string) function names when used during calls iv) total lines of code v) $branch-count$ is the number of branches for each module. Branches are defined as those edges that exit from a decision node.

## 5 Conclusion

This paper presents approaches to classification of software defect data using rough set algorithms, neuro-fuzzy decision trees and partial decision tree methods. We present classification results in terms of accuracy and number of rules applied to a software quality data set available in the public domain. The analysis includes a family-wise 10 fold paired t-test for the four different methods. The t-test shows that there is no significant difference between any of the methods in terms of accuracy. However, in terms of rules, there is a marked difference. The hybrid approach that combines rough set method with neuro-fuzzy decision trees has the most potential. This is particularly promising in the case of process metrics such as efficacy of review procedures, change control procedures and risk management. In contrast to product metrics which are easily obtained with automated metric tools, process metrics are harder to collect and project teams are reluctant to spend time gathering such data.

## References


1. Bhatt, R. B., Gopal, M.: Neuro-fuzzy decision trees, *International Journal of Neural Systems*, 16 (1)(2006) 63-78.
2. Bhatt, R. B.: Fuzzy-Rough Approach to Pattern Classification- Hybrid Algorithms and Optimization, Ph.D. Dissertation, Electrical Engineering Department, Indian Institute of Technology Delhi, India (2006)
3. Bezdek, J.C: Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Plenum (1981)
4. Chidamber, S. R., Kemerer, F.C.: A metrics suite for object-oriented design. IEEE Trans. Soft. Eng., v. 20 no. 6, (June 1994) 476-493.
5. Dick, S., Meeks, A., Last, M, Bunke, H, Kandel, A: Data mining in software metrics databases, *Fuzzy Sets and Systems* **145** (2004) 81110.
6. Halstead, M.H.: Elements of Software Science. Elsevier, New York, (1977)
7. Khoshgoftaar, T.M., Allen, E.B.: Neural networks for software quality prediction. In: Pedrycz, W., Peters, J.F.(Eds.), Computational Intelligence in Software Engineering, World Scientific, River Edge, NJ (1998) 3363.
8. McCabe, T.: A complexity measure. IEEE Trans. on Software Engineering SE-2(4), (1976) 308-320.
9. Pawlak, Z.: Rough sets. *International J. Comp. Information Science*, **11**(3)(1982) 341–356.
10. Peters, J.F., Ramanna, S: Towards a software change classification system: A rough set approach. *Software Quality Journal*, **11** (2003) 121-147.
11. Peters, J.F. Pedrycz, W.: Software Engineering: An Engineering Approach. John Wiley and Sons, New York (2000)
12. Quinlan, J.R: Induction of decision trees. *Machine Learning* 1(1), 1986, 81-106.
13. Tsang, E.C.C., Yeung, D.S., Lee, J.W.T., Huang,D.M., Wang, X.Z.: Refinement of generated fuzzy production rules by using a fuzzy neural network. IEEE Trans. on SMC-B, 34 (1)(2004) 409-418.
14. Wróblewski, J.: Genetic algorithms in decomposition and classification problem. Polkowski, L. and Skowron, A.(Eds.), *Rough Sets in Knowledge Discovery*, **1**. Physica-Verlag, Berlin, Germany (1998) 471-487.
15. Wang, X.Z., Yeung, D.S., Tsang, E.C.C.: A comparative study on heuristic algorithms for generating fuzzy decision trees. IEEE Trans. on SMC-B, 31 (2001) 215-226.
16. Yuan, X., Khoshgoftaar, T.M., Allen, E.B. Ganesan, K.: An application of fuzzy clustering to software quality prediction. Proc. 3rd IEEE Symp. on Application-Specific Software Engineering Technology (2000) 8590.

# Vaguely Quantified Rough Sets

Chris Cornelis[1], Martine De Cock[1], and Anna Maria Radzikowska[2]

[1] Computational Web Intelligence, Department of Mathematics and Computer Science, Ghent University, Krijgslaan 281 (S9), B-9000 Gent, Belgium
{Chris.Cornelis,Martine.DeCock}@UGent.be
[2] Faculty of Mathematics and Information Science, Warsaw University of Technology, Plac Politechniki 1, 00-661 Warsaw, Poland
annrad@mini.pw.edu.pl

**Abstract.** The hybridization of rough sets and fuzzy sets has focused on creating an end product that extends both contributing computing paradigms in a conservative way. As a result, the hybrid theory inherits their respective strengths, but also exhibits some weaknesses. In particular, although they allow for gradual membership, fuzzy rough sets are still abrupt in a sense that adding or omitting a single element may drastically alter the outcome of the approximations. In this paper, we revisit the hybridization process by introducing vague quantifiers like "some" or "most" into the definition of upper and lower approximation. The resulting vaguely quantified rough set (VQRS) model is closely related to Ziarko's variable precision rough set (VPRS) model.

**Keywords:** vague quantifiers, fuzzy sets, rough sets, VPRS model.

## 1 Introduction

In rough set theory, an object belongs to the upper approximation of a set as soon as it is related to *one* of the elements in the set, while the lower approximation only retains those objects related to *all* the elements in the set. This is due to the use of an existential quantifier in the definition of upper approximation, and of a universal quantifier for the lower approximation. In applications that use real-life data (which is usually noisy to some extent, and hence prone to classification errors and inconsistency), the definition of upper approximation might be too loose (easily resulting in very large sets), while the definition of lower approximation might be too strict (easily resulting in the empty set). A similar phenomenon can be observed at the level of fuzzy rough set theory, where the $\exists$ and $\forall$ quantifiers are replaced by the sup and inf operations (see e.g. [1,6]), which prove just as susceptible to noise as their crisp counterparts.

In his variable precision rough set (VPRS) model, Ziarko [8,9] introduced thresholds to deal with these problems in the crisp case. In general, given $0 \leq l < u \leq 1$, an element $y$ is added to the lower approximation of a set $A$ if at least $100 * u$ percent of the elements related to $y$ are in $A$. Likewise, $y$ belongs to the upper approximation of $A$ if more than $100 * l$ percent of the elements related to $y$

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 87–94, 2007.

are in $A$. This can be interpreted as a generalization of the rough set model using crisp quantifiers *at least* $100 * u$ *percent* and *more than* $100 * l$ *percent* to replace the universal quantifier (which corresponds to "at least 100 percent") and the existential quantifier (which corresponds to "more than 0 percent"). Also, some attempts have been made to pursue this approach within the fuzzy rough set model, having in common that they still rely on the use of crisp thresholds $l$ and $u$ (see e.g. [2,4]).

In this paper, we go one step further by introducing vague quantifiers like *most* and *some* into the model. In this way, an element $y$ belongs to the lower approximation of $A$ if most of the elements related to $y$ are included in $A$. Likewise, an element belongs to the upper approximation of $A$ if some of the elements related to $y$ are included in $A$. Mathematically, we model vague quantifiers in terms of Zadeh's notion of fuzzy quantifiers [7]. As such, the new model inherits both the flexibility of VPRSs for dealing with classification errors (by relaxing the membership *conditions* for the lower approximation, and tightening those for the upper approximation) and that of fuzzy sets for expressing partial constraint satisfaction (by distinguishing different *levels* of membership to the upper/lower approximation). Moreover, we illustrate that the model can be used in a meaningful way, regardless of whether the relation $R$ and the set $A$ to be approximated are crisp or fuzzy. In each case, the outcome of the approximations will be a pair of fuzzy sets delineating $A$ in a flexible way.

The remainder of this paper is structured as follows. In Section 2, we review basic notions of classical rough sets and VPRSs, while Section 3 introduces vaguely quantified rough sets in the crisp case, and illustrates their relevance in the context of information retrieval. In Section 4, we lift the VQRS paradigm to the level of fuzzy rough set theory, distinguish it from related work that combines VPRSs with fuzzy sets and detail an experiment on a benchmark dataset to show the performance of the proposed extension vis-à-vis the classical approach in a rough data analysis problem. Finally, in Section 5, we conclude.

## 2 Variable Precision Rough Sets

Recall that the traditional upper and lower approximation [5] of a set $A$ in the approximation space $(X, R)$ are defined by

$$y \in R{\uparrow}A \text{ iff } A \cap Ry \neq \emptyset \tag{1}$$

$$y \in R{\downarrow}A \text{ iff } Ry \subseteq A \tag{2}$$

in which $Ry$ is used to denote the equivalence class (also called $R$-foreset) of $y$. Furthermore, the rough membership function $R_A$ of $A$ is defined by

$$R_A(y) = \frac{|Ry \cap A|}{|Ry|} \tag{3}$$

$R_A(y)$ quantifies the degree of inclusion of $Ry$ into $A$, and can be interpreted as the conditional probability that $y$ belongs to $A$, given knowledge about the equivalence class $Ry$ that $y$ belongs to. One can easily verify that

$$y \in R{\uparrow}A \text{ iff } R_A(y) > 0 \tag{4}$$
$$y \in R{\downarrow}A \text{ iff } R_A(y) = 1 \tag{5}$$

In other words, $y$ is added to the upper approximation as soon as $Ry$ overlaps with $A$, while even a small inclusion error of $Ry$ in $A$ results in the rejection of the whole class from the lower approximation.

*Example 1.* Consider a document collection $D = \{d_1, \ldots, d_{20}\}$ in which the documents are arranged according to topic into four categories : $D_1 = \{d_1, \ldots, d_5\}$, $D_2 = \{d_6, \ldots, d_{10}\}$, $D_3 = \{d_{11}, \ldots, d_{15}\}$ and $D_4 = \{d_{15}, \ldots, d_{20}\}$. Hence, the categorization defines an equivalence relation $R$ on $X$. Suppose now that a user launches a query, and that the relevant documents turn out to be (automatically determined) the set $A = \{d_2, \ldots, d_{12}\}$. This suggests that the information retrieval system simply might have missed $d_1$ since all other documents from $D_1$ are in $A$. Furthermore, the fact that only $d_{11}$ and $d_{12}$ are retrieved from $D_3$ might indicate that these documents are less relevant to the query than the documents of $D_2$, which all belong to $A$. Pawlak's original rough set approach does not allow to reflect these nuances, since $R{\downarrow}A = D_2$ and $R{\uparrow}A = D_1 \cup D_2 \cup D_3$, treating $D_1$ and $D_3$ in the same way.

Since in real life, data may be affected by classification errors caused by humans or noise, Ziarko [9] relaxes the constraints in (4) and (5) to obtain the following parameterized definitions:

$$y \in R{\uparrow_l}A \text{ iff } R_A(y) > l \tag{6}$$
$$y \in R{\downarrow_u}A \text{ iff } R_A(y) \geq u \tag{7}$$

Formulas (4)–(7) can also be read in terms of quantifiers, i.e.

$$y \in R{\uparrow}A \text{ iff } (\exists x \in X)((x, y) \in R \wedge x \in A) \tag{8}$$
$$y \in R{\downarrow}A \text{ iff } (\forall x \in X)((x, y) \in R \Rightarrow x \in A) \tag{9}$$
$$y \in R{\uparrow_l}A \text{ iff more than } 100 * l\% \text{ elements of } Ry \text{ are in } A \tag{10}$$
$$y \in R{\downarrow_u}A \text{ iff at least } 100 * u\% \text{ elements of } Ry \text{ are in } A \tag{11}$$

Note that the quantifiers used above are all crisp: the existential quantifier $\exists$, the universal quantifier $\forall$, as well as two threshold quantifiers $> 100 * l\%$ and $\geq 100 * u\%$. As such, although the VPRS model warrants a measure of tolerance towards problematic elements, it still treats them in a black-or-white fashion: depending on the specific choice of $l$ and $u$, an element either fully belongs, or does not belong to the upper or lower approximation.

*Example 2.* Let us return to the document retrieval problem from Example 1. Ziarko's model offers more flexibility to distinguish the roles of $D_1$ and $D_3$, but the choice of the thresholds is crucial. In a symmetric VPRS model, $l$ is chosen equal to $1-u$ [9]. For $u = 0.8$ we obtain $R{\downarrow}_{.8} = D_1 \cup D_2$ and $R{\uparrow}_{.2} = D_1 \cup D_2 \cup D_3$. For $u = 0.9$, however, we obtain the same results as in Example 1.

## 3 Vaguely Quantified Rough Sets

The VPRS definitions for upper and lower approximation from the previous section can be softened by introducing vague quantifiers, to express that $y$ belongs to the upper approximation of $A$ to the extent that *some* elements of $Ry$ are in $A$, and $y$ belongs the lower approximation of $A$ to the extent that *most* elements of $Ry$ are in $A$. In this approach, it is implicitly assumed that the approximations are fuzzy sets, i.e., mappings from $X$ to $[0,1]$, that evaluate to what degree the associated condition is fulfilled.

To model the quantifiers appropriately, we use Zadeh's concept of a fuzzy quantifier [7], i.e. a $[0,1] \to [0,1]$ mapping $Q$. $Q$ is called regularly increasing if it is increasing and it satisfies the boundary conditions $Q(0) = 0$ and $Q(1) = 1$.

*Example 3.* Possible choices for $Q$ are the existential and the universal quantifier

$$Q_\exists(x) = \begin{cases} 0, & x = 0 \\ 1, & x > 0 \end{cases} \qquad Q_\forall(x) = \begin{cases} 0, & x < 1 \\ 1, & x = 1 \end{cases}$$

for $x$ in $[0,1]$, that will lead us to (4) and (5); or the quantifiers

$$Q_{>l}(x) = \begin{cases} 0, & x \leq l \\ 1, & x > l \end{cases} \qquad Q_{\geq u}(x) = \begin{cases} 0, & x < u \\ 1, & x \geq u \end{cases}$$

for $x$ in $[0,1]$, that will lead us to (6) and (7).

*Example 4.* The quantifiers in Example 3 are crisp, in the sense that the outcome is either 0 or 1. An example of a fuzzy quantifier taking on also intermediate values is the following parametrized formula, for $0 \leq \alpha < \beta \leq 1$, and $x$ in $[0,1]$,

$$Q_{(\alpha,\beta)}(x) = \begin{cases} 0, & x \leq \alpha \\ \frac{2(x-\alpha)^2}{(\beta-\alpha)^2}, & \alpha \leq x \leq \frac{\alpha+\beta}{2} \\ 1 - \frac{2(x-\beta)^2}{(\beta-\alpha)^2}, & \frac{\alpha+\beta}{2} \leq x \leq \beta \\ 1, & \beta \leq x \end{cases}$$

For example, $Q_{(0.1,0.6)}$ and $Q_{(0.2,1)}$ could be used respectively to reflect the vague quantifiers *some* and *most* from natural language.

Given sets $A_1$ and $A_2$ in $X$ and a fuzzy quantifier $Q$, Zadeh [7] computes the truth value of the statement "$Q$ $A_1$'s are also $A_2$'s" by the formula

$$Q\left(\frac{|A_1 \cap A_2|}{|A_1|}\right) \tag{12}$$

Once we have fixed a couple $(Q_l, Q_u)$ of fuzzy quantifiers, we can formally define the $Q_l$-upper and $Q_u$-lower approximation of $A$ by

$$R{\uparrow}_{Q_l} A(y) = Q_l\left(\frac{|Ry \cap A|}{|Ry|}\right) = Q_l(R_A(y)) \tag{13}$$

$$R{\downarrow}_{Q_u} A(y) = Q_u\left(\frac{|Ry \cap A|}{|Ry|}\right) = Q_u(R_A(y)) \tag{14}$$

for all $y$ in $X$. It is straightforward to verify that $R{\uparrow}_{Q_\exists}A = R{\uparrow}A$ and $R{\downarrow}_{Q_\forall}A = R{\downarrow}A$, and that $R{\uparrow}_{Q_{>l}}A = R{\uparrow}_l A$ and $R{\downarrow}_{Q_{\geq u}}A = R{\downarrow}_u A$. Moreover, if $Q_u \subseteq Q_l$, i.e., $Q_u(x) \leq Q_l(x)$ for all $x$ in $[0,1]$, then $R{\downarrow}_{Q_u}A \subseteq R{\uparrow}_{Q_l}A$.

*Example 5.* Let us return once more to the document retrieval problem discussed in Example 1 and 2. In our VQRS model with fuzzy quantifiers $Q_u = Q_{(0.2,1)}$ and $Q_l = Q_{(0.1,0.6)}$ the lower approximation $R{\downarrow}_{Q_u}A$ equals

$$\{(x_6,1),\ldots,(x_{10},1),(x_1,0.875),\ldots,(x_5,0.875),(x_{11},0.125),\ldots,(x_{15},0.125)\}$$

In this weighted list a document ranks higher if most of the elements in its topic category are in $A$. The gradations reflect the different roles of the categories in a desirable way. For example, category $D_3$ is not excluded but its documents are presented only at the bottom of the list. A similar phenomenon occurs with the upper approximation $R{\uparrow}_{Q_l}A = \{(x_1,1),\ldots,(x_{10},1),(x_{11},0.68),\ldots,(x_{15},0.68)\}$.

## 4 Vaguely Quantified Fuzzy Rough Sets

As the definition of vaguely quantified rough sets brings together ideas from fuzzy sets and rough sets, it is instructive to examine their relationship to, and combine them with existing work on fuzzy-rough hybridization. Throughout this section, we assume that $\mathcal{T}$ is a triangular norm (t-norm for short), i.e., any increasing, commutative and associative $[0,1]^2 \to [0,1]$ mapping satisfying $\mathcal{T}(1,x) = x$, for all $x$ in $[0,1]$, and that $\mathcal{I}$ is an implicator, i.e. any $[0,1]^2 \to [0,1]$-mapping $\mathcal{I}$ that is decreasing in its first, and increasing in its second component and that satisfies $\mathcal{I}(0,0) = 1, \mathcal{I}(1,x) = x$, for all $x$ in $[0,1]$. We also assume that the upper and lower approximation of a fuzzy set $A$ in $X$ under a fuzzy relation $R$ in $X$ are defined by [6]

$$R{\uparrow}A(y) = \sup_{x\in X} \mathcal{T}(R(x,y),A(x)) \tag{15}$$

$$R{\downarrow}A(y) = \inf_{x\in X} \mathcal{I}(R(x,y),A(x)) \tag{16}$$

for $y$ in $X$. Note how these formulas paraphrase the definitions (8) and (9) which hold in the crisp case. In particular, the sup and inf operations play the same role as the $\exists$ and $\forall$ quantifiers, and as such a change in a single element can still have a large impact on (15) and (16).

This observation has inspired some researchers to propose altered definitions of fuzzy-rough approximations in the spirit of the VPRS model. For example, Mieszkowicz-Rolka and Rolka [4] used the concept of a fuzzy inclusion set (based on an implicator) and the notion of $\alpha$-inclusion error (based on $\alpha$-level sets), while Fernández-Salido and Murakami [2] defined new approximations based on the so-called $\beta$-precision quasi minimum $\min_\beta$ and maximum $\max_\beta$ (aggregation operators dependent on a parameter $\beta$ in $[0,1]$). A serious drawback of these models is that they still rely on crisp thresholds $l$ and $u$ like Ziarko's model, which requires a fairly complex and not wholly intuitive mathematical apparatus.

The VQRS approach, on the other hand, lends itself to a much smoother and more elegant fuzzification. In fact, formulas (13) and (14) can simply be maintained in the fuzzy case, i.e., for $y$ in $X$ we have

$$R{\uparrow_{Q_l}}A(y) = Q_l\left(\frac{|Ry \cap A|}{|Ry|}\right) \tag{17}$$

$$R{\downarrow_{Q_u}}A(y) = Q_u\left(\frac{|Ry \cap A|}{|Ry|}\right) \tag{18}$$

with the conventions that the $R$-foreset $Ry$ is defined by $Ry(x) = R(x, y)$ for $x$ in $X$, the intersection $A \cap B$ of two fuzzy sets $A$ and $B$ in $X$ is defined by $(A \cap B)(x) = \min(A(x), B(x))$ and the cardinality $|A|$ of a fuzzy set $A$ in $X$ is defined by $\sum_{x \in X} A(x)$.

It is interesting that no implicator appears inside the VQRS lower approximation (18), as opposed to (16). In fact, $|Ry \cap A|/|Ry|$ and $\inf_{x \in X} \mathcal{I}(R(x, y), A(x))$ are considered in fuzzy set literature as two alternatives, to compute the inclusion degree of $Ry$ into $A$, the former set- or frequency-based and the latter logic-based (see e.g. [3]).

To demonstrate that the VQRS construct offers a worthwhile alternative to the traditional "logic"-based operations of fuzzy rough set theory in the context of rough data analysis, we ran an experiment on the `housing` benchmark dataset[1]. This dataset concerns housing prices in suburbs of Boston; it has 506 instances, 13 conditional attributes (12 continuous, one binary) and a continuous class attribute called MEDV (median value of owner-occupied homes in $1000s).

The setup of our experiment is as follows. Based on the distribution of the data, we defined a fuzzy partition on the universe of MEDV, containing three fuzzy classes *low*, *medium* and *high* in the range [0,50] as shown in Figure 1a. We also defined a fuzzy relation $R$ in the universe $X$ of instances expressing indistinguishability between instances $x_1$ and $x_2$ based on the conditional attributes:

$$R(x_1, x_2) = \min_{i=1}^{13} \max\left(0, \min\left(1, 1.2 - \alpha\frac{|c_i(x_1) - c_i(x_2)|}{l(c_i)}\right)\right) \tag{19}$$

in which $c_i$ denotes the $i^{\text{th}}$ conditional attribute, $l(c_i)$ is its range, and $\alpha$ is a parameter $\geq 1.2$ that determines the granularity of $R$ (the higher $\alpha$, the finer-grained the $R$-foresets).

We divided the instances into 11 folds for cross validation: in each step, we selected one fold as test set and used the remaining folds as training set $X'$ to compute the lower approximation of each decision class. For traditional fuzzy rough sets, we used formula (16), with three popular implicators $\mathcal{I}_L$ (Lukasiewicz), $\mathcal{I}_{KD}$ ( Kleene-Dienes), and $\mathcal{I}_G$ (Gödel) defined in Table 1. For the VQRS model, we used formula (18), with a fixed quantifier $Q_u = Q_{(0.2,1)}$ (shown in Figure 1b). We then predicted the membership of each test instance $y$ to each class $A$

[1] Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html

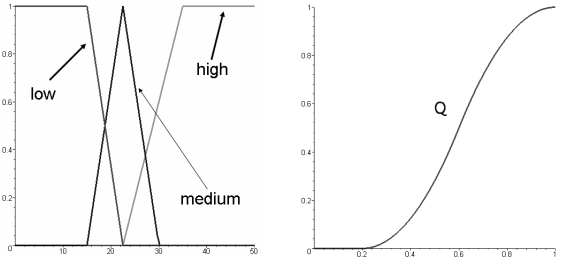



**Fig. 1.** a) Fuzzy partition of class attribute b) Fuzzy quantifier $Q_{(0.2,1)}$ for "most"

as the extent to which there exists a similar training instance $x$ belonging to the previously learned lower approximation $C$ of $A$:

$$\sup_{x \in X'} \mathcal{T}(R(x,y), C(x)) \tag{20}$$

In this formula, $\mathcal{T}$ is a t-norm; in our experiments, we used $\mathcal{T}_M$ (minimum) and $\mathcal{T}_L$ (Lukasiewicz), which are also shown in Table 1.[2]

The average absolute error between the predicted and the actual membership values of the test instances was used as a metric for comparing the approaches. Also, we let $\alpha$ in (19) range from 2 to 8. From the results in Table 2, we observe that all approaches perform better for increasing values of $\alpha$. This corresponds to the idea that a finer-grained relation allows for better approximation. However,

**Table 1.** Implicators and t-norms used in the experiment

| | |
|---|---|
| $\mathcal{I}_L(x,y) = \min(1-x+y,1)$ | $\mathcal{T}_L(x,y) = \max(0,x+y-1)$ |
| $\mathcal{I}_{KD}(x,y) = \max(1-x,y)$ | $\mathcal{T}_M(x,y) = \min(x,y)$ |
| $\mathcal{I}_G(x,y) = \begin{cases} 1, \text{ if } x \leq y \\ y, \text{ otherwise} \end{cases}$ | |

**Table 2.** Experimental results for 11-fold cross-validation

| $\alpha$ | $\mathcal{I}_L$-$\mathcal{T}_L$ | $\mathcal{I}_{KD}$-$\mathcal{T}_M$ | $\mathcal{I}_G$-$\mathcal{T}_M$ | VQRS-$\mathcal{T}_L$ | VQRS-$\mathcal{T}_M$ |
|---|---|---|---|---|---|
| 2 | 0.276 | 0.320 | 0.321 | 0.257 | 0298 |
| 3 | 0.264 | 0.301 | 0.315 | 0.238 | 0.280 |
| 4 | **0.258** | 0.288 | 0.299 | **0.236** | 0.265 |
| 5 | 0.263 | 0.268 | 0.274 | 0.246 | **0.256** |
| 6 | 0.272 | **0.264** | **0.270** | 0.261 | 0.258 |
| 7 | 0.282 | 0.269 | 0.271 | 0.274 | 0.266 |
| 8 | 0.291 | 0.280 | 0.280 | 0.286 | 0.279 |

[2] $\mathcal{I}_{KD}$ and $\mathcal{I}_G$ are, respectively, the S-implicator and R-implicator of $\mathcal{T}_M$, while the S- and R-implicator of $\mathcal{T}_L$ coincide in $\mathcal{I}_L$.

for a too fine-grained relation, the average errors start increasing again, indicating an overfit of the model.

Comparing $\mathcal{I}_L$-$\mathcal{T}_L$ with VQRS-$\mathcal{T}_L$, we notice that in both cases the smallest error is obtained for $\alpha = 4$. The corresponding relation is still relatively coarse-grained. We observe that our VQRS-$\mathcal{T}_L$ approach is least hampered by this: it in fact achieves the lowest average error of all approaches displayed in the table. The approaches with $\mathcal{T}_M$ score worse in general, but again the smallest error is obtained with our VQRS-$\mathcal{T}_M$ model.

## 5 Conclusion

In the VQRS model introduced in this paper, an element $y$ belongs to the lower approximation of a set $A$ to the extent that *most* elements related to $y$ are in $A$. Likewise, $y$ belongs to the upper approximation to the extent that *some* elements related to $y$ are in $A$. The use of vague quantifiers "most" and "some", as opposed to the traditionally used crisp quantifiers "all" and "at least one" makes the model more robust in the presence of classification errors. Experimental results on the `housing` dataset show that VQRS consistently outperforms the classical approach.

**Acknowledgment.** Chris Cornelis would like to thank the Research Foundation—Flanders for funding his research.

## References

1. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. International Journal of General Systems **17** (1990) 91–209
2. Fernández Salido, J.M., Murakami, S.: Rough set analysis of a general type of fuzzy data using transitive aggregations of fuzzy similarity relations. Fuzzy Sets and Systems **139(3)** (2003) 635–660
3. Fodor, J., Yager, R.: Fuzzy set theoretic operators and quantifiers. Fundamentals of Fuzzy Sets (D. Dubois, H. Prade, eds.), Kluwer, Boston, Mass. (2000) 125–193
4. Mieszkowicz-Rolka, A., Rolka, L.: Variable precision fuzzy rough sets. Transactions on Rough Sets I. Lecture Notes in Computer Science **3100** (2004) 144–160
5. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences **11(5)** (1982) 341–356
6. Radzikowska, A.M., Kerre, E.E.: A comparative study of fuzzy rough sets. Fuzzy Sets and Systems **126** (2002) 137–156
7. Zadeh, L.A.: A computational approach to fuzzy quantifiers in natural languages. Computers and Mathematics with Applications **9** (1983) 149–184
8. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences **46** (1993) 39-59
9. Ziarko, W.: Set approximation quality measures in the variable precision rough set model. Soft Computing Systems: Design, Management and Applications (A. Abraham, J. Ruiz-del-Solar, M. Koppen, eds.), IOS Press, (2002) 442–452

# A Fuzzy Search Engine Weighted Approach to Result Merging for Metasearch

Arijit De[1], Elizabeth D. Diaz[2], and Vijay Raghavan[1]

[1] Center of Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA 70504
[2] University of Texas of Permian Basin, Odessa, TX 79762
axd9142@cacs.louisiana.edu, diaz_e@utpb.edu, raghavan@cacs.louisiana.edu

**Abstract.** Each search engine queried by a metasearch engine returns results in the form of a result list of documents. The key issue is to combine these lists to achieve the best performance. The salient contribution of this paper is a result merging model that applies Yager's fuzzy aggregation Ordered Weight Average, OWA, operator in combination with the concept of importance guided aggregation to extend the OWA-based result merging model proposed by Diaz. Our result merging model, IGOWA, (Importance Guided OWA) improves upon the OWA model proposed by Diaz so as to allow weights to be applied to search engine result lists. To support our model we also explore a scheme for computing search engine weights. We call the weights obtained from our scheme Query-System Weights and we compare this with the scheme for computing search engine weights proposed by Aslam and Montague. We refer to Aslam's scheme as System Weights.

**Keywords:** Data Fusion, Metasearch engines, Metasearching, OWA, IGOWA.

## 1 Introduction

Metasearching is an application of standard data fusion techniques in the field of information retrieval. A metasearch engine is an Internet search tool that allows a user access to multiple search engines. It passes a query to multiple search engines, retrieves results from each of them in the form of a result list, and then combines these result lists into a single result list. Two key components of a metasearch engine are the query dispatcher and the result merger. The query dispatcher selects search engines to which the query is sent and the result merger merges the results from multiple search engines. In this paper we propose a fuzzy result-merging model (IGOWA) for metasearch, which extends the OWA model for result merging proposed by Diaz [3, 4] so as to allow us to give different weights to different search engine result lists prior to merging. Since our model needs search engine weights we also propose a scheme to compute search engine weights. We call the weights computed using this scheme Query System Weights. Aslam and Montague [2] suggest a simplistic scheme to compute search engine weights. Subsequently, we shall refer to weights computed using this scheme as System Weights. In our experiments we compare the performance of IGOWA model with the OWA model when System

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 95–102, 2007.

Weights and Query System Weights are used as inputs to the IGOWA model. The rest of the paper is organized in four sections. Section 2 describes a summary of related work. Section 3 describes our merging approach by using IGOWA. Section 4 discusses how to compute the weights by IGOWA. Section 5 presents experimental results achieved by using IGOWA, and compares them with OWA. Experimental data comes from TREC (Text Retrieval Conference).

## 2 Related Work

Data fusion techniques have been applied to develop result merging models in the past. Early research in this field can be attributed to Thompson [9], Fox & Shaw [5], Alvarez [1] have all explored this area. Other models include the Logistic Regression Model [6], and the Linear Combination Model.

Based on the political election strategy, Borda Count, Aslam and Montague [2] proposed two models. The first of these, the Borda-Fuse, works by assigning points to each document in each of the lists to be merged. Documents are ranked in descending order of total points accumulated by virtue of its rank in each result list. The second is the Weighted Board-Fuse model. This is a weighted version of the Borda-Fuse model and ranks documents in descending order of the linear combination of the product of points earned in each result list and the weight of the search engine who returns the result list. For the latter model, Aslam and Montague [2] proposed a scheme to learn search engine weights, based on the prior performance of the search engine.

A major shortcoming of the Borda-Count based models is the handling of missing documents. In these models every document, based on its position/rank in each result list, gets a certain number of points. However some documents appear in some but not all result lists. Reasons for missing documents have been discussed by Diaz [3, 4]. Borda-Fuse model assigns no points for these missing documents. This results in missing documents being ranked at the bottom of the list. According to Meng [7, 8] missing documents pose a major challenge to result merging as they make the content of the result lists heterogeneous. Diaz [3, 4] develops a fuzzy result merging model OWA which is based on the fuzzy aggregation OWA operator by Yager [11, 12]. The OWA model uses a measure similar to Borda points, called positional values. The positional value (PV) of a document $d_i$ in the result list $l_k$ returned by a search engine $s_k$ is defined as $(n - r_{ik} + 1)$ where, $r_{ik}$ is the rank of $d_i$ in search engine $s_k$ and n is the total number of documents in the result. Thus, higher the rank of a document in a result list, the larger the positional value of the document in that list. One key feature of the OWA model is that it provides two heuristics (H1 and H2) for handling missing documents. This is done by computing the positional value of a missing document in the result list and thereby effectively inserting the document in the result list in which it is missing. Diaz in [3] shows that the heuristic H1 provides the most effective way to handle missing documents. Let $PV_i$ be the positional values for a document d in the $i^{th}$ search engine. Let m be the total number of search engines. Let r be the number of search engines in which d appears. Let j denote a search engine not among the r search engines where d appears. In heuristic H1 $PV_j$ for all j is denoted by the average of the positional values of the documents in r search engines. In heuristic H2

$PV_j$ for all j is denoted by the average of the positional values of the documents in all m search engines.

## 3 Merging Approach

In this section we discuss our IGOWA result merging model that is the extension of the OWA model for metasearch. Let us define the OWA operator, F, of dimension n as a mapping F: $R^n$ -> R which has an associated weighting vector, $W = [w_1, w_2, ... w_n]^T$ given by:

$$w_j \in [0,1] \text{ and } \sum_{j=1}^{n} w_j = 1 \tag{1}$$

$$F(a_1, a_2,..., a_n) = \sum_{j=1}^{n} w_j b_j, \text{ where } b_j \text{ is the } j^{th} \text{ largest } a_i. \tag{2}$$

From equations (1) and (2) it can be seen that the OWA weights are key to the orness of the fuzzy aggregation. If $w_1 = 1$ and the other weights are all set to 0, then the maximum $a_i$ is selected. This is a case of high orness. In the same way is $w_n = 1$ and all other weights are set to zero then the minimum $a_i$ is selected. This is a case of low orness or high andness. Yager [11] characterizes the orness of the fuzzy aggregation in equation (3).

$$\text{orness}(w) = \frac{1}{n-1} \sum_{i=1}^{n} (n-i) w_i. \tag{3}$$

The concept of Importance Guided Aggregation was introduced by Yager [10, 11, 12] in the context of multi-criteria decision making. Looking at Importance Guided Aggregation from the multi-criteria decision making perspective, we can define the multi-criteria decision making problem to along the lists of n criteria that have importance weights attached to them. For each criterion, there is a set of alternatives. We can move to define the $i^{th}$ criterion as $A_i$, where $A_i(x)$ is the degree or extent to which the alternative x satisfies the criterion. When we use a RIM linguistic quantifier based aggregation model, we create an overall decision function that in effect models the phrase "Q criteria are satisfied by x" where Q is our quantifier. A RIM quantifier is of the form $Q(r) = r^{\alpha}$, where α is the order of the polynomial. Yager shows that the orness of the RIM quantifier is of the form:

$$\text{orness}(Q) = \frac{1}{1+\alpha} \tag{4}$$

So when $\alpha < 1$, we observe a condition of high orness. When $\alpha = 1$, the result is that of balanced orness and when $\alpha > 1$ the condition is that of low orness or high andness.

Let the importance weight attached to the $i^{th}$ criterion be $V_i$. The importance weight $V_i$ can be normalized to lie in the interval [0, 1]. Next we can proceed to evaluate the satisfaction of the alternative x. For the alternative x we can form n pairs $(V_i, A_i(x))$.

We now proceed to sort the $A_i(x)$s in descending order. Let $b_j$ be the $j^{th}$ largest $A_i(x)$. Let $u_j$ be the importance weight attached to the criterion that alternative x satisfies $j^{th}$ most. Thus, if $A_3(x)$ is the largest, then $b_1 = A_3(x)$ and $u_1 = V_3$. We can now associate, with alternative x, a collection of n $(u_j, b_j)$ pairs, where the $b_j$ s are degrees to which x satisfies the n criteria in descending order. We can now proceed to obtain the ordered weights by:

$$w_j(x) = Q\left(\frac{\sum_{k=1}^{j} u_k}{T}\right) - Q\left(\frac{\sum_{k=1}^{j-1} u_k}{T}\right) \tag{5}$$

where

$$T = \sum_{k=1}^{n} u_k \tag{6}$$

Once we have obtained the ordered weights we can now calculate a composite value of how well the alternative x satisfies the criteria using the following equation:

$$D(x) = \sum_{j=1}^{n} b_j w_j(x) \tag{7}$$

Here $b_j$ is the $j^{th}$ greatest $A_i(x)$. Our proposed model for metasearch is based on the principle of Importance Guided aggregation using the OWA operator described earlier. In this model each search engine is thought of as a criterion. Each document is said to be an alternative. Each search engine (criterion) returns results in the form of result lists of documents. Based on how high each document is ranked by a search engine, we compute the positional value of each document with respect to the search engine as described in the previous chapter. This measure is analogous to the degree to which an alternative (in our case a document) satisfies the criterion (search engine). However in this case each search engine (or search engine result list) has an importance weight attached to it. Let us illustrate the working of this model with an example. We start with four search engines $SE_1$ , $SE_2$ , $SE_3$ and $SE_4$ that have importance weights of 0.9, 0.6, 0.5 and 0.4 respectively and a set of 5 documents, some or all of which are returned by the search engines. Table 1 shows the result list of documents returned by each search engine. We use a quantifier $Q(r) = r^{\alpha}$.

**Table 1.** Result List from the four search engines

| | Rank 1 | Rank 2 | Rank 3 | Rank 4 | Rank 5 |
|---|---|---|---|---|---|
| $SE_1$ | $D_2$ | $D_1$ | $D_3$ | $D_4$ | $D_5$ |
| $SE_2$ | $D_2$ | $D_3$ | $D_4$ | | |
| $SE_3$ | $D_2$ | $D_5$ | $D_4$ | $D_1$ | |
| $SE_4$ | $D_5$ | $D_3$ | $D_2$ | $D_4$ | $D_1$ |

From this we can calculate the positional values of the documents with respect to the search engines. Using the heuristic H1 proposed by Diaz [3] missing documents is inserted into the ranked lists. Table 2 shows the positional values.

**Table 2.** Positional Values of the documents as returned by the search engines

| | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ |
|---|---|---|---|---|---|
| $SE_1$ | 4 | 5 | 3 | 2 | 1 |
| $SE_2$ | 2 | 3 | 2 | 1 | 3 |
| $SE_3$ | 1 | 4 | 3 | 2 | 3 |
| $SE_4$ | 1 | 3 | 4 | 2 | 5 |

Let us proceed to compute the final score for document $D_1$. We evaluate T to be 2.4 (0.9+0.6+0.5+0.4). The weights can be calculated using equation (5) based on the $u_j$ information in table 3. It should be noted that the rows in table 3 are sorted by $b_j$ values. With weights computed we can compute the score of document D(x) as in equation 8 where $x=D_1$. Similarly the scores for other documents can be calculated and the documents can be ranked in the final result list based by descending order of score.

**Table 3.** Positional Values and Importances for $D_1$

| | $b_i$ | $u_i$ |
|---|---|---|
| $SE_1$ | 4 | 0.9 |
| $SE_2$ | 2 | 0.6 |
| $SE_3$ | 1 | 0.5 |
| $SE_4$ | 1 | 0.4 |

$$D(x) = \sum_{j=1}^{4} b_j w_j(x) = 1.65687 \tag{8}$$

## 4 Computing Search Engine Weights

The IGOWA model for metasearch requires search engine weights. So let us compare our scheme for computing search engine weights with that used by Aslam and Montague [2].

### 4.1 System Weights

Aslam and Montague [2] proposed a scheme for computing the importance weights for search engine result lists based on the performance of search engines over a set of queries. A set of queries are passed to each search engine being covered by the metasearch engine. For each query, the search engine returns results in the form of a result list. For each search engine, the result list for each query is evaluated and the

average precision is computed. These are then averaged out to give a measure of the performance of the search engine. Correspondingly search engines importance weights can be determined according the performance measure over all queries.

### 4.2 Query System Weights

In the above approach, importance weights for search engine result lists are computed based on an overall performance of a search engine over a set of training queries. However, search engines respond to different queries in different ways. Thus a search engine that does poorly for one specific query or a group/cluster of similar queries might do better for another query or group of queries. In this work, we propose a scheme for computing importance weights that can assign different set of weights to the search engines based on the type of query being processed. We call these importance weights Query System weights or QSW for short.

Computing Query System Weights would be in two phases. The first would be a training phase to determine search engine importance weights for each cluster/group of queries. The second phase would be to fit an incoming query to a cluster. Let us say we have a set of search engines SE-SET = $\{se_i$, for all i, $1 \leq i \leq n\}$. Let us say we have a set of queries QUERY-SET = $\{q_i, 1 \leq i \leq m\}$. Let us consider a subset SUB-SE-SET of k randomly picked search engines SUB-SE-SET = $\{se_i$, for all i, $1 \leq i \leq k$ and $k < n\}$. The following steps are executed in phase 1 (training phase):

1. Pass all queries in QUERY-SET to search engines in SUB-SE-SET, and retrieve search results in the form of a result list. Evaluate the average precision of the result list.
2. So for a query $q_i$ in QUERY-SET for search engine $SE_j$ in SUB-SE-SET, we get result list $r_{ij}$. The average precision of the result list $r_{ij}$ is $p_{ij}$.
3. So for query $q_i$, we get a set of result lists $\{r_{ij}$, for all j, $1 \leq j \leq k\}$. For each list we can obtain an average precision. Thus we can form a precision vector $qv_i = \{ p_{ij}$, for all j, $1 \leq j \leq k \}$ for the query $q_i$.
4. We use the k-means algorithm to form the cluster of queries. For each cluster we pass all queries to all search engines in SE-SET, retrieve result lists, and compute average precision of result lists. For a search engine $se_j$ in SE-SET the average of the average precision can be computed as shown in equation 16.
5. We can compute importance weights for search engines for each cluster of queries based on the average performance of the search engine for all queries in the cluster.

The second phase is the testing phase. In this phase a new query q is made by the user. We pass the query to search engines in SE-SUB-SET. We obtain result lists for each search engine, compute the average precision for each result list, and form a feature vector $qv = \{ p_j$, for all j, $1 \leq j \leq k \}$. We then establish query membership. Once membership to a cluster is established, we can use the importance weights for the cluster.

## 5 Experiments and Results

In our experiments we used the data sets from the Text Retrieval Conference (TREC). We used the data sets TREC 3, TREC 5 and TREC 9. Each of the three data sets has a

set of 50 topics or queries and a number of systems or search engines. For each topic (query) and each system (search engine) a result list of documents retrieved is returned. Each data set also provided relevance information for each document with respect to a query. We use this relevance information to compute the average precision of the merged result list.

In order to learn search engine Query System Weights (QSW) and System Weights (SW), we use odd queries in the data sets. We used even queries for evaluating our result merging model IGOWA and comparing the average precision of the merged list returned by our IGOWA model with the average precision of the merged list returned by Diaz [3] OWA model.

In our experiments we pick a query at random and pick a certain number of search engines being merged. The number of search engines being merged varies from 2 to 12. We then pick the determined number of search engines randomly and merge the result lists returned by them for a query using the IGOWA model and the OWA model. We compute the average precision of the merged result list returned by each of the models. Both the OWA and the IGOWA models use a RIM quantifier of the form $Q(r) = r^{\alpha}$. In our experiments we vary the α parameter from high orness conditions of 0.25 and 0.5 through to balanced orness condition of 1.0 and low orness condition of 2.0, 2.5 and 5. The figures below show the graphical results for TREC 3, TREC 5 and TREC 9. We measure the effects of the quantifier parameter α (x-axis on the graphs) on the average precision (y-axis) of the merged list.

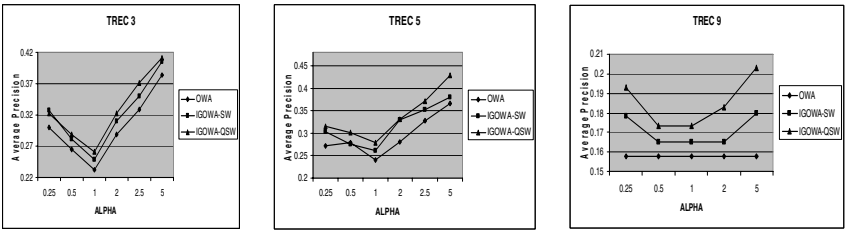



**Fig. 1.** Experimental Results showing a comparison of TREC3, TREC5 and TREC9

From the results of the experiments we can observe that the IGOWA model for metasearch outperforms the OWA model for metasearch irrespective of whether our scheme to compute Query System weights or Aslam's scheme to compute System weights is used. However, the IGOWA model for metasearch yields better performance in terms of average precision when the scheme to compute Query System Weights is employed over when the scheme to compute System Weights.

Previously we discussed how values of $\alpha < 1$, denotes a condition of high orness and values of $\alpha > 1$ shows conditions of low orness. In our experiments we use values of $\alpha = 0.25, 0.5, 1, 2, 2.5$ and 5. In our experiments we observe that when α is increased from 0.25 to 1 the performance, in terms of average precision of the merged list, of each of the models IGOWA with Query System Weights, IGOWA with System Weights and OWA all go down somewhat. When α is increased from 1 to 5, however there is a steady increase in the performance for all three models. For each of the models the average precision is the lowest when $\alpha = 1$. We also observe that the

fuzzy result merging models perform well under low or high orness condition, but not under balanced orness condition. Also from the experiments we can observe that the performance is best for high orness condition.

## References


1. Alvarez, S. A.: Web Metasearch as Belief Aggregation, AAAI-2000 Workshop on Artificial Intelligence for Web Search, Austin, TX, July (2000).
2. 2. Aslam, J. A., Montague, M.:. Models for Metasearch, Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, New Orleans, Louisiana, United States, September (2001), 276-284.
3. Diaz, E. D., De, A., and Raghavan, V. V.: A comprehensive OWA-based framework for result merging in metasearch. In Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing (RSFDGrC '05) (Regina, Canada, August 31 - September 3, 2005). Springer-Verlag, Heidelberg, DE, 193-201.
4. Diaz, E. D.: Selective Merging of Retrieval Results for Metasearch Environments. Ph.D. Dissertation, The Center of Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA, (2004).
5. Fox, E. A., Shaw J. A. : Combination of multiple searches, Proceedings of the 2nd Text Retrieval Conference (TREC-2), National Institute of Standards and Technology Special Publication 500-215, (1994), 243-252.
6. Hull, D. A., Pedersen, J. O. and Schütze, H.: Method combination for document filtering, Proceedings of the 19th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland, August 18-22, (1996), 279-287.
7. Meng, W., Yu, C., Liu, K.: Building Efficient and Effective Metasearch engines, ACM Computing Surveys, March (2002), 48-84.
8. Meng, W., Yu, C., Liu, K.: A Highly Scalable and Effective Method for Metasearc*h,* ACM Transactions on Information Systems*,* July (2001), 310-335.
9. Thompson, P.: A combination of expert opinion approach to probabilistic information retrieval, part 1: The conceptual model, Information Processing and Management: an International Journal, v.26 n.3, (1990), 371-382.
10. Yager, R. R.: On ordered weighted averaging aggregation operators in multicriteria decision making, Fuzzy Sets and Systems, vol. 10, (1983), 243-260.
11. Yager, R. R.: Quantifier guided Aggregating using OWA operators, International Journal of Intelligent Systems 11, (1996), 49-73.
12. Yager, R. R., Kreinovich V.: On how to merge sorted lists coming from different web search tools. Soft Computing Research Journal, 3, March (1999), 83-88.

# A Fuzzy Group Decision Approach to Real Option Valuation

Chen Tao, Zhang Jinlong, Yu Benhai, and Liu Shan

School of management, Huazhong University of Science and Technology, Wuhan, 430074, China
chentaohust@yahoo.com.cn, jlzhang@mail.hust.edu.cn, ybhai@163.com, liushan@163.com

**Abstract.** This paper develops a comprehensive but simple methodology for valuating IT investment using real options theory under the fuzzy group decision making environment. The proposed approach has the following advantages: (1) It does not need to formulate the distribution of expected payoffs, thus complex estimation tasks can be avoided. (2) It allows multiple stakeholders be involved in the estimation of real option value, therefore could alleviate the bias from particular evaluator's personal preference and could help decision makers achieve a more reliable valuation of the target investment. The author provides numerical illustration on the procedures mentioned above and discusses the strengths and possible extensions of this hybrid approach to IT investment analysis.

**Keywords:** fuzzy sets, real options analysis, information technology investment.

## 1 Introduction

Real options are developed from the concept of financial options. Financial options is a kind of contract, it gives its holders a right, but not the obligation, to buy or sell a specific quantity of financial assets with a fixed price. Later, it is recognized that the concept of options can also be extended to real (not financial) assets. Real options are often oriented to the right to invest on future possible opportunities. Thinking of future investment opportunities as "real options" has provided powerful new analysis tools that in many ways revolutionized modern corporate resource allocation. Real options analysis (ROA) is proved suitable for modeling IT investment involving an option, and its strengths have been illustrated by plenty of literatures [1-3]. However, there still exist some gaps between ROA and what is needed to effectively evaluate real world IT investment[4]. The lack of the knowledge about real options and several challenging preliminary requirements has prevented managers from utilizing this salient tool. For example, classical option pricing model require the variance per period of rate of return on the asset must be estimated. In fact, obtaining such a reliable estimation of the variance is usually very difficult[5]. Furthermore, option pricing model generally assumes that the expected payoffs are characterized by certain probably distributions, geometric Brownian motion, for instance. This

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 103–110, 2007.

assumption could be a rather strong one to make, and the misuse of probability would bring about a misleading level of precision.

In this paper, we propose a new methodology to value IT investment using the real options theory under the fuzzy group decision making environment. The valuation model we present is based on the assumption that the uncertainty of the expected payoffs from IT investments is not merely stochastic but also vague in nature. Fuzzy group decision-making theory is able to well formulate the uncertainty of the expected payoffs from an investment, and in addition it will alleviate the bias of possible personal preference or discrimination. This approach will help IT managers acquire a more objective and efficient valuation of the target investment.

## 2 Classical Real Option Pricing Model

There is a variety of real option pricing models that suits for different situations. Black–Scholes option pricing model is the most often used approach to price simple European growth option, which takes on the following form:

$$\mathrm{C} = \mathrm{V}N(d_1) - X\,e^{-rT}N(d_2) \tag{1}$$

Where $\mathrm{d}_1 = \dfrac{In(V/X) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}, \qquad d_2 = d_1 - \sigma\sqrt{T}$

Where C denotes the real option value (ROV) of the target projects, V is the present value of expected payoffs that is assumed to be log normally distributed, X is the expected costs, $\sigma$ is the volatility of the expected revenue, r is risk-free rate of interest, T represents maximum investment deferral time, and $N(\bullet)$ denotes the cumulative normal distribution function.

While expected costs and option's time are relatively straightforward to estimate, the expected payoffs and its deviation are usually very challenging to obtain. In the classical option valuation methods, this problem could be solved by seeking a "twin security"—a traded security the price of which is perfectly or highly correlated with the price of the target investment under consideration. However, in many cases, such a traded security does not exit and the reference price is not observable. The other way is to represent the uncertain payoffs expected using a probably distribution, geometric Brownian motion, for instance. This assumption may be defended for financial options, for which there could be an efficient market with numerous player and numerous stocks for trading. The law of large could apply to this efficient market, thus justify the use of probability theory[6]. Nevertheless, the situation for real options is rather different, especially for IT investment valuation. As to IT investment, the number of players producing the consequence is usually quite small. Moreover, decision makers cannot obtain historic date of past revenues and costs to formulate the distribution of expected payoffs. Therefore, the use of assumption on purely stochastic phenomena is not well-substantiated for IT investment valuation.

## 3 A Fuzzy Group Decision Making Approach to Real Option Valuation

In this section, we present a new real option approach using the tool of fuzzy group decision making. The investment valuation process consists of four steps as follows:

Step 1: estimate the expected payoffs
Each evaluator will give fuzzy estimation individually to the parameter of the expected payoffs, which will be not only the basis of calculation of real option value, but also the determination to the weight of each evaluator, which is how the model alleviates the bias of evaluators additionally. For simplicity of formulation, we adopt triangular fuzzy numbers to characterize the estimation of the payoffs. Supposing there are n evaluators in all, evaluator k estimates the expected payoffs by using a triangular possibility distribution of the form

$$V^k = (V_1^k, V_M^k, V_2^k) \qquad k=1, 2, \ldots, n \tag{2}$$

i.e. the most possible value of the payoffs is $V_M^k$, and $V_1^k$ is the downward potential and $V_2^k$ is the upward potential for the expected payoffs.

Step 2: calculate the weight of each evaluator
In the group decision making literature, the relative weight of each evaluator has been largely ignored. It is usually assumed that all members have the same importance. However, in many real life settings, specific members have recognized abilities and attributes, or privileged positions of power[7]. Thus, it is necessary to find the weights to be assigned to the members of the group. This is often a difficult task, especially when the target value is quite uncertain. These situations need an objective method to derive members' weights. This paper employs a simple and intuitively appealing eigenvector based method to determine the weights for group members using their own subjective opinions [8], which could help to avoid serious bias from particular evaluator's personal preference. The weight of each evaluator will be calculated through the distance from each other.

After each member gives estimation to the expected payoffs, the distance between evaluator k and l can be calculated as:

$$d(V^k, V^l) = \sqrt{\frac{1}{2}\left[(V_1^k - V_1^l)^2 + (V_M^k - V_M^l)^2 + (V_2^k - V_2^l)^2\right]} \tag{3}$$

In order to reflect the difference between each evaluator and others, construct the distance matrix as:

$$D = \begin{bmatrix} 0 & d(V^1,V^2) & d(V^1,V^3) & \cdots & d(V^1,V^n) \\ & 0 & d(V^2,V^3) & \cdots & d(V^2,V^n) \\ & & 0 & \cdots & \\ & symmetrical & & \ddots & \\ & & & & 0 \end{bmatrix} \tag{4}$$

Let $D_k = \sum_{j=1}^{n} d(V^k, V^j)$, which reflects the difference between the evaluation of evaluator k and those of others. The less is $D_k$, the nearer the evaluation of evaluator k to those of others. Thus the weight of evaluator k will be $W_k = \frac{1/D_k}{\sum_{k=1}^{n}(1/D_k)}$, $\sum_{k=1}^{m} W_k = 1$.

Step 3: fuzzy assessment aggregation

After obtaining the fuzzy assessment and the weight of each evaluator, we start to formulate the expected payoffs of the IT investment under consideration. Fuzzy weighted average (FWA) is a commonly used operation in decision analysis, and of the form: $V = \frac{\sum_{i=1}^{n} W_i \times V^i}{\sum_{i=1}^{n} W_i}$. The result of the calculated fuzzy weighted average is a fuzzy number $V = (V_1, V_M, V_2)$, which represents the expected payoffs of the IT investment under consideration. Since our purpose is to valuate the real option value of the investment, it's required to estimate the standard deviation of the expected payoffs.

Let $A$ be a fuzzy number with $[A]^\gamma = [a_1(\gamma), a_2(\gamma)]$, $\gamma \in [0,1]$. [6] introduced the possibilistic expected value of triangular fuzzy number $A = (a_1, a_M, a_2)$ as

$$E(A) = \frac{2}{3} a_M + \frac{1}{6}(a_1 + a_2) \tag{5}$$

And the possibilistic variance of fuzzy figure A as

$$\sigma^2(A) = \frac{1}{2}\int_0^1 \gamma(a_1(\gamma) - a_2(\gamma))^2 \, d\gamma \tag{6}$$

i.e. $\sigma^2(A)$ is defined as the expected value of the squared deviation between the arithmetic mean and the endpoints of its level sets. Thus, the possibilistic variance of the triangular fuzzy number V can be easily formulated

$$\sigma^2(V) = \frac{1}{2}\int_0^1 \gamma[(V_2 - V_1)(1-\gamma)]^2 \, d\gamma = \frac{(V_2 - V_1)^2}{24} \tag{7}$$

Step 4: the real option valuation of the investment

In the last step, we can assess the real option value of the investment based on the result obtained above. For the purpose of simplicity, we assume that only the expected payoff is uncertain and utilize the fuzzy term of the Black-Scholes pricing formula presented by Carlson etc. [9]. Then the fuzzy real option value of an investment is

$$FROV = VN(d_1) - X \quad e^{-rT} N(d_2) \tag{8}$$

$$\text{Where } d_1 = \frac{In(E(V)/X) + (r + \sigma^2/2)T}{\sigma\sqrt{T}},\ d_2 = d_1 - \sigma\sqrt{T}$$

Only V is fuzzy numbers. E(V) and $\sigma$ represent respectively the possibilistic expected value and the standard deviation of fuzzy figure V. The computing result FROV is also a fuzzy number, representing the real option value of the investment under consideration.

## 4 Numerical Examples

Benaroch and Kauffman [3] used Black-Scholes model to examine the decision to defer the deployment of point-of-case (POS) debit services at the Yankee 24 shared electronic banking network of New England . A series of interviews with decision makers of the company are conducted to determine the volatility of the expected revenues, and it was estimated to be between 50%-100%. In terms of costs, an initial investment of $400,000 was needed to develop the network, and there are an operational marketing cost of $40,000 per year. The firm was assumed to capture the revenues resulting from the market size 1 year after the initial investment, and the time horizon of the project was considered to be 5.5 years. Finally, 50% was used to compute the investment opportunity. The analysis results showed that the value of Yankee's American deferral option is $152,955. This value corresponds to the Europe option value at optimal deferral time, which is three years.

In this section, we apply the fuzzy real option valuation approach to analyze this case using the data provided by Benaroch and Kauffman. Since we do not have the firsthand data about this case, and out purpose is only to provide an illustration on calculating process, we simply assume there would be five evaluators involved in this decision making process. They give the following estimation for the expected cash flow from the projects when the deferral time is T=3:

**Table 1.** The assessment of the expected payoffs by each evaluator

| The expected payoffs | Evaluator 1 | Evaluator 2 | Evaluator 3 | Evaluator 4 | Evaluator 5 |
|---|---|---|---|---|---|
| The downward potential | $376,000 | $355,000 | $400,000 | $368,000 | $400,000 |
| The most possible value | $387,166 | $400,000 | $420,000 | $412,000 | $450,000 |
| The upward potential | $398,332 | $445,000 | $440,000 | $456,000 | $500,000 |

After each member gives estimation to the expected payoffs, the weight of each evaluator will be calculated through the distance from each other. The distance between evaluator 1 and 2 is:

$$
\begin{aligned}
d(V^1,V^2) &= \sqrt{\frac{1}{2}\left[(V_1^1-V_1^2)^2+(V_M^1-V_M^2)^2+(V_2^1-V_2^2)^2\right]} \\
&= \sqrt{\frac{1}{2}\left[(376000-355000)^2+(387166-400000)^2+(398332-445000)^2\right]} \\
&= 37307
\end{aligned} \tag{9}
$$

Similarly, all the distance between evaluators can be calculated and the distance matrix D is

$$
D = \begin{bmatrix}
0 & 37307 & 41172 & 44757 & 86199 \\
37307 & 0 & 35000 & 14731 & 61441 \\
41172 & 35000 & 0 & 25923 & 47434 \\
44757 & 14731 & 25923 & 0 & 46925 \\
86199 & 61441 & 47434 & 46925 & 0
\end{bmatrix} \tag{10}
$$

The difference between the evaluation of evaluator 1 and all those of others is

$$
D_1 = \sum_{j=1}^{n} d(V^1,V^j) = 0+37307+41172+44757+86199 = 209435 \tag{11}
$$

And $D_2$=148479, $D_3$=149529, $D_4$=132336, $D_5$=242000.

Then the weight of each evaluator is

$$
W_1 = \frac{1/d_1}{\sum_{k=1}^{n}(1/d_k)} = \frac{1/209435}{1/209435+1/148479+1/149529+1/132336+1/242000} = 0.1598 \tag{12}
$$

Similarly, $W_2$=0.2254, $W_3$=0.2238, $W_4$=0.2528, $W_5$=0.1383,

Using the equation $V = \dfrac{\sum_{i=1}^{n} W_i \times V^i}{\sum_{i=1}^{n} W_i}$, we can aggregate all the evaluators' judgments to obtain the final estimation of the expected payoffs, which is

$$
\begin{aligned}
V &= (V_1, V_M, V_2) = (\$377{,}934,\ \$412{,}372,\ \$446{,}811) \\
E(V) &= \frac{2}{3}\times 412372 + \frac{1}{6}\times(377934+446811) = \$412{,}372
\end{aligned} \tag{13}
$$

The standard deviation of expected payoffs can be calculate as

$$
\sigma(\mathrm{V}) = \sqrt{\frac{(V_2-V_1)^2}{24}} = \sqrt{\frac{(446811-377934)^2}{24}} = 14060 \tag{14}
$$

i.e. $\sigma$ (V)=14060/412372= 3.4%

The last step is to valuate the investment using real option pricing model. We set the other parameters required by Black-Scholes formula as the same as the data provided by Benaroch and Kauffman, e.g. T = 3, X = \$400,000, r = 7%.

$$FROV = \mathrm{V}N(d_1) - Xe^{-rT}N(d_2)$$

$$\text{Where} \quad \mathrm{d}_1 = \frac{In(412372/400000) + 3(0.07 + 0.034^2/2)}{0.034\sqrt{3}} = 4.113 \tag{15}$$

$$d_2 = d_1 - \sigma\sqrt{T} = 4.054$$

Thus, we can calculated that the fuzzy value of the real option is $FROV$ = ($53,701, $88,138, $122,577)

## 5 Concluding Remarks

The information technology investment are characterized by highly uncertainty, thus has imposed pressure on management to take into account the risks and payoffs in making their investment decision. Real option analysis is a tool well suited to evaluate the investment in uncertain environment. However, several minor limitations of ROA has prevented its application in practice, even could lead to incorrect valuation.

This paper developed a comprehensive but easy-to-use methodology based on fuzzy group decision-making theory to solve the complicated evaluation problem of IT investment. Fuzzy sets theory is not only able to well formulate the uncertainty of the expected payoffs from an investment, but also simplify the real option model in certain degree. Besides, option value calculation is sensitive to parameters. It is therefore necessary that multiple stakeholders be involved in the estimation of real option value [10]. By utilizing a simple and intuitively appealing eigenvector based method, we can intrinsically determine the weights for group members using their own subjective opinions, which is how the model avoid serious bias from particular evaluator's personal preference. Then it in turn provides a basis for a better evaluating and justifying of the target IT investment, and avoid complex estimation task at the same time. We are confident that this method is valuable to help IT managers produce a more well-structured and unbiased assessment.

**Acknowledgments.** This project is supported by National Natural Science Foundation of China (No. 70571025) and China Postdoctoral Science Foundation (No. 20060400103).

## References

1. Margrabe, W.: The value of an option to exchange one asset for another. Journal of Finance **33** (1978) 177-186
2. Panayi, S., Trigeorgis, L.: Multi-Stage real options: the cases of information technology infrastructure and international bank expansion. Quarterly Review of Economics and Finance **38** (1998) 675-692
3. Benaroch, M., Kauffman, R.J.: Justifying electronic banking network expansion using real options analysis. MIS Quarterly **24** (2000) 197–225
4. Benaroch, M.: Managing Information Technology Investment Risk: A Real Options Perspective. Journal of Management Information Systems **19** (2002) 43-84

5. Taudes, A., Feurstein, M., Mild, A.: Options analysis of software platform decisions: a case study. MIS Quarterly **24** (2000) 227– 243
6. Carlsson, C., Fuller, R.: On possibilistic mean value and variance of fuzzy numbers. Fuzzy Sets and Systems **122** (2001) 315-326
7. Cook, W.D.: Distance-based and ad hoc consensus models in ordinal preference ranking. European Journal of Operational Research **172** (2006) 369-385
8. Ramanathan, G.: Group preference aggregation methods employed in AHP: An evaluation and an intrinsic process for deriving members' weightages. European Journal of Operational Research **79** (1994) 249-265
9. Carlsson, C., Fuller, R.: A fuzzy approach to real option valuation. Fuzzy sets and systems **139** (2003) 297-312
10. Kumar, R.L.: Managing risk in IT project: an options perspective. Information and Management **40** (2002) 63-74

# Fuzzifying Closure Systems and Fuzzy Lattices

Branimir Šešelja and Andreja Tepavčević*

Department of Mathematics and Informatics
Faculty of Sciences, University of Novi Sad
Trg D. Obradovića 4, 21000 Novi Sad, Serbia
seselja@im.ns.ac.yu, etepavce@eunet.yu

**Abstract.** A fuzzifying closure system is introduced as a fuzzy set on the collection of subsets of a nonempty set. It is proved that this structure is a particular fuzzy lattice ordered poset. Conversely, every lattice ordered poset is isomorphic to a fuzzifying closure system. In particular, each complete fuzzy lattice is representable by a fuzzifying closure system.

**Keywords:** fuzzifying closure system, fuzzy lattice, fuzzy complete lattice, lattice ordered fuzzy poset.

## 1 Introduction

Fuzzy closure systems and related concepts have been investigated by many authors. These notions appear within the framework of various fields: fuzzy topology, fuzzified relational structures, fuzzy algebras etc. As it is the case with other set-theoretic notions, fuzzy generalizations of closure systems differ. We mention some papers in which these are investigated as lattice valued structures. The list of papers is not at all complete, still each of these papers is relevant in some sense to our approach.

Back in 1988, Achache [1] fuzzified a crisp closure space by means of Galois connections, using a complete lattice as the co-domain of fuzzy sets. In 1991, Swamy and Raju ([2]) introduced $L$-valued closure systems, $L$ being a complete Brouwerian lattice. In many recent approaches, $L$-fuzzy closure systems are investigated as particular fuzzy structures whose co-domain $L$ is a complete residuated lattice. Namely, in a series of papers (starting with [3]) Ying introduced and developed the framework of fuzzified topology. Gerla ([4,5]) and Bělohlávek ([6] and his book [7]) deal with fuzzy closure operators, considering fuzzy structures as mappings whose co-domain is a complete (residuated) lattice.

In the foregoing investigations, fuzzy closure systems and related structures are mostly considered to be collections of fuzzy sets, satisfying particular properties.

Our present investigation is focused on a different fuzzification of the notion of a closure system. Intuitively, we fuzzify the degree of closedness of a subset

* This research was partially supported by Serbian Ministry of Science and Environment, Grant No. 144011 and by the Provincial Secretariat for Science and Technological Development, Autonomous Province of Vojvodina, grant "Lattice methods and applications".

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 111–118, 2007.

belonging to a family. The framework are lattice valued structures, i.e., the codomains of mappings considered here are complete lattices. Our approach is cutworthy, in the sense that the cut sets of introduced structures are supposed to possess crisp properties being fuzzified. Therefore, we use complete lattices, since arbitrary infima and suprima support this cutworthy approach. We do not use additional operations (like those in e.g. residuated lattices), simply because they do not guarantee this transfer of properties to cuts.

We introduce a fuzzifying closure systems as a lattice valued (fuzzy) set defined on the crisp power set or crisp closure system. We prove that cut sets of a fuzzifying closure system are crisp closure systems on the same set. We also show that fuzzifying closure systems are so called fuzzy lattice-ordered posets. Conversely, we prove that every lattice ordered fuzzy poset is isomorphic to a particular fuzzifying closure system. Our construction, as well as the obtained structures and properties, are analogue to the corresponding crisp ones[1].

## 2 Preliminaries

Throughout the paper, $(L, \wedge, \vee, \leq)$ is a complete lattice, denoted usually by $L$, in which the top and the bottom element are 1 and 0 respectively (sometimes denoted by $1_L$ and $0_L$). All claims are valid also in a particular case when $L$ is the real interval $[0, 1]$.

If $x$ is an element of the complete lattice $L$, then $\downarrow x$ is the **principal ideal generated** by $x$:

$$\downarrow x := \{y \in L \mid y \leq x\} \ . \tag{1}$$

Recall that a **closure system** $\mathfrak{S}$ on a nonempty set $S$ is a collection of subsets of $S$, closed under arbitrary (hence also empty) set intersections.

We use the following known results.

**Lemma 1.** (*i*) *Every closure system is a complete lattice under the set inclusion.*
(*ii*) *Every complete lattice $M$ is isomorphic to a closure system $\mathfrak{S}_M$ consisting of all principal ideals generated by elements of $M$.*

A **fuzzy set** on a nonempty set $S$ is a mapping $\mu : S \to L$. A **cut set** (briefly **cut**), of a fuzzy set $\mu$ on $S$ is defined for every $p \in L$, as the subset $\mu_p$ of $S$, such that $x \in \mu_p$ if and only if $\mu(x) \geq p$ in $L$.

As defined in [8], a classical (crisp) property or a notion which is generalized to fuzzy structures is said to be **cutworthy** if it is preserved by all cuts of the fuzzified structure.

Here we also use the notion of a fuzzy lattice as a fuzzy algebra. If $M$ is a lattice and $L$ a complete lattice, then $\mu : M \to L$ is a **fuzzy lattice** on $M$ (**fuzzy sublattice of** $M$) if for al $x, y \in L$ we have that $\mu(x \wedge y) \geq \mu(x) \wedge \mu(y)$ and $\mu(x \vee y) \geq \mu(x) \wedge \mu(y)$. Operations $\wedge$ and $\vee$ on the left are respectively meet and join in $M$, and on the right by $\wedge$ is denoted meet in $L$; obviously, relation

[1] A third notion connected with lattices and closure systems are closure operators, which are not mentioned in the present article, due to the paper page limit.

$\geq$ is the order in $L$. The notion of a fuzzy lattice as a particular fuzzy ordered set and the connection among algebraic and relational approach to fuzzy lattices are investigated in [9]. In the next section we introduce the notion of a fuzzy lattice ordered poset, as a fuzzy generalization of a complete lattice. We recall that the notion of fuzzy lattice (more general, fuzzy (sub)algebra) is cutworthy: *every cut set of a fuzzy sublattice of $M$ is a crisp sublattice of $M$.*

We also use the notion of a **complete fuzzy lattice**, which we define to be a mapping $\mu$ from a complete lattice $M$ into another complete lattice $L$, so that for every family $\{x_i \mid i \in I\} \subseteq M$, the following holds:

$$\mu(\bigwedge\{x_i \mid i \in I\}) \geq \bigwedge_{i\in I} \mu(x_i) \text{ and } \mu(\bigvee\{x_i \mid i \in I\}) \geq \bigwedge_{i\in I} \mu(x_i).$$

In addition, since $\bigwedge \emptyset = 1$, we have that $\mu(1) \geq 1$, and hence $\mu(1) = 1$ in any complete fuzzy lattice.

As for other fuzzified crisp structures, $\mu$ can be called a complete fuzzy *sublattice of $M$*.

We conclude with the following fuzzy topological notion, introduced in [3] for real-interval valued fuzzy sets. Let $S$ be a nonempty set and $\mathcal{T}$ a mapping from the power set $\mathcal{P}(S)$ into a complete lattice $L$. Then $\mathcal{T}$ is said to be a **fuzzifying topology** if the following conditions are fulfilled:

$(i)$ $\mathcal{T}(S) = 1$;

$(ii)$ for any $X_1, X_2 \in \mathcal{P}(S)$, $\mathcal{T}(X_1 \cap X_2) \geq \mathcal{T}(X_1) \wedge \mathcal{T}(X_2)$;

$(iii)$ for a family $\{X_i \mid i \in I\} \subseteq \mathcal{P}(S)$, $\mathcal{T}(\bigcup\{X_i \mid i \in I\}) \geq \bigwedge_{i\in I} \mathcal{T}(X_i)$.

The fuzzifying topology is a special case of the smooth topology [10], since a smooth topology is a mapping from the fuzzy power set to a complete lattice (here we have a mapping from the crisp power set to a complete lattice).

## 3 Results

Let $S$ be a nonempty set and $L$ a complete lattice.

A **fuzzifying closure system** on $S$ is a mapping $\mathcal{C} : \mathcal{P}(S) \longrightarrow L$, such that for every family $\{X_i \mid i \in I\}$ of subsets of $S$,

$$\bigwedge_{i\in I} \mathcal{C}(X_i) \leq \mathcal{C}(\bigcap_{i\in I} X_i) \ . \tag{2}$$

**Lemma 2.** *$\mathcal{C}(S) = 1$, where $1$ is the top element of $L$.*

*Proof.* Indeed, when we consider the empty family, the infimum on the left side is infimum of empty family and it is equal to the top element of the lattice. On the other hand, the intersection of the empty family on the right is equal to $S$, so we obtain $1 \leq \mathcal{C}(S)$.

If $\mathfrak{S}$ is a crisp closure system on a set $S$, and $L$ a complete lattice, then a **fuzzifying closure subsystem** of $\mathfrak{S}$ is a mapping $\sigma$ from $\mathfrak{S}$ to $L$, such that for every family $\{X_i \mid i \in I\}$ of subsets from $\mathfrak{S}$,

$$\bigwedge_{i\in I} \sigma(X_i) \leq \sigma(\bigcap_{i\in I} X_i) \ . \tag{3}$$

As in Lemma 2, we can prove that for a fuzzyfying closure subsystem $\sigma$ of $\mathfrak{S}$, we have $\sigma(S) = 1$.

Obviously, a fuzzifying closure system on $S$ is a fuzzifying closure subsystem of $\mathcal{P}(S)$.

**Theorem 1.** *A mapping $\mathcal{C} : \mathcal{P}(S) \longrightarrow L$ is a fuzzifying closure system if and only if for every $p \in L$, the cut $\mathcal{C}_p$ is a crisp closure system on $S$.*

*Proof.* Let $\mathcal{C} : \mathcal{P}(S) \longrightarrow L$ be a fuzzifying closure system. Let $p \in P$. We have to prove that the cut $\mathcal{C}_p$ is a crisp closure system. Since $\mathcal{C}(S) = 1$, we have that $S \in \mathcal{C}_p$ for every $p$. Further, let $\{X_i \mid i \in I\}$ be a nonempty family of subsets belonging to $\mathcal{C}_p$, i.e., $\mathcal{C}(X_i) \geq p$ for each $i \in I$. Hence, $p \leq \bigwedge_{i\in I} \mathcal{C}(X_i) \leq \mathcal{C}(\bigcap_{i\in I} X_i)$. Therefore, $\mathcal{C}(\bigcap_{i\in I} X_i) \geq p$, and thus $\bigcap_{i\in I} X_i \in \mathcal{C}_p$.

On the other hand, suppose that all cuts of the fuzzy set $\mathcal{C} : \mathcal{P}(S) \longrightarrow L$ are crisp closure systems. Therefore, $S$ belongs to all cuts, in particular to 1-cut, and thus $\mathcal{C}(S) = 1$.

Let $\{X_i \mid i \in I\}$ be a nonempty family of subsets of $S$. Denote $\bigwedge_{i\in I} \mathcal{C}(X_i)$ by $p$. Since $\mathcal{C}(X_i) \geq p$ for all $i \in I$, we have that $X_i \in \mathcal{C}_p$ for all $i \in I$. Since $\mathcal{C}_p$ is a crisp closure system, it is closed under intersection, and thus $\bigcap_{i\in I} X_i \in \mathcal{C}_p$. Hence $\mathcal{C}(\bigcap_{i\in I} X_i) \geq p = \bigwedge_{i\in I} \mathcal{C}(X_i)$ and the theorem is proved.

It is straightforward that the same property holds for fuzzifying closure subsystems of an arbitrary closure system on $S$, as follows.

**Theorem 2.** *Let $\mathfrak{S}$ be a crisp closure system on $S$. Then a mapping $\sigma : \mathfrak{S} \longrightarrow L$ is a fuzzifying closure subsystem of $\mathfrak{S}$ if and only if for every $p \in L$, the cut $\mathcal{C}_p$ is a crisp closure subsystem of $\mathfrak{S}$.*

*Example 1.* Let $L$ be a lattice in Figure 1, and $S = \{a, b, c\}$.

The mapping

$$\mathcal{C} = \begin{pmatrix} \emptyset & \{a\} & \{b\} & \{c\} & \{a,b\} & \{a,c\} & \{b,c\} & \{a,b,c\} \\ 1 & t & s & t & p & p & q & 1 \end{pmatrix} \tag{4}$$

is a fuzzifying closure system on $S$ by Theorem 1, since its cuts are crisp closure systems on the same set:

$\mathcal{C}_\mathbf{1} = \{\emptyset, \{a, b, c\}\}$
$\mathcal{C}_p = \{\emptyset, \{a\}, \{c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}$
$\mathcal{C}_q = \{\emptyset, \{b\}, \{b, c\}, \{a, b, c\}\}$
$\mathcal{C}_r = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b, c\}\}$
$\mathcal{C}_s = \{\emptyset, \{b\}, \{a, b, c\}\}$

$\mathcal{C}_t = \{\emptyset, \{a\}, \{c\}, \{a, b, c\}\}$
$\mathcal{C}_\mathbf{0} = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}.$

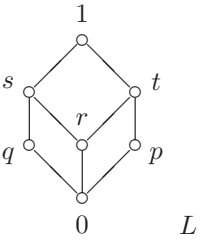



**Fig. 1.**

The following is the *Synthesis theorem* for fuzzifying closure systems. The proof is similar to the Theorem of synthesis of any fuzzy structure (fuzzy set, fuzzy (sub)group, fuzzy equivalence relation etc., see e.g., [11,12]) by crisp structures of the domain; we present it for the readers convenience.

**Theorem 3.** *Let $\mathcal{F}$ be a family of closure systems on the same set $S$ which is closed under intersections, and such that $\mathcal{P}(S) \in \mathcal{F}$.*

*Then, there is a lattice $L$ and a fuzzifying closure system $\mathcal{C} : \mathcal{P}(S) \to L$, such that the family $\mathcal{F}$ is the family of cuts of $\mathcal{C}$.*

*Proof.* Let $L$ be the poset $(\mathcal{F}, \leq)$, where $\leq$ denotes the dual of set inclusion. $L$ is a complete lattice, since its dual is by the assumption (it is closed under set intersection and contains the greatest element - $\mathcal{P}(S)$). Now we define $\mathcal{C} : \mathcal{P}(S) \to L$, so that for $X \subseteq S$

$$\mathcal{C}(X) = \bigcap \{g \in \mathcal{F} \mid X \in g\} \ . \tag{5}$$

$\mathcal{C}$ is a fuzzy set on $S$. We prove that its family of cut sets is precisely $\mathcal{F}$. Indeed, let $f \in \mathcal{F}$ and let $\mathcal{F}_f$ be the corresponding cut set of $\mathcal{C}$. Now, for an arbitrary $Y \subseteq S$, we have

$Y \in \mathcal{F}_f$ if and only if $\mathcal{C}(Y) \geq f$ if and only if $\mathcal{C}(Y) \subseteq f$ if and only if $\bigcap \{g \in \mathcal{F} \mid Y \in g\} \subseteq f$ if and only if $Y \in f$, proving that the cut $\mathcal{F}_f$ is equal to $f$. Hence the family of cut set of $\mathcal{C}$ is $\mathcal{F}$. Therefore, the fuzzy set $\mathcal{C}$ is a fuzzyfying closure system by Theorem 1, since all its cuts are crisp closure systems.

In an analogue way (using the same construction) one can prove the following *Synthesis theorem* for a fuzzifying closure subsystem of a crisp closure system $\mathfrak{S}$. The proof is (being similar to the previous one) omitted. Observe that a closure system $\mathfrak{S}_1$ on $S$ is a **subsystem** of a closure system $\mathfrak{S}_2$ on $S$ if $\mathfrak{S}_1 \subseteq \mathfrak{S}_2$.

**Theorem 4.** *Let $\mathfrak{S}$ be a closure system on a nonempty set $S$ and $\mathcal{F}$ a family of its subsystems which is closed under intersections, and such that $\mathfrak{S} \in \mathcal{F}$.*

*Then, there is a lattice $L$ and a fuzzifying closure system $\mathcal{C} : \mathfrak{S} \to L$, such that $\mathcal{F}$ is the family of cuts of $\mathcal{C}$.*

Next we investigate a connection between fuzzifying closure systems and fuzzy lattices.

We advance the following definition.

Let $(M, \leq)$ be a lattice, $(L, \leq)$ a complete lattices and

$$\mu : M \to L$$

a map such that for every $p \in L$ the cut set $\mu_p$ is a lattice under the order inherited from $M$. Then we say that $\mu$ is a **lattice-ordered fuzzy subposet** of $M$. If the lattices $M$ and $M_p$, for every $p \in L$ are all complete, then the lattice-ordered fuzzy subposet $\mu$ is said to be **complete**.

Observe that any fuzzy sublattice of a lattice $M$ (as defined in Preliminaries) is a lattice-ordered fuzzy subposet of $M$. In addition, the above notion is defined in purely set theoretic terms, like a lattice as an ordered set in the crisp case. Consequently, we do not require cut sets to be sublattices of $\mu$ (a sublattice is an algebraic notion, not a set-theoretic one). Therefore, a lattice-ordered fuzzy subposet is not generally a fuzzy sublattice of the same crisp lattice.

**Theorem 5.** *Let $S$ be a nonempty set, $\mathfrak{S}$ a closure system on $S$ and $(L, \leq)$ a complete lattice. Then the following are satisfied.*

*(i) Any* ***fuzzifying closure system*** *$\mathcal{C} : \mathcal{P}(S) \longrightarrow L$ on $S$ is a complete lattice-ordered fuzzy subposet of the Boolean lattice $(P(S), \subseteq)$.*

*(ii) Any* ***fuzzifying closure subsystem*** *$\sigma : \mathfrak{S} \to L$ of $\mathfrak{S}$ is a complete lattice-ordered fuzzy subposet of the lattice $(\mathfrak{S}, \subseteq)$.*

*Proof.* This is a direct consequence of Theorems 1 and 2, since every crisp closure system is a complete lattice under set inclusion.

It is easy to check that every cut of the fuzzifying closure system $\mathcal{C}$ in Example 1 is a lattice. Hence, $(\mathcal{C}, \subseteq)$ is a lattice-ordered fuzzy subposet of the Boolean lattice $\mathcal{P}(S)$. Still $\mathcal{C}$ is not its fuzzy sublattice, since not all cut lattices are crisp sublattices of $\mathcal{P}(S)$.

Conditions under which a closure system is a fuzzy lattice are given in the following theorem. The proof is straightforward, by the definitions of a complete fuzzy lattice and a fuzzifying topology.

**Proposition 1.** *A fuzzifying closure system $\mathcal{C} : \mathcal{P}(A) \longrightarrow L$ is a fuzzy complete lattice if and only if it is a fuzzifying topology.*

Next we define the notion of isomorphism among introduced fuzzy ordered structures.

Let $(M, \leq)$, $(N, \leq)$ and $(L, \leq)$ be lattices, $L$ being complete. Let also $\mu : M \to L$ and $\nu : N \to L$ be lattice-ordered fuzzy subposets of $M$ and $N$ respectively. Then we say that $\mu$ and $\nu$ are **isomorphic** if there is an isomorphism $f$ from $M$ onto $N$, such that for every $p \in L$ the restriction of $f$ to the cut lattice $\mu_p$ is an isomorphism from $\mu_p$ onto $\nu_p$.

Since a fuzzy lattice is a particular lattice-ordered fuzzy poset, the above notion of isomorphism includes also the case of fuzzy lattices.

In the following, if $f$ is a function from $A$ to $B$, and $C \subseteq A$, then the restriction of $f$ to $C$ is denoted by $f|_C$.

**Theorem 6.** *Let $(M, \leq)$, $(L, \leq)$ be complete lattices and $\mathfrak{S}_M$ the closure system consisting of all principal ideals generated by elements of $M$ (as in Lemma 1). Further, let $\mu : M \to L$ be an $L$–fuzzy complete lattice. Then the mapping $\mathcal{C} : \mathfrak{S}_M \to L$, defined by*

$$\mathcal{C}(\downarrow x) := \mu(x), \text{ for every } x \in M,$$

*is a fuzzifying closure subsystem of $\mathfrak{S}_M$.*

*In addition, $\mathcal{C}$ is isomorphic with the fuzzy lattice $\mu$.*

*Proof.* The mapping $\mathcal{C}$ is well defined, since from $\downarrow x = \downarrow y$ it follows that $x = y$.

Let $\{X_i \mid i \in I\}$ be a family of elements of $\mathfrak{S}_M$. Obviously, $X_i = \downarrow x_i$, for a family $\{x_i \mid i \in I\} \subseteq M$. Then,

$\bigwedge_{i\in I} \mathcal{C}(X_i) = \bigwedge_{i\in I} \mu(x_i) \leq \mu(\bigwedge_{i\in I} x_i) = \mathcal{C}(\bigcap_{i\in I} X_i)$,
by the fact that $\bigcap_{i\in I} \downarrow x_i = \downarrow(\bigwedge_{i\in I} x_i)$.

Also, in a fuzzy complete lattice, since $\mu(1_M) = 1_L$, we have that $\mathcal{C}(\downarrow 1_M) = \mu(1_M) = \mathcal{C}(M) = 1_L$.

To prove the second part, observe that the mapping $f : x \mapsto \downarrow x$, $x \in M$, is an isomorphism from the lattice $M$ onto the closure system $\mathfrak{S}_M$, which is a lattice under inclusion. For $p \in L$, the restriction of $f$ to the cut lattice $\mu_p$ is an isomorphism from $\mu_p$ onto the cut $\mathcal{C}_p$. Indeed, if $x, y \in \mu_p$, $x \neq y$, then $\downarrow x \neq \downarrow y$, hence $f|_{\mu_p}$ is an injection. It is obviously surjective, since $\downarrow x \in \mathcal{C}_p$ implies $\mathcal{C}(x) = \mu(x) \geq p$, and thus we have that $x \in \mu_p$, proving that $f|_{\mu_p}$ is a surjection. This restriction is obviously compatible with the order in both directions, hence it is an isomorphism.

Theorem 6 directly implies the following **Representation Theorem for fuzzy lattices**.

**Corollary 1.** *For every complete fuzzy lattice $\mu$ there is a closure system $\mathfrak{S}$ such that $\mu$ is isomorphic to a fuzzifying closure subsystem of $\mathfrak{S}$.*

## 4 Conclusion

Let us conclude with some comments concerning future investigations in this field.

As mentioned above, our approach is cutworthy and therefore the co-domains of considered fuzzy structures are complete lattices. Due to widely used $T$-norms in applications of fuzzy systems, it would be important to investigate analogously defined structures as mappings from a power set to a residuated lattice. In addition, our approach could be generalized to systems connected with smooth topologies ([10]), with both mentioned lattices as co-domains.

This investigation would be our task in the future.

## References


1. Achache, A.: How to Fuzify a Closure Space, Journal of Mathematical Analysis and Applications Vol.130 No.**2** (1988) 538-544.
2. Swamy, U.M., Raju, D.V.: Algebraic fuzzy systems, Fuzzy Sets and Systems **41** (1991) 187-194.
3. Ying, M.: A new approach for fuzzy topology, Fuzzy Sets and Systems **39** (1991) 303-321.
4. Biancino, L., Gerla, G.: Closure systems and $L$-subalgebras, Inform. Sci. **33** (1984) 181-195.
5. Gerla, G.: Mathematical Tools for Approximate Reasoning, Kluwer Academic Publishers, Dordrecht 2001.
6. Bělohlávek, R.: Fuzzy Closure operators, Journal of Mathematical Analysis and Applications **262** (2001) 473-489.
7. Bělohlávek, R.: Fuzzy Relational System, Kluwer Academic Publishers, Dordrecht 2002.
8. Klir, G., Yuan B.: Fuzzy sets and fuzzy logic, Prentice Hall P T R, New Jersey, 1995.
9. Tepavčević, A., Trajkovski, G.: $L$-fuzzy lattices: an introduction, Fuzzy Sets and Systems **123** (2001) 209-216.
10. Ramadan, A. A., Smooth topological spaces, Fuzzy Sets and Systems **48** (1992) 371 - 375.
11. Šeselja, B.: Tepavčević, A., Completion of Ordered Structures by Cuts of Fuzzy Sets, An Overview, Fuzzy Sets and Systems **136** (2003) 1-19.
12. Šeselja, B.: Tepavčević, A., Representing Ordered Structures by Fuzzy Sets, An Overview, Fuzzy Sets and Systems **136** (2003) 21-39.

# Evolution of Fuzzy System Models: An Overview and New Directions

Aslı Çelikyılmaz[1] and I. Burhan Türkşen[1,2]

[1] Dept. of Mechanical and Industrial Engineering, University of Toronto, Canada
asli.celikyilmaz@utoronto.ca, turksen@mie.utoronto.ca
[2] Dept. of Industrial Engineering TOBB-Economics and Technology University, Turkey
bturksen@etu.edu.tr

**Abstract.** Fuzzy System Models (FSM), as one of the constituents of soft computing methods, are used for mining implicit or unknown knowledge by approximating systems using fuzzy set theory. The undeniable merit of FSM is its inherent ability of dealing with uncertain, imprecise, and incomplete data and still being able to make powerful inferences. This paper provides an overview of FSM techniques with an emphasis on new approaches on improving the prediction performances of system models. A short introduction to soft computing methods is provided and new improvements in FSMs, namely, Improved Fuzzy Functions (IFF) approaches is reviewed. IFF techniques are an alternate representation and reasoning schema to Fuzzy Rule Base (FRB) approaches. Advantages of the new improvements are discussed.

**Keywords:** Fuzzy systems, soft computing, data mining, knowledge discovery.

## 1 Introduction

Knowledge discovery (KD) is commonly viewed as the general process of discovering valid, novel, understandable, and useful knowledge about a system domain from empirical data and background knowledge where the discovered knowledge is implicit or previously unknown. Data mining is the most important step in KD, which involves the application of data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns over the given data.

Zadeh's pioneering work [20] in mid 1960s, in which he introduced fuzzy sets to replace non-probabilistic uncertainties, opened a new window in the area of KD. Since then, fuzzy sets and logic have started to be employed in system modeling applications. In recent years, there have been major developments in fuzzy system modeling (FSM) approaches, which have been applied to solve many different scientific and engineering problems. FSMs have been the main constituents of the *soft computing* (SC) methods along with neural networks, probabilistic modeling, evolutionary computing such as genetic algorithms, and learning theory [7]. Soft computing, unlike hard computing, treats the concepts such as imprecision, uncertainty, partial truth, and approximation. In order to achieve robustness,

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 119–126, 2007.

interpretability, and low cost solutions, soft computing methods exploit the tolerance for such concepts.

This paper briefly overviews some of the well-known soft computing techniques including fuzzy logic, neural networks, and genetic algorithms. Then, FSM models are reviewed with an emphasis on the new developments, their advantages, and possible drawbacks. Finally conclusions are drawn.

## 2 Soft Computing for Knowledge Discovery

SC, unlike classical methods, is an area of computing with imprecision, uncertainty and partial truth, which can achieve robustness and low cost solutions. Among many fields of soft computing, fuzzy logic, neural networks, and genetic algorithms of evolutionary computing techniques are the top three commonly preferred methods in scientific research. Fuzzy systems [7],[13],[15], as the central component of the soft computing methods which implement fuzzy sets and logic theory, provide methodology to capture valid patterns from the data. They also allow integration of expert knowledge during knowledge discovery. From this point of view, FSM models are considered as gray box approaches. In traditional rule-base formulation, a system is represented with sequence of rules that describe the relationships between the input and output variables, which are expressed as a collection if-then rules that utilize linguistic labels, i.e., fuzzy sets. The general FRB structure is represented as:

$$R_i : \text{IF } antecedents_i \text{ THEN } consequents_i. \quad (1)$$

where $i$ represents each fuzzy rule, $R$. To derive conclusions from a fuzzy rule set an inference procedure must be determined with approximate reasoning [21].

Neural networks (NN) [10], on the other hand, are popularly known as black-box function approximates. They allow a system to learn from examples through the process of training. The learning process in a NN implies the adjustment of weights of the network in an attempt to minimize an error function. Generic algorithms (GA), [5], most widely known evolutionary algorithms, are theoretically and empirically proven to provide the means for efficient search and optimization in complex space. GA has the power of finding the global minim of the defined objective function. Support Vector Machines [18] have recently been accepted as soft computing methods by some researchers.

Recent research emphasizes the capabilities of fuzzy function approximations by training with neural networks as a good decision support tool [8],[10]. Utilization of genetic algorithms in evolutionary fuzzy systems has been a powerful global optimization tool due to its success in several problem domains [7]. Due to the complementarities of neural networks, fuzzy systems, and evolutionary computation, the recent trend is to fuse various systems to form a more powerful integrated system, to overcome their individual weaknesses [22], e.g. hybridization (Fig. 2).

The rest of the paper briefly reviews our recent FSM approaches based on Fuzzy Function (FF)s for structure identification of fuzzy models and reasoning with them [2],[3],[16],[17]. The proposed FFs approach is a new branch of FSM, which is an alternate modeling to hybridization.

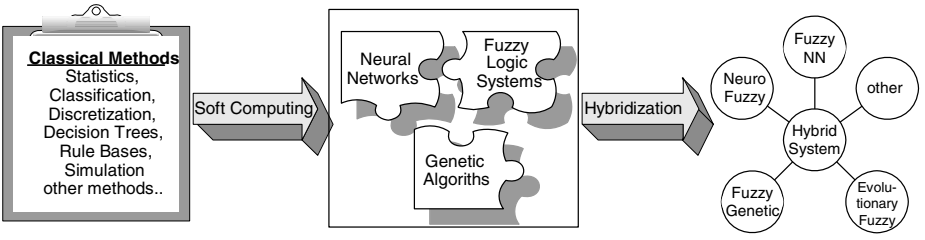



**Fig. 1.** Evolution in Data Mining Techniques of Knowledge Discovery

## 2 A New Fuzzy System Model Approach

Traditional FSMs are rule-based systems that use fuzzy logic, rather than Boolean logic, to reason with expert knowledge and available data. Based on the Zadeh's fuzzy logic foundations [21], FSM with fuzzy rule bases has been the most prominent method to date to simulate the actual systems of various domains, e.g., engineering, business, financial problems, etc.

Traditional FSMs are knowledge based system models structured with linguistic if-then rules, which can be constructed using either an expert knowledge of the system that we want to model or system data acquired through experimentation [7]. FSM family is classified mainly upon structure of the antecedent (*if*) and consequent (*then*) part of the rules, known as "Fuzzy Rule Bases (FRB)". They can be roughly categorized into three parts:

(i) linguistic fuzzy models, where both parts are linguistic variables, e.g., [14],[21],

(ii) fuzzy relational models, where the mapping from antecedent fuzzy sets, $A_i$, to the consequent fuzzy sets $B_i$ is represented with a fuzzy relation, e.g., [12],

(iii) Takagi-Sugeno models, where the antecedents are fuzzy sets but the consequents are represented with crisp functions such as, i.e., if $x$ is $A_i$, then $y_i=f(x)$., e.g., [15].

Typical traditional FSM development steps include; data preparation, similarity based supervised or unsupervised learning to determine the FRB structure, and fine-tuning of the FRB parameters by minimization of approximation error [7]. There are different methods to identify the fuzzy model structure such as neural learning methods, least squares, inductive learning, or fuzzy clustering (FC). The most commonly used FRB models based on FC is the zero or first order Takagi-Sugeno (TS) models [15]. Recently, we presented a different structure identification method for FSM, called the Fuzzy Functions (FF)s approaches, which do not fall into any of the FSM categories listed above. The FF approaches are new methods for identification of system structure, which are proposed to be alternatives to FRB approaches. They are more easily interpretable and easily understandable by engineers, system modelers, business people, etc. The foundations of the FFs approaches were introduced by Türkşen [16] and Çelikyılmaz & Türkşen [2]. The framework for the generalization of the FSM with FFs is shown in Fig. 3.

FSM with FFs and the traditional FSM approaches based on FRB structures share similar system design steps [17], but they differ in structure identification. FFs do not require most of the learning and inference steps that are needed in the FRB. FFs only

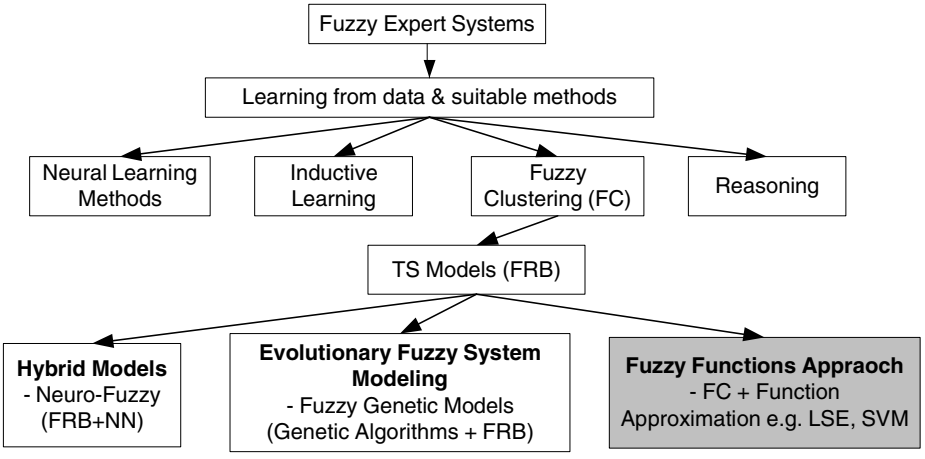



**Fig. 2.** Evolution of Fuzzy System Models

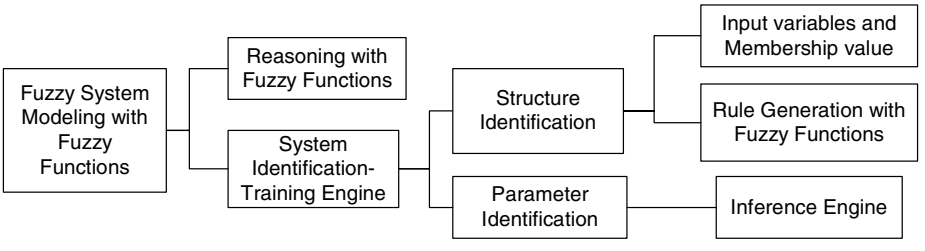



**Fig. 3.** Evolution of Fuzzy System Models

require two well-known learning algorithms, i.e., any type of FC to discover the hidden structures associated with membership values in the given dataset and a function estimation method.

FRB approaches approximate the given non-linear system behavior by local (non)linear models structured with either heuristic expert knowledge or a type of FC method. Thus, essentially the FFs and FRB approaches differ in structure identification process of the system modeling.

## 3 Fuzzy Functions (FFs)

In fuzzy logic theory, the FFs are defined as fuzzy relations, which can map both crisp and approximate values and they allow us to describe relationships between approximate objects approximately [4],[11], e.g. If…THEN rules. The FFs of the new FSM with FFs approach also map the fuzzy data to a non-fuzzy output value. The novelty of the FF approaches is that the membership values of each input data vector from a FC approach e.g., generally standard Fuzzy *c*-Means Clustering (FCM) [1], are used as additional predictors of the system model along with the original input variables to estimate the local relations of the input-output data. For each local model

captured from the FC, we use a linear regression method [16] as simple and interpretable as least squares estimation (LSE), or a non-linear regression method such as non-linear polynomial function. In order to obtain more powerful FFs, while sacrificing the interpretability of the models, we propose using other soft computing algorithms such as neural networks or support vector regression methods [2] with kernel functions, to model local input-output relations. In short, in the FFs methods, [2],[16],[17], there are as many regression methods as the number of clusters identified by the FC approach and the input variables of each regression method include membership values for a given cluster as well as the original scalar input variables. This is equal to mapping the original input space, $\boldsymbol{x}$, of $nv$ dimensions, $\Re^{nv}$, onto a higher dimensional feature space, $\Re^{nv+nm}$, and searching for regression models in this new feature space identified by the number of membership values, $nm$, i.e., $\boldsymbol{x} \rightarrow \Phi_i(\boldsymbol{x},\mu_i)$. Hence, each data vector is represented in $(nv+nm)$ feature space. The decision boundary is sought in this new space. Let $\mu_i$ represent the column vector of membership values of the input vectors, $\boldsymbol{x}$, in $i^{\text{th}}$ cluster. For each cluster $i$, a different dataset is structured by using membership values ($\mu_i$) and/or their transformations as additional dimensions. The structure of the input matrix of the $i^{th}$ cluster for a special case (Fig. 4) using one-dimensional input matrix and only membership values as additional dimensions is:

$$\Phi_i(\boldsymbol{x},\mu_i) = \begin{bmatrix} x_1 & \mu_{i\times 1} \\ \vdots & \vdots \\ x_{nd} & \mu_{i\times nd} \end{bmatrix} \in \Re^{nv+nm} \begin{bmatrix} y_1 \\ \vdots \\ y_{nd} \end{bmatrix} \in \Re \tag{2}$$

where $y$ represents the output variable. The proposed FF approaches are a variation of FFs defined in [4]. Each FF corresponds to a fuzzy rule in any FRB.

During approximation of FFs, one only needs to determine a FC algorithm and a function estimation method. Hence, it can be easily structured and easily understood if simple function estimation methods are used to find the parameters of such functions. Our recent studies indicate that [2],[16],[17], FFs applications, outperform the FRB approaches as well as the standard regression methods.

Some of the major challenges and issues of classical FSMs based on FRBs (*if-then rules)* [16] are: identification of membership functions, identification of most suitable combination operator (*t-norm* or *t-conorm),* choosing between fuzzy or non-fuzzy conjunctions, disjunction and implication operators to capture the uncertainty associated with the linguistic "AND", "OR" and "IMP" for the representation of rules as well as for reasoning with them, defuzzification. Thus, to overcome these challenges, the FFs approaches [2],[16],[17] are developed to estimate regression and classification models, which do not require construction of *if-then* rules. The system modeling with FFs approach consequently requires training and inference engines. During the training step of the system models for the determination of FFs approaches, the data is fuzzy partitioned into $c$ number of fuzzy clusters and one FF is approximated for each fuzzy cluster using membership values as additionaldimensions of the input space. The inference algorithms of the FFs approaches are used to infer about the output values of new data vectors using the optimum model parameters, which are captured during the learning (training)

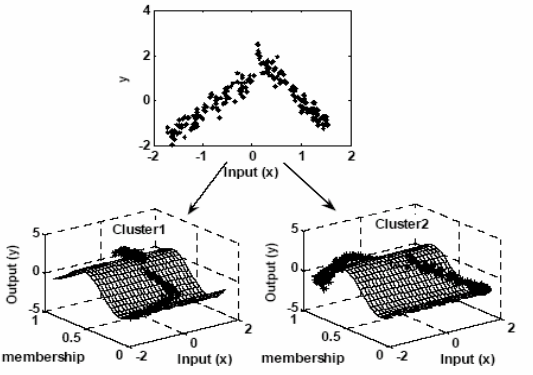



**Fig. 4.** Graphical Representation of the fuzzy functions of each cluster using the data mapped onto each individual cluster using individual membership values for a single input-single output artificial dataset

exercises. Before FFs of each cluster is applied on the new data to estimate their output values in each cluster, the membership values of each new data in each cluster is calculated using the membership function of the FC algorithm that is implemented during the training algorithm of the FSM with FFs approach. To calculate a single output value for each data vector, in a sense defuzzify the output value, the output values from each cluster is weighted with their corresponding membership values and are summed up.

## 4 Improved Fuzzy Clustering with Fuzzy Functions

In the new FSM with FFs approaches the membership values from a FCM clustering method are used as additional predictors [2],[16] in each cluster. In this regard, one might argue that, the FCM [1] algorithm is not designed to find optimal membership values, which are at the same time good predictors for the FF models. In response, we introduced a new FC algorithm, namely the Improved Fuzzy Clustering (IFC) algorithm, [3] which carries out two objectives:

(i) to find good representation of the partition matrix which captures the fuzzy model structure of the given system by identifying the hidden overlapping patterns,
(ii) to find the membership values, which are good predictors of the regression models (FFs) of each cluster.

Therefore the objective function of the new IFC,

$$J_m^{IFC} = \underbrace{\sum_{i=1}^{c}\sum_{k=1}^{nd}\mu_{ik}^m d_{ik}^2}_{FCM-FirstTerm} + \underbrace{\sum_{i=1}^{c}\sum_{k=1}^{nd}\mu_{ik}^m \left(y_k - f_i(\boldsymbol{\tau}_{ik})\right)^2}_{\textbf{Second term- Fuzzy Function}} \quad ,k=1,..,nd,i=1,..,c \tag{3}$$

will be minimized by balancing two terms: (first term) the distance of $k^{th}$ data vector to $i^{th}$ cluster center, $d_{ik}{}^2$, and at the same time, (second term) the residual error, which

is the difference between the actual given output, $y_k$ and the approximated output from the fuzzy functions of each cluster, $\hat{y}_i = f_i(\boldsymbol{\tau}_{ik})$, $i=1,..,c,$ where $c$ represents the total number of clusters. In this sense, the new IFC combines the FC methods and the regression methods within one clustering-optimization model. During the optimization of IFC, the regression models, to be approximated for each cluster, will use only the membership values and their user defined transformations calculated at particular iteration and form the matrix $\tau_i$ of each cluster $i,$ e.g., $\tau_i = [\mu_i \mu_i^2 e^{\mu i}]$, but not the original input variables. Alienating the original input variables and building regression models with membership values will only shape the membership values into candidate inputs during IFC to explain the output variable for each local model.

The convergence of the new IFC depends on how well these membership values are shaped to explain the output variable at each local fuzzy model. These membership values will be candidate input variables for the system modeling with FFs approaches. Therefore, in the proposed IFC, it is hypothesized that the calculated membership values can increase prediction power of the system modeling with FFs.

## 5 Conclusion

This paper briefly presented the evolution of Fuzzy System Models with an emphasis on the recent Fuzzy Functions (FFs) methods and reviewed new improvements. Comparisons to earlier rule base methods and advantages are discussed. The FFs method is a new approach to fuzzy system modeling in which different traditional learning algorithms are combined for system identification, e.g., FCM, or IFC and rule generation e.g., LSE, SVM, NN, etc. FFs are unique structures, which utilize membership values as additional predictors to form approximate relation between the inputs and output at each local model. To improve the approximation accuracy of these FFs, an improved clustering algorithm is used in FSM with FFs approach. Our new research is based on capturing the uncertainties in identification of the IFC parameters including degree of fuzziness ($m$), structure of the fuzzy functions, etc.

## References

1. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: Fuzzy C-Means Algorithm. Computers and Geoscience 10 (1984) 191-203
2. Çelikyılmaz, A., Türkşen, I.B.: Fuzzy Functions with Support Vector Machines. Information Sciences Special Issue (2007) to be published
3. Çelikyılmaz, A., Türkşen, I.B.: A New Fuzzy System Modeling Approach with Improved Fuzzy Clustering Algorithm. IEEE Trans. on Fuzzy Systems (2006) Under review
4. Demirci, M.: Fuzzy functions and their fundamental properties. Fuzzy Sets and Systems 106 (1999) 239-246
5. De Jong, K.A.: Evolutionary Computation: A Unified Approach. The MIT Press (2006)
6. Emami, M.R., Türkşen, I.B. and Goldenberg, A.A.: Development of a Systematic Methodology of Fuzzy Logic Modeling. IEEE Transactions on Fuzzy Systems **63** (1998) 346-361
7. Hellendoorn, H. and Driankov, D.: Fuzzy Model Identification: Selected Approaches. Springer, Berlin, Germany (1997)

8. Jang, J-S.R.: ANFIS: Adaptive Network Based Fuzzy Inference System. IEEE Trans. On System, Man and Cybernetics 23 (1993) 665-685
9. Kecman, V.: Learning and Soft Computing. The MIT Press, Cambridge MA (2001)
10. Kosko, B.: Neural Networks and Fuzzy Systems. Englewood Cliffs, Prentice Hall (1992)
11. Niskanen, V.A.: Soft Computing Methods in Human Sciences: Studies in Fuzziness and Soft Computing. Springer (2004)
12. Pedrycz, W.: Applications of fuzzy relational equations for methods of reasoning in presence of fuzzy data. Fuzzy Sets and Systems 16 (1985) 163-175
13. Pedrycz, W., Lam, P.C.F, Rocha, A.F.: Distributed Fuzzy System Modeling. IEEE Transactions On Systems, Man, and Cybernetics 25 (1995) 769-780
14. Sugeno, M., Yasukawa, T.: A Fuzzy Logic Based Approach to Qualitative Modeling. IEEE Transaction on Fuzzy Systems 1 (1993) 7-31
15. Takagi, T. and Sugeno, M.: Fuzzy Identification of Systems and Its Applications to Modeling and Control. IEEE Transactions on Systems, Man and Cybernetics 15 (1985) 116-132
16. Türkşen, I.B.: Fuzzy Functions with LSE (Accepted for publication) Applied Soft Computing (2007) to appear
17. Türkşen, I.B., Çelikyılmaz, A.: Comparison of Fuzzy Functions with Fuzzy Rule Base Approaches. International Journal of Fuzzy Systems 8 (2006) 137-49
18. Vapnik, V.: Statistical Learning Theory, New York, Wiley (1998)
19. Weiss, S.M., Kulikowski, C.A.: Computer Systems that Learn. Morgan Kaufmann Publishers Inc. (1991)
20. Zadeh, L.A.: Fuzzy Sets. Information and Control 8 (1965) 338-353
21. Zadeh, L.A.: Concept of a Linguistic Variable and Its Application to Approximate Reasoning-I. Information Sciences 8 (1975) 199-249
22. Zadeh, L.A.: Fuzzy Logic, Neural Networks, and Soft Computing. Communications of the ACM 37 (1994) 77-84
23. Zarandi, M.H.F., Türkşen, I.B., Razaee, B.: A systematic approach to fuzzy modeling for rule generation from numerical data. IEEE Annual Meeting of the Fuzzy Information Proceedings NAFIPS '04 (2004) 768-773

# A New Cluster Validity Index for Fuzzy Clustering Based on Similarity Measure

Mohammad Hossein. Fazel Zarandi [1], Elahe Neshat [1], and I. Burhan Türkşen [2]

[1] Department of Industrial Engineering, Amirkabir University of Technology (Polytechnic of Tehran) P.O. Box 15875-4413, Tehran, Iran
[2] Department of Mechanical and Industrial Engineering, University of Toronto 5 King College Road, Toronto, Canada M5S2H8
zarandi@aut.ac.ir, Neshat_elahe@yahoo.com, turksen@mie.utoronto.ca



**Abstract.** In this paper, first, the main problems of some cluster validity indices when they have been applied to Gustafson and Kessel (GK) clustering approach are review. It is shown that most of these cluster validity indices have serious shortcomings to validate Gustafson Kessel algorithm. Then, a new cluster validity index based on a similarity measure of fuzzy clusters for validation of GK algorithm is presented. This new index is not based on a geometric distance and can determine the degree of correlation of the clusters. Finally, the proposed cluster validity index is tested and validated by using five sets of artificially generated data. The results show that the proposed cluster validity index is more efficient and realistic than the former traditional indices.

**Keywords:** Fuzzy cluster analysis, similarity measure, cluster validity index.

## 1 Introduction

The objective of a fuzzy clustering approach is to partition a data set into $c$ homogeneous fuzzy clusters. The most widely used fuzzy clustering algorithm is the fuzzy c-means (FCM) algorithm proposed by Bezdek [1]. The FCM detects clusters having centroid prototypes of a roughly same size. The Gustafson–Kessel (GK) algorithm is an extension of the FCM, which can detect clusters of different orientation and shape in a data set by employing norm-inducing matrix for each cluster [9]. Both FCM and GK algorithms require the number of clusters as an input, and the analysis result can vary greatly depending on the value chosen for this variable. However, in many cases the exact number of the clusters in a data set is not known. Both FCM and GK algorithm may lead to undesired results if a wrong cluster number is given. It is necessary to validate each of the fuzzy $c$-partitions once they are found [2]. This validation is carried out by a cluster validity index, which evaluates each of the fuzzy $c$-partitions and determines the optimal partition or the optimal number of the clusters ($c$). Many validation criteria have been proposed for evaluating fuzzy $c$-partitions [1–14]. In particular, Bezdek's partition coefficient (PC) [3] and partition entropy (PE) [4], and Xie-Beni's index [5] have been frequently used in recent research. Cluster properties such as compactness (or variation) and separation

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 127–135, 2007.

(or isolation) are often considered as major characteristics by which to validate clusters. Compactness is an indicator of the variation or scattering of the data within a cluster and separation is an indicator of the isolation of the clusters from one another. However, conventional approaches to measuring compactness suffer from a tendency to monotonically decrease when the number of the clusters approaches to the number of data points [2, 7]. In addition, conventional separation measures have a limited capacity to differentiate the geometric structures of the clusters because the calculation is based only on the centroid information and does not consider the overall cluster shape.

In the case of the GK algorithm, there are only a few validation indices found in the literature [11,12] with poor performance, and most validation indices proposed for the FCM cannot be applied to the GK clustering directly because they highly depend on the centroid information of the clusters and they do not use the covariance information of the clusters. Most of the validity indices proposed for the FCM [1, 3–8] measure intra-cluster compactness and inter cluster separation using cluster centroids. However, interpretation of inter cluster separation of these indices is problematic because such indices quantify cluster separation based on only the distance between cluster centroids. Thus, they are not appropriate for the clusters found by the GK algorithms which often are in the shape of hyper ellipsoids of different orientation and shapes [20].

This paper presents a new cluster validity index based on a similarity measure. This similarity measure can quantify the degree of overlap between fuzzy clusters by computing an inter-cluster overlap. Here the similarity between fuzzy sets can be defined as the degree to which the fuzzy clusters are more similar.

The organization of the paper is as follows: section 2 reviews the GK algorithm, some cluster validity indices and their main problems when they are applied to validate the GK algorithm. In section 3 we propose a new cluster validity index based on a similarity measure for fuzzy clustering. In section 4, the performance of the new validity index is tested by applying it to 5 data sets and comparing the results with those obtained using traditional validity indices.

## 2 Backgrounds

In this section, we briefly review the GK algorithm and some traditional cluster validity indices.

### 2.1 The Gustafson-Kessel Algorithm

Assume the vector $x_k$ , $k$=1,2, …, $N$, contained in the columns of data matrix $X$, will be partitioned into $c$ clusters, represented by their prototypical vectors $v_i$=[$v_{i,1}$, $v_{i,2}$,…,$v_{i,n}$]$^T$∈ $R^n$ , $i$=1,2,…,$c$. Denote $V$∈$R^{n\times c}$ the matrix having $v_i$ in its $i$-th column. This matrix is called the prototype matrix. The fuzzy partitioning of data among the $c$ clusters is represented as the fuzzy partition matrix $U$∈$R^{n\times c}$ whose element $\mu_{i,k}$∈[0,1] are the membership degree of the data vector $x_k$ in the $i$-th cluster. A class of clustering algorithms searches for the partition matrix and the cluster prototypes such that following objective function is minimized:

$$J(X;V,U)=\sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{i,k})^m d^2(x_k,v_i),$$
$$\text{subjet to}:\quad \sum_{i=1}^{c}(\mu_{i,k})=1,k=1,2,...,N \qquad 0<\sum_{k=1}^{N}(\mu_{i,k})<N,j=1,2,...,c \tag{1}$$

where, $m>1$ is a parameter that controls the fuzziness of the clusters. The function $d(x_k, v_i)$ is the distance of the data vector $x_k$ from the cluster prototype $v_i$.
Gustafson and Kessel (1979) extended the fuzzy $c$-mean algorithm for an inter-product matrix norm:

$$d^2(x_k, v_i) = (x_k - v_i)^T M_i (x_k - v_i) \tag{2}$$

where, $M_i$ is a positive definite matrix adapted according to the actual shapes of the individual clusters, described approximately by the cluster covariance matrices $F_i$ :

$$F_i=\frac{\sum_{k=1}^{N}(\mu_{i,k})^m(x_k-v_i)(x_k-v_i)^T}{\sum_{k=1}^{N}(\mu_{i,k})^m} \tag{3}$$

$$M_i=\det(F_i)^{\frac{1}{n}}F_i^{-1} \tag{4}$$

The objective of GK algorithm is to obtain a fuzzy $c$-partition and its corresponding norm-inducing matrices ($M_i$) by minimizing the evaluation function $J$ [10].

### 2.2 Traditional Cluster Validity Indices

Cluster validity indices are used to establish which partition best explains the unknown cluster structure in a given data set [17]. A fuzzy clustering algorithm is run over a range of $c$ values, 2,...,$c_{max}$ and the resulting fuzzy partition is evaluated with the validity indices to identify the optimal number of the clusters. Bezdek proposed two cluster validity indices for fuzzy clustering [3, 4]. These indices, which are referred to as the Partition Coefficient ($V_{PC}$) and Partition Entropy ($V_{PE}$), are defined as:

$$V_{PC}=\frac{\sum_{k=1}^{n}\sum_{i=1}^{c}\mu_{ik}^2}{n} \tag{5}$$

$$V_{PE}=-\frac{1}{n}\left(\sum_{k=1}^{n}\sum_{i=1}^{c}[\mu_{ik}\ \log_a(\mu_{ik})]\right) \tag{6}$$

The optimal fuzzy partition is obtained by maximizing $V_{PC}$ (or minimizing $V_{PE}$) with respect to $c = 2; \ldots; c_{max}$.

Partition Index $V_{SC}$ which is presented by Bensaid and Hall [19] is the ratio of the sum of compactness and separation of the clusters. It is a sum of individual cluster validity measures normalized through division by the fuzzy cardinality of each cluster:

$$V_{SC}=\sum_{i=1}^{c}\frac{\sum_{j=1}^{n}(\mu_{ij})^m\left\|x_j-v_i\right\|^2}{n_i\sum_{k=1}^{c}\left\|v_k-v_i\right\|^2} \tag{7}$$

$V_{SC}$ is useful when comparing different partitions having equal number of clusters. A lower value of $V_{SC}$ indicates a better partition.

Xie and Beni proposed a validity index ($V_{XB}$) that focuses on two properties: compactness and separation [5]. $V_{XB}$ is defined as:

$$V_{XB} = \frac{\sum_{i=1}^{c}\sum_{k=1}^{n}\mu_{ik}^{2}\left\|x_i - v_k\right\|^{2}}{n(\underbrace{\min}_{i\neq j}\{\left\|v_i - v_j\right\|^{2}\})} \tag{8}$$

In this equation, the numerator is the sum of the compactness of each fuzzy cluster and the denominator is the minimal separation between fuzzy clusters. The optimal fuzzy partition is obtained by minimizing $V_{XB}$ with respect to $c = 2,\ldots,c_{max}$, $V_{XB}$ decreases monotonically as $c \to n$. Kwon extended $V_{XB}$ to eliminate this decreasing trend [7] by adding a penalty value to the numerator of $V_{XB}$. Kwon's index is as follows:

$$V_k = \frac{\sum_{k=1}^{n}\sum_{i=1}^{c}\mu_{ik}^{2}\left\|x_k - v_i\right\|^{2} + \frac{1}{c}\sum_{i=1}^{c}\left\|v_i - \bar{v}\right\|^{2}}{\underbrace{\min}_{i\neq k}(\left\|v_k - v_i\right\|^{2})} \tag{9}$$

Most validity indices shown above focus only on the compactness and the variation of the intra-cluster distance [5–8]. Fuzzy hyper volume and density criteria use the hyper volume to assess the density of the resulting clusters measuring mainly compactness of the given fuzzy partition [13]. Some indices, for example $V_{XB}$ and $V_K$, use the strength of separation between clusters; however, interpretation of these indices is problematic because they quantify cluster separation based only on the distance between cluster centriods. Since the GK clustering involves Mahalanobis distance norm for each cluster, validity indices like $V_{XB}$, $V_K$, cannot discriminate the separation of two different pairs of clusters with different clusters and with different orientation.

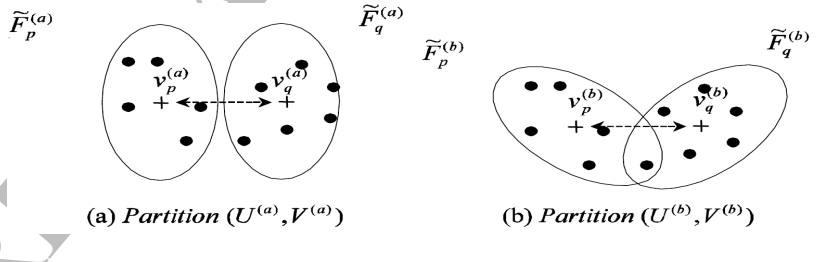



**Fig. 1.** Two different fuzzy partitions ($U^{(a)}$, $V^{(a)}$)and($U^{(b)}$ ,$V^{(b)}$), with the same distance between cluster centroids with different orientations for the same data

In Fig.1 even though the pair ($U^{(a)}$,$V^{(a)}$)provides a better partitioning than the pair ($U^{(b)}$ ,$V^{(b)}$), this cannot be reflected properly in $V_{XB}$, and $V_K$ because they calculate the separation between clusters using only centroid distances.

### 2.3 Similarity

The concept of similarity is interpreted in different ways depending on the context. The interpretation of similarity in everyday language is "having characteristics in common" or "not different in shape, but in size or position". This interpretation of similarity differs from the one we use. We define similarity between fuzzy sets as the degree to which the fuzzy sets are equal. This definition is related to the concepts represented by the fuzzy sets.

Different similarity measures have been proposed for fuzzy sets, and a study of some measures can be found in [15] and [16]. In general, they can be divided into two main groups: 1) geometric similarity measures, 2) set-theoretic similarity measures.

The proposed similarity measure is not based on a geometric distance such as the Minkowski distance, and it provides answers for the following question: at what degree can two clusters (or groups) be co-related?

The similarity measure satisfies the following properties [18]:

**Property1.** $S\left(A_p, A_q\right)$ is the maximum degree of similarity between $A_P$ and $A_q$.

**Property2.** The similarity degree is bounded, $0 \le S\left(A_p, A_q\right) \le 1$

**Property3.** If $A_p$ and $A_q$ are normalized and $A_p$ and $A_q$, $S\left(A_p, A_q\right) = 1$. If $A_p \cap A_q = \phi$, $S\left(A_p, A_q\right) = 0$.

**Property4.** The measure is commutative, i.e. $S(A_p, A_q) = S(A_q, A_p)$

**Property5.** When $A_p$ and $A_q$ are crisp, $S = 0$ if $A_p \cap A_q = \phi$, S = 1 if $A_p \cap A_q \neq \phi$.

Thus, when the similarity between fuzzy clusters $A_p$ and $A_q$ is, for example, $S(A_p, A_q) = 0.4$, then the interpretation is that the clusters $A_p$ and $A_q$ are similar or co-related with a degree of at least 0.4. Conversely, we can say that the two fuzzy cluster sets are unrelated or isolated with a degree of 0.6.

## 3 Cluster Validity Based on Fuzzy Similarity

Let $A_p$ and $A_q$ be two fuzzy clusters belonging to a fuzzy partition $(U, V)$ and $c$ be the number of clusters.

**Definition 1.** The relative similarity between two fuzzy sets $A_p$ and $A_q$ at $x_j$ is defined as:

$$S_{rel}(x_j : A_p, A_q) = \frac{f(x_j : A_p \cap A_q)}{f(x_j : A_p \cap A_q) + f(x_j : A_p - A_q) + f(x_j : A_q - A_p)} \tag{10}$$

In (10): $f\left(x_j : A_p \cap A_q\right) = \mu_{A_p}(x_j) \wedge \mu_{Aq}(x_j)$ where, $\wedge$ is minimum. Moreover:

$$f\left(x_j : \mathrm{A_p} - \mathrm{A_q}\right) = Max(0, \mu_{A_p}(x_j) - \mu_{Aq}(x_j)) \tag{11}$$

and

$$f\left(x_j : \mathrm{A_q} - \mathrm{A_p}\right) = Max(0, \mu_{A_q}(x_j) - \mu_{A_p}(x_j)) \tag{12}$$

**Definition 2.** The relative similarity between two fuzzy sets $A_p$ and $A_q$ is defined as:

$$S_{rel}(A_p, A_q) = \sum_{j=1}^{n} S_{rel}\ (xj : A_p\ ,\ A_q\ )\ h(x_j) \tag{13}$$

where,

$$h(x_j) = -\sum_{p=1}^{c} \mu_{A_p}(x_j) \log(\mu_{A_p}) \tag{14}$$

Here, $h(x_j)$ is the entropy of datum $x_j$ and $\mu_{A_p}(x_j)$ is the membership value with which $x_j$ belongs to the cluster $A_p$. in (13), $h(x_j)$ measures how vaguely (unclearly) the datum $x_j$ is classified over $c$ different clusters. $h(x_j)$ is introduced to assign a weight for vague data. Vague data are given more weight than clearly classified data. $h(x_j)$ also reflects the dependency of $\mu_{A_p}(x_j)$ with respect to different $c$ values. This approach makes it possible to focus more on the highly-overlapped data in the computation of the validity index than other indices do.

**Definition 3.** The proposed validity index is as follows:

$$V_{FNT}(U, V; X) = \frac{2}{c(c-1)} \sum_{p \neq q}^{C} S_{rel}(A_p, A_q) \tag{15}$$

The optimal number of the clusters is obtained by minimizing $V_{FNT}(U, V; X)$ over the range of $c$ values: 2,…,$c_{max}$.

Thus, $V_{FNT}$ is defined as the average value of the relative similarity between $c(c-1)/2$ pairs of clusters, where the relative similarity between each cluster pair is defined as the weighted sum of the relative similarity at $x_j$ between two clusters in the pair. Hence, the less overlap there is in a fuzzy partition, and the less vague the data points in that overlap, the lower the value of $V_{FNT}(U, V; X)$ is resulted.

## 4 Numerical Examples

To test the performance of $V_{FNT}$, we use it to determine the optimal number of clusters in six artificially generated data sets and compared the results with those obtained using $V_{PC}$, $V_{PE}$, $V_{SC}$, $V_{XB}$, $V_K$.

$V_{XB}$, $V_K$ and $V_{SC}$ are modified to accommodate Mahalanobis distance norm instead of Euclidean norm in calculating the distance from each data point to the cluster centers. The parameters of the GK algorithm were set as follows: termination criterion $\varepsilon = 10^{-5}$, weighting exponent m = 2, $c_{max}$= 12 and the initial cluster centers were selected by the FCM. Figures 2–6 show scatter plots of the seven artificially generated data sets used in the experiments.

Table 1 summarizes the optimal cluster numbers identified by each validity index. For example for Data set 5 all validity indices $V_{PC}$, $V_{PE}$, $V_{SC}$, $V_{XB}$ and $V_K$ incorrectly identified the optimal cluster number and only $V_{FNT}$ identified it correctly. This result indicates that the proposed validity index is very reliable.

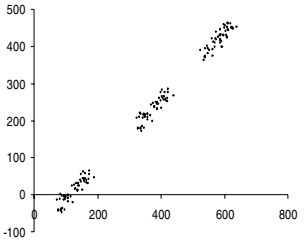



**Fig. 2.** Data set 1 ($c$*= 3.)

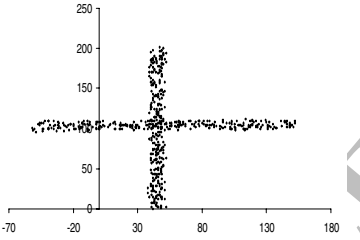



**Fig. 3.** Data set 2 ($c$*=2.)

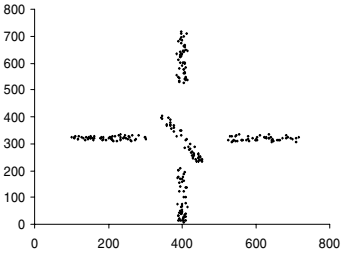



**Fig. 4.** Data set 3 ($c$*=5.)

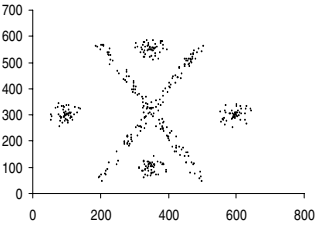



**Fig. 5.** Data set 4 ($c$*=6.)

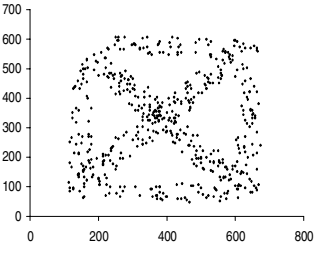



**Fig. 6.** Data set 5 ($c$*=6.)

**Table 1.** Optimal cluster number for 5 data sets

| | $V_{PC}$ | $V_{PE}$ | $V_{SC}$ | $V_{XB}$ | $V_K$ | $V_{FNT}$ | c |
|---|---|---|---|---|---|---|---|
| Data set 1 | 3 | 3 | 8 | 11 | 11 | 3 | **3** |
| Data set 2 | 2 | 2 | 12 | 4 | 4 | 2 | **2** |
| Data set 3 | 5 | 3 | 11 | 10 | 4 | 5 | **5** |
| Data set 4 | 6 | 2 | 10 | 8 | 3 | 6 | **6** |
| Data set 5 | 2 | 2 | 12 | 12 | 3 | 8 | **8** |

## 5 Conclusions

In this paper, the problems of some traditional validity indices when applied to the GK clustering are reviewed. A new cluster validity index for the GK algorithm based on similarity measure is proposed. This validity index is defined as the average value of the relative degrees of sharing of all possible pairs of fuzzy clusters in the system. The optimal number of clusters is obtained by minimizing the validity index. Finally, the performance of the proposed validity index was tested by applying it to 7 data sets and comparing the results with those obtained using several other validity indices. The results indicate that the proposed validity index is reliable.

## References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, (1981).
2. Pal, N.R., Bezdek, J.C.: On cluster validity for the fuzzy c-means model, IEEE Trans. Fuzzy Syst. 3(3) (1995) 370 379.
3. Bezdek, J.C.: Numerical taxonomy with fuzzy sets, J. Math. Biol. 1 (1974) 57–71.
4. Bezdek, J.C.: Cluster validity with fuzzy sets, J. Cybernet. 3 (1974) 58–72.
5. Xie, X.L., Beni, G.: A validity measure for fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 13 (8), (1991) 841– 847.
6. Fukuyama, Y., Sugeno M.: A new method of choosing the number of clusters for the fuzzy c-means method, in: Proc. of the Fifth Fuzzy Systems Symposium, (1989), pp. 247–250.
7. Kwon, S.H.: Cluster validity index for fuzzy clustering, Electron. Lett. 34(22) (1998) 2176–2177.
8. Rezaee, M.R, Lelieveldt, B.P.F., Reiber J.H.C.: A new cluster validity index for the fuzzy c-mean, Pattern Recognition Lett. 19 (1998) 237–246.
9. Boudraa, A.O.: Dynamic estimation of number of clusters in data sets, Electron. Lett. 35(19) (1999) 1606–1607.
10. Gustafson, D., Kessel, W.: Fuzzy clustering with a fuzzy covariance matrix, in: Proc. IEEE CDC, San Diego, CA, USA, (1979), pp. 761–766.
11. Lee-Kwang H., Seong, K.A., Lee K.M.: Hierarchical partition of non structured concurrent systems, IEEE Trans. Systems Man Cybernet. 27(1) (1997) 105–108.
12. Shahin A., Menard, M., Eboueya M.: Cooperation of fuzzy segmentation operators for correction aliasing phenomenon in 3D color doppler imaging, Artif. Intell. 19(2) (2000) 121–154.
13. Baduska, R.: Fuzzy Modeling for Control, Kluwer Academic Publishers, 1998.Y.-I. Kim et al. / Information Sciences 168 (2004) 225–242 241
14. Bezdek, J.C., Pal, N.R.: Some new indexes of cluster validity, IEEE Trans. Systems Man Cybernet. 28(3) (1998) 301–315.
15. Cross,V. V.: An analysis of fuzzy set aggregators and compatibility measures, Ph.D. dissertation, Wright State Univ. Dayton, OH, (1993).
16. Setnes, M.: Fuzzy rule-base simplification using similarity measures, M.Sc. thesis, Dept. Elect. Eng., Contr. Lab., Delft Univ. Technol., July (1995).
17. Dumitrescue, D. Lazzerini, B. Jain, L.C.: Fuzzy Sets and their Application to Clustering and Training, CRC Press, (2000).
18. Setnes, M. Babuska, R. Kaymak, U. and van Nauta Lemke, H R.: Similarity Measures in Fuzzy Rule Base Simplification (1998).

19. Bensaid A.M., Hall L.O., Bezdek, J.C. Clarke, L.P. Silbiger, M.L. Arrington, J.A. and Murtagh, R.F.. Validity-guided (Re) Clustering with applications to image segmentation. IEEE Transactions on Fuzzy Systems, 4 (1996) 112-123.
20. Kima, D.W. Lee, K.H. Leeb, D.: On cluster validity index for estimation of the optimal number of fuzzy clusters Pattern Recognition 37 (2004) 2009-2025.

# A New Classifier Design with Fuzzy Functions

Aslı Çelikyılmaz[1], I. Burhan Türkşen[1,2], Ramazan Aktaş[3], M. Mete Doğanay[4], and N. Başak Ceylan[5]

[1] Dept. of Mechanical and Industrial Engineering, University of Toronto, Canada
{asli.celikyilmaz,turksen}@mie.utoronto.ca
[2] Dept. of Industrial Engineering TOBB-Economics and Technology University, Turkey
[3] Dept. of Business Administration TOBB-Economics and Technology University, Turkey
[4] Dept. of Business Administration Çankaya University, Turkey
[5] Dept. of Business Administration Atılım University, Turkey

**Abstract.** This paper presents a new fuzzy classifier design, which constructs one classifier for each fuzzy partition of a given system. The new approach, namely Fuzzy Classifier Functions (FCF), is an adaptation of our generic design on Fuzzy Functions to classification problems. This approach couples any fuzzy clustering algorithm with any classification method, in a unique way. The presented model derives fuzzy functions (rules) from data to classify patterns into number of classes. Fuzzy *c*-means clustering is used to capture hidden fuzzy patterns and a linear or a non-linear classifier function is used to build one classifier model for each pattern identified. The performance of each classifier is enhanced by using corresponding membership values of the data vectors as additional input variables. FCF is proposed as an alternate representation and reasoning schema to fuzzy rule base classifiers. The proposed method is evaluated by the comparison of experiments with the standard classifier methods using cross validation on test patterns.

**Keywords:** Fuzzy classification, fuzzy c-means clustering, SVM.

## 1 Introduction

Numerous classical classifiers such as logistic regression (LR), support vector machines (SVM) [9, 13, 23], etc., are widely used to approximate linear or non-linear decision boundaries of the system under study. The main assumption in these classification methods and their possible drawback is that, the data with possible multi-model structure is classified using a single classifier. More recently, fuzzy classifier methods based on *if-then* rules have been applied to solve classification problems by constructing multi-model structures, which yield a class label for each vector in the given space. Some examples of well known fuzzy classifiers are fuzzy clustering methods [1, 19], adaptive neuro-fuzzy inference system (ANFIS) for classification [12], evolutionary algorithms [34,11], etc., which have been employed to automatically learn fuzzy *if-then* rules from the data.

Classical fuzzy system models based on fuzzy rule bases (FRB) (*if-then rules)* have some challenges [20], e.g., identification of membership functions, and most suitable combination operator (*t-norm* or *t-conorm*), the choice between fuzzy or non-fuzzy

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 136–143, 2007.

conjunctions, defuzzification and implication operators to capture the uncertainty associated with the linguistic "AND", "OR" and "IMP" for the representation of rules, reasoning with them, defuzzification, etc. Hence, to overcome these challenges, we present a new Fuzzy Classifier Function (FCF) approach to estimate decision boundaries, which does not require construction of *if-then* rules. The proposed FCF approach, based on our previous novel "Fuzzy Functions" approaches for regression problems [6, 20, 21], is an alternative approach to Fuzzy Classifiers with "Fuzzy Rule Base" approaches [15].

The presented FCF first clusters the given data into several overlapping fuzzy clusters, each of which is used to define a separate decision boundary. Our approach is unique because during fuzzy classifier design, the similarity of the objects is enhanced with additional fuzzy identifiers, i.e., the membership values, by utilizing them as additional input variables. Thus, the membership values and their possible transformations are augmented to the original training dataset as new dimensions to structure different datasets for each cluster. With this approach, during training algorithm, the classifier employs valuable information such that the objects that are closer to each other with opposite labels are assigned different membership values in the same cluster. The proposed approach builds one classifier for each cluster (see Figure 1). Any classifier method to build a linear or a non-linear classifier can be used depending on the structure of the dataset. Hence, we used Logistic Regression to represent a linear classifier and SVM to build non-linear fuzzy classifiers to observe their affects on the models.

The remainder of this paper is organized as follows. First, we present description about the idea and formulation of FCF algorithm. Next, the discussion and conclusions on the empirical analysis of the proposed FCF models using various datasets in comparison to well-known classifier methods is presented. Finally, conclusions are drawn.

## 2 Proposed Fuzzy Classifier Function (FCF) Approach

In the novel FCF approach, the given dataset is first fuzzy partitioned using fuzzy *c*-means clustering (FCM) [2] algorithm, so that each cluster can be represented by a separate decision boundary using classification methods. FCM clustering algorithm enables to apply supervised partitioning of the dataset based on the given class labels. Therefore, membership values encapsulate the class label information as well as spatial relationships between the input data vectors and they are used as additional dimensions to structure different datasets for each cluster $i$, $i=1,..,c$.

Let the given dataset contains $nd$ data points, $XY=\{(x_k, y_k)\}^{nd}$, where $x_k$ represents input data vectors of $nv$ features, $\boldsymbol{x}_k=(x_{k,1},\ldots, x_{k,nv})\in X\subseteq\Re^{nv}$, $k=1,..,nd$, and class labels $y_k\in\{0,1\}\in Y$.

We first find hidden structures in the given training dataset with FCM [2] clustering algorithm, where $m$ is the level of fuzziness ($m>1$), and $c$ is the number of clusters ($c\geq 2$). FCM clustering algorithm computes partition matrix $U\subseteq\Re^{nd\times c}$ of membership values $0\leq\mu_{ik}\in U\leq 1$ of every data sample $k$ in each cluster $i$, using the following membership function for a given pair of $(m,c)$ parameters:

$$\mu_{ik}(\boldsymbol{xy}) = \sum_{j=1}^{c} \left[ \left\| (\boldsymbol{xy})_k - \upsilon_i(\boldsymbol{xy}) \right\| / \left\| (\boldsymbol{xy})_k - \upsilon_j(\boldsymbol{xy}) \right\| \right]^{2/(1-m)} . \quad (1)$$

using the cluster centers;

$$\upsilon(\boldsymbol{xy})_i^{(m,c)} = \{x_{1,i},\ldots, x_{nv,i}, y_i\},\ i=1,\ldots,c. \quad (2)$$

Then, we identify the cluster centers of the "input space" for given $(m,c)$ as :

$$\upsilon(\boldsymbol{x})_i^{(m,c)} = \{x_{1,i},\ldots, x_{nv,i}\}. \quad (3)$$

We compute membership values of the *x-input* domain, $\mu_{ik}(\boldsymbol{x})$, using the membership function in (1) and their cluster centers $\upsilon(\boldsymbol{x})_i^{(m,c)}$ in input domain and then normalize them to interval [0,1]. The $\gamma_i(x)$'s are the normalized membership values of $x$-domain as column vectors of $i^{th}$ cluster, which in turn indicate the membership values that will constitute as new input variables in our proposed scheme of classifier identification for the representation of $i^{\text{th}}$ cluster.

For each cluster *i,* a different dataset is structured by using membership values ($\gamma_i$) and their transformations as additional dimensions. This is same as mapping the original input space, $\Re^{nv}$, onto a higher dimensional feature space $\Re^{nv+nm}$, i.e., $\boldsymbol{x} \rightarrow \Phi_i(\boldsymbol{x}, \gamma_i)$. Hence, each data vector is represented in $(nv+nm)$ feature space. *nm* is the number of dimensions added to the original input space. The decision boundary is sought in this new space, see Fig. 1. The structure of the input matrix of the $i^{th}$ cluster for a special case using one dimensional input matrix and only membership values as additional dimensions is as follows:

$$\Phi_i(\boldsymbol{x}, \gamma_i) = \begin{bmatrix} x_1 & \gamma_{i\times 1} \\ \vdots & \vdots \\ x_{nd} & \gamma_{i\times nd} \end{bmatrix} \in \Re^{nv+nm} \begin{bmatrix} y_1 \\ \vdots \\ y_{nd} \end{bmatrix} \in \Re. \quad (4)$$

Intercept is encapsulated in $x$-domain. Then, depending on the system, a linear or a non-linear classifier, e.g., LR or SVM for classification, etc., is build for each cluster in $(nv+nm)$ space, i.e., $\Re^{nv} \rightarrow \Re^{nv+nm}$. The new augmented input matrices, $\Phi_i(\boldsymbol{x}_i, \gamma_i)$ for each $i^{th}$ cluster as in (4), could take on several forms depending on which transformation of membership values we want/need to include in our system structure identification. Possible examples would include; logit transformations, $\log((1-\gamma_{ik})/\gamma_{ik})$, or exponential transformations, $exp(\gamma_{ik})$ of the membership values. A prominent feature of the novel FCF is that, a linearly inseparable data in the original input space can be separated in the $\Re^{nv+nm}$ space with the additional information induced by the membership values. Hence, data points with opposite class labels, which are closer in the input space, will be farther away pointing separate directions in the feature space after mapping is employed.

The membership values also affect the design of the classifier of each cluster in the following way. The data points, which are closer to the cluster center have larger $\mu_i$'s ($\mu_i \in [0,1] \mid \mu_i > 0.5,\ i=1,..,c \geq 2$) than those that are farther away from the cluster center. Data points around the cluster center would have more impact on their corresponding cluster decision surface.

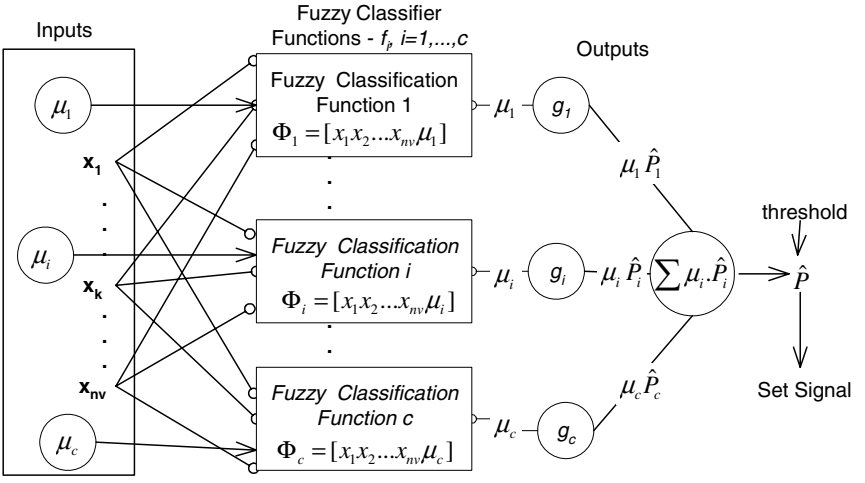



**Fig. 1.** Fuzzy Classifier Function (FCF) of a system with *c*-models

In order to estimate the posterior probability $P_i(y_i=1 \in Y_i|\Phi_i)$ of the binary output for each cluster, we use logistic regression or support vector machine for classification (SVC) to represent one linear or non-linear classifier function of a cluster. Here we show SVC implementation. The optimal SVC hyperplane is found for each cluster and dual optimization problem of SVC is represented as follows:

$$\begin{aligned} &Max\ Q_i = \sum_k^{nd} \beta_{ik} - \sum_{k,l=1}^{nd} \beta_{ik}\beta_{il} y_{ik} y_{il} K\left(\Phi_k\left(\boldsymbol{x}_{ik}, \gamma_{ik}\right), \Phi_l\left(\boldsymbol{x}_{il}, \gamma_{il}\right)\right) \\ &s.t.\ \sum_k^{nd} \beta_{ik} y_{ik} = 0, \quad 0 \le \beta_{ik} \le C_i, \quad k = 1,...,nd, i = 1,..,c \end{aligned} \tag{5}$$

where $K(\Phi_{ik}(X_{ik},\gamma_{ik}), \Phi_{il}(X_{il},\gamma_{il}))$ is a kernel function between two vectors, $\Phi_{ik}(X_{ik},\gamma_{ik})$ and $\Phi_{il}(X_{il},\gamma_{il})$, of $i^{th}$ cluster in feature space, $\beta$ are Lagrange multipliers, which are introduced to solve SVC optimization algorithm. The kernel is used to determine the level of proximity between any two points. We used three most popular kernel functions, i.e., linear(*FCFLin*): $K(x_k, x_j) = x_k^T x_j$, polynomial kernel:(*FCFPol*) : $K(x_k,x_j) = (x_k^T x_j+1)^d$, $d>0$ and radial basis kernel (*FCFRbf*): $K(x_k,x_j) = \exp(-\delta|x_k-x_j||^2)$, $\delta>0$, where d, and $\delta$ are user defined kernel parameters. In this paper, default values of kernel parameters, ($d=2, \delta=1/nv$), are used.

User defined penalty term, $C_i$, which may or may not be same for each cluster, determines the complexity of the classifier and it bounds the $\beta_{ik}$, which represents the Lagrange multiplier assigned to $k^{th}$ train vector for $i^{th}$ cluster, $C_i \ge \beta_{ik} \ge 0$. The decision hyperplane for each cluster, *i,* is given below:

$$\hat{y}_i = sign\left(f_i\left(\Phi_i\left(\boldsymbol{x}_i, \gamma_i\right)\right)\right) = \sum_k^{nd} \beta_{ik} y_{ik} K\left(\Phi_{ik}\left(\boldsymbol{x}_{ik}, \gamma_{ik}\right), \Phi_i\left(\boldsymbol{x}_i, \gamma_i\right)\right). \tag{6}$$

The learning problem in expression (6) is only expressed in terms of unknown Lagrange multipliers $\beta_{ik}$, the known mapped inputs vectors, $\Phi_i(X_i,\gamma_i)$ of each cluster, and their output values. Hence, these vectors with $\beta_{ik}>0$ are called the support vectors.

Since the classifiers estimated for each cluster make some part of the overall decision, we calculate the posterior probabilities using improved Platt probability method [16, 18], which is based on the approximation of the $\hat{y}_i$ with the sigmoid function as:

$$\hat{P}_i\ (\hat{y}_i = 1 | \Phi_i(X_i, \gamma_i)) = 1/(1+\exp(a_1 f_i + a_2)) \tag{7}$$

where the parameters $a_1$ and $a_2$ are found by maximum likelihood estimation [16, 18]. Since the aim of the algorithm is find a crisp probability output, each probability output $\hat{P}_i$ from each cluster is weighted with their membership values as:

$$\hat{P} = \left(\sum_{i=1}^{c} \gamma_i \hat{P}_i\right) / \left(\sum_{i=1}^{c} \gamma_i\right). \tag{8}$$

The common way of finding the optimum cluster size is through suitable validity index measures. In this paper, the optimal pair, $(m^*, c^*)$, is determined through an exhaustive search technique as follows: For each given $(m,c)$ pair, one FCF model is build using the training dataset and its performance is evaluated based on maximum AUC (Area under the ROC curve). The FCF model parameter set with the best performance are set as the optimum parameters. In this paper, we optimized the FCF parameters, through cross validation method. Next, we illustrate the application of our algorithm to various benchmark and real life data.

## 3 Numerical Experiments

We used 5 classification data from UCI repository [17] including *breast-cancer, pima-diabetes, liver-disorders, ionosphere,* and *credit-application* and a real life bank failure data to build an early warning system (EWS). For benchmark datasets, approximately 45% of observations from each dataset are randomly selected for training, %35 to optimize the parameters (validation) and 20% observation, which has not been used in training or validation data, is used for testing the model performance. Experiments are repeated 10 times with the above combinations.

The real-life *bank failure* sparse dataset [8] consists of 27 financial ratios of 42 Turkish banks based on 1998-2002 period. The binary output variable, $y \in \{0,1\}$, is 1 if the bank was bankrupted at the 4th year (2002). Since each financial ratio from three prior years before 2002, i.e., *(t-1)*, *(t-2)* and *(t-3),* are used to estimate the 4th year output variable, the model comprised a total of 81 input variables, i.e., (27X3=81) [78]. Based on series of feature selection methods 6 input variables are selected from three different years: $x_1^{(t-1)}$, $x_2^{(t-2)}$, $x_3^{(t-2)}$, $x_2^{(t-3)}$, $x_3^{(t-1)}$, $x_4^{(t-3)}$ *($x_1$:liquid_assets; $x_2$:net_income/paid_in_capital; $x_3$:interest_revenue/interest_expense; $x_4$:interest_income/ #employees).* We applied 6 fold *cv* by dividing the data into 6 mutually exclusive sub-samples, each containing 7 observations, preserving the overall fail/success ratio as much as possible. Then we built a classification model using 5 sub-samples (35 observations) as training and the remaining sub-sample as validation. Each sub-sample is used once as validation dataset and average *cv* performance over 10 repetitions are calculated.

As it is usually difficult to identify single best algorithm reliably, it a good practice to provide a rank ordering of different algorithms applied on the set of datasets [3]. AUC is closely related to Wilcoxon-Mann-Whitney [10] statistic to measure the

performance of the models, which is much preferred to simple accuracy measure [7]. SVC and LR classifier functions are separately used to build 4 different types of proposed FCF models, namely, FCFLin, FCFPol, FCFRbf, FCFLR. Their performances are evaluated in comparison to the model results of FRB based classifier, ANFIS [1114], a single model approach with standard LR, SVC models [5] with three different kernels: Linear (SVCLin), polynomial (SVCPol), RBF kernel (SVCRBF), and multi-layer perceptron Neural Networks (NN) of Matlab 7.0.1 toolbox.

For each experiment, we evaluated AUC using different SVC cost parameters, $C$=[$2^7$,$2^6$,…,$2^{-4}$] for FCF models which implement SVC during structure identification, and FCM parameters, $m$=[1.3, 1.4, …,2.4] and $c$=[2,3,…,8]. For the construction of fuzzy functions, membership values and their exponential transformations are used as additional inputs.

Average ranking method [3], inspired by Freidman, is used to compare different classification algorithms. For each dataset, we order the algorithms according to average AUC and assign ranks accordingly. Let $r_j^q$ be the rank of algorithm $j(j=1,..,10)$ on dataset $q(1,..,n=6)$. Average Rank (AR) for each algorithm is calculated as $\bar{r}_j=(\sum_q r_j^q)$/n. Table 1 displays the AUC and AR results as follows:

**Table 1.** Classfication Performances based on average test AUC of 10 repetitions. The ranks in parenthesis are used in computation of Freidman's' rank test.

| *Method* | *Diabet* | *CreditS* | *Liver* | *BreastC* | *IonSp.* | *Bank* | *AR* | *Rank* |
|---|---|---|---|---|---|---|---|---|
| *LR* | 0.848(9) | 0.925(7.5) | 0.752(7) | 0.560(6) | 0.895(9) | 0.902(10) | 8.1 | 10 |
| *ANFIS* | 0.82(10) | 0.91(10) | 0.72(10) | **0.665**(1) | 0.935(6) | 0.928(7.5) | 7.4 | 9 |
| *NN* | 0.854(8) | 0.928(3) | 0.746(8) | 0.558(7) | 0.932(8) | 0.930(5.5) | 6.6 | 6 |
| *SVCLin* | 0.860(5) | 0.926(5.5) | 0.745(9) | 0.540(8) | 0.88(10) | 0.930(5.5) | 7.2 | 7 |
| *SVCPol* | 0.858(6) | 0.920(9) | 0.759(6) | 0.52(10) | 0.950(4) | 0.919(9) | 7.3 | 8 |
| *SVCRbf* | 0.865(3) | 0.927(4) | 0.777(5) | 0.519(9) | 0.976(2) | 0.928(7.5) | 5.1 | 5 |
| *FCFLR* | 0.857(7) | **0.940**(1) | 0.778(4) | 0.567(5) | 0.940(5) | 0.945(2) | 4.0 | 3 |
| *FCFLin* | 0.864(4) | 0.926(5.5) | 0.788(2) | 0.593(3) | 0.933(7) | 0.936(4) | 4.3 | 4 |
| *FCFPol* | 0.866(2) | 0.925(7.5) | **0.794**(1) | 0.581(4) | 0.965(3) | 0.937(3) | 3.4 | 2 |
| *FCFRbf* | **0.875**(1) | 0.932(2) | 0.786(3) | 0.639(2) | **0.984**(1) | **0.962**(1) | 1.7 | 1 |

## 4 Discussions

On average, the proposed model using non-linear SVC (*FCFRbf)* ranked the first. Our proposed models outperform the traditional classifiers in all datasets. For bank failure system, the single model approach, LR, is not as accurate as the Fuzzy Classifier models, where there are as many fuzzy functions as the optimum number of clusters. ANFIS models have the highest AUC for the breast cancer data only. Traditional non-linear SVM models have high AUC for all data; however, proposed FCF models can separate non-linear cases better in feature spaces within granular structures. Using fuzzy membership values as new predictors for local fuzzy classifiers, we can create better models with higher generalization ability. In the experiments we used one type of fuzzy classifier function structure, the membership values and their exponential transformations. Further research with different combinations of fuzzy function

structures and different model approximators within one model will be investigated for possible performance improvements. Furthermore, with the application of the proposed method and the comparison methods on various other empirical datasets, we will conduct statistical significance tests 3] to measure the level of improvement of the proposed approach as opposed to other classifiers.

## 5 Conclusions

We presented a novel classifier design, the Fuzzy Classifier Functions (FCF) methodology, which is an adaptation of our earlier Fuzzy System Modeling with Fuzzy Functions for regression approaches to classification problems. The novel FCF couple fuzzy clustering and traditional classifier algorithms to represent systems with multi-models. Firstly, the membership values from the FCM are used to map the original input matrix to a user defined feature space by augmenting the membership values and their transformations to the original input variables. Then, a separate classifier function approximation technique, i.e., logistic regression or support vector classification, is applied on this new space for each cluster. The multi-model system identification approach of FCF enables to design local classifiers, which uses the information on natural grouping of data samples, i.e., the membership values, as explanatory variables. Empirical comparisons using one real world problem and five benchmark data indicate that the proposed FCF is a robust method in terms of yielding more accurate results than the traditional classification methods. FCF can be considered as a new perspective for the applications of *Fuzzy Functions* on real world problems.

## References

1. Abe, S., Thawonmas, R.: A fuzzy classifier with ellipsoidal regions. IEEE Trans. Fuzzy Syst. 5 (1997) 358-368
2. Bezdek, J.-C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press (1981)
3. Bradzil, P.B., Soares, C.: A Comparison of Ranking Methods for Classification Algorithm Selection. In Machine Learning: ECML 2000, 11th European Conference on Machine Learning, ECML 2000, LNAI 1810, Springer Verlag (2000)
4. Chang, X., Lilly, J.H.: Evolutionary Design of a fuzzy classifier from data. IEEE Trans. On System, Man and Cyber. B. 34 (2004) 1894-1906
5. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines (2001). Software available http://www.csie.ntu.edu.tw/~cjlin/libsvm
6. Çelikyilmaz, A., Türkşen, I.B.: Fuzzy Functions with Support Vector Machines. Information Sciences (2007)
7. Demsar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research 7 (2006) 1-30
8. Doganay, M.M, Ceylan, N.B., Aktas, R.: Predicting Financial Failure of the Turkish Banks, Annals of Financial Economics 2 (forthcoming)
9. Ducker, H.D, Wu, D., Vapnik, V.: Support Vector Machines for spam categorization. IEEE Transaction on Neural Networks 10 (5) (1999) 1048-1054

10. Hanley, J.A., McNeil, B.J.: The meaning and use of area under a receiver operating characteristic (ROC) curve. Radiology (1992)
11. Huerta, E.B., Duval, B., Hao, J.-K.: A Hybrid Bayesian Optimal Classifier Based on Neuro-fuzzy Logic. Applications of Evolutionary Computing Vol. 3907. Springer-Verlag, Berlin Heidelberg New York (2000) 34-77
12. Huang, M.-L., Chen, H.-Y., Huang, J.-J.: Glaucoma detection using adaptive neuro-fuzzy inference system. Expert Systems with Applications 32 (2007) 458-468
13. Kecman, V.: Learning and Soft Computing: Support Vector Machines. Neural Networks, and Fuzzy Logic Models, Cambridge, Mass. MIT Press (2001)
14. Klawonn, F., Nauck, D., Kruse, R.: Generating rules from data by fuzzy and neuro-fuzzy methods. In Proc. Fuzzy Neuro-System (1995) 223-230
15. Kuncheva, L.I.: Fuzzy Classifier Design. Studies in Fuzziness and Soft Computing (2000)
16. Lin, H.-T., Lin, C.-J., Weng R.C.: A note on Platt's probabilistic outputs for support vector machines. Technical report (2003)
17. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science (1998)
18. Platt, J.: Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. MIT Press, Cambridge, MA (2000)
19. Setnes, M., Babuska, R.: Fuzzy relational classifier trained by fuzzy clustering. IEEE Trans. Syst. Man, Cybern. B. 29 (1999) 619-625
20. Türkşen, I.B., Çelikyılmaz, A.: Comparison of Fuzzy Functions with Fuzzy Rule Base Approaches. Int. Journal of Fuzzy Systems 8 (3) (2006) 137-149
21. Türkşen, I.B.: Fuzzy Functions with LSE. Applied Soft Computing (forthcoming)
22. Wang, L.P. (ed.): Support Vector Machines: Theory and Application. Springer, Berlin Heidelberg New York (2005)
23. Vapnik, V.: Statistical Learning Theory. New York, Wiley (1998)

# Image Analysis of Ductal Proliferative Lesions of Breast Using Architectural Features

Haegil Hwang, Hyekyoung Yoon, Hyunju Choi, Myounghee Kim, and Heungkook Choi

School of Computer Engineerng, Inje University, Gimhae, 621-749, Korea
Dept. of Pathology, Inje University, Gimhae, 621-749, Korea
Dept. of Computer Engineering, Ewha Womans University, Seoul 120-170, Korea
{seaload,pathyoon,hjchoi}@mitl.inje.ac.kr, mhkim@mm.ewha.ac.kr, cschk@inje.ac.kr

**Abstract.** We propose a method to classify breast lesions of ducatal origin. The materials were tissue sections of the intraductal proliferative lesions of the breast: benign(DH:ductal hyperplasia), ductal carcinoma in situ(DCIS). The total 40 images from 70 samples of ducts were digitally captured from 15 cases of DCIS and 25 cases of DH diagnosed by pathologist. To assess the correlation between computerized images analysis and visual analysis by a pathologist, we extracted the total lumen area/gland area, to segment the gland(duct) area used a snake algorithm, to segment the lumen used multilevel Otsus method in the duct from 20x images for distinguishing DH and DCIS. In duct image, we extracted the five texture features (correlation, entropy, contrast, angular second moment, and inverse difference moment) using the co-occurrence matrix for a distribution pattern of cells and pleomorphism of the nucleus. In the present study, we obtained classification accuracy rates of 91.33%, the architectural features of breast ducts has been advanced as a useful features in the classification for distiguishing DH and DCIS. We expect that the proposed method in this paper could be used as a useful diagnostic tool to differentiate the intraductal proliferative lesions of the breast.

**Keywords:** Intraductal proliferative lesions of the breast, Texture features, Gray level co-occurrence matrix(GLCM), Sanke algorithm, Multilevel Ostus method, Architectural features of breast ducts.

## 1 Introduction

The breast cancer is a malignant-tumour that can develop metastatic tumours in women. The diagnosis of ductal hyperplasia (DH) and ductal carcinoma in situ (DCIS) still remains a problem in the histological diagnosis of breast lesions. Image analysis of tissue sections holds promise for diagnosing cancer and tracking the progression of the disease. In traditional cancer diagnosis, pathologists use histopathological images of biopsy samples taken from patients, examine them under a microscope, and make judgments based on their personal experience. However, intra- and interobserver variability(considerable variability) is

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 144–152, 2007.

presented in some situation[1], and it is difficult to accurately reproduce descriptions of tissue texture[2]. Therefore, we attempted to create a more objective and highly reproducible system for the morphological classification of breast diseases.

To create an optimized classifier for breast lesions, significant features that accurately describe the order/disorder of nuclear details must be extracted from images. The cancer cells show pleomorphism of the nucleus, and mitotic cell division[3], [4], [5]. In addition, nuclear features such as granularity and regularity of chromatin, irregularity of nuclear size and shape, distance between nuclei, and the change in the distribution of the cells across the tissue, are important for determining or predicting the progress of cancer[6],[7]. Histology-based statistical analyses of textural features are frequently based on a gray level co-occurrence matrix(GLCM)[8], [9]. Structural analysis methods describe the properties of texture elements as well as the placement of the texture elements.

The established criteria for distinguishing DH from DCIS are subjective and include features such as the architectural pattern of ducts. Anderson et al[10] used disciminat analysis of the duct characteristics for DH and DCIS groups selected the lumen/duct area ration and the duct area as significant discriminatory variables. We extracted the gland area, the ration of total lumen area/gland area, the average of the lumen area, the major axis, the minor axis, the number of the lumen in the duct. To segment the gland(duct) area used a snake algorithm, to segment the lumen used multilevel Otsus method. In duct image, we extracted the five texture features(correlation, entropy, contrast, angular second moment, and inverse difference moment) using the co-occurrence matrix for a distribution pattern of cells. We were interested in architecture the glands and the shape of individual lumen not the information of individual cell. Figure 1 show the structure diagram of the image classification system. It is clear that ductal characteristics carry useful diagnostic information for the discrimination with DH and DCIS. DH and DCIS.

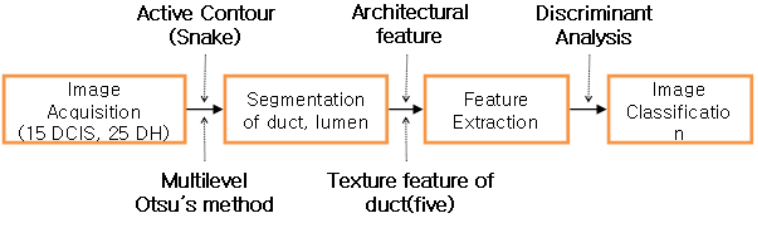



**Fig. 1.** The structure diagram of the image classification system

## 2 Materials and Methods

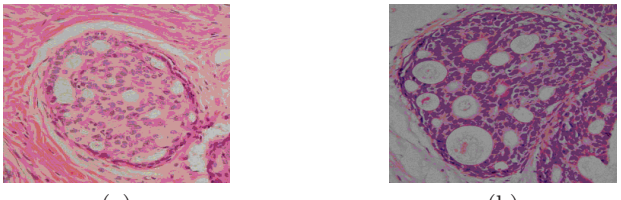
### 2.1 Tissue Samples and Image Acquistion

The samples of breast tissue were obtained from breast cancer patients at Busan Paik Hospital, Inje University in Korea. The breast tissue was stained with

hematoxylin and eosin (H&E) using an autostainer XL (Leica, UK). The digital images of the sections were acquired by a pathologist at a magnification of 20× with a 0.3 NA objective using a digital camera (Olympus C-3000) that was attached to a microscope (Olympus BX-51), and we obtained a total of 40 from 70 samples (25 images(DH), 15 images(DCIS) of tissue). A region of interest (ROI)-duct of in each digital image (1280×960, 20x) was selected by a pathologist (see the Fig. 2).

### 2.2 Segmentation of Duct and Lumen

The permitted duct profiles and intraduct lumen to be identified and their size[9] and texture features computed. It is clear that these groups form a spectrum of histological change; a proportion of cases fall into the intermediate category of atypical ductal hyperplasia (ADH). Once the segmented components of the histological scene are identified, they are recombined to form identifiable architectural structures such as glands.



(a) (b)

**Fig. 2.** Representative images of histological sections of breast tissue used in the present study (20x). Images correspond to selected regions of interest (ROI) ; (a) DH and (b) DCIS.

The segmentation requires higher magnification images to resolve the exact details of objects and the success of the next steps becomes more sensitive to the success of the segmentation. In segmentation, one difficulty is the complex nature of image scenes e.g., a typical tissue consists of touching and overlapping cells, lumen and ducts.

There are mainly two different approaches in object segmentation: the region-based (ex.thresholding) and boundary-based(ex.snakes[11]) approaches. The region- based approach is based on determining whether a pixel belongs to a duct (or lumen in the duct) or not, whereas the boundary based approach is based on finding the boundary points of a duct. The segmentation method should be chosen depending on the type of the features to be extracted. For example, in the case of morphological feature extraction, determining the exact locations of cells is more important, boundary-based approaches are more suitable than the region-based approaches[12].

Thresholding is an important technique for image segmentation that tries to identify and extract a target from its background on the basis of the distribution

of gray levels or texture in image objects. But, thresholding can not work well if a histogram valley is narrow or wide, or if a histogram is not bimodal. To obtain the optimal threshold, we used the Otsu method[13] for automatic threshold selection. The Otsu method is based on the relationship of variances derived from probability theory and determines an optimal threshold which minimizes the within-class variance or which maximizes the between-class variance[14]. For a given image with L different gray levels, the Otsu method computes the within-class variance for a threshold T as follows ;

$$\sigma_w^2(T) = \omega_0\sigma_0^2 + \omega_1\sigma_1^2 \tag{1}$$

where $\sigma_0^2$ and $\sigma_1^2$ are the variances of the pixels below and above the threshold, respectively. The Sahoo et al. study on global thresholding[15], concluded that the Otsu method was one of the better threshold selection methods for general real world images with regard to uniformity and shape measures. However, the Otsu method uses an exhaustive search to evaluate the criterion for maximizing. For bi-level thresholding, the Otsu verified that the optimal threshold $t^*$ is chosen so that the between-class variance $\sigma_B^2$ is maximized, $t^* = Arg\ \ Max\left\{\sigma_B^2(t)\right\}, 1 \leq t \leq L$. Assuming that there are $M-1$ thresholds, $\{t_1, t_2, \ldots, t_{M-1}\}$, which divide the original image into $M$ classes, the optimal thresholds $\left\{t_1^\bullet, t_2^\bullet, \ldots, t_{M-1}^\bullet\right\}$ are chosen by maximizing $\sigma_B^2$ as follows;

$$\begin{aligned}
&\left\{t_1^*, t_2^*, \ldots, t_{M-1}^*\right\} = Arg\ \ Max\left\{\sigma_B^2\left(\{t_1, t_2, \ldots, t_{M-1}\}\right)\right\} \\
&, 1 \leq t_1 < \ldots < t_{M-1} < L \\
&where\ \ \sigma_B^2 = \sum_{k=1}^{M} \omega_k \left(\mu_k - \mu_r\right)^2 \\
&with\ \ \omega_k = \sum_{i \in C_k}^{T-1} p\left(i\right), \mu_k = \sum_{i \in C_k}^{T-1} ip\left(i\right)/\omega\left(k\right).
\end{aligned} \tag{2}$$

This method were tested on 30 of the images; Fig. 3 shows the thresholding images when we used the multilevel (five-level) Otsus method[16].

To find the total lumen area and gland area, it is necessary to segment the lumen and gland(duct). We used the region-based approaches ('multilevel Otsu' method') and then, boundary-based approaches (active contour models ('snakes')[17]) for the duct segmentation (Fig. 4(b)). Filled duct area (Fig. 4(c)) and account the gland area. To segment each lumen, we processed the 'arithmetic operation(morphological operators-ADD)' to the applied multilevel Otsu' method images and the segmented duct image (Fig. 4(d)) and filling small holes in the lumen, and each the lumen was labeled (Fig. 4(e)), the lumen size(area) was added for the lumina area.

### 2.3 Extraction of Texture Features in Duct Image

In duct the texture features extracted (correlation, entropy, contrast, angular second moment, and inverse difference moment) using the gray-level co-occurrence matrix (GLCM) for a distribution pattern of cell. The five features used that were

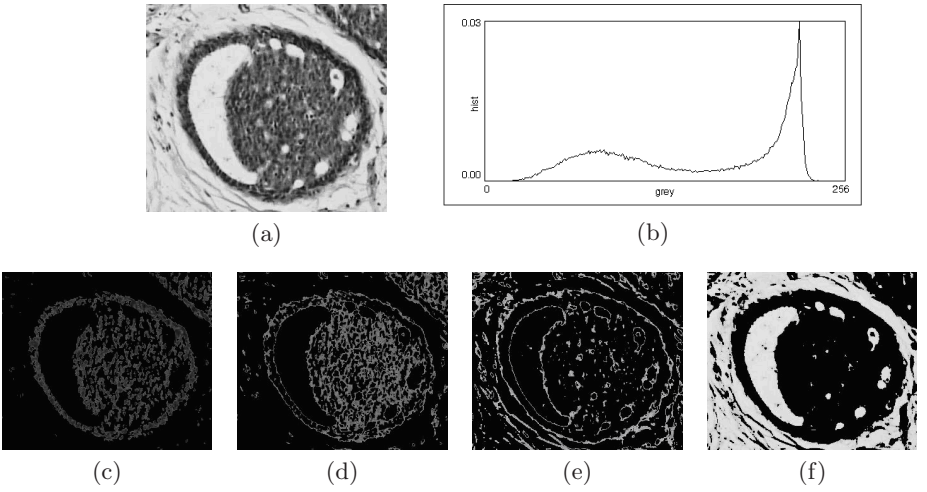



**Fig. 3.** The result image using the Otsu method for automatic threshold selection; (a) Green image of original. (b) Image histogram of (a). (c) bi-level of (a). (d) tri-level of (a). (e) four-level of (a). and (f) five-level of (a).

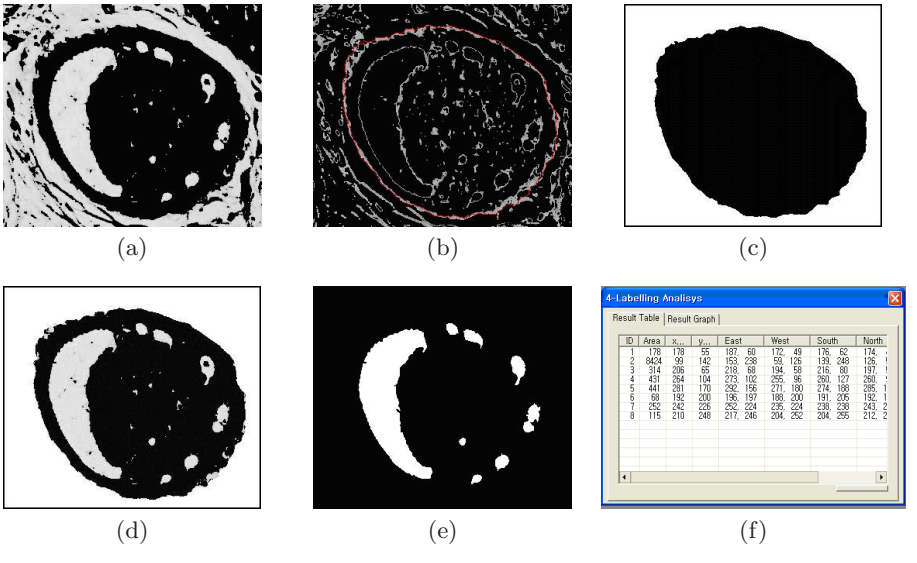



**Fig. 4.** To segment the lumen and gland(duct); (a) The result image using the multi-level Otsus method for automatic threshold selection (Green image of original 4 level). (b)using snakes algorithm, (c)duct segmentation of (a). (d) arithmetic operation(ADD) of (a) and (c). (e) each the lumen was labeling. and (f) the calculated result table.

a correlation function to calculate the linearity of the gray-level dependencies, entropy to measure randomness, contrast function to measure local variation, angular second moment(energy) to characterize the homogeneity of the image and inverse difference moment to identify the local homogeneity of the image. A region of interest (ROI) of in each duct image (20x) was selected 128×128 image for processing. To ensure that the co-occurrence matrices were calculated from major changes in grayscale, the images were scaled linearly from 256 gray levels to 32 gray levels. This had the added advantage of minimizing the calculation time of the co-occurrence matrices and reducing the size of the matrices.

We calculated five texture features from GLCM[8], [9], as follows

$$Correlation = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{[(i-\mu_x)(i-\mu_y)PM]}{(\sigma_x \sigma_y)} \tag{3}$$

$$Entropy = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} PM \log(PM) \tag{4}$$

$$Contrast = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i-j)^2 PM \tag{5}$$

$$Angular\ \ Second\ \ Moment = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} PM^2 \tag{6}$$

$$Inverse\ \ Difference\ \ Moment = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{PM}{\left[1+(i-j)^2\right]} \tag{7}$$

in where PM denotes the GLCM that contains the probability, $i$ = integer pixel number along a row ($1\ to\ N-1$), $j$ = integer pixel number along a column ($1\ to\ N-1$), $\mu_x$ = mean of row sums, $\mu_y$ = mean of column sums(with square matrix $\mu_x = \mu_y$), $\sigma_x$ = standard deviation of row sums, $\sigma_y$ = standard deviation of column sums, and $N-1$ = total number of pixels in row or column.

## 3 Results

We extracted the gland area(feature 1), the ratio of total lumen area/gland area(feature 2), the average of the lumen area(feature 3), the major axis(feature 4), the minor axis(feature 5), the number of the lumen in the duct(feature 6) and the five texture features(correlation(feature 7), entropy(feature 8), contrast(feature 9), angular second moment(feature 10), and inverse difference moment(feature 11)) in the duct from 20x images for distinguishing DH and DCIS. The image resolution was 1280×960(20x) pixels and 24 bits per pixel. We obtained a total of 40 from 70 samples (25 images (DH), 15 images (DCIS) of tissue).

To evaluate what quantitative features of the DH and DCIS could contribute to separate information and how it could increase the accuracy, we analyzed the statistical difference of eleven features between the DH and DCIS. The analysis of variance (ANOVA) was used to determine the levels of statistical significance in the differences in distribution. Te best feature will have the largest F-value and the smaller the p-value, the greater the inconsistency. Traditionally, the p-value is less than 0.05, we were given p-value (<0.005). So seven features were selected in Feature 1,2,3,4,5,6 and 11.

To find the vector of features that best characterized the difference in distribution, sequential stepwise selection was applied. The results are shown in Table 1, where, Wilks Lambda is the ratio of the determinants of the within-class and total covariance matrices [18].

We created the three classifiers using discriminant analysis: one is using the total features (eleven), the other is using the seven feature, other is using the two features(feature 1 and 2). Table 2, 3 and 4 shows the correct classification rate, 91.33%, 89.33% and 86.00%, respectively, and columns is the computer-based classification and rows is subjective classification by a pathologist.

**Table 1.** Stepwise discriminant analysis

| Step | Entered | Partial R-Square | F Value | Pr > F | Wilks Lambda | Pr < Lambda |
|---|---|---|---|---|---|---|
| 1 | Feature 2 | 0.4047 | 25.83 | <.0001 | 0.59533863 | <.0001 |
| 2 | Feature 1 | 0.2335 | 11.27 | 0.0018 | 0.4563069 | <.0001 |

**Table 2.** The classification result using the total eleven features

| | DH | DCIS | Total(%) |
|---|---|---|---|
| DH | 24 | 1 | 96.00 |
| DCIS | 2 | 13 | 86.67 |
| Total | 26 | 14 | 91.33 |

**Table 3.** The classification result using the seven features (feature 1,2,3,4,5,6 and 11)

| | DH | DCIS | Total(%) |
|---|---|---|---|
| DH | 23 | 2 | 92.00 |
| DCIS | 2 | 13 | 86.67 |
| Total | 25 | 15 | 89.33 |

**Table 4.** The classification result using the two features (feature 1 and 2)

| | DH | DCIS | Total(%) |
|---|---|---|---|
| DH | 23 | 2 | 92.00 |
| DCIS | 2 | 12 | 80.00 |
| Total | 26 | 14 | 86.00 |

## 4 Conclusion

We extracted the gland area, the ration of total lumen area/gland area, the average of the lumen area, the major axis, the minor axis, the number of the lumen in the duct and the five texture features (correlation, entropy, contrast, angular, second moment, and inverse difference moment) in the duct from 20x images for distinguishing DH and DCIS. Varying architectural features (the ration of total lumen area, gland area and duct area) can be associated with both DH and DCIS ducts and even visually the basic gland/lumen shape is not sufficient to distinguish clearly between DH and DCIS lesions.

It is clear that ductal characteristics carry useful diagnostic information for the discrimination with DH and DCIS. However, duct architecture alone is not sufficient to identify and discriminate DH and DCIS, and it is necessary to identify the more reliable features to discriminant them. Hitopathologically, the presence of swirls, necrosis, and Roman Bridges are conventionally used as additional clues in the differentiation between DH and DCIS[10]. We expected the efforts to derive additional quantitative data from these segmented glands which provide data on nuclear orientation and spatial distribution which can be combined with architectural features of the duct. An addition, we make an estimate that the misclassified data fall into the intermediate category of atypical ductal hyperplasia (ADH). To improve the accuracy, we will find the significant discriminatory variables of duct characteristics for distinguishing among the DH, DCIS and ADH. A Compartive analysis with neural network and SVM(Support Vector Machine) will be studied to evalute the classification result.

## Acknowledgments

This work was supported by the Korea Institute of Science & Technology Evaluation and Planning (KISTEP) under the Real Time Molecular Imaging program.

## References

1. Andrion A., Magnani C., Betta P.G., Donna A, Mollo F, Scelsi M, Bernardi P, Botta M, Terracini B.: Terracini, Malignant Mesothelioma of the Pleura: Interobserver Variability. J. Clin. Pathol 48 (1995) 856-860
2. Rodenacker K., Bengtsson E.: A Feature Set for Cytometry on Digitized Microscopic Images. Anal. Cell. Pathol. 25 (2003) 1-36
3. Dalton L.W., Page D.L., Dupont W.D.: Histological Grading of Breast Carcinoma: A Reproducibility Study. Cancer 73 (1994) 2765-2770
4. Kronqvist P., Kuopio T., Collan Y.: Effect of Freezing on Histologic Grading of Invasive Ductal Breast Cancer. Analyt. Quant. Cytol. Histol. 25 (2003) 47-52
5. Palcic B., Jaggi B., MacAulay C.: The Importance of Image Quality for Computing Texture Features in Biomedical Specimens. Proc. SPIE 1205(1990) 155-162
6. Choi H.K, Vasko J., Bengtsson E., Jarkrans T., Malmstrom P., Wester K., Busch C.: Grading of Transitional Cell Bladder Carcinoma by Texture analysis of Histological Section. Anal. Cell. Pathol. 6 (1994) 327-343

7. Hwang H.G., Choi H.J., Lee B.I., Yoon H.K., Nam S.H., Choi H.K.: Multiresolution Wavelet-transformed Image Analysis of Histological Sections of Breast Carcinomas. Cell. Oncol27(4) (2005) 237-244
8. Haralick R.M., Shanmugam K., Dinstein I.: Texture Feature for Image Classification. IEEE Trans. On System Man and Cybernetics SMC-3(6) (1973) 610-624
9. Rodenacker K.: Invariance of Texture Features in Image Cytometry under Variation of Size and Pixel Magnitude. Anal. Cell. Pathol. 8 (1995) 117-133
10. Anderson N.H., Hamilton P.W., Bartels P.H., Thompson D., Montironi R., Sloan J.M. et al.: Computerized Scene Segmentation for the Discrimination of Architectural Features in Ductal Proliferative Lesions of the Breast. J. Pathol 181 (1997) 374-380
11. Kass M., Witkin A., Terzopoulos D.: Snakes: Active contour models. International J. Computer Vision 1 (1988) 321-331
12. Demir C., Yener B.: Automated Cancer Diagnosis Based on Histopathological Images: a Systematic Survey. Technical report Renesselaer polytechnic institute Dep. Computer Scie (2005) 1-15
13. Otsu N.: A Threshold Selection Method from Gray Level Histograms. IEEE Trans Syst Man Cybern 9(1) (1979) 62-69
14. Weyn. B., Wouwer G. V., A. Van Daele, et al.: Automated Breast Tumor Diagnosis and Grading Based on Wavelet Chromatin Texture Description. Cytometry 33 (1998) 32-40
15. Sahoo P.K., Soltani S., Wong A.K.C, Chen Y.C. : A Survey of Thresholding Techniques. Comp Vis Graph Image Process. 41 (1988) 233-260
16. Liao P.S., Chen T.S, Chung P.C.: A Fast Algorithm for Multilevel Thresholding. J.Inform. Sci .Engin 17 (2001) 713-727
17. Street W.N.,Wolberg W.H., Mangasarian O.L.: Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology 1905 (1993) 861-870
18. Johnson. R.A., Wichern D.W.: Applied Multivariate Statistical Analysis. Prentice-Hall Inc. (1998)

# Nucleus Segmentation and Recognition of Uterine Cervical Pap-Smears

Kwang-Baek Kim[1], Doo Heon Song[2], and Young Woon Woo[3]

[1] Dept. of Computer Engineering, Silla University, Busan, Korea
gbkim@silla.ac.kr
[2] Dept. of Computer Game and Information, Yong-in SongDam College, Korea
dsong@ysc.ac.kr
[3] Dept. of Multimedia Engineering, Dong-Eui University, Busan, Korea
ywwoo@deu.ac.kr

**Abstract.** The classification of the background and cell areas is very important but difficult problem due to the ambiguity of boundaries. In this paper, the cell region is extracted from an image of uterine cervical cytodiagnosis using the region growing method. Segmented images from background and cell areas are binarized using a threshold value. And the 8-directional tracking algorithm for contour lines is applied to extract the cell area. Each extracted nucleus is transformed to the original RGB space. Then the K-Means clustering algorithm is employed to classify RGB pixels to the R, G, and B channels, respectively. Finally, the Hue information of nucleus is extracted from the HSI models that are transformed using the clustering values in R, G, and B channels. The fuzzy RBF Network is then applied to classify and identify the normal or abnormal nucleus. The result shows that the accuracy of our method is 80% overall and 66% in 5-class problem according to the Bethesda system.

## 1 Introduction

Cervical cancer is one of the most frequently found genital diseases in Korean women but is curable if it is diagnosed early enough. Previous researches show that cervical cancer occupies 16.4 $\sim$ 49.6% of malignant tumors and especially occupies 26.3 $\sim$ 68.2% in Korean women [1][2]. The best method to completely cure cervical cancer is to prevent the cell from developing into cervical cancer. For this purpose, there have been many efforts to automate the process of cytodiagnosis at least partially during the last 40 years[3].

Diagnosis of the region of interest in a medical image consists of area segmentation, feature extraction and characteristic analysis. In area segmentation, a medical specialist detects abnormal regions of a medical image based on his expertise. In feature extraction, features are extracted from the separated abnormal region. A medical doctor diagnoses a disease by using character analysis which deciphers the extracted features to analyze and compare clinical information. Area segmentation methods can be taxonomized to the pixel-based methods

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 153–160, 2007.

and the region-based methods[3][4]. Pixel-based methods assign an independent meaning to each pixel according to a predefined criterion. Pixel-based methods can use global features[3]. Meanwhile, region-based methods catch the meaning of the region by analyzing neighbour pixels with local characteristics but typically require more calculations[4].

In this paper, we propose a new nucleus segmentation and recognition of uterine cervical pap-smears with region growing technique and neural network. Segmented images from background and cell areas are binarized using a threshold value. And then the 8-directional tracking algorithm for contour lines is applied to extract the cell area[5]. Each extracted nucleus is transformed to the original RGB space. Then the K-Means clustering algorithm is employed to classify RGB pixels to the R, G, and B channels, respectively. Finally, the Hue information of nucleus is extracted from the HSI models that are transformed using the clustering values in R, G, and B channels. The fuzzy RBF network is then applied to classify and identify the normal or abnormal nucleus.

## 2 Segmentation of Nucleus Area

Fig. 1 shows the process of extracting the nucleus of cervix uteri cytodiagnosis.

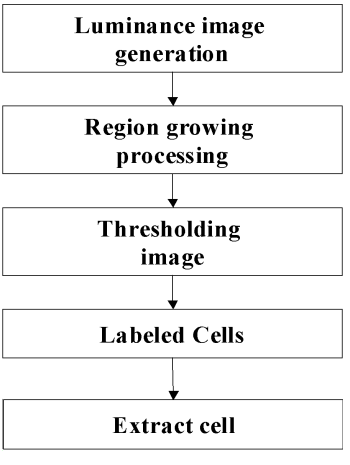



**Fig. 1.** Extracting the nucleus of cervix uteri cytodiagnosis

### 2.1 Region Growing Technique

Region growing technique used in this paper expands or segments the area by analyzing pixel similarity. First, we set a center area and check the neighbor pixels if they belong to the center area. The decision criterion of area membership is shown in formula (1) where G(A), G(B) denote the gray level brightness of pixel A and B and T is a threshold.

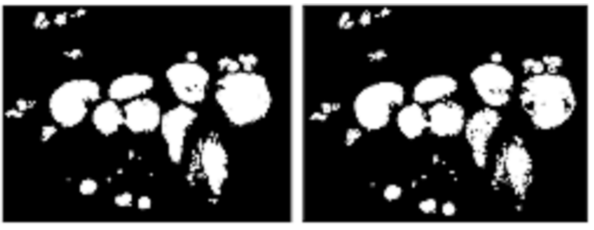



**Fig. 2.** Cell binarization

$$G(A) - G(B) < T \tag{1}$$

The center area is expanded including pixel A and B if the formula (1) holds. The area is expanded until there does not exist any neighbor pixels whose dissimilarity is less than the threshold T. Figure 2 shows the results of binarized images with and without the region growing technique. One can easily see that Fig. 2(b) has more damage in cell area than (a).

### 2.2 Image Binarization and Cell Area Extraction with Thresholding Technique

Thresholding technique is simple and fast image binarization technique with fixed threshold but has difficulty in area segmentation of complex images. However, we can find the center coordinates by extracting part of nucleus area with this technique since the brightness histogram of cell image can differentiate nucleus, cytoplasm, and background shown as fig. 3.

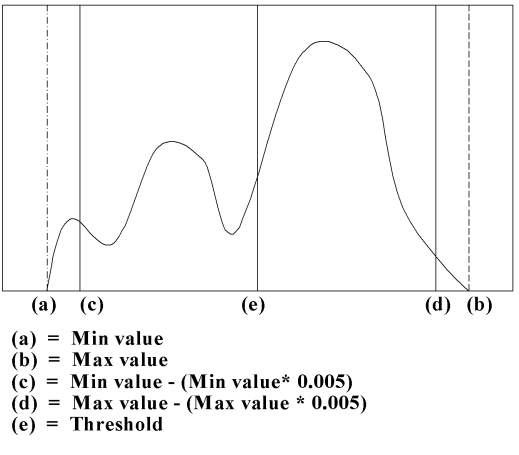



**Fig. 3.** Histogram and threshold

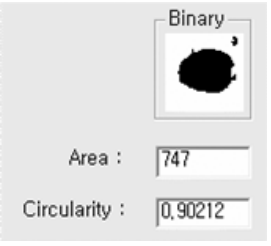



**Fig. 4.** Area and circularity of a cell

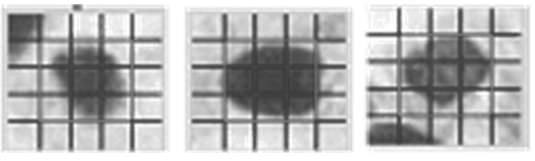


**Fig. 5.** Segmented cell image

We assume that the pixels that have up to 0.5% higher intensity from the minimum and 0.5% lower intensity from the maximum value are noises and set the new maximum and minimum (fig. 3 (c) and (d)). The center of the new limits becomes the threshold of the binarization. Next, 8-directional tracking algorithm is applied to extract the cell area. We compute the area and the circularity as features of the classification of the normality of cells.

The circularity is computed by equation (2) and the result is shown as fig. 4. where, perimeter in equation (2) means the perimeter of the nucleus.

$$Circularity = \frac{Perimeter^2}{4\pi Area},\ perimeter = 2\pi r \tag{2}$$

The extracted cell is divided to nucleus, cytoplasm, and background shown as fig. 5.

### 2.3 Nucleus Feature Extractions Using K-Means Clustering Algorithm

Both the cells and the nuclei characteristically display pleomorphism - variation in size and shape. Cells are often many times larger than their neighbors, and other cells may be extremely small and primitive appearing. Characteristically the nuclei contain an abundance of DNA and are extremely dark staining (hyperchromatic). The nuclei are disproportionately large for the cell, and the nuclear-cytoplasmic ratio may approach 1:1 instead of the normal 1: 4 or 1: 6. The nuclear shape is extremely variable, and the chromatin is often coarsely clumped and distributed along the nuclear membrane. Large nucleoli are usually present in these nuclei.

We can extract morphometric features, densitometric features, colorimetric features, and a textural feature from cervix cell. Each extracted nucleus is transformed to the original RGB space. Then the K-Means clustering algorithm is employed to classify RGB pixels to the R, G, and B channels, respectively. Finally, the Hue information of nucleus is extracted from the HSI models that are transformed using the clustering values in R, G, and B channels.

The main reason that we use the color model and hue information as the basis of classification is by the observation that the nucleous of cancer cell is much larger in size and more pachychromatic than normal cells.

K-Means is a well-known clustering algorithm that classifies the input into K groups of similar groups. We perform clustering to the R, G, and B channels, which are composed of R, G, and B values of pixels in each block. In our experiment, K = 10 and for 25 blocks, the codebook of 9 blocks in the center which composes a nucleus are created. Then, we transform the cluster values of codebook into HSI model with equation (3) and the Hue information becomes the input pattern of the fuzzy RBF network in order to recognize and classify nucleuses.

$$H = \cos^{-1}\left[\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right] \tag{3}$$

### 2.4 Nucleus Recognition and Classification Using Fuzzy RBF Network

We apply the fuzzy RBF network in classifying normal and abnormal cells using normalized Hue information obtained through the process explained in section 2.3.

Our system uses the fuzzy C-Means algorithm[6] to generate the middle layer. It is often criticized by consuming too much time when applied to character recognition. In character recognition, a binary pattern is usually used as the input pattern.

The fuzzy RBF networks can be summarized as follows.

1. The connection structure of input layer and middle layer is the same as in the fuzzy C-Means algorithm whose output layer is the middle layer of the proposed learning structure.
2. The node of middle layer denotes a class. Thus, though being a complete connection structure as a whole, we adopts the winner node method which back-propagates the weight connected with the representative class in terms of comparing the target vector with the actual output vector.
3. The fuzzy C-Means algorithm selects the node of middle layer with the highest membership degree as the winner node.
4. The generalized delta learning method is applied to the learning structure of middle layer and output layer in terms of supervised learning.

## 3 Experiment

The environment of the experiment is embodied by Visual C++ 6.0 in Pentium IV PC of IBM compatible. The specimen was 20 samples of 640*480 cervix uteri

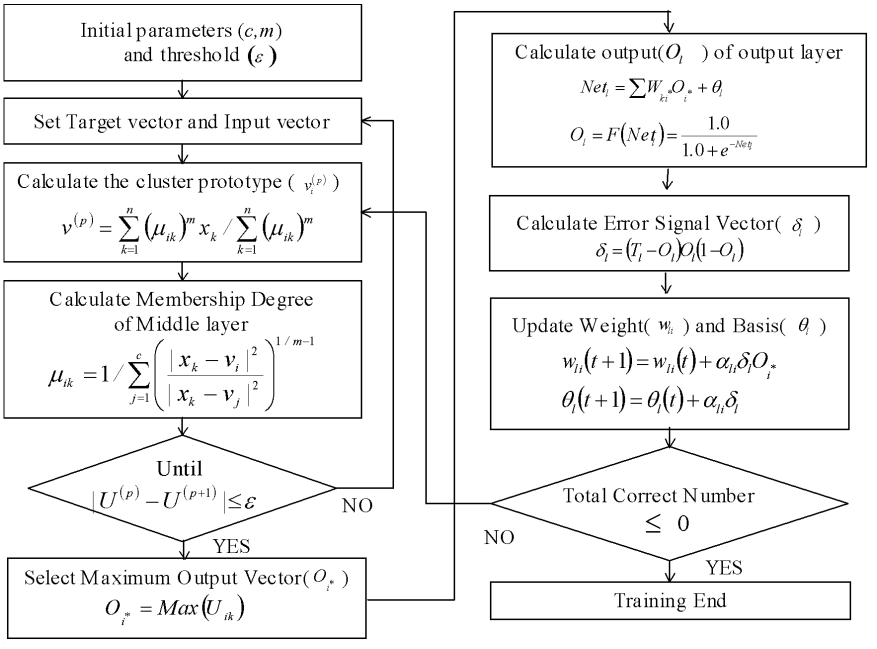



**Fig. 6.** Fuzzy RBF network

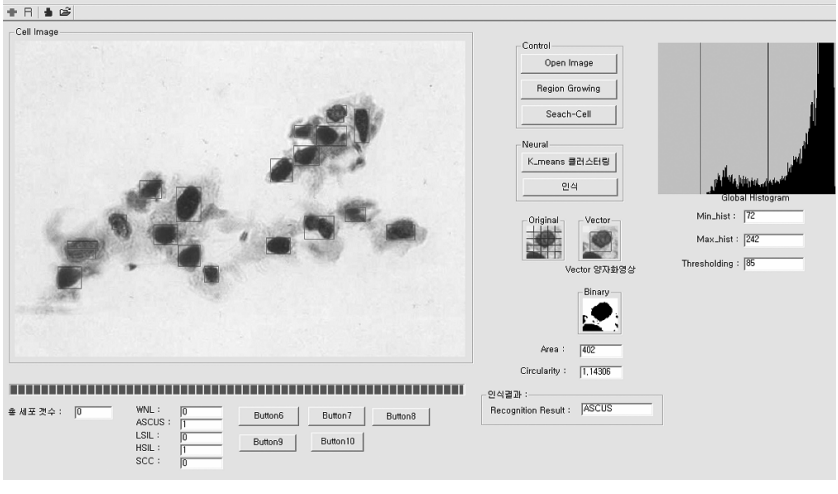



**Fig. 7.** Extraction of cervix uteri cytodiagnosis

cytodiagnosis image size. The nucleus result of cervix uteri cytodiagnosis image by proposed method is shown as fig. 7.

Table 1 shows the result of extracting nucleus by proposed method. Overall, the extraction rate is 85% , which is fairly good in practice.

**Table 1.** Extraction rate of nucleus

| | Population | Extracted by our method | Extraction rate |
|---|---|---|---|
| Number of nucleus | **316** | 259 | 85 |

**Table 2.** Training and test data

| Cell class | Training set | Test set |
|---|---|---|
| WNL | 60 | 94 |
| ASCUS | 60 | 78 |
| LSIL | 60 | 65 |
| HSIL | 20 | 25 |
| SCC | 10 | 7 |
| Total | 210 | 269 |

**Table 3.** Result of the classification

| | WNL | ASCUS | LSIL | HSIL | SCC | Overall-accuracy | Class-accuracy |
|---|---|---|---|---|---|---|---|
| WNL | 73 | 9 | 10 | 2 | 0 | 77.6% | 77.6% |
| ASCUS | 21 | 43 | 5 | 9 | 0 | 55.1% | 73.0% |
| LSIL | 13 | 5 | 39 | 7 | 1 | 60.0% | 80.0% |
| HSIL | 1 | 2 | 2 | 17 | 3 | 68.0% | 96.0% |
| SCC | 0 | 0 | 0 | 1 | 6 | 85.7% | 100.0% |

Table 2 shows the training and test data distribution for the fuzzy RBF network. We follow the Bethesda system classification in that cells are classified into five classes such as WNL, ASCUS, LSIL, HSIL, and SCC where WNL is normal and others are abnormal (2-class problem) and SCC is the cancer cell. The total size of training data is 210 and test data is 269. The training data is obtained by medical experts of Pusan National University Hospital.

The neural net used in this experiment has 90 input nodes, 20 middle layer nodes, and 3 output nodes. The number of training repetition is 1102 and the learning rate is 0.65. The evaluation of our method is done in two ways. Overall accuracy is the result of 2-class problem that discriminates normal (WNL) vs. abnormal (all others). The class accuracy denotes the recognition accuracy over five classes defined by the Bethesda system (5-class problem). Table 3 summarizes the experimental result.

In summary, our approach shows 80% overall accuracy (2-class problem) and 66% class accuracy by the Bethesda system (5-class problem). The source of misclassification might be the training data obtained by human expert's judgment. Since the previous study[7] shows that the performance of human expert is just as good as that of a fuzzy clustering method in clinical data.

## 4 Conclusions

Cervix uteri cytodiagnosis is complicated and varies differently so that it is difficult to extract and identify cell nucleus efficiently by existing image processing methods. In this paper, we proposed a region-growing technique to segment cell area efficiently and a fuzzy RBF network to identify abnormal cells. In order to provide informative input to the neural network, we applied K-Means clustering algorithm and transformed that information to HSI model. The Hue information obtained in that process played an important role in abnormal cell recognition. The fuzzy RBF Network is then applied to classify and identify the normal or abnormal nucleus. The result seems to be acceptable for the practioner's viewpoint.

In the future, we will try to analyze more morphometric features such as structural change and color change of abnormal cells to improve our algorithm in nucleus cell segmentation and will try to compare it with real world clinical data.

## References

1. Heinz, K.G., Nassem, H.: Automated Cervical Cancer Screening. Igaku-shoin, (1994)
2. Seo, C.W., Choi, S.J., Hong, M.K., Lee, H.Y., Jeong, W.G.: Epidemiologic Observation of Diagnosis of Cervical Cancer and Comparative Study of Abnormal Cytologic Smear and Biopsy Result. Journal of The Korean Academy of Family Medicine, 17(1) (1996) 76-82
3. Hugo, B.G., Ian, R., Alistair, Kudair, C., James, H., Tucker, H., Nasseem, H.: Automation in cervical cytology: an overview. Analytical Cellular Pathology, 4 (1992) 25-48
4. Lee, J.D.: Color Atlas Diagnostic Cytology. Press of Korea Medical Publishing Company, (1989)
5. Kim, H.Y., Kim, S.A., Choi, Y.C., Kim, B.S., Kim, H.S., Nam, S.E.: A Study on Nucleus Segmentation of Uterine Cervical Pap-Smears using Multi Stage Segmentation Technique. Journal of Korea Society of Medical Informatics, 5(1) (1999) 89-95
6. Kulkarmi, A. D.: Computer Vision and Fuzzy-Neural Systems: Prentice Hall, (2001)
7. Kim, K.B., Kim, C. S., and Kim, S.: Nucleus Classification of uterine cervical papsmears using FCM clustering algorithm. Proceedings of SCIS & ISIS2006, Tokyo, Japan, (2006) 1084-1088

# A Study: Segmentation of Lateral Ventricles in Brain MRI Using Fuzzy C-Means Clustering with Gaussian Smoothing

Kai Xiao, Sooi Hock Ho, and Qussay Salih

Faculty of Engineering and Computer Science
The University of Nottingham Malaysia Campus
Jalan Broga, 43500 Semenyih, Selangor Darul Ehsan, Malaysia
kcx2kx@nottingham.edu.my,
Ho.Sooi-Hock@nottingham.edu.my,
Qussay.A@nottingham.edu.my

**Abstract.** This paper demonstrates a study on lateral ventricles segmentation in brain Magnetic Resonance Imaging (MRI). The method applies Gaussian smoothed image data as additional features into the feature space of Fuzzy C-Means (FCM) algorithm. With the aid of the smoothing effect from Gaussian filters, FCM is able to segment lateral ventricular compartments by reducing inappropriate clustering caused by noise and inhomogeneous intensity distribution. The results demonstrate both noise insensitivity and more homogeneous clustering.

**Keywords:** Fuzzy C-Means, Clustering, Gaussian Smoothing, Segmentation, Validity Functions, brain MRI, Lateral Ventricles

## 1 Introduction

Image segmentation is the process of assigning pixels to regions sharing common properties. It is one fundamental process in computer vision and pattern recognition because further processing steps have to rely on the segmentation results. Despite the numerous segmentation techniques, image segmentation is still a subject requiring intensive exploration due to the diversity within each application[1-5].

In brain Magnetic Resonance Imaging (MRI), noise, spurious blobs, inhomogeneous pixel intensity distribution and blunt region boundary in the ventricular compartments are the main challenges for lateral ventricles segmentation[1,2]. As a consequence, selecting an appropriate segmentation method is of utmost importance.

Fuzzy C-Means (FCM) clustering[1,6,12] is an unsupervised method that has been effectively applied to several application areas including image segmentation. In applying FCM algorithm to image segmentation, images represented in feature space are classified by FCM through grouping data points with similar property, e.g., intensity value in MRI application. With iterative minimization of the cost function that is dependent on distances between pixels and group

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 161–170, 2007.

centers, its membership function and the cluster centers are updated until a termination criterion is reached.

Gaussian 2-dimensional convolution operator can be used to smooth images and remove detail and noise. The smoothing degree is determined by the standard deviation of the Gaussian[13,14]. The Gaussian smoothed image data can be added into the feature space associated with the original image and used as input feature space data in FCM algorithm. The cost function in FCM algorithm will be updated according to the distance between pixels in multiple dimensions, as a result of combining all the feature data into the multiple feature space. Clustering will be affected by both the original image and the smoothed image data which leads to better clustering results.

The aim of this study is to find a more appropriate method to apply FCM clustering in brain MRI lateral ventricle segmentation. The experimental results demonstrate that this approach performs better than that of FCM without Gaussian combination.

## 2 Methods

### 2.1 FCM Clustering

The FCM algorithm groups one piece of data to two or more clusters, where data is represented by image pixels. Let X($x_1$, $x_2$,..., $x_N$) denotes an image with $N$ pixels to be partitioned into $c$ clusters, where $x_i$ represents multi-feature data. The algorithm is an iterative optimization that minimizes the cost function defined as follows:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{c} \mu_{ij}^m \left\| x_i - c_j \right\|^2 \qquad , 1 \leq m < \infty \tag{1}$$

where $\mu_{ij}$ represents the degree of membership of $x_i$ in the $j$th cluster, $x_i$ is the $i$th measured data, $c_j$ is the $j$th cluster center, $\| * \|$ is any norm expressing the similarity between any measured data and the cluster center[3,6,12], where standard Euclidean distance metric is generally applied for the multi-feature data, and $m$ is a real constant greater than 1 which controls the fuzziness of the resulting partition.

When pixels close to the center of their clusters are assigned high membership values, and low membership values are assigned to pixels with data far from the center, cost function is minimized. The membership function represents the probability that a pixel belongs to a specific cluster. For each pixel, the sum of probabilities in each cluster will remain the same constant as '1' in this study. In the FCM algorithm, the probability is dependent on the distances between the pixel and each individual cluster centers in the feature domain. The membership functions and cluster centers are iteratively updated as follows:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}} \tag{2}$$

and

$$C_j = \frac{\sum_{i=1}^{N} \mu_{ij}^m x_i}{\sum_{i=1}^{N} \mu_{ij}^m} \tag{3}$$

The iteration will stop when $\max\{|\mu_{ij}^{(k+1)} - \mu_{ij}^{(k)}|\} < \sigma$, where $\sigma$ is a termination criterion between 0 and 1, and $k$ is the iteration step. This procedure converges to a local minimum or a saddle point of $J_m$ in equation (1).

### 2.2 Applying Gaussian Smoothing Operator into FCM

The Gaussian image smoothing operator is a widely used 2-Dimension convolution operator that is used to blur or smooth image and remove detail and noise. A 2-D circularly symmetric Gaussian has the form:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

In its blurring action, Gaussian filter provides gentler smoothing and preserves edges better than a similarly sized mean filter because Gaussian function outputs a 'weighted average' of each pixel neighborhood, with the average weighted more towards the value of the central pixels[14].

With the fact that the standard FCM algorithm supports multiple feature inputs, the original image and its Gaussian filtered image pixel values can be combined as a multi-dimensional matrix. These are used as input data for the FCM algorithm. The cost function in an FCM algorithm will be updated according to the distance between pixels in multiple dimensions. As a result of combining all the features into the multi-dimensional space, clustering will be affected by both the original image and the smoothed image data, leading to more noise-insensitive and more homogeneous clustering results.

Because of the fact that Gaussian smoothing can preserve edges better, the boundary in the region of interest (ROI) will remain, though Gaussian filtered data feature has a blurring effect on the clustering results.

### 2.3 Cluster Validity Functions

In this study, fuzzy partition is used to evaluate the performance of clustering. The representative functions for the fuzzy partition are partition coefficient $V_{pc}$[7] and partition entropy $V_{pe}$[8]. They are defined as follows:

$$V_{pc} = \frac{\sum_{j}^{N} \sum_{i}^{c} \mu_{ij}^2}{N} \tag{5}$$

and

$$V_{pe} = \frac{-\sum_{j}^{N} \sum_{i}^{c} [\mu_{ij} \log \mu_{ij}]}{N} \tag{6}$$

The idea of these validity functions is that the partition with less fuzziness means better performance. In both equation (5) and (6), $\mu_{ij} (i = 1, 2, ...c;\ j =$

$1, 2, ...N$) is the membership of data point $j$ in cluster $i$. The closer this value is to unity the better the data are classified. As a result, the best clustering is achieved when $V_{pe}$ is minimal and $V_{pc}$ is maximal[6].

To quantify the ratio of total variation within clusters and the separation of clusters, another validity function $V_{xb}$[9,10] is used as:

$$V_{xb,m} = \frac{\sum_{j=1}^{N} \sum_{i=1}^{c} (\mu_{ij})^m \left\| x_j - v_i \right\|^2}{N * \left( \min_{i,k} \left\{ \left\| v_k - v_i \right\|^2 \right\} \right)} \tag{7}$$

where $\|x_j - v_i\|$ denotes the Euclidean distance between the pattern, $x_j$ and the cluster center, $v_i$, and $\min_{i,k} \left\{ \|v_k - v_i\|^2 \right\}$ represents the minimum Euclidean distance between cluster centers, $v_i \neq v_k$.

An optimal clustering result generates samples that are within one cluster and samples that are separated between different clusters. Minimized $V_{xb}$ represents a good clustering result.

### 2.4 Image Data

For this study, the images were collected from the Internet MRI atlas [11]. To focus on ventricle segmentation, one pair of T1-weighted and T2-weighted MRIs in the trans-axial view with the same slice number (which indicates they were taken from the same area of the brain) and displaying the most noticeable lateral ventricular compartments were selected. To demonstrate the effect of noise on the segmentation process, noisy images have been created by adding Gaussian white noise with different SNR to the original images.

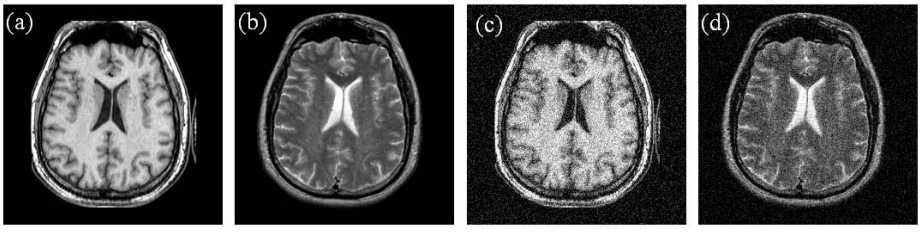



**Fig. 1.** (a) T1 and (b) T2 original images. (c) T1 and (d) T2 images added with noise of SNR=10.

## 3 Results and Discussion

Fig. 1(a) and (b) illustrate the T1 and T2 original images selected for this study, respectively. Fig. 2(a) and (b) show the segmentation results obtained by using a standard FCM with T1 and T2 image data under 3 and 5 clusters, respectively. Fig. 2(c) and (d) show the segmentation results obtained by applying combined

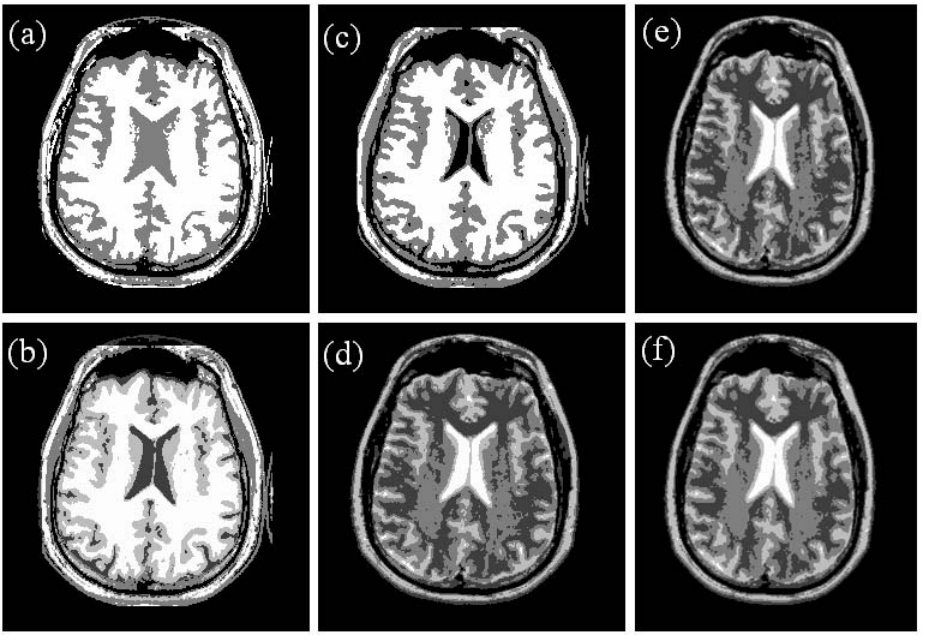



**Fig. 2.** Segmented images of MRI images using FCM with features of (a) T1 + T2 images under 3 clusters; (b) T1+ T2 images under 5 clusters; (c) T1+ its Gaussian smoothed images under 3 clusters, Gaussian filter kernel in size of 5 and sigma of 5; (d) T2 + its Gaussian smoothed images under 5 clusters, Gaussian filter kernel in size of 5 and sigma of 5; (e) T2 + its Gaussian smoothed images under 5 clusters, Gaussian filter kernel in size of 5 and sigma of 10; (f) T2 + its Gaussian smoothed images under 5 clusters, Gaussian filter kernel in size of 10 and sigma of 5.

features of T1 image and its Gaussian smoothed image under 3 clusters and T2 image with its Gaussian smoothed image under 5 clusters, respectively. Fig. 3 illustrates and compares the extracted ventricular compartments segmentation results from clusters after FCM with different feature inputs and cluster number.

Conventional FCM with the combined features of T1 and T2 images is able to classify MRI images. However, the two parts of the lateral ventricular compartments are joined together in different levels; by combining the original image and the Gaussian smoothed image features, its counterpart shows the two fully separated compartments. Adding Gaussian smoothed image data into FCM reduces the number of spurious blobs, and the segmented images are more homogeneous. The possible disadvantage of applying Gaussian smoothing filters is the blurring effect on some fine details, especially when the Gaussian filter is of bigger filter kernel size. In Fig. 2(f) the two compartments are again joined together while the clustering result shows further removals of small spurious blobs when a Gaussian filter kernel size of 10 is applied. In Fig. 2(e), the clustering result does not change much when compared to Fig. 2(d), although a doubled sigma value has been used.

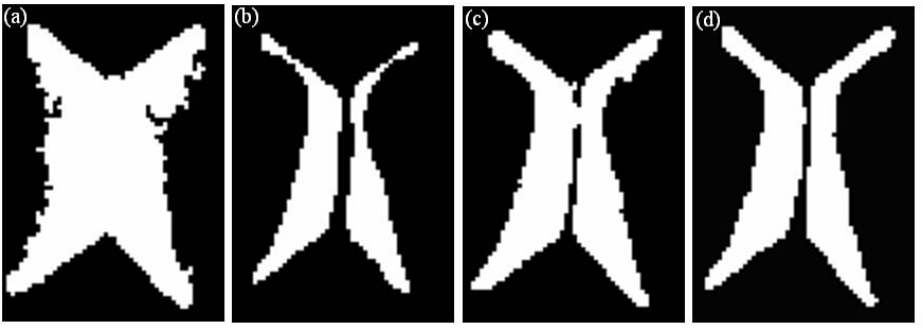



**Fig. 3.** Extracted lateral ventricular compartments after FCM clustering with features of (a) T1 + T2 images under 3 clusters; (b) T1+ its Gaussian smoothed images under 3 clusters, Gaussian filter kernel in size of 5 and sigma of 5; (c) T1+ T2 images under 5 clusters; (d) T2 + its Gaussian smoothed images under 5 clusters, Gaussian filter kernel in size of 5 and sigma of 5.

Fig.1(c) and 1(d) show the T1 and T2 added with a noise of SNR=10. Fig. 4 shows the segmented result of applying noisy T1 + T2 image data and T2 + Gaussian smoothed image data into FCM, respectively.

As can be seen, the standard FCM technique misclassifies ventricular compartments at numerous places because the added noise changes the location of pixels inside the ventricular compartments in the feature space, causing the misclassification of these noisy pixels. When the T1 image data is replaced by the Gaussian smoothed image as input feature, the weight of the noisy cluster is

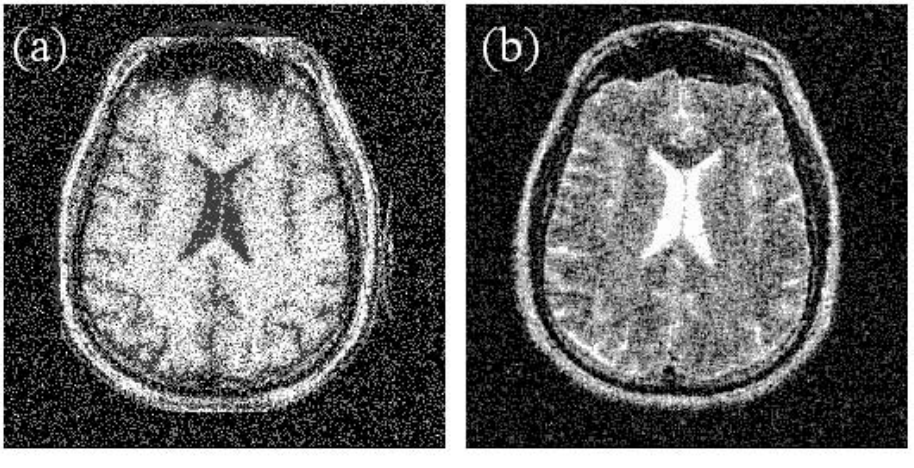



**Fig. 4.** Segmented images of noisy MR images using FCM with features of (a) T1+ T2 images under 5 clusters; (b) T2 + its Gaussian smoothed images under 5 clusters

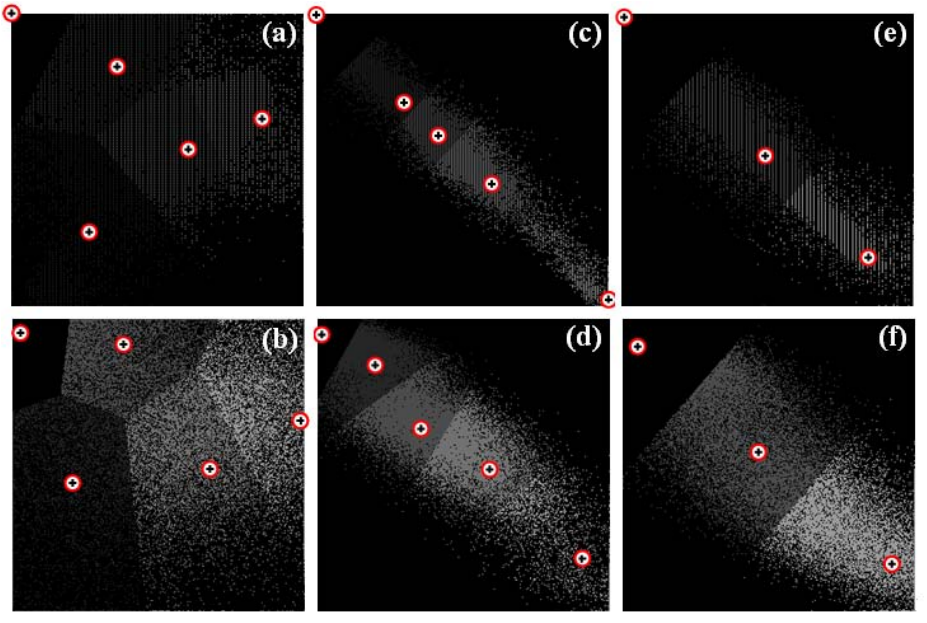



**Fig. 5.** The calculated centroids (⊙) of cluster in the feature domain (where two features pixel values are coordinates) for FCM with features of (a) original T1 and T2 images; (b) noisy T1 and T2 images; (c) original T2 image and its Gaussian smoothed image; (d) noisy T2 image and its Gaussian smoothed image; (e) original T1 image and its Gaussian smoothed image; (f) noisy T1 image and its Gaussian smoothed image

greatly reduced. Furthermore, the membership of the correct cluster is enhanced by the cluster distribution in the combined feature spaces. As a result, image as feature input effectively corrects misclassifications caused by noise.

Fig. 5 shows the segmentation results of MRI images in the feature domain for FCM using different features. Fig. 5(a) shows the result of using T1 and T2 images without added noise as a feature input. Fig. 5(b) shows the result of using T1 and T2 images with added noise of SNR=5 as an input feature. Fig. 5(c) shows the result by using T2 image and its respective Gaussian smoothed image without added noise as the input features. Fig. 5(d) shows the result of using T2 image and its respective Gaussian smoothed image with added noise of SNR=5 as an input feature. As can be seen, the center distribution pattern in FCM with Gaussian smoothed image is less changed than that in the FCM with two original images as the input data. The techniques of input features combined with original image data and its Gaussian smoothed image successfully correct the misclassified pixels and kept the cluster centers less affected by noise.

Table 1 tabulates the standard deviations of the clustering centroids direction change rate caused by noise. As can be seen in Fig. 6, each cluster center direction can be represented by the ratio of two coordinate differences that are calculated by the coordinates of the current centroid to its previous cluster centroid, e.g.,

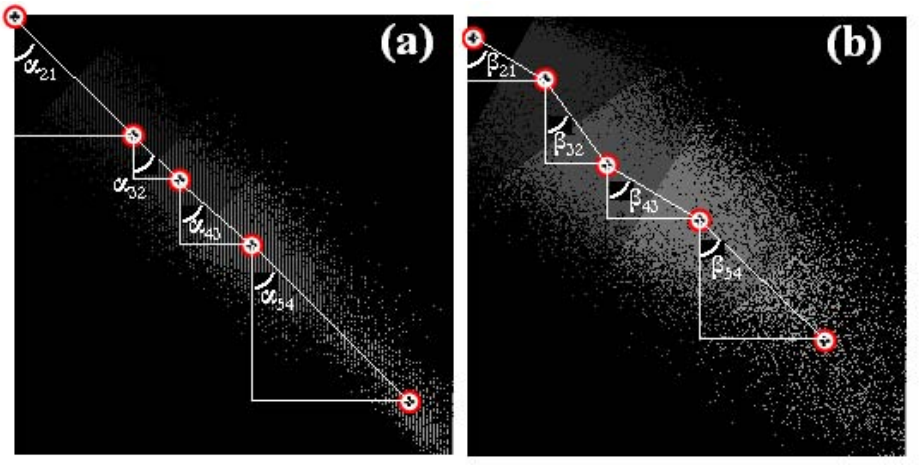



**Fig. 6.** An illustration of FCM clustering centroids direction distribution under input features of (a) original T2 + Gaussian smoothed image; (b) noisy T2 + Gaussian smoothed image

**Table 1.** Standard deviations of the clustering center direction change rate

| Cluster Number | SNR of Noise Added | STDEV on direction change ratio of T1 + T2 images clustering centers | STDEV on direction change ratio of T1/T2 + Gaussian smoothed image clustering centers |
|---|---|---|---|
| 3 | 5 | 0.431897641 | 0.0356923166 |
| | 10 | 0.164503921 | 0.093785356 |
| | 20 | 0.006078361 | 0.014945789 |
| 5 | 5 | 0.514171017 | 0.393363852 |
| | 10 | 0.603437368 | 0.371229271 |
| | 20 | 0.058867462 | 0.044899522 |

$\alpha_{32}$ and $\beta_{32}$ represents the direction from FCM clustering centroid number 3 to centroid number 2 under input features of original image and noisy image, respectively. After calculation the ratio for each respective $\alpha$ and $\beta$, we get a dataset of direction change ratios. Finally the standard deviation (STDEV) of this dataset is used to represent the cluster centers distribution change pattern by the effect of added noise. The effect of wrong clustering from noise input will be minimized when the cluster centers distribution change pattern value approaches 0. In Table 1, most of the results in the cases when T1 or T2 together its Gaussian smoothed image are applied in the feature domain are less than that when T1 and T2 images without Gaussian are applied. This further explains that adding the Gaussian smoothed image into the FCM algorithm leads to more noise-insensitive clustering results.

Table 2 tabulates the validity functions used to evaluate the performance of FCM clustering for six images. In all cases, the validity functions based on the

**Table 2.** The clustering results of six images using FCM with different feature data

| Images | Cluster Number | Features used in FCM | $V_{pc}$ | $V_{pe}$ | $V_{xb}$ |
|---|---|---|---|---|---|
| Original MR images | 3 | T1 + T2 | 0.8400 | 0.1234 | 9.5464 |
| | | T1+Gaussian | 0.8872 | 0.0894 | 4.8160 |
| | 5 | T1+T2 | 0.7967 | 0.1789 | 6.6535 |
| | | T1+Gaussian | 0.8438 | 0.1323 | 4.0191 |
| Noise added MR images SNR=5 | 3 | T1 + T2 | 0.7119 | 0.2252 | 16.2653 |
| | | T1+Gaussian | 0.7906 | 0.1672 | 9.3009 |
| | 5 | T1+T2 | 0.6198 | 0.3349 | 11.7261 |
| | | T1+Gaussian | 0.6914 | 0.2646 | 7.4524 |
| Noise added MR images SNR=10 | 3 | T1 + T2 | 0.7787 | 0.1769 | 12.9993 |
| | | T1+Gaussian | 0.8408 | 0.1302 | 7.2333 |
| | 5 | T1+T2 | 0.6557 | 0.2962 | 13.3426 |
| | | T1+Gaussian | 0.7457 | 0.2164 | 7.4782 |

fuzzy partition were better for the FCM with Gaussian smoothed image feature than the conventional FCM with features of T1 and T2 images.

## 4 Summary

FCM clustering is an unsupervised clustering technique to segment images into clusters with similar spectral properties. It utilizes the distance between pixels and cluster centers in the spectral domain to compute the membership function. With the capacity of multi-feature support in FCM algorithm, Gaussian smoothed images can be added into the feature domain as an additional feature to affect the clustering and to lead to more homogeneous and noise-insensitive segmentation results.

In this paper, focusing on brain ventricular compartments segmentation, we did an intensive study on applying Gaussian smoothing into FCM and attempting to incorporate the smoothed image as an additional feature for the FCM algorithm in order to improve the segmentation results. This correlation between Gaussian smoothed image and original image in the membership and cost function of FCM algorithm reduces the number of spurious blobs and biases the solution toward homogeneous labeling. The method was tested on MRI images and evaluated by using various cluster validity functions. Preliminary results showed that the effect of noise in segmentation was less with the approach we proposed than with the FCM without additional Gaussian smoothed image.

## References

1. Andrew J. Worth, Nikos Makris, Mark R. Patti, Julie M. Goodman, Elizabeth A. Hoge, Verne S. Caviness, Jr., David N. Kennedy.: Precise Segmentation of the Lateral Ventricles and Caudate Nucleus in MR Brain Images using Anatomically Driven Histograms. IEEE Transactions on Medical Imaging (1997)

2. Wu Y, Pohl K, Warfield SK, Cuttmann CRG.: Automated Segmentation of Cerebral Ventricular Compartments. International Society for Magnetic Resonance in Medicine Eleventh Scientific Meeting and Exhibition, Toronto, Ontaril, Canada. (2003)
3. Bezdek J, Hall L, Clarke L.: Review of MR image segmentation using pattern recognition. Med Phys. **20** (1993) 1033-1048
4. Brandt ME, Bohan TP, Kramer LA, Fletcher JM.: Estimation of CSF, white matter and gray matter volumes in hydrocephalic children using fuzzy clustering of MR images. Comput Med Imaging Graph. **18** (1994) 25-34
5. Pham DL, Prince JL.: Adaptive fuzzy segmentation of magnetic resonance images. IEEE Trans Med Imaging **18** (1999) 737–752
6. Chuang KS, Tzeng HL, Chen S, Wu J, Chen TJ.: Fuzzy c-means clustering with spatial information for image segmentation. Computerized Medical Imaging and Graphics **30** (2006) 609–633
7. Bezdek JC, P.: Cluster validity with fuzzy sets. J Cybern . **3** (1974) 58–73
8. Bezdek JC, P.: Mathematical models for systematic and taxonomy. proceedings of eigth international conference on numerical taxonomy, San Francisco. (1975) 143–166
9. Xie XL, Beni GA.: Validity measure for fuzzy clustering. IEEE Trans Pattern Anal Mach Intell. **3** (1991) 841–846
10. Chen CF, Lee JM.: The Validity Measurement of Fuzzy C-Means Classifer for Remotely Sensed Images. Asian Conference on Remote Sensing. (2001)
11. Harvard Medical School: The Whole Brain Atlas.
*http* : *//www.med.harvard.edu/AANLIB/home.html*
12. Dipartimento di Elettronica e Informazione, Politecnico di Milano: Clustering Fuzzy C-Means.
*http* : *//www.elet.polimi.it/upload/matteucc/Clustering/tutorial*$_h$*tml/cmeans.html*
13. Wikipedia.: Gaussian blur.
*http* : *//en.wikipedia.org/wiki/Gaussian*$_b$*lur/*
14. School of informatics, The University of Edinburgh: Spatial Filters Gaussian Smoothing.
*http* : *//homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm*
15. Wikipedia: Signal-to-noise ratio.
*http* : *//en.wikipedia.org/wiki/Signal* $-$ *to* $-$ *noise*$_r$*atio*

# Ischemic Stroke Modeling: Multiscale Extraction of Hypodense Signs

Artur Przelaskowski[1], Pawel Bargiel[2], Katarzyna Sklinda[3], and Elzbieta Zwierzynska[4]

[1] Institute of Radioelectronics, Warsaw University of Technology Nowowiejska 15/19, Warszawa, Poland
arturp@ire.pw.edu.pl
[2] Institute of Radioelectronics, Warsaw University of Technology Nowowiejska 15/19, Warszawa, Poland
pbargiel@ire.pw.edu.pl
[3] Department of Radiology CMKP, CSK MSWiA, Woloska 137, Warszawa, Poland
hogaforever@gmail.com
[4] Department of Neurology CMKP, CSK MSWiA, Woloska 137, Warszawa, Poland
elzbieta.zwierzynska@cskmswia.pl

**Abstract.** Multiscale extraction of the subtlest signs of hypodensity, which were often undetected in standard CT scan review was the subject of our research. Proposed method is as follows: evidence-based description of hypodense changes, the investigation of hypodensity across scales, basing on a set of over 20 hyperacute stroke exams, the improvement of wavelet-based display of ischemic stroke. Considered problems were: –extension of the brain tissues for marginal and missing space after deskulling and segmenting of unusual areas; –best basis selection;–non-perfect reconstruction across scales as an extraction of hypo-attenuating tendency. Increased visibility of hypodense signs on CT scans performed in patients with hyperacute stroke was noticed in subjective rating. In opinion of radiologists and image processing experts, enhanced perception of hypodense area was noticed for all test exams. The rate of unique, clear and doubtless extraction of hypodense area was 92% in 13 tested cases of hyperacute ischemic stroke.

**Keywords:** Ischemic stroke detection, wavelet-based image processing, medical image perception.

## 1 Introduction

Stroke is the clinical syndrome of rapid onset of focal, or sometimes global, cerebral deficit with a vascular cause, lasting more than 24 hours or leading to death. Infarction may occur in any area of the brain following vascular territory or watershed distribution. Brain imaging is required to guide the selection of acute interventions to treat patients with a stroke, which is very important for the stroke emergency centers. The recent advent of thrombolytic therapy for acute

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 171–181, 2007.

stroke treatment makes as early as possible detection of areas of hypoattenuating ischemic parenchyma exceedingly important [1,2,3,4].

For most cases, CT remains the most important brain imaging test. Irreversible ischemic injury would be represented by a focal hypodense area, in cortical, subcortical, or deep gray or white matter. A hypodense area is defined as any area in the brain with density lower than normal brain tissues. On the initial CT-scan, performed during the hyperacute phase of stroke (0-6h), the mentioned hypodensity does not have to be seen. Early indirect findings, like obscuration of gray/white matter differentiation and effacement of sulci, or "insular ribbon", may be noticed instead.

Focal hypodense changes were found to be the most frequent and reliable signs of early ischemia. A decline in cerebral blood flow causes the brain tissue to take up water immediately. Thus, in the early stage of cerebral ischemia, the tissue changes consist mainly in alteration of water and electrolyte content. Parallel intracellular increase of sodium and a decrease of potassium concentration occur. A 2-4% increase in brain tissue water within 4 h of MCA occlusion was noticed in several experiments [3,5,6]. Increase of water content causes the lowering of brain attenuation coefficients in acute ischemia, which leads to a discrepant decrease of about 1.3-2.6 HU for 1% change in water content [1,7,6]. The discrepancy of water uptake and density changes might suggest an incompleteness of ischemic physiology model and unclear impact of other factors, e.g. decreased lipids, increased protein and electrolyte changes.

However, subtle hypodense changes are often masked due to artifacts, noise and other tissue abnormalites. The attenuation coefficients of brain parenchyma vary, mainly due to the differing thickness of the cranial vault. Dense bone lowers the energy of the beam and thus, increases attenuation. M. Bendszus et al. [7] found inter-individual differences, i.e. bone artifacts, of up to 14 HU in brain parenchyma at comparable scan levels. The CT number (HU) for water should ideally be zero, but the actual value changes because of variations in the stability of the detector system or x-ray source. Normally, these variations (i.e. standard deviation of the water value) are very small and most scanners should be able to stay within 2HU of zero for water. The mean CT number measured over the central test ROI (region of interests) should be in the range of 4HU [8], which is close to the early changes within ischemic region.

It is evident that the early changes with ischemia occur, but may vary within the limited range of HU scale depending on cerebral infarct case, discrepant patient characteristics, and acquisition conditioning. The hypodense changes are slight, and ischemic area is not well-outlined or contrasted (with slow edges characterized by low-frequency spectrum). Because of the human eye limitations, these first ischemic signs can often be out of that range. Typical preview window of width 80 HU gives maximum noticeable change of 1-2 grey shade within the first 4 h of ischemia. Diffusely interspersed changes in grey shade can hardly be distinguished in noisy areas because of low brightness contrast, bone artifacts, non-optimum scanning.

Display of ischemic stroke as a kind of computer aided interpretation tool was designed to uncover, model and exploit hyposensity as a signature of pathology. We investigated multiscale wavelet-like methods for identifying the signatures of hypodensity. Signal and noise separation based on spatially distributed properties over different scales and subbands is usually much more effective than in image domain. Lower frequency parts offer distinguished information about poor textures and mean value estimates in regions. A correlation of high frequency information across scales, portrays even very weak edges and region distinction [9,10]. Therefore, noise and artifacts may effectively be reduced in multi-scale data processing [11,10,9]. Post-processing in wavelet domain was less susceptible to local perturbations, and beneficial noise suppression and selective contrast enhancement was possible. Especially, wavelet-based algorithms with adaptive histogram equalization were investigated as a method of automatic simultaneous display of the full dynamic contrast range of CT images (chest exams with lymph nodes, pericardial disease, air cysts) [12]. However, we found the adaptive histogram equalization in multi-scale data domain too coarse a method to enhance subtle distinction of attenuation coefficients because of high level of noise and artifacts presence in brain images.

The purpose of our study was to improve the diagnosis of hyperacute ischemic brain parenchyma on emergency CT scans. The method was the enhanced visibility of more distinguished or extracted subtle and hidden hypodense signs. Suggested wavelet-based post-processing algorithm was based on –extension of the brain for marginal (border effects) and missing space after deskulling and segmenting of unusual areas (e.g. sulci); –best basis selection; –non-perfect data reconstruction across scales to extract hypo-attenuating tendency.

## 2 Materials and Methods

Multiscale hypodensity modeling was based on the following data processing stages: –the initial gray-to-white tissue segmentation; –the next segmentation of potentially hypodense areas (e.g. sulci or the aged lesions); –noise suppression in selected ROIs through the non-perfect signal reconstruction in successive scales, basing on middle band suppressing orthogonal filter bank; – successively scaled orthogonal filtering with adaptive soft thresholding in a set of middle-frequency subbands for increasing the local mean data variability. Local contrast of the processed images was additionally improved by adaptive histogram equalization.

The proposed method was based on a concept of stroke display [13] implied as a kind of intelligent data visualization method that communicates selected, extracted, and enhanced ischemic hypodensity signs to the observers, especially for "radiologically silent" cases (really difficult to diagnose). It complements conventional CT display with additional display highly specific in infarct cases.

Initial two-stage segmentation of the regions susceptible to ischemic density changes was used to eliminate false diagnostic indications. Tissue features masking ischemic changes such as density distinctions caused by non-ischemic reasons unfavorably effecting diagnosis, may be suppressed by post-processing. False

positives have to be avoided, since treating ineligible patients with intravenous thrombolysis is associated with an unacceptable risk of hemorrhage and death. 3D region growing methods with interactively controlled distribution of the seeds was applied. Significant improvement of brain tissue segmentation over all slices, in comparison to the adaptive threshold methods [13] was noticed. Next, the susceptible-to-stroke ROI was detected in successive slices, through local adaptive thresholding based on the gray-to-white tissue histogram models.

Modeling of hypodense regions over successive scales and subbands with increased distinction from noisy background, was the subject of theoretical and experimental study. Decomposition scheme was firstly optimized in order to perceptually extract hypodense regions including orthogonal, biorthogonal, integer bases, undecimated wavelets, contourlets and data grid converters (i.e. hexagonal 2D kernels). Noise suppression and contrast enhancement was obtained by adjusting multi-scale coefficients of interests at some particular spatial-frequency locations, by modeling a distribution of their magnitudes in a context of surrounding data. Distortions caused by wavelet transform implementation with designed non-perfect reconstruction filter banks, shaped noised hypodensity signs in multiscale domain more effectively than typical denoising methods. Additional soft thresholding of the wavelet coefficients over scales was applied, in order to achieve sharper, non-spread outline of hypodense regions.

Preliminary subjective tests were performed to maximize hypodense signs extraction according to diagnostic performance criteria. Firstly, the subjective rating of over 30 test CT exams was used to select the most effective wavelet bases and multiresolution schemes. Next, the mentioned elements of proposed algorithm were designed and optimized basing on common neurological, radiological, and engineering consultations. Uniqueness and clearance of hypodensity determining was the most important optimization criterion of wavelet-based image post-processing as much as avoiding false indications. Four radiologists and two neurologists participated in that process, which resulted in proposed detailed algorithm of multiscale extraction of the signs of hypodensity.

### 2.1 Non-perfect Multiscale Image Processing

Although wavelet analysis possesses many attractive features, its numerical implementation is not as straightforward as that of FFT, STFT or e.g. DCT. However, wavelet transform in dyadic, 2D form can be implemented with specific types of digital filter banks known as two- channel perfect reconstruction (PR) filter banks (fig. 1).

Filters are associated with scaling functions and the wavelets of the transform kernel according to the following two equations (scaling and wavelet, respectively):

$$\phi(t) = \sqrt{2} \sum_n h_n \phi(2t - n) \tag{1}$$

and

$$\psi(t) = \sqrt{2} \sum_n g_n \phi(2t - n) = \sum_n (-1)^{1-n} h_{1-n} \phi(2t - n) \tag{2}$$

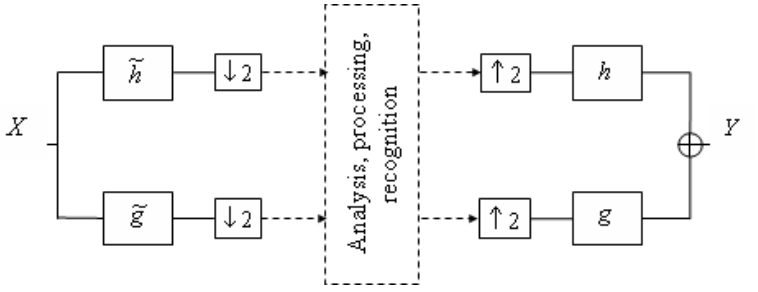



**Fig. 1.** Two-channel filter bank (FB) used with wavelet-based data analysis, processing and signature recognition. The signal $X$ is decomposed, processed and reconstructed to form $Y$ with low-pass and high-pass filters for analysis$(\tilde{h}, \tilde{g})$ and synthesis $(g, h)$, respectively. Downsampling and upsampling operators are $\downarrow 2$ and $\uparrow 2$, respectively.

Conditions of the perfect reconstructions ($Y$ is almost, i.e. according to the assumed precision, equal to $X$) with $l$ delays for two-channel FB are as follows:

$$h(z)\tilde{h}(-z) + g(z)\tilde{g}(-z) = 0 \tag{3a}$$

$$h(z)\tilde{h}(z) + g(z)\tilde{g}(z) = 2z^{-l} \tag{3b}$$

It often requires the filters from eq. 3 to be FIR (finite impulse response), linearly phased and form orthogonal FBs. The first term (eq. 3a) traditionally called the alias (cancellation) term is often fulfilled by using quadrature mirror filters (QMFs) with conditions: $h(z) = \tilde{g}(-z)$ and $g(z) = -\tilde{h}(-z)$, as we did. However, the second term (eq. 3b) called the distortion (elimination) term was used to control the distortion introduced in data processing to denoise and differentiate signal features. Magnitude responses of applied 3 taps spline FB, non–PR, called TSpline2 and 8 taps symmetric (almost PR) FB, called Atrial were presented in Fig. 2.

### 2.2 Algorithm of Hypodensity Extraction

The proposed algorithm is as follows:

1. Segmentation of susceptible-to-stroke ROI to be processed in successive slices
   - the brain extraction to remove non-brain tissue from a CT volume (to de-skull the brain in the image) through region growing, arranged in 3D space of successive slices; interactively controlled distribution of the seeds in order to control any irregular, untypical cases was applied;
   - selection of the only tissue regions which are probable to include ischemic stroke with adaptively set range of $[water + 18, ROImean + 15]$ HU to extract the brain tissue of gray matter to low white matter density and to get rid of clear brain sulci, old ischemic scars and other structures useless in early stroke detection; all pixels out of stroke tissue ROI are set to adaptively computed $ROImean$.

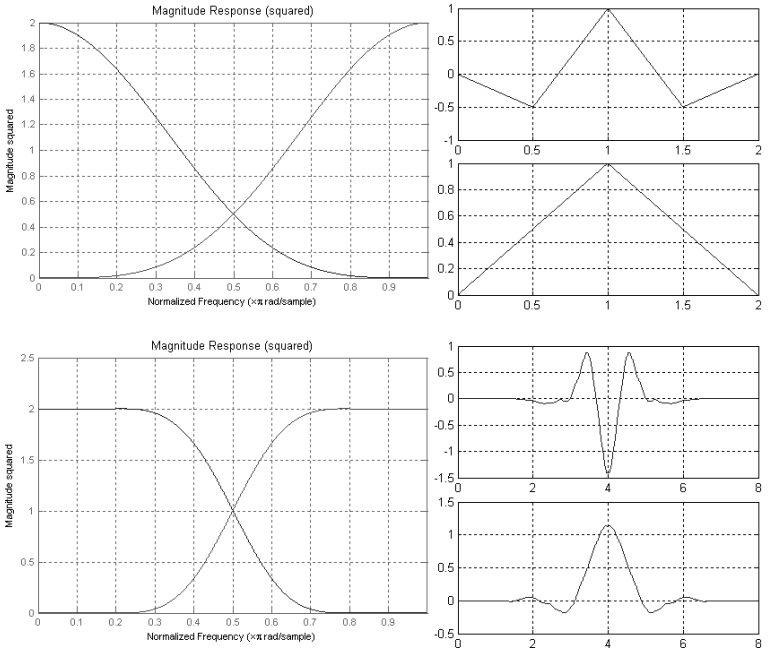



**Fig. 2.** Characteristics of two applied filter banks: magnitude responses, associated wavelets and scaling functions of short spline FB TSpline2 (top) and longer symmetric FB Atrial (bottom)

2. Wavelet-based image enhancement
   Two subsequent dyadic wavelet decompositions with 6 scales were implemented with symmetric wavelet basis because of the minimized distortions caused by the necessary extension of the ROIs to regular domain of data processing
   - controlled distortions propagated across scales for coarse denoising; TSpline2 transform kernel defined by $\tilde{h} = [1/4, 2/4, 1/4]$ was applied;
   - adaptive soft thresholding for subtle denoising and increasing the local mean data variability; shrinkage of detailed scales data with Atrail FB defined by $\tilde{h} = [0.01995, -0.04271, -0.05224, 0.29271, 0.56458, 0.29271, -0.05224, -0.04271, 0.01995]$;
3. Visualization of processed image
   Conditions of data visualization were set to increase the perception of tissue density distinction. Window of HU values including only the susceptible-to-hypodensity tissue was rescaled to 8 bit display with contrast enhancement by histogram equalization. Useless soft brain tissue was set to 255 level in order to increase contrast resolution. Non-brain tissue was reconstructed according to a conventional bone window of source image.

### 2.3 Experimental Study

Preliminary subjective tests were performed to maximize hypodense signs extraction according to clearly defined diagnostic performance criteria. Presence and location of ischemic changes were determined according to follow-up CT scan. Thus, the extracted hypodense areas could be simply verified even by engineers with the aim of optimization of processing procedure.

A set of over 30 test CT exams (including 18 hyperacute stroke cases, non-ischemic changes, normal CT) was subjectively rated in preliminary experiments due to the selection of the most effective wavelet bases and multiresolution schemes. Next, a set of selected 13 test CT exams of brain, including clinically confirmed cases of acute stroke appearance which contained a variety of ischemic abnormalities, was used for the assessment of hypodensity extraction suggestiveness. The exams of 13 patients aged 55-84 (mean 75.6) including an hyperacute MCA territory and pons infarct were used. Mostly, the cases of stroke which was difficult to detect (i.e. "silent" cases of acute stroke) were selected. The time between the onset of symptoms and the early CT examination ranged from 1 to 5 hours (mean 2.9 hours). Follow-up CT (from 1 to 10 days after the ictus) was used to determine the location and size of the infarct.

## 3 Results

Firstly, the subjective rating of over 30 processed test CT exams (over 600 images) was used to select the most effective wavelet bases and multiresolution schemes. Transform kernel impact on hypodensity enhancement was verified to optimize the algorithm presented in p. 2.2. Two experts in image processing and interpretation methods and a radiologist took part in the experiment. Relative scale of 1 to 5 was used where "1" indicated definite disappearance of hypodense signs in comparison to source exam, "2" indicated slight disappearance of hypodense signs, "3" indicated the same perception of hypodense signs, "4" indicated the enhancement of hypodensity, "5" indicated definite extraction of hypodensity. Close to 70 multiscale decomposition schemes with 1D and 2D kernels were rated. The best scored FBs were TSpline3[1] (mean score of 4.39), TSpline2 (4.36), TBi25_15[2] (4.32) and well-known Antonini 7_9 FB (4.28)[14] while the lowest FB score was under 3. The superiority of experimentally modified, non-PR FBs (the three first) was confirmed by observers' consensus. Thus, we decided to apply them in the proposed algorithm. The example of a diversified degree of hypodensity enhancement was given in Fig. 3).

[1] $\tilde{h} = [.125, .375, .375, .125]$.

[2] $\tilde{h} = [0.0010535,-0.0011583,-0.0135193,0.0139009,0.0812568,-0.066076,-0.2573154, 0.2590495, 0.04798992, -1.0713295, -0.8579303, 4.4011473, 8.2041789, 4.4011473, -0.8579303, -1.0713295, 0.4798992, 0.2590495, -0.2573154, -0.0660760, 0.0812568, 0.0139009, -0.0135193, -0.0011583, 0.0010535]*10^{-1}$
$h = [-0.0335799, -0.0369214, 0.2415515, 0.2348697, 0.8380562, -0.3530644, 4.1656185, 7.3813001, 4.1656185, -0.3530644, 0.8380562, 0.2348697, 0.2415515, -0.0369214,-0.00335799]*10^{-1}$.

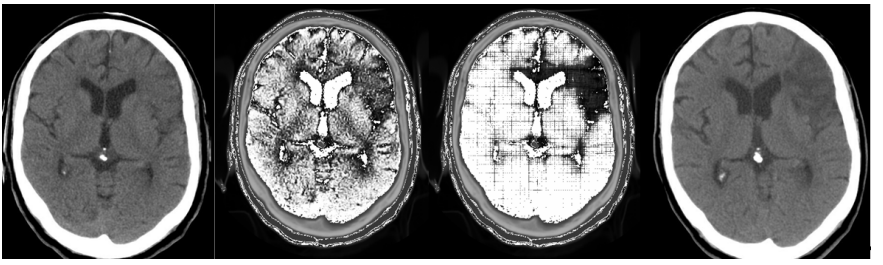


**Fig. 3.** Different displays of ischemic stroke cases. Source image of hyperacute stroke exam with unperceptable hypodensity (left), processed images with Antonini 7_9 FB (middle–left) or TBi25_15 (middle–right) filtering and denoising (soft thresholding), and follow–up CT scan that confirms stroke with clear location (right).

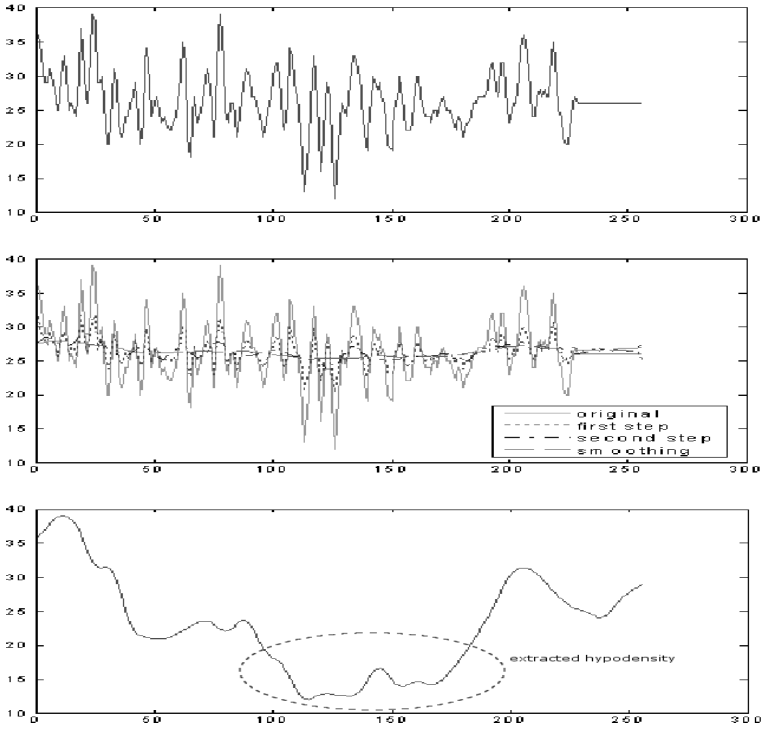



**Fig. 4.** Extraction of hypodensity in a line of hyperacute stroke exam. It contains noisy selected row of CT exam with hidden hypodense changes in the right site of the signal (top), subsequent processing stages of the first and second denoising steps, and smooth local data variability (middle), the resulting denoised approximation of signal extended to source signal range with clearly enhanced hypodensity, a hole between 100 and 190 samples (bottom).

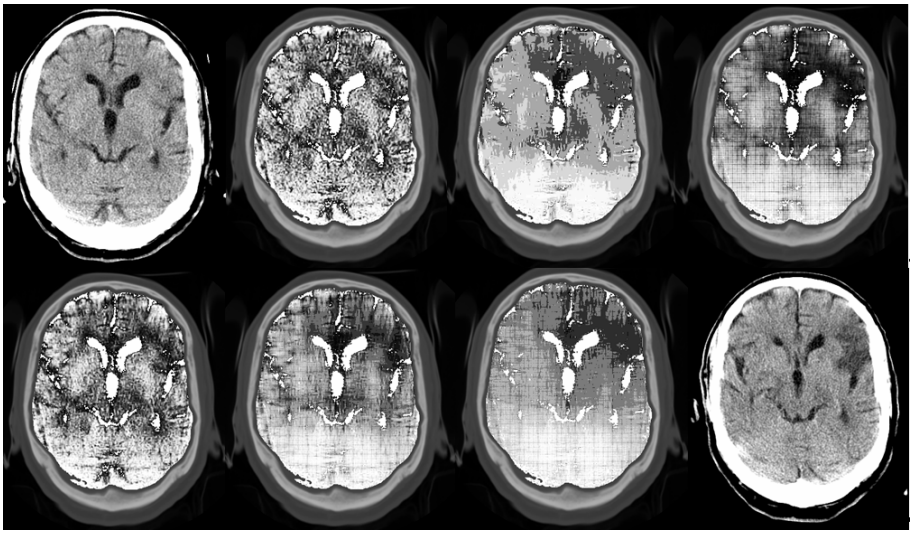


**Fig. 5.** The examples of improved perception of hypodensity, in sequence: acute stroke with undetectable hypodensity, four images processed with different FB and denoising (Antonini, TSpline2, TBi25_15 and Atrial, respectively), two images processed with stroke display (according to proposed algorithm with TSpline2 and alternative TSpline3, respectively), and follow-up CT scans.

Next, the mentioned elements of the proposed algorithm were designed and optimized basing on common neurological, radiological, and engineering consultations. Uniqueness and clearance of hypodensity determining was the most important optimization criterion of wavelet-based image post-processing as much as avoiding false indications. Four radiologists and two neurologists participated in that process, which resulted in proposed detailed algorithm of multiscale extraction of the signs of hypodensity. Subjective rating, according to the above mentioned procedure, was done on subsequent steps of the algorithm optimization. Finally, the set of 13 test stroke cases was used to determine the uniqueness and suggestiveness of hypodensity area indication. The effect of hypodense change extraction "in microscale" was shown in Fig. 4. The examples of the use of stroke display with improved, suggestive perception of hypodensity in the images were presented in Fig. 5.

According to subjective rating, stroke display improved the diagnosis of early ischemic changes because of the increased visibility of hypodense signs in 100% of test exams. Enhanced perception of the hypodense area through signal denoising as well as the increased local contrast resolution were noticed in the opinion of radiologists and image processing experts. Additionally, the rate of unique, clear and doubtless extraction of hypodense area was estimated. The rate of 92%(12/13) in 13 cases of acute ischemic stroke was noticed.

## 4 Conclusions

Reported results indicate that hypodensity-oriented enhancement based on image processing in wavelet domain may facilitate the interpretation of CT scans in hyperacute infarction. Improved segmentation of processed ROIs and controlling of the introduced distortions, reduced the possibility of false positives and significantly reduced the number of false negatives. However, the interactive verification of automated display, user-defined binarization of indications and work experience related to stoke display are necessary to eliminate any false indication and to make hypodensity extraction most useful for diagnosis.

Therefore, the reliable display of hypodense signs can considerably accelerate the diagnosis of hyperacute ischemic stroke with increased sensitivity and minimized possibility of false positives. Further optimization of automatic understanding of hypodensity phenomenon, modeling and detection of hyperacute cases is possible and desired. Clinical tests are necessary to consider the display as possible to be accepted for medical practice.

## References

1. N. Tomura, K. Uemura et al.: Early CT finding in cerebral infarction. Radiology **168** (1988) 463–7
2. L. Bozzao, S. Bastianello et al.: Correlation of angiographic and sequential CT findings in patients with evolving cerebral infarction. AJNR Am J Neuroradiol **10** (1989) 1215–22
3. R. von Kummer: The impact of CT on acute stroke treatment. Book chapter (draft), december (2005)
4. R. von Kummer, K.L.Allen et al.: Acute stroke: usefulness of early CT findings before thrombolytic therapy. Radiology **205** (1997) 327–333
5. F.J. Schuier, K.A. Hossmann: Experimental brain infarcts in cats. II. Ischemic brain edema. Stroke **6** (1980) 593-602
6. Dzialowski , J. Weber et al., Brain tissue water uptake after middle cerebral artery occlusion assessed with CT, J Neuroimaging 14:42-48, (2004)
7. M. Bendszus, H. Urbach , B. Meyer, R. Schultheiss, L. Solymosi: Improved CT diagnosis of acute middle cerebral artery territory infarcts with density-difference analysis. Neuroradiology **39**(2)(1997) 127–31
8. European guidelines in quality criteria for computed tomography, Report EUR 16262, Office for Official Publications of the European Communities, Brussels (1999)
9. D.K. Hammond, E.P. Simoncelli: Nonlinear image representation via local multiscale orientation. Courant Institute Technical Report (2005) TR2005–875
10. N. Bonnier, E.P. Simoncelli: Locally adaptive multiscale contrast optimization. Proc IEEE ICIP **2** (2005) 1001–1004
11. J.L. Starck, F. Murtagh, E. J. Candes., D.L. Donoho: Gray and color image contrast enhancement by the curvelet transform. IEEE Trans Image Proc **12**(6) (2003) 706–717
12. L.M. Fayad, Y. Jin et al.: Chest CT window settings with multiscale adaptive histogram equalization: pilot study. Radiology **223** (2002) 845-852

13. A. Przelaskowski, K. Sklina, P. Bargiel, J. Walecki, M.Biesiadko-Matuszewska, M.Kazubek: Improved early stroke detection: wavelet-based perception enhancement of computerized tomography exams. Comp Biol Med **37** (2007) 524-533
14. M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies: Image coding using wavelet transform. IEEE Trans Image Proc, **1**(2) (1992) 205–220

# Supporting Literature Exploration with Granular Knowledge Structures

Yiyu Yao[1,2], Yi Zeng[2], and Ning Zhong[2,3]

[1] Department of Computer Science, University of Regina
Regina, Saskatchewan S4S 0A2, Canada
yyao@cs.uregina.ca
[2] International WIC Institute, Beijing University of Technology
Beijing, 100022, P.R. China
yzeng@emails.bjut.edu.cn
[3] Department of Information Engineering, Maebashi Institute of Technology
Maebashi-City, 371-0816, Japan
zhong@maebashi-it.ac.jp

**Abstract.** Reading and literature exploration are important tasks of scientific research. However, conventional retrieval systems provide limited support for these tasks by concentrating on identifying relevant materials. New generation systems should provide additional support functionality by focusing on analyzing and organizing the retrieved materials. A framework of literature exploration support systems is proposed. Techniques of granular computing are used to construct granular knowledge structures from the contents, structures, and usages of scientific documents. The granular knowledge structures provide a high level understanding of scientific literature and hints regarding what has been done and what needs to be done. As a demonstration, we examine granular knowledge structures obtained from an analysis of papers from two rough sets related conferences.

**Keywords:** Granular computing, research support systems, research methods, literature exploration, granular knowledge structures.

## 1 Introduction

Literature exploration plays an important role in scientific research. Many scientists devote much of their valuable time exploring and digesting the scientific literature. With the over-increasing volume of scientific documents, the study and analysis of them becomes a real challenge for any scientist. Solso envisioned an intelligent system that "may tell us what research has been done, so we can avoid redundant studies, and it also may tell us what needs to be done, so we can put our valuable time to good use" [8]. Mjolsness and DeCoste suggested that machine learning can be used to support every phase of the research process [3].

Traditional information retrieval systems and Web search engines support the basic tasks of browsing and retrieval, so that a scientist can easily navigate the

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 182–189, 2007.

Web, browse digital libraries, and find relevant documents. They normally do not support the knowledge intensive tasks of analyzing, organizing and digesting the retrieved documents. Although many authors have pointed out the ineffectiveness of retrieval systems and Web search engines, the real problems may not lie on the classical issue of "retrieval". That is, the real problems are no longer retrieval, but post-processing of retrieved results.

In order to resolve the difficulties of current retrieval systems and to better support scientists, many proposals of next generation intelligent systems have been made, including information retrieval support systems [9,10] and research support systems [11]. The main objective of this paper is to propose a framework of literature exploration support systems, as a sub-system of a research support system. Such systems help scientists understand scientific literature in a structured and knowledgeable way.

Many authors have studied the problem of supporting literature exploration from different perspectives. Robert and Alfonso examined the connection and relation among different literature by domain characteristics [7]. Kuznetsov analyzed literature from its content view using concept lattice [2].

Based on these studies, we introduce the notion of literature exploration support systems. Such a system needs to analyze and organize scientific literature in multiple views. It supports a scientist to make explicit the granular knowledge structures embedded in scientific literature from its contents, structures, and usages perspectives. Techniques of granular computing are used to construct and represent granular knowledge structures.

## 2 An Overview

This section introduces the notion of literature exploration support systems and two important technologies for building such systems.

### 2.1 Literature Exploration Support Systems

Knowledge structures play a central role in problem solving [1,6]. They may help scientists to see the contributions of a particular study and its relationships to other studies. One of the objectives of reading and literature exploration is to construct these knowledge structures or concept maps. This is evident from many survey papers and literature review sections in many scientific writings.

A set of documents from different sources (e.g., digital libraries, conference proceedings, journal databases, results from retrieval systems or Web search engines, etc.) may be viewed as the space of exploration. Through multi-view analysis of its contents, structures and usages, a literature support helps a scientist to organize the literature into a structured and knowledgeable way so that it can be better understood and used in future research. The results may be represented as granular knowledge structures.

A literature exploration support system may be viewed as a sub-system of a research support system. Such a system can be seamlessly integrated with a

retrieval system or a Web search engine, by treating the retrieved results as a collection of scientific documents. Thus, a literature exploration support system may also be viewed as a sub-system of an information retrieval support system.

### 2.2 Granular Computing

Knowledge of a well-established field can normally be organized in a hierarchical way [6]. More abstract knowledge can be built upon more concrete knowledge. Knowledge at different levels represents differing granularity. Furthermore, at each level of the hierarchical structure, one associates rules regarding how to apply such knowledge [6]. It becomes clear that a literature exploration support system must help us to construct such granular knowledge structures.

As an emerging field of study, Granular Computing (GrC) is consistent with human problem solving based on knowledge structures [13]. Granular computing covers theories, methodologies, and tools that explore data granules, information granules and knowledge granules in problem solving. By viewing literature exploration as a problem solving task, one can immediately apply granular computing to literature exploration support systems. The three perspectives of granular computing are very relevant to literature exploration. In the philosophical perspective, it leads to structured thinking for understanding and organizing scientific literature. In the methodological perspective, it offers language and methods to build and represent granular knowledge structures from the literature. In the computational perspective, it deals with structured processing of granular knowledge structures.

### 2.3 Multi-view and Multi-level Exploration

Different views provide various unique understanding of the literature. By drawing results from Web mining, we propose to support literature exploration in multiple views and at multiple levels, based on the contents, structures, and usages of the literature.

The contents of literature can be organized based on different levels of granularity. Each granule represents a specific level of details of the literature. By comparing different levels of granules, the system can find the relationships among different papers or between a specific paper and a given topic.

Scientific literature is closely linked together by cross references. Such structural information needs to be explored when generating knowledge structures. For example, citation information has long been used in many studies of the structures of the literature.

Literature usage is another source that may be useful for building knowledge structures. The relationships among different documents could be investigated through user access behaviors. For example, if some papers are always viewed or studied together, one may establish a connection between them.

Based on the multi-view and multi-level in each view, a literature exploration support system can provide visualization for knowledge navigation and browsing.

## 3 Granular Knowledge Structures Generation

The essential issue in implementing a literature exploration support system is the generation of granular knowledge structures.

### 3.1 Granular Knowledge Structures

Knowledge structures can be built based on concepts. A concept is considered to be the basic unit of human thought and knowledge. A concept can be conveniently interpreted as a granule, namely, the extension of the concept. The representation, interpretation, connection and organization of concepts lead to granular knowledge structures [12].

We can use an information table and a language with respect to the table to represent knowledge. Consider a generalized decision logic language [14], *GDL*-language, which is an extension of a decision logic language used by Pawlak [4]. A specific information granule is represented by an atomic formula $(a, r, l)$, where $r$ denotes a particular relationship between an attribute value and a label. Let $L_a$ be the set of labels for all granules on the domain of attribute $a$. We have the relation set $R_a = \{=, \in\}$. Thus, the atomic formulas are of the two forms, $(a, =, l)$ and $(a, \in, l)$.

A concept in an information table can be jointly represented as $(\phi, m(\phi))$. The formula $\phi$ represents the intension of the concept, while the set $m(\phi)$ consists of those objects satisfying the formula and represents the extension of the concept [14].

Knowledge granules can be defined as relations on concepts:

$$G(\{\Re_i | i \in I^+\}, \{(\phi_n, m(\phi_n)) | n \in I^+\}), \tag{1}$$

where $\Re_i$ denotes the relations between concept granules and $I^+$ the set of positive integers. Different levels of relations among concepts induce a hierarchical structure called a granular knowledge structure. In particular, we need to consider three levels of structures, namely, internal structure of a granule, collective structure of a family of granules, and hierarchical structure of a web of granules [13]. They form the integrated knowledge structures of the literature.

### 3.2 A Granular Knowledge Structures Generation Process

Building knowledge structures based on isolated papers by using traditional Web mining methods may not be satisfactory. They may not be able to represent the connections between different levels of granules, such as subtopics and disciplines [5]. As a knowledge intensive system, a literature exploration support system must consider semantic information and user involvement.

One issue is the weights of different documents in the literature. It is a well known fact that some scientific papers are more important than others, because they have major impact on later research. Therefore, those documents should

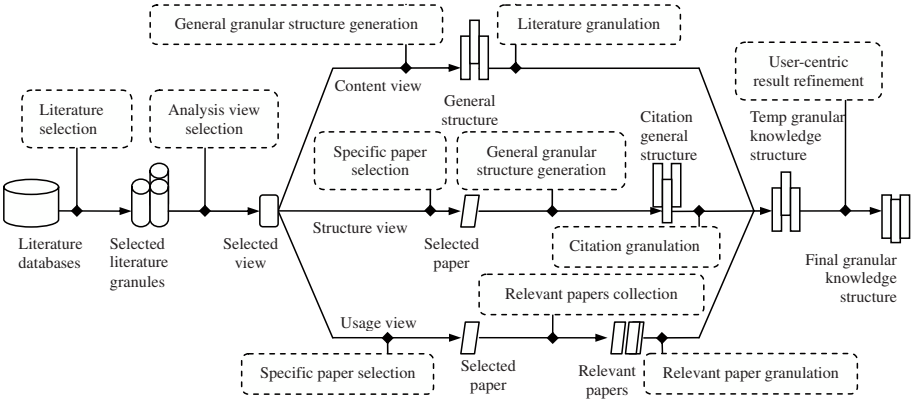



**Fig. 1.** Main Steps for Granular Knowledge Structures Generation

play a major role in forming the knowledge structures. A set of such documents may be easily obtained from citation information. A related issue is the definition of semantic relations between concepts, documents, and sets of documents. Typically, a scientific document has a well defined granular structure, consisting of title, abstract, section titles, and subsection titles. Such in formation may be incorporated. In other words, we can associate different weights to different concepts in a document.

A literature exploration support system must incorporate domain knowledge and user background knowledge. Although the construction process is the same, the knowledge base used is domain specific and personalized. We take a human-centric approach that allows a scientist to add new, to improve existing knowledge, and to refine granular knowledge structures. A support system needs to seek for the right balance between automation and user intervention [15].

Figure 1 shows the main steps for generating granular knowledge structures:

- In the *literature selection* step, the system or a user collects a set of documents to be explored.
- In the *view selection* step, a user selects a particular view for building multi-view based granular knowledge structures.
- In the *structure generation* step, the system generates different granular structures.
- In the *user-centric result refinement* step, a user can refine the results from the previous steps.

For exploration from content view, we build the general granular structures according to information about a single document and a sub-collection of documents, as well as domain knowledge. For exploration from structure view, we focus on building citation relation structure. For exploration from usage perspective, we find the connections and external structures of relevant papers based on literature access logs and domain knowledge.

**Table 1.** A Partial Information Table for Generating Figure 2

| Paper | Initial Page | Theory | Application | Domain |
|---|---|---|---|---|
| No.05 | p1-94 | Rough-Algebra | – | Rough Set |
| No.12 | p1-345 | Rough-Fuzzy Hybridization | – | Rough Set |
| No.25 | p2-342 | Logics and Reasoning | Medical Science | Rough Set |
| No.21 | p2-263 | Data Reduction | Image Processing | Rough Set |
| No.29 | p2-383 | Logics and Reasoning | Bioinformatics | Rough Set |
| No.97 | p3-522 | Formal Concepts | – | Rough Set |
| No.30 | p2-430 | Data Reduction | Bioinformatics | Rough Set |

## 4 An Illustrative Example

To demonstrate the proposed framework, we extract related information from RSFDGrC 2005 and RSKT 2006 proceedings to form the granular knowledge structures of Rough Sets.

The diagram shown in Figure 2 is formed based on information granules at different level of granularities from the two proceedings. Table 1 contains some examples of the information table used to construct Figure 2.

Examples of information granules from Table 1 are given as:

$$G(Theory, =, Formal\ Concepts) = \{No.97\},$$
$$G(Application, \in, l_1) = \{No.25, No.29, No.30\},$$
$$G((Theory, =, Data\ Reduction) \wedge (Application, \in, l_1)) = \{No.30\},$$
$$G((Page, =, 2-383) \Rightarrow (Application, =, Bioinformatics)) = \{No.29\}.$$

The label $l_1$ is the granule containing {Medical Science, Bioinformatics}. The symbol $\Rightarrow$ denotes the connection between two granules [14]. The concept granules for the first two formulas can be represented as:

$$((Theory, =, Formal\ Concepts), m(Theory, =, Formal\ Concepts),$$
$$((Application, \in, l_1), m(Application, \in, l_1)).$$

An example of granular knowledge structure based on the partial ordering is given as:

$$((Theory, =, Formal\ Concepts), m(Theory, =, Formal\ Concepts)$$
$$\subseteq ((Domain, =, Rough\ Sets), m(Domain, =, Rough\ Sets)).$$

Figure 2 shows a multi-level granular structure from the content view. For example, the coarsest granule is "Rough Sets", finer granules are subtopics related to "Rough Sets", and papers falling under each subtopic form the basic granules. The fact that "Nine theory subtopics are related to Rough Sets" reflects a coarser knowledge structure. The fact that "Bioinformatics and Data Reduction are related" reflects a finer knowledge structure. Figure 3 provides a structural

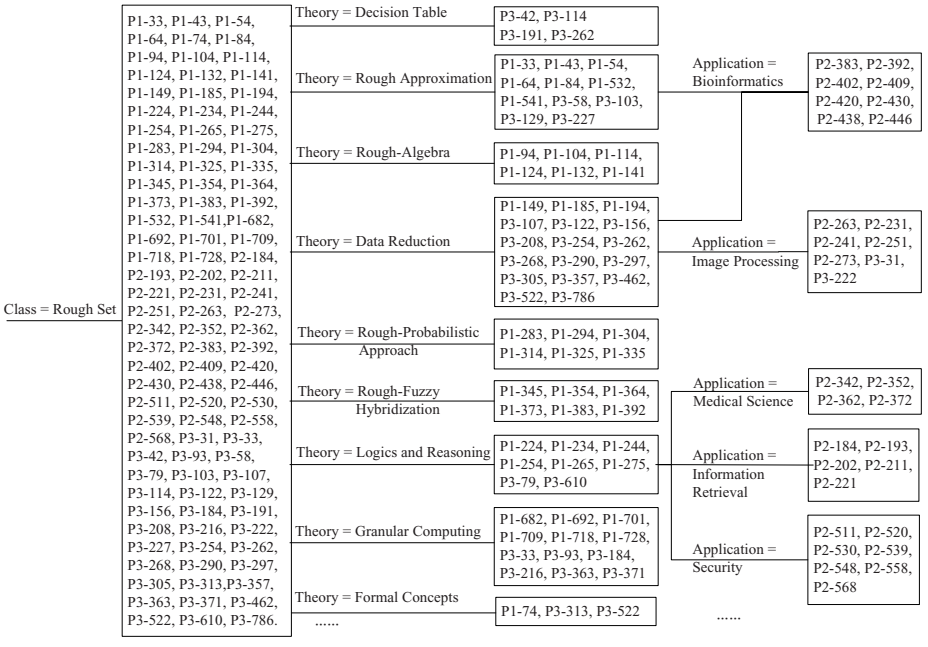



**Fig. 2.** Granular Knowledge Structure of Rough Sets from the Content View of the RSKT 2006 and RSFDGrC 2005 Proceedings

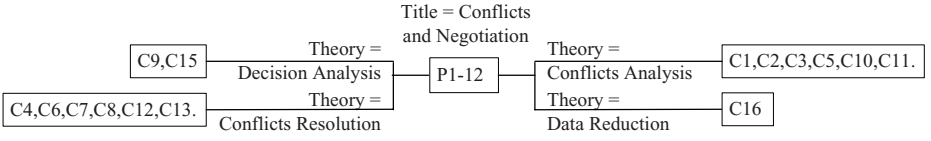



**Fig. 3.** A Single Paper's Granular Knowledge Structure from Citation View

view of a single paper's citations. Other views of granular structures could be further investigated.

The granular knowledge structures not only provide a relation diagram of specified discipline, but also help researchers to find the contribution of each study and possible future research topics. For example, as shown in Figure 1, many studies concentrate on data reduction and rough set approximations, and research of applications does not receive much attention. It can also be concluded that one may apply some of the theoretical studies (e.g., Rough-Algebra).

## 5 Conclusion

This paper proposes a framework of literature exploration support systems. Such a system constructs granular knowledge structures of the literature by using the

theory and techniques of granular computing. This enables scientists to explore literature in multiple views and at multiple levels, in order to see the contributions of a particular study and its relationships to other studies.

Literature exploration support systems focus on the post-processing of retrieved results of current retrieval systems and search engines. These systems may have great impact in helping scientists to meet the challenge of over-increasing literature growth.

## References


1. Gordon, S.E., Gill, R.T.: *The Formation and Use of Knowledge Structures in Problem Solving Domains*. Idaho University, Moscow (1989).
2. Kuznetsov, O.S.: Galois Connections in Data Analysis: Contributions from the Soviet Era and Modern Russian Research. In: *Formal Concept Analysis*. Springer, Berlin (2005) 196–225.
3. Mjolsness, E., DeCoste, D.: Machine Learning for Science: State of the Art and Future Prospects. *Science* **14** (2001) 2051-2055.
4. Pawlak, Z.: *Rough Sets, Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht (1991).
5. Pedrycz, W.: *Knowledge-Based Clustering: From Data to Information Granules*. John Wiley & Sons, Inc., New York (2005).
6. Reif, F., Heller, J.: Knowledge Structure and Problem Solving in Physics. *Educational Psychologist* **17** (1982) 102-127.
7. Robert, H., Alfonso, V.: Implementing the iHOP Concept for Navigation of Biomedical Literature. *Bioinformatics* **21** (2005) 252-258.
8. Solso, R.L., MacLin, M.K, MacLin, O.H.: *Cognitive Psychology*. Pearson Education, Inc. (2004).
9. Yao, J.T., Yao, Y.Y.: Web-based Information Retrieval Support Systems: Building Research Tools For Scientists in the New Information Age. In: Proc. of the IEEE/WIC Int. Conf. on Web Intelligence 2003, Halifax, Canada (2003) 570-573.
10. Yao, Y.Y.: Information Retrieval Support Systems. In: Proc. of FUZZ-IEEE'02, Hawaii, USA (2002) 773-778.
11. Yao, Y.Y.: A Framework for Web-based Research Support Systems. In: Proc. of COMPSAC'03, Washington, DC, USA (2003) 601-606.
12. Yao, Y.Y.: Concept Formation and Learning: A Cognitive Informatics Perspective. In: Proc. of the IEEE-ICCI'04, Victoria, Canada (2004) 42-51.
13. Yao, Y.Y.: Three Perspectives of Granular Computing. In: Proc. of IFTGrCRSP2006, Nanchang, China (2006) 16-21.
14. Yao, Y.Y., Liau, C.-J.: A Generalized Decision Logic Language for Granular Computing. In: Proc. of FUZZ-IEEE'02, Hawaii, USA (2002) 1092-1097.
15. Zhao, Y., Chen, Y.H., Yao, Y.Y.: User-centered Interactive Data Mining. In: Proc. of the IEEE-ICCI'06, Beijing, China (2006) 457-466.

# Ordinal Credibility Coefficient – A New Approach in the Data Credibility Analysis

Roman Podraza[1] and Krzysztof Tomaszewski[2]

[1] Institute of Computer Science
Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
R.Podraza@ii.pw.edu.pl
[2] Javart Sp. z o.o.
8 Jana Christiana Szucha Av., 00-582 Warsaw, Poland
krzysztof.tomaszewski@javart.com.pl

**Abstract.** The Data Credibility Analysis is a computer science domain aimed at discovering universal algorithms for identifying improper or unusual data. It is done by calculating credibility coefficients for individual records. In recent years many different methods of computing these coefficients were presented. In the paper we propose a transformation of credibility coefficients to ordinal credibility coefficients. By developing this idea we propose another credibility coefficient computing algorithm, which benefits from incorporating arbitrary many other credibility coefficient computing methods. The preliminary tests showed that this approach leads to better results.

**Keywords:** credibility coefficients, information system, classification, emerging patterns.

## 1 Introduction

Data credibility analysis [1] is a computer science research area in a domain of knowledge acquisition. It focuses on the problem of detecting improper or somehow outlying data in arbitrary data sets. As it is usually hard to make difference between invalid and valid but outlying data, these cases should be filtered out and indicated to domain experts. On the other hand the data set without such data should give better results when inferring general knowledge or processing with general rules.

The main goal of the research is estimating credibility of individual records of analyzed datasets. Evaluation of credibility of data is done by specialized heuristic algorithms. Some of them were described in [2] [3] [4]. The most important aspect of these algorithms is unawareness on meaning of the processed data. This makes them general, universal and ready to operate on any data. For a given dataset they assign to each data record a relative credibility estimation known as a credibility coefficient [1]. This is just a real number from range [0, 1]. The intention of the proposed data credibility assessment is to assign lower credibility values to less typical record, as lesser typicality is obviously associated with higher risk that data is improper or outlying.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 190–198, 2007.

Assuming that credibility coefficient computing algorithm is able to distinguish data typicality, records with lower credibility coefficients are commonly invalid, outlying or abnormal data. In any of these cases it is good to identify such records. Invalid data are obviously incredible and outlying data do not match well to typical schemes, so should not be used to infer a general knowledge. For example, if in a medical application an outlying patient record denotes a special case, it probably should be treated with some extra care and likely will require different remedies.

Values of credibility coefficients are relative to the analyzed data set. Obtained coefficients would be probably a bit different if the considered data set would have some records changed, added or removed. It is a consequence of the approach when a data set is the only input and having no additional information about the data domain. Thus for one data set a single credibility coefficient of 0.5 could mean a relatively high credibility while for the other one it could be one of lowest coefficients. This is the reason, why the credibility analysis system cannot itself decide how low coefficient value denotes an incredible record. Nevertheless, an expert can revise a chosen number of records (e.g. 10% of the data set), which were given the lowest credibility coefficients. Then the expert should decide how significant are the records and what to do with them (e.g. neglect, correct, start thorough investigation of cases).

It is important to notice how ordinary credibility coefficients are used. Assuming we do not focus on certain coefficients calculating algorithm's properties or, in other words, assuming credibility coefficients are produces just by some “black box” absolute values of these coefficients have no meaning. We use credibility coefficients to order records of the data set to focus on the least credible ones. Probably it would be common to inspect from 1% to 10% of such records. Hence record's credibility is really represented by the record's ordering position not by the value of the coefficient itself. A record is less credible if there are many records with higher position. As a consequence we do not use obtained credibility coefficients separately. Sorting operation requires all coefficients to be already calculated and each coefficient's value is important because it influences ordering positions of other data records.

## 2 Motivation

There were designed at least five different methods of computing credibility coefficients. In chronological order these are:

1. *Statistical/Frequency Method* [1][2] –based on number of objects having the same value for each attribute individually;
2. *Method Based on Class Approximation* [1][2] – based on rough set theory [5]. The value of coefficient is calculated using measures of positive and negative regions of each decision class;
3. *Method Based on Frequent Set* [3]– applies contribution of objects to frequent sets;
4. *Method Based on Decision Rules* [4] – applies relations of objects to decision rules inferred from decision system;
5. *Voting Classifier Method* – based on any voting classifier (like Bayesian classifier, k-NN, neural network, SVM); a classifier CAEP [6] (Classification by Aggregating Emerging Patterns) is a voting one and was implemented in KT Data Analysis System [7] to employ Emerging Patterns [8] to the data credibility investigation.

It can be expected that the number of these algorithms will increase in near future. So many diverse approaches are considered, because it is not obvious how to estimate data credibility, especially in a way independent on data meaning and/or their domain.

As could be expected there is no the best algorithm of computing credibility coefficients, although wide experiments comparing all of these approaches with the same objective criteria still have to be done. Such criteria are briefly described in section 4. Currently we can state that each of the calculating method of credibility coefficient has some advantages and behaves well in certain cases.

Because any algorithm calculating credibility coefficients is perfect, probably one would not like to stop after performing data credibility analysis using only one method. It would be reasonable to repeat the analysis using one or two methods more. However this is inconvenient and unpractical. Although this is possible to support this procedure by computer program such a scheme has its flaws. It would only acquire not one but few coefficients for each record and the conclusions would be left to an expert anyway.

It would be desirable to benefit simultaneously from a whole set of arbitrary chosen coefficients computing algorithms. Such a procedure should generate a single coefficient for each record and this coefficient should be a result of a "smart" aggregation of credibility coefficients obtained for the record from all chosen basic algorithms. The trivial aggregation like average or median cannot be applied because coefficients coming from different algorithms are hardly comparable.

## 3 Proposition

### 3.1 Ordinal Credibility Coefficient

We propose a new kind of credibility coefficient, namely ordinal credibility coefficient. Its values are not computed directly from the input data set but they are rather based on credibility coefficient's values obtained for the data set from an arbitrary chosen calculating method, for example one of listed above.

Let us have a non-empty data set $D$ and a method of calculating credibility coefficients. For a data record $x \in D$ let $cred(x)$ mean the value of credibility coefficient obtained from a given method for $x$. A value:

$$\frac{\left|\{y \in D : cred(y) \leq cred(x)\}\right|}{|D|} \tag{1}$$

for $x \in D$ is named ordinal credibility coefficient for record $x$ relative to data set $D$. We will denote it as $cred_{ORD}(x)$.

As defined above ordinal credibility coefficient for a given record from a given data set with use of a given coefficient computing method expresses the relative amount of records with credibility coefficients less or equal to the credibility coefficient for this record. To obtain ordinal credibility coefficients for all data records we perform this transformation to each record.

**Example**

Let us consider a data set with six objects identified by letters from *a* to *f*. Let us assume some credibility coefficient calculating method produced values, denoted as *cred(x)*, as shown in Table 1. Following columns of the table show sets of objects with coefficients less or equal to the given object, cardinalities of these sets and values of ordinal credibility coefficient.

**Table 1.** Calculation of ordinal credibility coefficients

| x | cred(x) | $Y_x = \{ y: cred(y) \le cred(x) \}$ | $\lvert Y_x \rvert$ | $cred_{ORD}(x)$ |
|---|---|---|---|---|
| *a* | 1.00 | *{ a, b, c, d, e, f }* | 6 | 6/6 = 1 |
| *b* | 1.00 | *{ a, b, c, d, e, f }* | 6 | 6/6 = 1 |
| *c* | 0.99 | *{ c, d, e, f }* | 4 | 4/6 = 0.67 |
| *d* | 0.98 | *{ d, e, f }* | 3 | 3/6 = 0.5 |
| *e* | 0.80 | *{ e, f }* | 2 | 2/6 = 0.33 |
| *f* | 0.78 | *{ f }* | 1 | 1/6 = 0.17 |

**Properties**

Ordinal credibility coefficients have the following properties.

1. The range of their values is (0, 1].
2. Credibility coefficient values and their ordinal counterparts introduce the same data set object ordering.
3. Ordinal credibility coefficient has a well defined interpretation. For a given credibility coefficient calculating method and dataset *D,* value $cred_{ORD}(x)$, for $x \in D$, shows what part of dataset *D* has estimated credibility less or equal to the estimated credibility of object *x*. Value $1\text{-}cred_{ORD}(x)$ shows what part of dataset *D* has estimated credibility grater then estimated credibility of object *x*. Thus ordinal credibility coefficient can be expressed by percentage.

As a result of the two first properties ordinal credibility coefficient is credibility coefficient itself. Thus it is possible to use ordinal credibility coefficients as base coefficients to obtain ordinal credibility coefficients again (of the next degree). Such transformation leads to exactly same values of ordinal credibility coefficient.

As a result of the third property we can aggregate ordinal credibility coefficient's values. For example, we have two ordinal credibility coefficients for object $x \in D$. The first one was obtained from credibility coefficients for dataset *D* computed by method $M_1$, and the second derived from credibility coefficients for *D* computed by $M_2$. It is possible to take average of these two ordinal credibility coefficient values. The obtained average is meaningful. It denotes an average part of data set *D* with credibility less or equal to the credibility estimated for object *x* while considering both computing methods $M_1$ and $M_2$. Finally, thanks to the definition of ordinal credibility coefficient, its value for a single record provides useful information itself without a need of referencing to credibility coefficients for other objects. This is not a case for an “ordinary” credibility coefficient, where its value for an individual record is

meaningful only in the context of the other coefficient values or if certain properties of used computing method are known and can be interpreted.

### 3.2 Multi Credibility Coefficient Method

So far no methodology of coincident cooperation of many methods of computing credibility coefficients was proposed. Here we propose a new approach of combining a number of credibility calculation algorithms using the notion of ordinal credibility coefficient to obtain an aggregate outcome.

Let us have a non-empty data set $D$. Let $M_{1, \dots,} M_N$ denote $N$ arbitrary chosen methods of computing credibility coefficients. Let $cred^{(i)}(x)$, $1 \le i \le N$ denotes a credibility coefficient for object $x \in D$ obtained by applying method $M_i$ to dataset $D$. Let $cred^{(i)}{}_{ORD}(x)$, $1 \le i \le N$ denotes an ordinal credibility coefficient for object $x \in D$ derived from all values $cred^{(i)}(y)$, $y \in D$. The Multi Credibility Coefficient Method evaluates credibility coefficient for object $x \in D$ as the value:

$$\frac{1}{N}\sum_{i=1}^{N} cred_{ORD}^{(i)}(x) \tag{2}$$

Although values computed by Multi Credibility Coefficient Method are results of aggregating ordinal credibility coefficients, these values are not ordinal credibility coefficients themselves. Of course they can be transformed to ordinal coefficients.

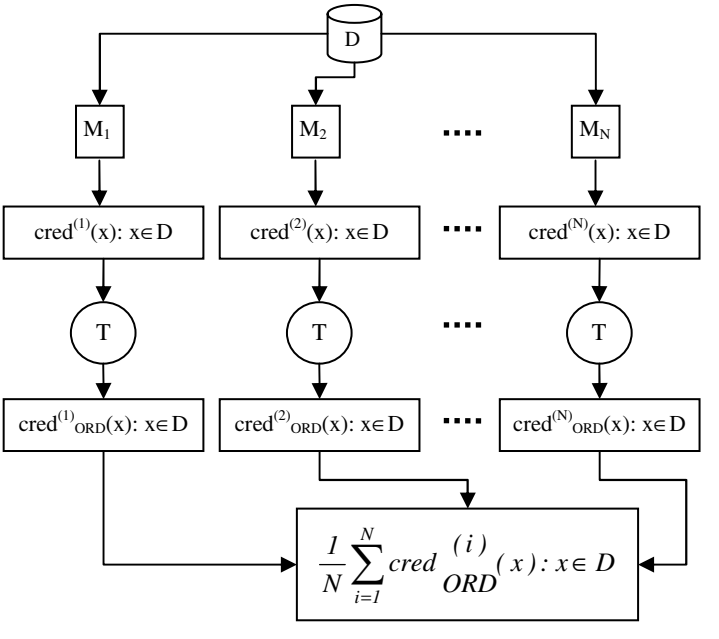



**Fig. 1.** Symbolic scheme for Multi Credibility Coefficient Method

In Fig. 1 a symbolic scheme for the proposed method is presented. Symbol "T" denotes transformation from credibility coefficients to ordinal credibility coefficients. As the result of this process one obtains credibility coefficient for each record of input data set and this coefficient is influenced by each "base" coefficient's computing algorithm used. As it was previously mentioned, the main idea of proposing such solution was to combine all advantages of various methods.

## 4 Results

The Multi Credibility Coefficient Method was implemented on an entirely new platform (KT Data Analysis System [7]) and it could be tested only with two variants of the Voting Classifier Method using Emerging Patterns [6] based classifier. The two variants (denoted further as I and II) were differing in a way Emerging Patterns were discovered in input data set. The first version employed methodology of frequent set and the second used decision tree approach. As it was hard to predict or evaluate which one Emerging Patterns discovering algorithm gives better results it was interesting to use them both.

We have performed experiments of two types, both innovative and never used before in the domain of data credibility analysis. In both cases input data were randomly modified/generated so each test configuration was repeated from 30 to 50 times to get averaged results.

Experiments of the first type used publicly available data sets *Iris*, *Heart* and *Glass* [9]. At first we have cleaned these data sets by removing 10% to 30% of the least credible records. For each run of the experiment we have injected synthetic, randomly generated falsified records to a given data set, such that there were always 10% of false records. They were generated by randomly choosing two records and copying values of a given number of attributes from the second one to the first one. Thus obtained false records were not so simple to detect. After computing credibility coefficients we were selecting at least 10% of all records having the least credibility coefficients – it was a subset of records considered as improper (falsified) by the tested method. Then we were evaluating two measures defined by us, namely *perfection* and *precision*. Perfection and precision for method $M$ computing credibility coefficients for objects from data set $D$ are defined respectively as

$$perf_M = \frac{|A \cap B|}{|B|} \qquad prec_M = \frac{|A \cap B|}{|A|} \tag{3}$$

where $B \subset D$ denotes a subset of all improper objects, and $A \subset D$ is set identified by the credibility coefficients that

$$|A| \geq |B| \wedge \max_{x \in A}(cred(x)) < \min_{x \in D-A}(cred(x))$$

Perfection is a ratio of number of false objects detected to the number of all false objects. Precision is a ratio of number of false objects detected to the number of all objects identified by credibility coefficients as improper. For both perfection and precision higher values means better results. Although similar experiments were

performed previously they never made use of such measures. Results from these experiments for two variants of the Voting Classifier Method and the Multi Credibility Coefficient Method are shown in Tables 2, and 3.

**Table 2.** False record detection for Iris dataset

| Number of modified attributes. | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|
| *perfection* | 22,20% | 46,40% | 81,30% | 93,10% | *Variant I* |
| *precision* | 22,20% | 46,40% | 81,30% | 93,10% | |
| *perfection* | **45,80%** | 70,00% | 90,00% | **99,30%** | *Variant II* |
| *precision* | 36,80% | 67,50% | 87,50% | **99,00%** | |
| *perfection* | 43,30% | **76,20%** | **92,00%** | 98,70% | *Multi Credibility* |
| *precision* | **42,90%** | **75,80%** | **91,20%** | 98,20% | *Coefficient M* |

**Table 3.** False record detection for Heart dataset

| Number of modified attributes. | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|
| *perfection* | 11,90% | 30,10% | 64,90% | **86,50%** | *Variant I* |
| *precision* | 11,80% | 30,10% | **64,90%** | **86,50%** | |
| *perfection* | 10,20% | 34,30% | 57,60% | 80,70% | *Variant II* |
| *precision* | 10,20% | 34,30% | 57,60% | 80,70% | |
| *perfection* | **13,70%** | **35,40%** | **65,30%** | 85,30% | *Multi Credibility* |
| *precision* | **13,70%** | **35,20%** | 64,40% | 85,10% | *Coefficient M* |

Experiments of the second type used synthetic, randomly generated data sets. Data sets were generated along with multidimensional normal distribution randomly parameterized for each run. Each data set contained two decision classes with its own distribution. We have performed this experiment with number of attributes set to 2 and to 5. Having a probability distribution for a data set it is possible to assess a probability of appearance of a given object. The credibility coefficients should be in close correlation with the probability values. We have measured a divergence

**Table 4.** Mean absolute error and linear correlation coefficient between probability and credibility coefficients

| Dimensions | 2 | 5 | |
|---|---|---|---|
| *MAE* | 0,27 | **0,27** | *Variant I* |
| *LCC* | 0,35 | 0,3 | |
| *MAE* | 0,31 | 0,29 | *Variant II* |
| *LCC* | 0,25 | 0,23 | |
| *MAE* | **0,26** | **0,27** | *Multi Credibility* |
| *LCC* | **0,37** | **0,32** | *Coefficient M.* |

between them with the mean absolute error (MAE) and the linear correlation coefficient (LCC). MAE should strive to 0.0 and LCC should be positive with higher values meaning better correlation. Results of these experiments are shown in Table 4.

As could be seen in Tables 2 and 3 the Multi Credibility Coefficient Method produced better results in most cases. From the Table 4 we can conclude that the Multi Credibility Coefficient Method calculated coefficients are closest to probability values although its superiority was rather slight.

## 5 Conclusions

The paper presents a concept of ordinal credibility coefficient. It is derived from values of basic credibility coefficients calculated by one of the available algorithms. Ordinal credibility coefficient of a given record defines its relative position within the data set according to ranking given by the values of the basic credibility coefficients. Thus ordinal credibility coefficients preserves and emphasizes the sorting of objects of the data set according to their estimated credibility. One of the possible applications is identification of data considered as doubtful. This is of course the main goal of basic credibility coefficient, but their data are meaningful only in context of all other credibility coefficients. In contrary the value of the ordinal credibility coefficient explicitly demonstrates if an object belongs to the least credible records.

The "worst" objects can be removed to improve the quality of the remaining data or they can be inspected with a special care (to understand why they are exceptions) – both concepts are appealing for research and can find many reasonable applications.

The ordinal credibility coefficient was a mechanism used to combine a number of different algorithms evaluating credibility coefficients. The proposed Multi Credibility Coefficient Method can produce a synthetic measure of different approaches. We hope that for most cases single information on data credibility would be desired.

The credibility coefficients were aimed to aid in automatic detection of improper data without using any interpretation of the analyzed data. The methodology is general one and can be applied to any data set. Ordinal credibility coefficients enable incorporating many credibility assessment methods to produce a compound signature.

## References

[1] Podraza R., Walkiewicz M., A. Dominik A.: Credibility Coefficients in ARES Rough Set Exploration System. Proc. 10th Int. Conf. on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, RSFDGrC 2005, Regina, Canada, Lecture Notes in Artificial Intelligence, LNAI 3642, Part II. Springer-Verlag, Berlin Heidelberg New York (2005) 29-38.

[2] Podraza R., Dominik A.: Problem of Data Reliability in Decision Tables. Int. J. of Information Technology and Intelligent Computing (IT&IC), Vol. 1 No. 1, (2006) 103-112.

[3] Podraza R., Walkiewicz M., A. Dominik A.: Credibility Coefficients Based on Frequent Sets", Conf. on Comp. Sci.– Research and Applications, Kazimierz Dolny, Poland, 2006, to be published in Annales UMCS, AI Informatica, Lublin, Poland.

[4] Podraza R., Walkiewicz M., Dominik A.: Credibility Coefficients Based on Decision Rules. Proceedings of the Int. Multiconference on Comp. Sci. and Inf. Technology, Vol. 1, XXII Autumn Meeting of Polish Information Processing Society, Wisła, Poland, (2006) 179-187.
[5] Pawlak Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer, (1991).
[6] Dong G., Zhang X., Wong L., Li J.:CAEP: Classification by Aggregating Emerging Patterns. Proc. of 2nd Int. Conf. on Discovery Science, Tokyo, Japan, (1999) 30-42.
[7] Podraza R., Tomaszewski K.: KTDA: Emerging Patterns Based Data Analysis System. Annales UMCS, Informatica, AI, Vol.4, Lublin, Poland, (2006) 279-290.
[8] Dong G., Li J.: Efficient Mining of Emerging Patterns: Discovering Trends and Differences. Proc. of the SIGKDD (5th ACM Int. Conf. on Knowledge Discovery and Data Mining), San Diego, USA, (1999) 43-52.
[9] Blake C.L., Merz, C.J.: UCI Repository of machine learning databases, Irvine, University of California, (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html

# *F*uzzyPR: An Effective Passage Retrieval System for QAS

Hans Ulrich Christensen and Daniel Ortiz-Arroyo

Computer Science Department
Aalborg University Esbjerg
Niels Bohrs Vej 8, 6700 Denmark
huc1405@student.aaue.dk, do@cs.aaue.dk

**Abstract.** In this paper we present *FuzzyPR*, a novel fuzzy logic based passage retrieval system for *Question Answering Systems (QAS)*. *FuzzyPR* employs a fuzzy logic based similarity measure that includes the best performing models to implement the question reformulation intuition. Our experiments show that *FuzzyPR* achieves consistently better performance in terms of coverage than JIRS on the TREC corpora and slightly better on the CLEF corpora.

**Keywords:** Information Retrieval, Question Answering Systems, Passage Retrieval, Fuzzy Logic.

## 1 Introduction

A *Question Answering System (QAS)* is one type of information retrieval (IR) system that attempts to find exact answers to user's questions expressed in natural language. In an *Open-Domain Question Answering System (ODQAS)*, questions are not restricted to certain domains and answers have to be found within an unstructured document collection. The *Passage Retrieval (PR) system*, one component of a QAS, extracts text segments from a group of retrieved documents and ranks these passages in decreasing order of computed likelihood for containing the correct answer to a question. Typically, such text segments are referred to as *candidate passages*.

A QAS is bound by the performance of its PR component. A PR system that fails to retrieve any answering passages to a question or returns many, large candidate passages will have a negative impact on the effectiveness of a QAS [1].

Previous research has proposed to use the *question reformulation* intuition: "*frequently, an answer to a (factoid) question can be found as a reformulation of the same question*" to build QAS. An example of the application of the *reformulation intuition* is the question "How much is the international space station expected to cost?" of QA@TREC 11 (QID: 1645)[1]. The answering passage contains the snippet: "(...)United States and Russia, are working together to build

[1] TREC's Question Answering collections are available from: http://trec.nist.gov/data/qa.html

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 199–207, 2007.

the SPACE STATION, which is EXPECTED TO COST between $40 billion and $60 billion.(...)".

This paper presents *FuzzyPR*, a language-independent PR system for ODQAS. *FuzzyPR* includes a fuzzy logic based implementation of the reformulation intuition. The paper is organized as follows. Section 2 briefly describes related work on passage retrieval systems. Section 3 describes and analyzes the main component mechanisms of a PR system. Section 4 describes *FuzzyPR* and presents its performance results. Finally, Section 5 presents some conclusions and future work.

## 2 Related Work

JIRS [2] is a PR system that employs a $n$-gram model. JIRS supports two extensions to the basic $n$-gram matching mechanism (called *Simple Model*): term weights (called *Term Weight*) and both term weights and a distance measure (called *Distance Model*). JIRS basically ranks higher passages containing larger sequences of the terms contained in the questions. Brill et al.'s Web QAS [3] builds queries constructed as permutations of the terms employed in the question. Kong et al. [4] use fuzzy aggregation operators in a passage-based retrieval system for documents, where the relevance of a document is re-calculated taking into account the retrieved passages. Other research [**?**] [4] has also explored the application of fuzzy logic in a QAS.

Although the application of the *reformulation intuition* has been previously explored to build QAS [2] [3] to our knowledge we are the first to propose a fuzzy logic question-passage similarity measure to model such intuition.

## 3 Analysis of Main Component Mechanisms in a Passage Retrieval System

The *reformulation intuition* can be modeled using two characteristics of a candidate passage: "*most (important) question terms*" and "*close proximity*". The feature *"most (important) question terms"* is modeled by the fuzzy subset: *The degree to which candidate passages contain all question terms.* The degree of membership varies from 1 when all important question terms occur within a candidate passage to 0 if no question terms occur within the passage. *"Close proximity"* is modeled by the fuzzy subset: *The degree to which the question terms contained in a candidate passage are juxtaposed.* If all question terms of the passage are juxtaposed, then the passage's membership degree in this fuzzy subset is 1. Otherwise, the more distributed the terms are, the lower the degree of proximity approaching 0.

The third vague concept that can be used in the reformulation intuition is *term matching*. In ODQAS, questions and documents commonly suffer from grammatical inflections and typos that have a negative impact on performance. The fuzzy logic interpretation of binary term similarity is the fuzzy subset: *The*

*degree to which two terms are identical* yielding 1 if the two terms are identical, a value in $]0, 1[$ if they have some letters in common, and 0 if they are very different. In the following subsections we briefly analyze fuzzy models to implement: *proximity of question terms occurring in a passage* and *automatic detection of term variations*. Further details can be found in [5].

### 3.1 Proximity of Question Terms Occurring in a Passage

*Fuzzy proximity measures* calculate the degree of proximity within a document of two or more question terms, based on the following two intuitions: 1) if all matching document terms are juxtaposed then the measure yields 1, and 2) the farther away the matching document terms occur, the lower the degree of proximity.

We evaluated three different fuzzy proximity measures as to their ability in finding answering passages for the first 50 questions of TREC11's question set using the AQUAINT corpus. We used the standard QAS evaluation metrics *Mean Reciprocal Rank (MRR)* and *coverage*. *MRR* is defined as the average of the reciprocal rank $r_i$ of the first hit to each question within the top 5 candidate passages:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} RR_i \,. \tag{1}$$

where $RR_i = \frac{1}{r_i}$ if $r_i \leq 5$ or 0 otherwise and $Q$ is the set of questions. As is done in the JIRS system [2], we measured coverage on the first top 20 passages. *Coverage* is defined as the proportion of questions for which an answer can be found within the $n$ top-ranked passages:

$$cov(Q, D, n) \equiv \frac{|\{q \in Q | R_{D,q,n} \cap A_{D,q} \neq \emptyset\}|}{|Q|}. \tag{2}$$

where $Q$ is the set of questions, $D$ is the passage collection, $A_{D,q}$ the subset of $D$ containing correct answers for $q \in Q$ and $R_{D,q,n}$ the $n$ top ranked passages.

Fig. 1 shows that Mercier and Beigbeder's *Fuzzy Proximity Measure* [6] achieves the same level of coverage at ranks 1-20 as the Extended Distance Factor [5], but performs 7.2% better in terms of MRR.

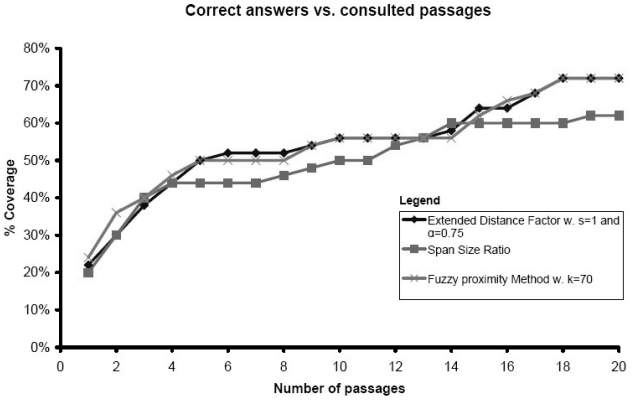
### 3.2 Automatic Detection of Term Variations

*Term variations* are lexical differences—in terms of meaning and spelling—between a word of the question typed by a user and an equivalent word contained in a document in the corpus. Reasons for term variations include grammatical inflection and spelling mistakes. Two main features are needed in a mechanism to handle term variations effectively: 1) *language-independence* and 2) *effectiveness*, measured as tolerance toward common misspellings and grammatical inflections, which are interpreted as a type of misspelling.

*Fuzzy term similarity algorithms* determine the degree of similarity between two strings. Reflexivity and symmetry are desired properties of these algorithms.

| Proximity Measure | MRR |
|---|---|
| Span Size Ratio | 0.2933 |
| Fuzzy Proximity Measure | 0.3363 |
| Extended Distance Factor | 0.3137 |

(a)




(b)

**Fig. 1.** The MRRs (a) and coverages (b) of the 3 fuzzy proximity measures

We performed a comparative evaluation on the effectiveness of six different algorithms when set to calculate the similarity between 300 English homophone[2] pairs. The average of the similarity computations yields the score of the fuzzy term matching algorithm.

**Table 1.** Average similarity scores of 8 Fuzzy similarity algorithms (sorted in decreasing order)

| Algorithm | Average similarity score |
|---|---|
| Normalized longest common subsequence | 0.5984 |
| Inverse normalized DD | 0.5569 |
| Inverse normalized LD | 0.5513 |
| Szczepaniak and Gil | 0.4395 |
| Reciprocal DD | 0.3751 |
| Reciprocal LD | 0.3720 |
| Improved trigram algorithm | 0.2477 |
| Trigram algorithm | 0.1691 |

Table 1 shows that the normalized longest common subsequence (nLCS) [5] performed best, giving an average homophone pair similarity rate of 0.5984.

[2] A *homophone pair* is two terms pronounced the same but differing in meaning and spelling, thus reflecting misspellings and typos. Examples include "advice vs. advise" and "cite vs. site".

## 4 FuzzyPR System and Performance Results

*FuzzyPR* consists of two components: 1) a question–passage similarity measure module and 2) a passage identification and extraction mechanism adapted to the special needs of QAS. The following subsections describe these components.

### 4.1 Similarity Measure

The similarity measure we propose is the fuzzy logic-based interpretation of the *reformulation intuition*: "*a passage p is relevant to the user's question q if many question terms or variations of these question terms occur in close proximity*" described by Equation 3.

$$\mu_{rel}(p,q) = wMin\left((v_1, \mu_f(p,q)), (v_2, \mu_p(p,q))\right). \tag{3}$$

This similarity measure combines lexical and statistical data extracted at *term-level* into the two fuzzy measures: $\mu_f(p,q)$ the weighted fraction of question terms $q$ occurring in the passage $p$ and $\mu_p(p,q)$ the proximity of question terms $q$ within the passage. Using the results of the performance analysis described in Section 3, $\mu_f(p,q)$ and $\mu_p(p,q)$ are defined in equations 4 and 5.

$$\mu_f(p,q) = h_{\alpha_f}\left((v_1^f, sat(t_{q_1},p)) \ldots (v_n^f, sat(t_{q_n},p))\right). \tag{4}$$

where $h$ is the AIWA importance weighted averaging operator [7] with an ANDness of $\alpha_f = 0.65$, $t_{q_i}$ is a question term, $v_i^f = NIDF(t_{q_i}) = 1 - \frac{log(n_i)}{1+log(N)}$ [3], $n$=frequency of $t_{q_i}$ in $\Omega$ the set of documents, $N = |\Omega|$. $sat(p, t_{q_i})$ measures the degree to which $p$ contains $t_{q_i}$ using the normalized longest common subsequence (nLCS), i.e. $sat(p, t_{q_i}) = \max_{\forall t_p \in p}\left(\mu_{sim}^{nLCS}(t_p, t_{q_i})\right)$, where $\mu_{sim}^{nLCS}(t_p, t_{q_i}) = \frac{|LCS(t_p,t_{q_i})|}{\max(|t_p|,|t_{q_i}|)}$, $LCS$ being the longest common subsequence. Finally,

$$\mu_p(p,q) = \frac{s(p,q)}{\max\limits_{\forall p_i \in \Omega} s(p_i,q)}. \tag{5}$$

where $\mu_p(p,q)$ is a max-normalization of Mercier and Beigbeder's *fuzzy proximity* method [6] described by $s(p,q) = \int_1^n \mu_t^p(x)dx$, $t \in q$ with the term influence function $\mu_t^p(x) = \max\limits_{i \in Occ(t,p)}\left(\max\left(\frac{k-|x-i|}{k}, 0\right)\right)$, where the parameter adjusting the support $k = 70$. The values of $v_1$, $v_2$, $\alpha_f$ and $k$ are determined experimentally. Aggregating these two fuzzy measures using the weighted minimum gives the overall relevance score $wMin$, which is defined as:

$$wMin(v_1, v_2, \mu_f, \mu_p) = \min\left(\max(1 - v_1, \mu_f(p,q)), \max(1 - v_2, \mu_p(p,q))\right). \tag{6}$$

with the importance weights $v_1 = 1$, $v_2 = 1$ and both the passage $p$ and the question $q$ represented as sets of terms: $\{t_{p_1}, t_{p_2}, ..., t_{p_n}\}$ and $\{t_{q_1}, t_{q_2}, ..., t_{q_m}\}$,

[3] NIDF is an abbreviation of normalized inverse document frequency.

respectively. $wMin$ aggregates $\mu_f(p,q)$ and $\mu_p(p,q)$ into a single fuzzy value $\mu_{rel}(p,q)$ as described by Equation 3. $\mu_{rel}(p,q)$ is the fuzzy subset of passages providing a correct answer to the question $q$, where $p$ is a specific passage. $\mu_{rel}(p,q)$ has the advantage of being *language-independent*.

### 4.2 Mechanism for Passage Identification and Extraction

A fuzzified variation of the concept *arbitrary passages*[4] is employed in *FuzzyPR*. An arbitrary passage is modeled as its membership function in the ideal set of passage sizes as stated in equation 7.

$$\mu_{Ideal\ passage\ size}(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq d \\ \frac{x-b}{d-b} & \text{if } d < x < b \\ 0 & \text{if } x \geq b \end{cases} . \tag{7}$$

$x$ is a term's location in the passages and $d$ and $b$ adjust the crisp support and the fuzzy support respectively. Due to efficiency concerns, the membership function of the ideal passage size set is transformed into an equivalent symmetric membership function, where the center term of a passage is required to have a question term similarity greater than $\alpha$ and a normalized IDF greater than $\beta$. This restriction is justified by the intuition that a passage containing none or very few of the question's terms is unlikely to provide an answer to the question.

### 4.3 Performance Results

We measured the effectiveness of *FuzzyPR* by comparing its ability to find correct answers to questions in a document corpora with both an adapted PR system that we have integrated within Lucene—a popular vector space search engine—and the JIRS PR system [2]. We decided to evaluate the simple model and the distance model of JIRS, because we found that the term weighted model and the simple model perform almost identically.

Both JIRS and the PR system implement an index of 3 sentence passages with 1 sentence overlapping. Llopis et al. in [8] report that this approach achieves good results. The PR system allows Lucene to be used as a PR module in a QAS by employing a simple query expansion method. In this method the question term with the lowest IDF is removed until $\geq 20$ passages are retrieved from the index of 3 sentence passages.

Because *FuzzyPR* defines a passage as a number of consecutive terms, we computed and used the arithmetic mean of the average passage sizes of the top 100 passages retrieved by both Lucene, JIRS Distance Model and JIRS Simple Model. In table 2 the numbers in parenthesis are the actual passage sizes used by *FuzzyPR*.

As test data we used TREC12's set of 495 questions and the corpus called AQUAINT consisting of $1,033,461$ documents of English news text and

[4] Arbitrary passages are defined as: *"any sequence of words of any length starting at any word in the document"*.

**Table 2.** The average passage sizes of the PR systems used for comparison

| PR system Test data | TREC12 | CLEF04 |
|---|---|---|
| Lucene | 55.91 | 74.74 |
| JIRS Distance Model | 132.23 | 105.87 |
| JIRS Simple Model | 166.96 | 111.48 |
| Arithmetic mean | 118.37 (119) | 97.36 (98) |

CLEF04's 180 question and the AgenciaEFE corpus of 454, 045 Spanish newswire documents. To answer questions automatically for TREC12 we used Ken Litkowsky's regular expression patterns of correct answers[5] and for CLEF4 we used the patterns supplied with JIRS[6].

The TREC12 question set was reduced to 380, since 115 questions do not have a recognizable pattern. As evaluation metrics we used *Mean Reciprocal Rank (MRR)* and *coverage* defined in Section 3. *%impr.* is the improvement (or worsening) *FuzzyPR* achieves compared to a PR system expressed as an percentage.

**Table 3.** MRRs obtained with TREC12's and CLEF04's QA test data

| PR system / QA test data | TREC12 | %impr. | CLEF04 | %impr. |
|---|---|---|---|---|
| *FuzzyPR* | 0.3394 | - | 0.3726 | - |
| JIRS Distance Model | 0.3180 | 6.73% | 0.3721 | 0.13% |
| JIRS Simple Model | 0.2724 | 24.60% | 0.3771 | −1.19% |
| Lucene | 0.2910 | 16.63% | 0.3399 | 9.62% |

Tables 3 and 4 show that *FuzzyPR* consistently performs better than Lucene's vector space PR system independently of the number of top-ranked passages consulted tested with both TREC12 and CLEF04 QA test data. MRR is improved at least 9.62% and coverage@20 at least 14.47%.

Comparing the performance of *FuzzyPR* and the two variations of JIRS shows that for TREC12 QA test data in terms of both MRR and coverage *FuzzyPR* performs consistently better. Compared to the second best PR system: JIRS Distance Model, MRR is improved by 6.73% and coverage@20 by 4.15%. As Table 4(b) shows, *FuzzyPR* tested with CLEF04 QA test data in general (18 out of 20 cases) achieves slightly better coverage than JIRS. Table 4 reveals that although *FuzzyPR* fails to boost coverage at the ranks 1 to 3, at ranks 4 to 20 it achieves a 0%-7.87% higher coverage than number two: JIRS Distance Model.

[5] Ken Litkowsky's patterns are available from the TREC website: `http://trec.nist.gov`.

[6] Patterns of correct answers to CLEF QA test data are available from JIRS' web site: `http://jirs.dsic.upv.es/`.

**Table 4.** The PR systems' coverages tested with (a) TREC12 and (b) CLEF04 data

(a)

| Rank | FuzzyPR | Lucene | %impr. | JIRS_SM | %impr. | JIRS_DM | %impr. |
|---|---|---|---|---|---|---|---|
| 1 | 0.2500 | 0.2237 | 11.76% | 0.2222 | 12.51% | 0.2434 | 2.71% |
| 2 | 0.3579 | 0.3053 | 17.23% | 0.2698 | 32.65% | 0.3201 | 11.81% |
| 3 | 0.4184 | 0.3500 | 19.54% | 0.2989 | 39.98% | 0.3836 | 9.07% |
| 4 | 0.4500 | 0.3711 | 21.26% | 0.3466 | 29.83% | 0.4206 | 6.99% |
| 5 | 0.4868 | 0.4026 | 20.91% | 0.3704 | 31.43% | 0.4497 | 8.25% |
| 6 | 0.5184 | 0.4237 | 22.35% | 0.4048 | 28.06% | 0.4788 | 8.27% |
| 7 | 0.5421 | 0.4342 | 24.85% | 0.4312 | 25.72% | 0.4921 | 10.16% |
| 8 | 0.5684 | 0.4526 | 25.59% | 0.4471 | 27.13% | 0.5079 | 11.91% |
| 9 | 0.5816 | 0.4789 | 21.44% | 0.4788 | 21.47% | 0.5317 | 9.38% |
| 10 | 0.5947 | 0.4947 | 20.21% | 0.4894 | 21.52% | 0.5476 | 8.60% |
| 11 | 0.6105 | 0.5053 | 20.82% | 0.4947 | 23.41% | 0.5582 | 9.37% |
| 12 | 0.6158 | 0.5237 | 17.59% | 0.5053 | 21.87% | 0.5688 | 8.26% |
| 13 | 0.6211 | 0.5289 | 17.43% | 0.5212 | 19.17% | 0.5794 | 7.20% |
| 14 | 0.6237 | 0.5368 | 16.19% | 0.5265 | 18.46% | 0.5899 | 5.73% |
| 15 | 0.6237 | 0.5474 | 13.94% | 0.5291 | 17.88% | 0.5952 | 4.79% |
| 16 | 0.6263 | 0.5500 | 13.87% | 0.5317 | 17.79% | 0.6032 | 3.83% |
| 17 | 0.6316 | 0.5579 | 13.21% | 0.5476 | 15.34% | 0.6085 | 3.80% |
| 18 | 0.6368 | 0.5605 | 13.61% | 0.5556 | 14.61% | 0.6111 | 4.21% |
| 19 | 0.6368 | 0.5605 | 13.61% | 0.5635 | 13.01% | 0.6164 | 3.31% |
| 20 | 0.6447 | 0.5632 | 14.47% | 0.5714 | 12.83% | 0.6190 | 4.15% |

(b)

| Rank | FuzzyPR | Lucene | %impr. | JIRS_SM | %impr. | JIRS_DM | %impr. |
|---|---|---|---|---|---|---|---|
| 1 | 0.2833 | 0.2722 | 4.08% | 0.3222 | -12.07% | 0.3000 | -5.57% |
| 2 | 0.3778 | 0.3722 | 1.50% | 0.3889 | -2.85% | 0.3722 | 1.50% |
| 3 | 0.4389 | 0.3944 | 11.28% | 0.4111 | 6.76% | 0.4444 | -1.24% |
| 4 | 0.4944 | 0.4222 | 17.10% | 0.4500 | 9.87% | 0.4833 | 2.30% |
| 5 | 0.5333 | 0.4389 | 21.51% | 0.4722 | 12.94% | 0.4944 | 7.87% |
| 6 | 0.5556 | 0.4556 | 21.95% | 0.4944 | 12.38% | 0.5278 | 5.27% |
| 7 | 0.5611 | 0.4722 | 18.83% | 0.5222 | 7.45% | 0.5444 | 3.07% |
| 8 | 0.5722 | 0.4722 | 21.18% | 0.5278 | 8.41% | 0.5667 | 0.97% |
| 9 | 0.5722 | 0.4833 | 18.39% | 0.5333 | 7.29% | 0.5722 | 0.00% |
| 10 | 0.5944 | 0.4889 | 21.58% | 0.5611 | 5.93% | 0.5833 | 1.90% |
| 11 | 0.6000 | 0.4889 | 22.72% | 0.5611 | 6.93% | 0.5833 | 2.86% |
| 12 | 0.6167 | 0.4889 | 26.14% | 0.5667 | 8.82% | 0.5944 | 3.75% |
| 13 | 0.6222 | 0.4889 | 27.27% | 0.5667 | 9.79% | 0.6000 | 3.70% |
| 14 | 0.6278 | 0.5000 | 25.56% | 0.5778 | 8.65% | 0.6056 | 3.67% |
| 15 | 0.6278 | 0.5056 | 24.17% | 0.5778 | 8.65% | 0.6167 | 1.80% |
| 16 | 0.6389 | 0.5056 | 26.36% | 0.5778 | 10.57% | 0.6167 | 3.60% |
| 17 | 0.6389 | 0.5056 | 26.36% | 0.5778 | 10.57% | 0.6167 | 3.60% |
| 18 | 0.6389 | 0.5167 | 23.65% | 0.5778 | 10.57% | 0.6222 | 2.68% |
| 19 | 0.6444 | 0.5222 | 23.40% | 0.5833 | 10.47% | 0.6278 | 2.64% |
| 20 | 0.6500 | 0.5333 | 21.88% | 0.5833 | 11.43% | 0.6333 | 2.64% |

However, in terms of MRR, JIRS Simple Model achieves a MRR of 0.3771, which is 1.2% better than *FuzzyPR*. This indicates that sometimes answering passages in this collection do not conform to the *reformulation intuition*. However, this only seems to affect the ability to boost answering passages to higher ranks because JIRS Simple Model falls behind JIRS Distance Model and *FuzzyPR* for coverage@4–20.

*FuzzyPR* has been optimized using TREC11 QA test data, which might bias the TREC12 results. However, table 4(b) shows that *FuzzyPR* achieves the highest coverage at ranks 4 to 20 for CLEF04 QA test data, too. Because Gómez-Soriano et al. [2] evaluated JIRS with CLEF's Spanish, Italian, and French QA test data it is reasonable to assume that JIRS' system parameters have been optimized for these languages. *FuzzyPR* performs better than JIRS due to the incorporation of two additional fuzzy concept besides those included in the JIRS Distance Model: 1) terms are importance-weighted using inverse document frequencies and 2) instead of $n$-grams the similarity method uses subsequences of $n$ question terms together with a proximity method yielding the highest similarity when the terms are juxtaposed. Furthermore, compared to JIRS's Distance Model *FuzzyPR* also fuzzifies 3) the definition of passage size and 4) question terms' occurrences in a passage. A last difference is that *FuzzyPR* computes the proximity of the question terms occurring in a passage rather than relying on $n$-gram or subsequence matching.

## 5 Conclusions and Future Work

In this paper we presented *FuzzyPR*. *FuzzyPR* implements a fuzzy logic based interpretation of the *reformulation intuition*. *FuzzyPR* has three main advantages: 1) its *passage identification and extraction methods* that enables it to retrieve candidate passages from documents at retrieval time thus avoiding the

time-consuming indexing process[7] 2) its *language-independence* property, and 3) its ability to handling spelling errors and grammatical inflections.

Our experiments show that *FuzzyPR* achieves a consistently higher MRR and coverage than Lucene's PR system and JIRS on TREC corpora. Furthermore it performs better in terms of coverage than JIRS on the CLEF corpora at ranks 4 to 20. In future work we plan to evaluate *FuzzyPR* with CLEF's French and Italian corpora to test its performance when compared to JIRS.

## References

1. Gaizauskas R., Greenwood M., Hepple M., and Roberts I.: The university of sheffields trec 2003 q&a experiments. *Proceedings of the 12th Text REtrieval Conference*, 2003.
2. Gómez-Soriano J., Montes y Gómez M., Arnal E., and Rosso E.: A passage retrieval system for multilingual question answering. *Proceedings of 8th International Conference of Text, Speech and Dialogue 2005 (TSD'05)*, Lecture Notes in Computer Science, Springer-Verlag, 2005. Karlovy Vary, Czech Republic.
3. Brill E., Lin J., Banko M., Dumais S. , and Ng A.: Data-intensive question answering. *In Proceedings of the Tenth Text REtrieval Conference (TREC 2001), Gaithersburg, Maryland*, pages 443–462, November 2001.
4. Kong K., Luk R., Ho K., and Chung F.: Passage-based retrieval using parameterized fuzzy set operators. *ACM SIGIR Workshop on Mathematical/Formal. Methods for Information Retrieval*, 2004.
5. Christensen H. U.: Exploring the use of fuzzy logic and data fusion techniques in passage retrieval for QA. Master's thesis, Aalborg University Esbjerg, December 2006.
6. Beigbeder M. and Mercier A.: An information retrieval model using the fuzzy proximity degree of term occurrences. *Proceedings of the 2005 ACM symposium on Applied computing*, March 2005.
7. Larsen H. L: Efficient andness-directed importance weighted averaging operators. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, pages 67–82, 2003.
8. Llopis F., Ferrández A., and Luis Vicedo J.: Text segmentation for efficient information retrieval. *In Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2002), Mexico City, Mexico*, Lecture Notes in Computer Science, Springer-Verlag:373–380, 2002.

[7] An unoptimized method in Java for segmenting and indexing the AQUAINT corpus took 4 hours on an AMD64 3400+ w. 2 GB RAM and RAID 0.

# Parallel Artificial Immune Clustering Algorithm Based on Granular Computing*

Keming Xie[1], Xiaoli Hao[2], and Jun Xie[1]

[1] College of Information Engineering, Taiyuan University of Technology
[2] College of Computer and Software, Taiyuan University of Technology, 030024 Taiyuan, Shanxi, P.R. China
kmxie@tyut.edu.cn

**Abstract.** When samples number, classification and dimension of clustering are much more, traditional clustering algorithm usually leads to unharmonious character between clustering and transcendent knowledge. Therefore, a new clustering algorithm is proposed, which is parallel artificial immune clustering algorithm based on granular computing. Artificial immune system model has the characteristics, such as parallel, random searching and maintaining diversity, which can solve premature problem in latter evolution and converge to a global optimization solution faster. Besides, we unite it to dynamic granulation model and apply granulation description to clustering. In the process of granulation changing, we can choose appropriate granulation size by adjusting to ensure clustering efficiency and quality. Tests show that the algorithm is more effective and more reasonable when we handle clustering of some data with it.

**Keywords:** Granular computing, artificial immune algorithm, clustering, harmony degree.

## 1 Introduction

Clustering is an important research topic in machine learning [1]. Clustering can divide a set into a several of classes. Objects of similar characters are involved in one class. There are many traditional methods, for example K-means based clustering, leveling clustering and fuzzy clustering. In essence, these methods are local searching optimization, in which climbing strategy is adopted. Therefore, When dealing with problems of multi-samples, multi-properties and multi-classification, we easily get into local solutions other than global ones. If initial values of clustering are chosen incorrectly, it easily leads to different results and not reflects clustering character.

Artificial immune algorithm is illuminated by body cell theory and network theory, and is a developed algorithm based on natural immune system. It simulates to realize excellent function of antigen recognition, cell division, memory

* Project supported by Special Foundation of Doctor's Subject for Colleges and Universities (2006112005), National Natural Science Foundation of China (60374029), Visiting Scholar Foundation of Shanxi Province, P.R.C. (2004-18).

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 208–215, 2007.

and self-adjustment. By repeated evolution process, the algorithm maintains the superior individuals to keep diversity and obtain optimal antibodies in the end [2]. The introduction of artificial immune model into clustering not only solves latter premature, which often occurred in conventional clustering algorithm, but also converge to global optimization quickly. Especially in the occasion of parallel computing, it shows superior advantage. However, clustering results are objective and transcendent knowledge directed by respects is subjective. There is no correspondence between them. So we can not deal with the problem only by artificial immune algorithm.

In the paper, by the advantage of these characters of artificial immune algorithm, we combine it to dynamic granular model. In the point of information granularity [3], we construct a new clustering algorithm, which can eliminate inharmonious degree between clustering results and transcendent knowledge. In the new algorithm, we can adjust granularity size by synthesis computing. On each granularity layer, we can get their correspondence. If inharmonious character between them still exists, we continue to generate new granularity size in partial order relation until we achieve ideal clustering. Finally, we take two examples to test its validity comparing with other algorithms, and tests show that it is more effective and superior than others.

## 2 Clustering Algorithm Analysis Based on Dynamic Granularity

### 2.1 Clustering Harmony Degree in Meaning of Uniform Granularity

Clustering defines equivalence relation among samples. It means that two samples belonging to a class are considered as equivalent. They have resembling characters and have no difference under the same threshold. An equivalence relation is respondent to a division of a set. According to feature space and similarity function, samples belonging to a class described in transcendent knowledge should be grouped. However, in most cases, we can not achieve the ideal conclusion. It often happens that for those samples classified into a category by experts, their distance in feature space is far. While for those belonging to different classes, their distance is near. That is to say, there exists unharmonious character between clustering results and transcendent knowledge.

Therefore, we should abandon the thoughts of uniform granularity [4] and construct a dynamic granularity. For a problem, we adopt multiply granularity size in analysis. Clustering spectrum figure defines an equivalence relation sequence in which granularity size is changing. Choosing a threshold is equal to choosing an equivalence relation $R^{'}$. Then quotient set is obtained, which is knowledge structure $U/R^{'}$ [5]. If the class $X$ described in transcendent knowledge can be precisely expressed by present knowledge structure, we can say that it is harmony between them. But if the upper approximation and the lower approximation are not the same, there is inharmonious.

### 2.2 Clustering Algorithm Based on Dynamic Granularity

Clustering aims to search a knowledge structure, which not only precisely describes the class $X$ defined in transcendent knowledge, but also further reveals its rules. Therefore, we should abandon the thoughts of uniform or average granularity, and adopt the knowledge system of dynamic granularity. In another words, we can construct granularity layers in which different granularity sizes are involved. In the changing of granularity layers, there exist coarser size and finer size. When problem is described too roughly which leads to some characters are lost, we should adopt finer granularity layer. On the other hand, if we describe problem too delicately and each sample is involved in itself class, we should adopt coarser granularity layer. In conclusion, we should choose proper granularity size to the problem.

The introduction of granularity theory leads to cluster efficiently. If granularity size is too fine, each sample constitutes a class and we can not mining rules of samples. If granularity size is too rough, some characters of samples are omitted. Therefore the key to clustering is to choose a proper granularity size. In order to transform among different granularity size conveniently, granularity synthesis method is given following.

**Definition 1.** *Suppose $R_1$ and $R_2$ represent two equivalence relation on universe $X$ respectively. If two following conditions are satisfied, we call $R$ product of $R_1$ and $R_2$, denoted as $R = R_1 \otimes R_2$.*

*(1) $R < R_1$ and $R < R_2$*
*(2) There exists $R^{'}$. Let $R^{'} < R_1$ and $R^{'} < R_2$, and $R^{'} < R$.*

**Definition 2.** *Suppose $R_1$ and $R_2$ represent two equivalence relation on universe $X$ respectively. If two following conditions are satisfied, we call $R$ sum of $R_1$ and $R_2$, denoted as $R = R_1 \oplus R_2$.*

*(1) $R < R_1$ and $R < R_2$*
*(2) There exists $R^{'}$. Let $R^{'} < R_1$ and $R^{'} < R_2$, and $R^{'} < R$.*

Above all, $R_1 \otimes R_2$ is the coarsest division which can subdivide $R_1$ and $R_2$, while $R_1 \oplus R_2$ is the finest division which is divided by $R_1$ and $R_2$. That is to say, $R_1 \otimes R_2$ is the coarsest upper boundary which divide $R_1$ and $R_2$, while $R_1 \oplus R_2$ is the finest lower boundary which divide $R_1$ and $R_2$.

For clustering problem, firstly we set an equivalence relation $R_0$ beforehand to divide sample points, whose corresponding granularity is $\Delta_0$. Then quotient space $S_0$ and clustering result $A_0$ are obtained. If it satisfies the requirement of classification, it shows that clustering granularity is proper. Otherwise, we can adjust granularity size according to following rules. Its theory figure is showed in Fig.1.

(1) Comparing with $\Delta_0$, if granularity is coarser, we should adopt a finer equivalence relation $R_0^{'}$ and set $R_1 = R_0 \otimes R_0^{'}$. Then we can analyze further on $R_1$ and get result $A_1$ and granularity $\Delta_1$. If $A_1$ is still coarser, we can repeat the process above until granularity become finer enough.

(2) Comparing with $\Delta_0$, if granularity is finer, we should adopt a coarser equivalence relation $R_0^{'}$, and set $R_1 = R_0 \oplus R_0^{'}$. Then we can analyze further on $R_1$ and get result $A_1$ and granularity $\Delta_1$. If $A_1$ is still finer, we can repeat the process above until granularity become coarser enough.

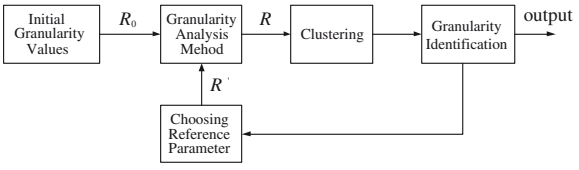



**Fig. 1.** Theory figure of granularity synthesis method

By the method, we can get an equivalence relation family $P = \{R_1,\ R_2, \cdots, R_n\}$, which satisfy partial order relation $R_1 \leq R_2 \leq \cdots \leq R_n$. Then corresponding quotient set order is obtained, which is knowledge system family $U/R_i\ \ (i = 1,\ 2, \cdots, k)$. The class $X$ defined in transcendent knowledge can be expressed by knowledge system $U/R_1$ firstly, and we take its information granularity sets as new research objects. Then it is expressed by $U/R_2$, and so on until we reach the most precision degree and $X$ is expressed precisely by current knowledge system.

**Definition 3.** *As for $K = (U,\ R)$ and $\forall X \subseteq U$, an equivalence relation family of partial order relation $P = \{R_1,\ R_2, \cdots, R_n\}$ is given which satisfy $R_1 \leq R_2 \leq \cdots \leq R_n$. From it harmony degree between clustering results and $X$ is*

$$H(P,\ X) = \frac{|\underline{P}(X)|}{|X|} \tag{1}$$

*Where $|\cdot|$ denotes the cardinal of the sets.*

Obviously, here $H(P,\ X) \in [0,\ 1]$. When $H(P,\ X) = 0$ is satisfied, it shows there does not exist harmony between results and transcendent knowledge. However, when $H(P,\ X) = 1$ is satisfied, it shows that there is the most harmonious between them, that is to say, the current knowledge system can describe transcendent knowledge precisely.

### 2.3 Defects of the Algorithm

Clustering algorithm based on dynamic granularity not only diminishes inconsistence, but also improve correct ratio of clustering. However, there are still some defects. (1) It mainly depends on initial classification. If initial classification is severely far away from global optimization, it easily leads to local optimization. (2) Whether the choice of feature is correct or not and clustering dimension directly affect clustering results. (3) It is not available in data mining of large scale.

## 3 Parallel Immune Clustering Algorithm Based on Dynamic Granularity

In the paper, we introduce parallel immune model into clustering analysis based on dynamic granularity, and propose a developed clustering algorithm. It takes advantage of dynamic granularity to get harmony, and parallel artificial immune to improve efficiency in data mining of large scale.

### 3.1 Clustering Algorithm Based on Parallel Artificial Immune Algorithm

The application of immune algorithm to clustering analysis can solve the problem of low efficiency, high initial sensitive degree and easily dipping into local optimization [6]. Here, some operators involved in new algorithm are introduced in detail.

(1) **Evaluation of antibody.** In order to improve clustering effect, we take distance between clusters and within clusters as factors to construct fitness function. Bigger is fitness value of an antibody, larger is its choice probability. It ensures to maintain antibodies of larger fitness value and accelerate convergence of the algorithm.

N samples $x_j$ $(j = 1,\ 2, \cdots, n)$ are grouped into $c$ classes $G_i$ $(i = 1,\ 2, \cdots, c)$. Therefore each antibody in the population responds to $c$ centers. We take Euclid distance as similar index between vector $x_k$ and center $z_j$ in group $j$. Its fitness function is defined as

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij} ||x_j - z_i||^2 \tag{2}$$

Where $u_{ij}$ denotes the degree of $x_j$ belonging to $G_i$, and its value is 0 or 1.

For $c$ centers, we can group sets into $c$ classes. Distance within $G_i$ is defined as

$$S_i = \frac{1}{|G_i|} \sum_{x \in c_i} |x - \overline{c}_i| \tag{3}$$

Where $\overline{c}_i$ is average value of $G_i$. We define $d_{ij} = ||\overline{c}_i - \overline{c}_j||$ as distance between $G_i$ and $G_j$. Clustering index is defined as

$$D_B = \frac{1}{C} \sum_{i=1}^{C} R_i \tag{4}$$

Where $R_i = max\{\frac{(S_i + S_j)}{d_{ij}}\}$.

Fitness function of an antibody is defined as

$$f(c) = \frac{1}{D_B} \tag{5}$$

(2) **Selection operator.** In evolution process, when the scale of the antibodies reach some degree, which is not yet optimization individuals, we confine them in order to avoid premature. Antibody concentration is proposed to limit antibodies which is in large scale but not optimization solution.

Antibody concentration denotes the scale of the antibodies with similar character, which is defined as

$$Density(c_i) = \frac{1}{\rho(c_i)} = \frac{1}{\sum_{j=1}^{N} |f(c_i) - f(c_j)|} \tag{6}$$

Choice probability of an antibody is defined as

$$P_S(c_i) = \frac{\rho(c_i)}{\sum_{i=1}^{n} \rho(c_i)} = \frac{\sum_{j=1}^{N} |f(c_i) - f(c_j)|}{\sum_{i=1}^{N} \sum_{j=1}^{N} |f(c_i) - f(c_j)|} \tag{7}$$

Supposing initial population is $A(k)$, We calculate each antibody's concentration in the population. According formula 6, we select $m$ ones as memory antibodies to constitute population $B(k)$, whose has the largest selection probability. From above, we know that the larger is concentration of an antibody, the smaller is selection probability. It ensures diversity of antibodies in evolution and avoid premature, which make antibodies of lower fitness value have reproduction choice.

(3) **Similar-taxis operator.** the operator is produced within sub-population. The individuals within a group compete each other until winner individual comes forth. Antibodies of $B(k)$ is operated by similar-taxis operator within radius $R_Q$ to generate new population $C(k)$.

(4) **Dissimilation operator.** As a sub-population easily get into balance state after several generations, in order to break the balance, we should select the optimization individuals representive of their sub-population respectively, and make them compete each other. By this, we can exchange optimization information among sub-population to generate new population $D(k)$.

(5) **Optimization Heritage operator.** In the algorithm, it can maintain and take advantage of optimization antibodies in each sub-population to supervise searching. When the algorithm is carried out every $k$ generations, by the operator we can distribute the optimization individuals into other sub-populations to realize evolution in span.

### 3.2 Artificial Immune Clustering Algorithm Based on Dynamic Granularity

We combine the operators above with dynamic granularity model to produce a new clustering algorithm. In the changing of granularity size, we choose proper granularity by adjusting its size. Algorithm is described as following:

Step 1: Firstly all samples are classified according to transcendent experience.

Step 2: Initialize the parameters of clustering model: $\varepsilon$, $N$, $R_Q$, centers. Centers represent gravity data point of classes, which are continuously adjusted dynamically in the clustering process.

Step 3: According to formula 5 6 7, we select antibodies of high fitness based on concentration. By similar-taxis, dissimilation and optimization operators, we can obtain the optimization antibodies.

Step 4: According to encoding and decoding theory, we can decode the optimization antibodies get by parallel immune algorithm to obtain new clustering centers.

Step 5: According to new clustering centers by decoding, we adjust centers population. If data points within a class belong to a group by testifying, we turn to step 6. Otherwise we turn to step 3.

Step 6: After clustering, we get clustering spectrum figure and a series of threshold. To choose a threshold is responding to choose a equivalence relation family. According to definition 1 and definition 2, we can synthesize granularity to search for proper size and get a equivalence family $P = \{R_1,\ R_2, \cdots, R_n\}$, which satisfy partial order relation $R_1 \leq R_2 \leq \cdots \leq R_n$.

Step 7: $i = 1$

Step 8: Calculate the harmony degree between clustering results and $X$, which is $H_i(P,\ X) = \frac{|P_i(X)|}{|X|}$.

Step 9: If there exists $H_i(P,\ X) = 1$, then clustering results is output. Otherwise set $i = i + 1$ and turn to step 6.

## 4 Tests and Analysis

We testify the two following examples by traditional artificial immune clustering algorithm (AI), clustering algorithm based on dynamic granularity (DG), parallel artificial immune clustering based on dynamic granularity (DGAI).

(1) Test 1 Choose criterion data of IRIS as test samples, which are composed of 150 points of four dimension space. It includes three classes, which are Setosa, Versicolor and Virginica. Each class has 50 samples. We carry out 10 times tests. The comparison of the minimum of value function $J$ by different clustering algorithm is showed in Table 1.

**Table 1.** comparison of clustering conclusion by different algorithm

| algorithm | $J$ | Error classification | Correct classification percent |
|---|---|---|---|
| AI | 6259.21 | 16 | 89.33 |
| DG | 6147.53 | 12 | 92.00 |
| DGAI | 6013.04 | 10 | 93.33 |

Owing to the application of dynamic granularity theory and parallel artificial immune to clustering, it make value function $J$ by DGAI is smaller than by AI and DG. It is obvious that correct classification percent of DGAI is higher.

(2) Test 2 The data in reference [7] is the samples of stones in iron mine district, which has eleven indexes. After several times of clustering by AI, we get value

function $J = 160.24$. However, the value of $J$ was reached by DGAI in high speed. Only by DG, quantity of calculation is easily caused, while by DGAI we can obtain global optimization classification quickly. When samples and classification is in large scale, the developed algorithm shows its advantage.

## 5 Conclusion

In the paper, we combine parallel artificial immune algorithm with dynamic granularity, and produce evolution clustering algorithm based on granularity. We introduce granularity adjustment strategy to choose proper size quickly. Besides, owing to the character of parallel and searching at random in artificial immune algorithm, we apply it to dynamic granularity to construct new clustering algorithm. Tests show that the algorithm's superiority is obvious when dealing with data clustering problem in large scale.

## References

1. Jain, A.K., Dubes, R.C.: Algorithms for clustering. N J Prentice Hall, Englewood Cliffs, New Jersey, USA (1988)
2. Tang, Z., Yamaguchi, T.: Multiple-value immune network model and its simulation. Proceedings of The 27th international symposium on multi-valued logic, Autigonish, Canada (1997) 1: 233-238
3. Lin, T.Y., Hu, X.H., Louie, E.: A Fast Association Rule Algorithm Based on Bitmap and Granular Computing. Proceedings of The 12th IEEE International Conference on Fuzzy, St. Louis, MO, USA (2003) 678–683
4. Xie, K.M., Chen, Z.H., Xie, G.: BGrC for Superheated Steam Temperature System Modeling in Power Plant. Proccedings of the 2006 IEEE International Conference on Granular Computing, Atlanta, Georgia, USA (2006) 708–711
5. Xu, F., Zhang, L.: An Analysis of Uneven Granules Clustering Based on Quotient Space. Journal of Computer Engineering, 31(3) (2005) 26-53.
6. Castro, L.N.D., Zuben, F.J.V.: An evolutionary immune system for data clustering. Proceedings of The Sixth Brazlilian Symulation on Neural Network, Los Alamitos, CA, USA (2000) 84–89
7. Zhang, W., Pan, F.Z.: Fuzzy Clustering Based on Genetic Algorithm. Journal of Hubei University, 24(2) (2002) 101–104

# C-DBSCAN: Density-Based Clustering with Constraints

Carlos Ruiz[1], Myra Spiliopoulou[2], and Ernestina Menasalvas[1]

[1] Facultad de Informatica, Universidad Politecnica, Madrid, Spain
cruiz@cettico.fi.upm.es, emenasalvas@fi.upm.es
[2] Faculty of Computer Science, Otto-von-Guericke-Universität Magdeburg, Germany
myra@iti.cs.uni-magdeburg.de

**Abstract.** Density-based clustering methods are of particular interest for applications where the anticipated groups of data instances are expected to differ in size or shape, arbitrary shapes are possible and the number of clusters is not known a priori. In such applications, background knowledge about group-membership or non-membership of some instances may be available and its exploitation so interesting. Recently, such knowledge is being expressed as constraints and exploited in *constraint-based clustering*. In this paper, we enhance the density-based algorithm DBSCAN with constraints upon data instances – "Must-Link" and "Cannot-Link" constraints. We test the new algorithm C-DBSCAN on artificial and real datasets and show that C-DBSCAN has superior performance to DBSCAN, even when only a small number of constraints is available.

**Keywords:** constraint-based clustering, semi-supervised clustering, instance-level constraints, clustering with constraints, background knowledge.

## 1 Introduction

In the last years, clustering with instance-level constraints has received a lot of attention, because it allows the incorporation of domain knowledge to the knowledge discovery process [1]. Constraint-based clustering exploits the fact that many applications deliver background information in the form of a small set of labeled data, i.e. records that should belong to the same cluster or be in distinct clusters. Such information can be exploited to guide the clustering of the non-labeled data. Examples of constraint-based clustering include the detection of road lanes from GPS data [2] and helping the navigation of a Sony Aibo Robot [3]. Constraints improve clustering quality [2], enhance computational performance [3] and prevent the construction of empty clusters [4].

Most works on constraint-based clustering have been designed for partitioning algorithms [2] and hierarchical algorithms [5]. In this study, we apply the principle of constraint-driven cluster formation upon density-based clustering and show that instance-level constraints enhance performance with respect to cluster quality.

Density-based algorithms identify dense data areas of separated by sparse areas. "Density" may refer to a high concentration of proximal data points or to the closeness of a data point to the mean of a Gaussian. We concentrate on one prominent example of density-based clustering, DBSCAN [6], which discovers neighbourhoods of proximal points in a metric space. DBSCAN requires no background knowledge on the number

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 216–223, 2007.

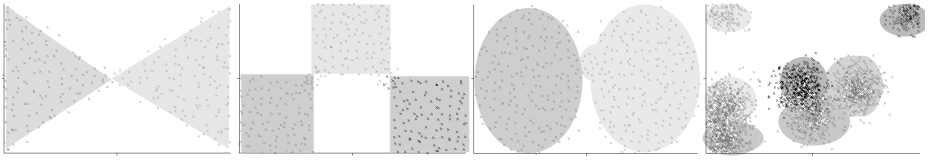


**Fig. 1.** (a) DS1 (b) DS2 (c) DS3 (d) DS4. Samples where DBSCAN does not perform well.

or shape of clusters nor on the data distribution. However, it performs poorly if the clusters are diffuse, partially overlapping, connected by "bridges" or having very different densities. These shortcomings are depicted in the examples of Fig 1 (artificial data).

We extend DBSCAN into *Constraint-driven DBSCAN "C-DBSCAN"*. We first partition the data space into subspaces and then enforce "instance-level constraints" on the data, i.e. constraints that dictate whether some points may appear in the same cluster or not. We use these constraints to drive cluster construction and show experimentally, that the clusterings produced are of higher quality than DBSCAN clusters.

The rest of the paper is organized as follows: Section 2 discusses related literature. We describe C-DBSCAN in Section 3. Section 4 contains our first experiments that compare C-DBSCAN with the DBSCAN for various datasets and sets of constraints. Section 5 summarizes our results and proposes directions for further research.

## 2 Related Work

Clustering with constraints [7,8], also referred to as semi-supervised clustering [9], is a relatively new research direction, in which background knowledge in the form of constraints is used to enhance the clustering process. An overview of the types of constraints proposed in literature can be found in [7]. In this study, we focus on "instance-level constraints": They express background knowledge on the cluster membership of some data instances, i.e. that some instances must appear in the same cluster ("Must-Link" constraints) or in separate clausters ("Cannot-Link" constraints) [2].

There are two approaches for constraint enforcement. In the first one, the objective function is modified into one that satisfies as many constraints as possible [2,10,11,3,5]. In the second one, sometimes called "distance-based", the algorithm is trained on the data involved in constraints, learns a new metric and uses it for clustering [12,13,14].

A challenging aspect is the interplay between achieving a feasible solution (i.e. ensuring convergence) and satisfying all constraints. Davidson et al have proven in [3] that the satisfaction of all Must-Link and Cannot-Link constraints when clustering with K-means is an NP-complete problem; Cannot-Link constraints may prevent the algorithm from converging. When hierarchical clustering is used instead, constraint satisfaction becomes a P-complete problem [5]. In this study, we enhance a density-based algorithm with constraints, i.e. an algorithm that focusses on local optima, similarly to a hierarchical algorithm. This allows us to build clusters that satisfy *all* constraints.

A number of successful density-based clustering algorithms can be found in the literature, starting with DBSCAN in 1996 [6]. DBSCAN has been designed for large and noisy datasets and is able to discover clusters of arbitrary shapes: DBSCAN introduced

the concept of "neighbourhood" as a region of given radius (i.e. a sphere) and containing a minimum number of data points. Connected neighbourhoods form clusters, thus departing from the notion of spherical cluster [6]. The idea of dense neighbourhood has inspired further research on density-based clustering algorithms, including [15,16,17]. In this study, we have opted for the DBSCAN as a reference representative.

## 3 Instance-Level Constraints for Density-Based Clustering

Our constraint-based algorithm builds upon DBSCAN [6]. DBSCAN identifies *neighbourhoods* around *core points*, i.e. points having at least *MinPts* neighbours within a radius *Eps*. Points within the same neighbourhood are *density-reachable*, those in overlapping neighbourhoods are *density-connected*. A cluster is a maximal group of overlapping neighbourhoods.

C-DBSCAN extends DBSCAN in three steps: We first apply a KD-Tree [18] to divide the data space into dense partitions, similarly to DESCRY [17]. We enforce Cannot-Link constraints within each tree leaf, thus producing "local clusters". Next, we merge adjacent local clusters, thereby enforcing the Must-Link constraints. Finally, we merge adjacent clusters hierarchically, while enforcing the remaining Cannot-Link constraints.

### 3.1 Partitioning the Data Space

For the partitioning step (cf. Algorithm 1), we use the KD-Tree construction algorithm [18], which divides the data space iteratively into cubes by splitting planes that are perpendicular to the axes. Each cube becomes a node and is further partitioned, as long as it contains a minimum number of data points (the *MinPts* threshold of DBSCAN). The result is an unbalanced tree: small leaves capture locally dense subareas while large ones cover the less dense subareas. With this partitioning, thin bridges between dense subareas are avoided. Instead of connecting arbitrary adjacent neighbourhoods to build clusters, only neighbourhoods within the same node are considered at first. They are merged into "local clusters", subject to the instance-level concstraints described below.

### 3.2 Introducing Instance-Level Constraints

C-DBSCAN supports two types of constraints among data instances/points: A *Must-Link constraint* for the data points $x$ and $y$ states that they must be assigned to the same cluster. A *Cannot-Link constraint* states that they must be assigned to different clusters.

*Creating Local Clusters under Cannot-Link Constraints.* Step 2 of C-DBSCAN (cf. Algorithm 1) groups density-reachable points into "local clusters" while enforcing Cannot-Link constraints. The leaf nodes of the KD-Tree are traversed: If there is a Cannot-Link constraint involving data points within the same leaf, then each data point of the leaf becomes a singleton local cluster. If there is no Cannot-Link constraints, the conventional DBSCAN is invoked for each data point $p$ in the leaf: It is checked whether there is a neighbourhood with at least *MinPts* points in a radius *Eps* around $p$.

**Algorithm 1.** C-DBSCAN

**Data:**
A set of instances, $D$.
A set of must-link constraints, $ML$, and a set of cannot-link constraints, $CL$.
**Result:** The set $D$ partitioned into clusters that satisfy $ML$ and $CL$.
**begin**
  **Step 1: Partitioning the data space.** *kdtree* := BuildKDTree($D$).
  **Step 2: Creating local clusters under Cannot-Link constraints.**
  **repeat**
    **for** *(all unlabeled points in a leaf X)* **do**
      Select an arbitrary point $p_i$ from $X$.
      $X_{p_i} \leftarrow$ all points in $X$ that are within $Eps$ radius of $p_i$.
      **if** *($X_{p_i}$ contains less than MinPts points)* **then**
        Label $p_i$ as NOISE.
      **else if** *(exists a Cannot-Link constraint in CL among points in $X_{p_i}$)* **then**
        Create a local cluster for each point in $X$. Break.
      **else**
        Label $p_i$ as CORE. Label $X_{p_i}$ as LOCAL CLUSTER.
  **until** *(all leaves of the kdtree have been processed)*
  **Step 3a: Merging local clusters under Must-Link constraints.**
  **for** *(each constraint $m \in ML$)* **do**
    Join clusters involved in constraint $m$ into cluster $Y$.
    Label $Y$ as CORE LOCAL CLUSTER.
  **Step 3b: Merging clusters under Cannot-Link constraints.**
  **for** *(each core (local) cluster Y)* **do**
    **while** *(number of local clusters NLC decreases)* **do**
      *closestCluster* $\leftarrow$ closest local cluster to $Y$.
      **if** *($\nexists$ Cannot-Link constraint in CL between points of Y and closestCluster)* **then**
        $Y \leftarrow Y \cup$ *closestCluster*. Label $Y$ as CORE CLUSTER.
        NLC-=1.
**end**

If the neighbourhood has too few points, then $p$ is a noise point and is ignored. Otherwise, all data points that are density-reachable from it become members of the same local cluster.

*Merging Local Clusters under Must-Link Constraints.* Must-Link constraints are enforced in Step 3a (cf. Algorithm 1). If the data points involved in a Must-Link constraint belong to different local clusters, the clusters are merged into a "core local cluster". At the end of Step 3a, points that should appear together belong either to the same core local cluster (by constraint enforcement) or are already members of the same local cluster – one of the data points is a core point and the other is density-reachable from it.

*Merging Clusters under Cannot-Link Constraints.* In Step 3b (cf. Algorithm 1), we further merge local clusters (also: singleton local clusters and core local clusters output

by Step 3a) to enforce Cannot-Link constraints that are not satisfied yet. For cluster merging, we use hierarchical agglomerative clustering with single linkage, but only consider density-reachable data points when we compute distances. Moreover, we let the core local clusters drive the merging process, in the sense that we do not consider arbitrary clusters as candidates but only the local clusters that are close to each core local cluster. For each such pair of candidates, we check whether they contain data points involved in a Cannot-Link constraint. If this is the case, the clusters are not merged. The algorithm stops when the number of clusters does not change any more.

*Discussion on Constraint Enforcement.* Most constraint-based clustering algorithms make a best effort to enforce all constraints. C-DBSCAN ensures that *all* constraints are satisfied: Step 3a enforces all Must-Link constraints; Step 2 enforces Cannot-Link constraints within the same KD-tree node; Step 3b enforces the remaining Cannot-Link constraints by preventing the merging of clusters. This is achieved at the cost of building many small clusters, since each Cannot-Link constraint results in a singleton local cluster. We believe that some of those singleton clusters can be merged with other ones without constraint violation, so we intend to work on heuristics to this purpose.

## 4 Experimental Results

We have studied the impact of constraints on the clustering results by applying C-DBSCAN and the original DBSCAN on four artificial and three real datasets with a priori known clusters. The artificial datasets contain data, for which DBSCAN is known to perform poorly (cf. Section 1 and Fig. 1). The real datasets are from the Machine Learning Repository UCI [19]. They are depicted in Fig. 2 and summarized in Table 1.

In a real scenario, constraints are derived by studying a small subset of the data; they are the result of human insight. Our datasets do not contain such constraints, so we have generated some randomly. We show that even random constraints do increase cluster quality. For each dataset, we have generated a percentage $x\% = 5\%, 10\%, 15\% \ldots 100\%$ of records involved in Must-Link constraints. The Cannot-Link constraints were derived from the Must-Link constraints and are thus interdependent. However, this only means that the percentage of independent constraints is smaller than $x\%$.

### 4.1 Evaluation method

The datasets we use are labeled, so we can compute cluster quality towards the known classes. For this, we use the Rand Index measure [22], which takes as input two

**Table 1.** Data sets used in the experiments

| **Artificial data sets** | **Real data sets** |
|---|---|
| DS1 (2 classes, bridge between clusters) | Iris [19] (3 classes, 2 overlapped) |
| DS2 (3 classes, 2 overlapped, bridge) | Cure [20] (5 classes, 4 overlapped, 2 by 2) |
| DS3 (2 classes, bridge) | Chameleon [21] (2 classes, bridge) |
| DS4 (7 gaussian clusters, 4 overlapped, 2 by 2, bridge, different notions of density | |

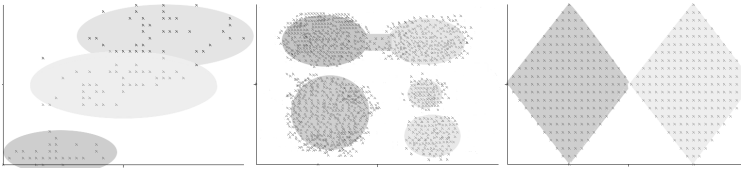


**Fig. 2.** (a) IRIS (b) CURE (c) CHAMELEON

partitionings $\zeta_1$ and $\zeta_2$ , computes the number of "agreements" and "disagreements" among them, and takes the highest value 1 if the partitionings are identical. The "agreements" are the number $a$ of data points that appear together in the same partition for both $\zeta_1$ and in $\zeta_2$ plus the number $b$ of data points that appear in different partition for both $\zeta_1$ and $\zeta_2$. The "disagreements" are the number $c$ of data points that appear in the same partition of $\zeta_1$ and in different partitions of $\zeta_2$ plus the number $d$ of data points that appear in the same partition of $\zeta_2$ and in different partitions of $\zeta_1$. Then:

$$Rand(\zeta_1, \zeta_2) = \frac{agreements}{agreements + disagreements} = \frac{a+b}{a+b+c+d}$$

### 4.2 Results Using Artificial and Known Data Sets

We show the performance of C-DBSCAN for the artificial datasets DS1,...,DS4 in Fig. 3 and for the datasets IRIS, CURE and CHAMELEON in Fig. 4. In both figures, the horizontal axis depicts the percentage of data points involved in constraints. The vertical axis shows the Rand Index value of each clustering towards the classes (the *true* partitioning). The Rand Index for DBSCAN is constant. We have placed this value at point Zero of the axes: Deviations from this point reflect the impact of the constraints. The curves show that C-DBSCAN improves DBSCAN and achieves very good partitionings, reaching RandIndex values of more than 0.8 for all datasets. We see that a small number of arbitrary, labeled records suffices to guide C-DBSCAN towards a good partitioning and that no more constraints are necessary (saturation at 10%). This result strengthens the findings of [2,10] that a small amount of domain information is sufficient to achieve high performance.

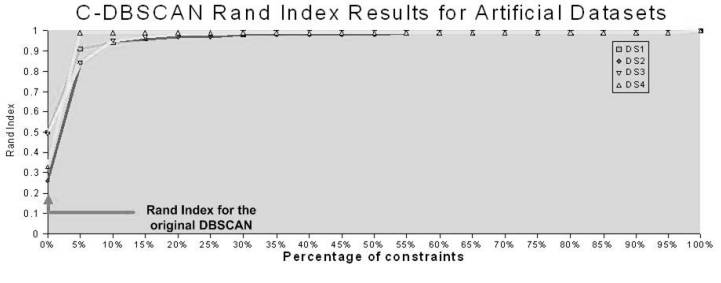



**Fig. 3.** Rand Index values for C-DBSCAN with different constraint-sets - Artificial datasets

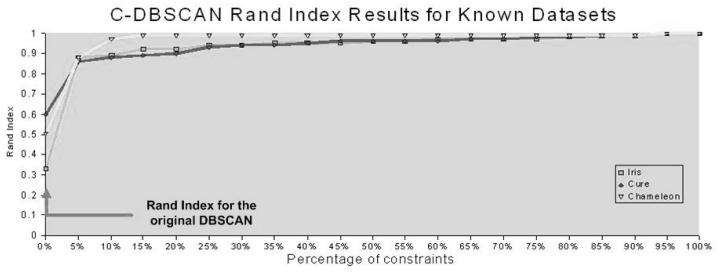



**Fig. 4.** Rand Index values for C-DBSCAN with different constraint-sets - Real datasets

## 5 Conclusions

Constraint-based clustering methods exploit background knowledge to guide the grouping of data into clusters. We have presented C-DBSCAN, a constraint-based extension of the density-based algorithm DBSCAN [6]. C-DBSCAN enforces Must-Link and Cannot-Link constraints among data points and, Differently from best-effort constraint-based clustering algorithms, it guarantees that *all* constraints are satisfied. Our first experiments show that even constraints improve the clustering quality substantially, notably on datasets where the original DBSCAN performs poorly.

We have observed that few constraints suffice to improve quality. We intend to study next the impact of constraint type (Must-Link vs Cannot-Link) on quality and the interplay of constraint type and number of constraints, using larger and more complex datasets. Further, we want to design heuristics that minimize the number of singleton clusters built by C-DBSCAN by merging clusters, while still satisfying all constraints.

A direct comparison of C-DBSCAN to other algorithms, like constraint-based K-means, would not be informative, since the relative behaviour of the basic algorithms for different data distributions is already well studied. However, we plan to study whether C-DBSCAN exploits constraints more effectively than other constraint-based algorithms.

## References

1. Kopanas, I., Avouris, N.M., Daskalaki, S.: The Role of Domain Knowledge in a Large Scale Data Mining Projects. In: SETN'02: Proc. of the Methods and Applications of Artificial Intelligence, Second Hellenic Conf. on AI. Volume 2308 of LNCS, Springer. (2002)
2. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained K-means Clustering with Background Knowledge. In: ICML'01: Proc. of 18th Int. Conf. on Machine Learning. (2001) 577–584
3. Davidson, I., Ravi, S.S.: Clustering with Constraints: Feasibility Issues and the k-Means Algorithm. In: SIAM'05: Proc. of the SIAM Int. Conf. on Data Mining. (2005)
4. Bennett, K., Bradley, P., Demiriz, A.: Constrained K-Means Clustering. Technical report, Microsoft Research (2000) MSR-TR-2000-65.
5. Davidson, I., Ravi, S.S.: Agglomerative Hierarchical Clustering with Constraints: Theoretical and Empirical results. In: PKDD'05: Proc. of Principles of Knowledge Discovery from Databases. (2005) 59–70

6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A Density-Based Algortihm for Discovering Clusters in Large Spatial Database with Noise. In: KDD'96: Proc. of 2nd Int. Conf. on Knowledge Discovery in Databases and Data Mining. (1996)
7. Davidson, I., Basu, S.: Clustering with Constraints. In: ICDM'05: Tutorial at The 5th IEEE Int. Conf. on Data Mining. (2005)
8. Davidson, I., Basu, S.: Clustering with Constraints: Theory and Practice. In: KDD'06: Tutorial at The Int. Conf. on Knowledge Discovery in Databases and Data Mining. (2006)
9. Gunopulos, D., Vazirgiannis, M., Halkidi, M.: From Unsupervised to Semi-supervised Learning: Algorithms and Evaluation Approaches. In: SIAM'06: Tutorial at Society for Industrial and Applied Mathematics Int. Conf. on Data Mining. (2006)
10. Wagstaff, K., Cardie, C.: Clustering with Instance-level Constraints. In: ICML'00: Proc. of 17th Int. Conf. on Machine Learning. (2000) 1103–1110
11. Basu, S., Banerjee, A., Mooney, R.J.: Semi-supervised Clustering by Seeding. In: ICML'02: Proc. of the Int. Conf. on Machine Learning. (2002)
12. Basu, S., Bilenko, M., Mooney, R.J.: A Probabilistic Framework for Semi-Supervised Clustering. In: KDD'04: Proc. of the 10th Int. Conf. on Knowledge Discovery in Databases and Data Mining. (2004) 59–68
13. Bilenko, M., Basu, S., J.Mooney, R.: Integrating Constraints and Metric Learning in Semisupervised Clustering. In: ICML'04: Proc. of the 21th Int. Conf. on Machine Learning. (2004) 11–19
14. Halkidi, M., Gunopulos, D., Kumar, N., Vazirgiannis, M., Domeniconi, C.: A Framework for Semi-Supervised Learning Based on Subjective and Objective Clustering Criteria. In: ICDM'2005: Proc. of the IEEE Int. Conf. on Data Mining. (2005) 637–640
15. Hinneburg, A., Keim, D.A.: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In: KDD'98: Proc. of the 4th Int. Conf. on Knowledge Discovery in Databases and Data Mining. (1998) 58–65
16. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: Ordering Points to Identify the Clustering Structure. In: SIGMOD'99: Proc. of the 1999 ACM SIGMOD Int. Conf. on Management of Data. (1999) 49–60
17. Angiulli, F., Pizzuti, C., Ruffolo, M.: DESCRY: A Density Based Clustering Algorithm for Very Large Data Sets. In: IDEAL'04: Proc. of the Intelligent Data Engineering and Automated Learning. (2004) 203–210
18. Bentley, J.L.: Multidimensional Binary Search Trees Used for Associative Searching. Communications of ACM **18** (1975) 509–517
19. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI Repository of ML Databases (1998)
20. Guha, S., Rastogi, R., Shim, K.: CURE: An Efficient Clustering Algorithm for Large Databases. In: SIGMOD98: Proceeding of the 1998 ACM SIGMOD Int. Conf. on Management of Data. (1998) 73 – 84
21. Karypis, G., Hang, E.H., Kumar, V.: Chameleon: Hierchachical Clustering Using Dynamic Modeling. Computer **32** (1999) 68–75
22. Rand, W.M.: Objective Criteria for the Evalluation of Clustering Methods. In: Journal of the American Statistical Association. 66 (1971) 846–850

# A New Cluster Based Fuzzy Model Tree for Data Modeling

Dae-Jong Lee[1], Sang-Young Park[2], Nahm-Chung Jung[2], and Myung-Geun Chun[3]

[1] CBNU BK21 Chungbuk Information Technology Center, Korea
djmidori@empal.com
[2] Korea Water Resources Corporation, Korea
{sypark119,chung}@kwater.or.kr
[3] Dept. of Electrical and Computer Engineering, Chungbuk National University, Korea
mgchun@chungbuk.ac.kr

**Abstract.** This paper proposes a fuzzy model tree, so-called c-fuzzy model tree, consisting of local linear models using fuzzy cluster for data modeling. Cluster centers are calculated by fuzzy clustering method using all input and output attributes. And then, linear models are constructed at internal nodes with fuzzy membership grades between centers and input attributes. The expansion of internal node is determined by comparing the error calculated at the parent node with the sum of ones at the child nodes. On the other hand, data prediction is performed with the linear model having the highest fuzzy membership value between input attributes and cluster centers at the leaf nodes. To show the effectiveness of the proposed method, we have applied this method to real world data set. We found that the proposed method showed better performance than the widely used methods, such as model tree and artificial neural networks.

**Keywords:** Model Tree, Fuzzy Clustering, Data Modeling.

## 1 Introduction

In recent years there has been a growing tendency of using the data-driven modeling to complement or even replace deterministic models, especially for forecasting. The basic idea of data-driven modeling is to work with real world data of the domain where they are given, and to find a form of relationship that properly explains the specific data sets. Relationships can be derived between parameters that may have little or nothing to do with the physical principles of the underlying processes. A simple example of data-driven model is a linear regression model. However, more complex data-driven models are usually highly non-linear and require sophisticated techniques. Nowadays the data-driven modeling borrows techniques developed in such various areas as statistics, computer science, artificial intelligence, and soft computing. Among them, the neural networks often predict with higher accuracy than the other techniques because of the capability of approximating any continuous function. However, one major drawback often associated with neural networks is their lack of explanation power. It is difficult to explain how the networks arrive at their solutions due to the complex nonlinear mapping of the input data by the networks [1].

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 224–231, 2007.

In many applications, it is required to extract knowledge from trained networks for users to gain better understanding of the problems.

On the other hand, a model tree based decision method has been applied widely in the field of machine learning and data mining. One noticeable advantage is that the model tree mechanism is transparent. Thus, one can follow a tree structure easily to explain how a decision is made [2]-[4]. For building model tree, however, one should consider three major conditions such as choosing the best partition of a region of the feature space, determining the leaves of the tree and choosing a model for each leaf [5] [6]. In design procedure to choose the best attribute, one attribute is chosen at a time in design procedure to choose the best splitting of a region of the feature space. More specifically, one selects the most "discriminative" attribute and expands the tree by adding the node whose attribute's values are located at the branches originating from this node. The growth of the tree relying on a choice of a single attribute can be also considered as a drawback. While being quite simple and transparent, considering two or more attributes as a individual group of variables occurring as the discrimination condition located at some node of the tree may lead to the better tree. In addition, the problem of overall tree optimization can be computationally costly since each attribute should be tested across a number of possible split values [7]-[9].

To alleviate these problems, we propose a cluster based fuzzy model tree(c-fuzzy model tree) which makes it possible to split the input space into several subspaces by taking all input and output attributes rather than an input attribute. More specifically, the subspace (each cluster center) is determined by fuzzy clustering method considering all input and output attributes. And then, linear models are constructed in each subspace at internal nodes having fuzzy membership values between predefined centers and input attributes. The expansion of internal node is determined by comparing errors calculated at the parent node with the sum of ones at the child nodes. And then, data prediction is performed with the linear model having the highest fuzzy membership value between input attributes and cluster centers at the leaf nodes.

The rest of the paper is organized as follows. In section 2, we introduce the proposed method named c-fuzzy mode tree in detail. In section 3, we present our results and compare them with those from other methods for regression. Finally, some concluding remarks are given in Section 4.

## 2 Cluster-Based Fuzzy Model Tree

The architecture of the cluster-based fuzzy model tree develops around fuzzy clusters that are treated as generic building blocks of the tree. The training data set **X** is clustered into *c* clusters so that the similar data points are put together. These clusters are completely characterized by their prototypes [6]. We start with them positioned at *c* top nodes of the tree structure. The way of building the clusters implies a specific way in which we allocate elements of **X** to each of them. In other words, each cluster comes with a subset of **X**, namely $\mathbf{X_1}$, . . . , **Xc**. The process of growing the tree is guided by a certain heterogeneity criterion that quantifies a diversity of the data (with respect to the output variable *y*) falling under the given cluster (node). In growth (splitting) process, an intuitive appealing criterion takes into account of the

performance improvement obtained by comparing performance before partition with it after partition. The growth process will terminate if performance improvement varies only slightly or only a few instances remain. In the regression problem, the simplest performance criterion is the root mean square (RMS) of the output error with respect to an independent set of testing data [10]. The essence of diversity (performance improvement) criterion is to quantify a dispersion of the data "allocated" to the given clusters so that higher dispersion of data results in higher value of criterion. Recall that individual data points belong to the clusters with different membership grades; however, for each data, there is a dominant cluster to which they exhibit the highest degree of membership.

As mentioned before, we use the split criterion based on the performance improvement. In addition, we consider the split criterions presented in Table 1. As seen in Table 1, the performance (RMS) is calculated in the parent node prior to splitting and then splitting procedure is actually performed if the performance is higher than the predefined threshold, otherwise goes next procedure and then produces tree structure with child nodes by fuzzy clustering. In the splitting tree, we should consider not only performance with respect to least square error but also the number of instances after splitting and depth since a few instances and long depth lead to inaccurate liner coefficient and overfitting, respectively.

**Table 1.** Node splitting criterion considered in c-fuzzy model tree

| |
|---|
| - $S_1$: RMS errors in the candidate parent node before splitting |
| - $S_2$: minimum instance numbers remained in each candidate child node after splitting |
| - $S_3$: performance improvement |
| - $S_4$: the depth of tree |

The proposed c-fuzzy model tree algorithm is summarized as follows.

**[Step 1]** Select the parameters ($S_1$, $S_2$, $S_3$, $S_4$) for node splitting as shown in Table 1.
**[Step 2]** Linear coefficients are calculated by least square method (LSE) for input-output pairs {X, Y} with h ($h \geq S_2$) instances in a candidate parent node and then we obtained RMS error ($E_b$) as Eq. (2). Stop if the $E_b$ is below the predefined threshold $S_1$, otherwise go next step.

$$\hat{y}(k) = a_0 + a_1 x_1(k) + a_2 x_2(k) + \cdots + a_q x_q(k) \tag{1}$$

$$E_b = \sqrt{\sum_{k=1}^{h} (\hat{y}(k) - y(k))^2 / h} \tag{2}$$

**[Step 3]** Create c candidate child nodes by FCM. Each child node contains input-output pair {$X_i$, $Y_i$}. For simplicity, let us denote the ith node of tree $N_i$ as $N_i$ ={$X_i$, $Y_i$} where, $X_i$ denotes all elements of the data set belonging to this node in virtue of the highest membership grade.

$$X_i = \{x(k) \mid u_i(x(k)) > u_j(x(k)) \quad \text{for all} \quad j \neq i \tag{3}$$

where the index "j" pertains to the nodes originating from the same parent.

The second set $Y_i$ collects the output coordinates of the elements assigned to $X_i$ as follows.

$$Y_i = \{ y(k) \mid x(k) \in X_i \} \tag{4}$$

**[Step 4]** Compute the number of instances ($n_1$, $n_2$, . . . $n_c$) remained in the candidate child nodes ($N_1$, $N_2$, . . . $N_c$). Stop if the smallest instance number is below $S_2$, and then candidate parent node is considered as leaf node. Otherwise, go next step.

**[Step 5]** Computer performance improvement $\delta$ obtained by comparing performance ($E_b$) before partition with it ($E_f$) after partition. Stop if the performance improvement varies only slightly as S3. In case of stopping, candidate parent node is considered as leaf node with no child node. Otherwise, go next step. In case of that, the candidate parent and child nodes are considered as parent node and internal nodes, respectively.

$$\delta = E_b - E_f \tag{5}$$

where

$$\hat{y}_i(k) = b_{i0} + b_{i1}x_{i1}(k) + b_{i2}x_{i2}(k) + , \cdots , b_{iq}x_{iq}(k), \qquad for\ k = 1,2,\ldots,n_i \tag{6}$$

$$E_f = \sqrt{(\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\hat{y}_i(j) - y_i(j))^2) / \sum_{i=1}^{c} n_i} \tag{7}$$

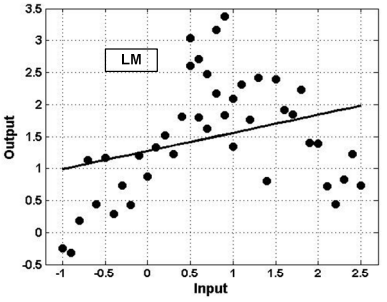
**[Step 6]** Stop if the depth of tree is deeper than $S_4$. Otherwise, go back [Step 1] so the growth process of the tree is repeated for internal nodes satisfying the splitting criterion in Table 1. We consider these nodes as candidate parent nodes for next depth.

## 3 Experimental Studies

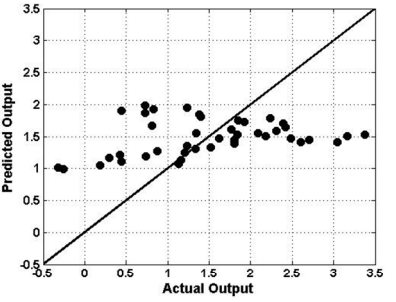
### 3.1 Illustrative Examples with Regression Problems

To illustrate the regression problem with one dimensional input data, let us consider the data set plotted in Fig. 1. We obtain linear coefficients from the least mean square error (RMS error) at the root node. After the first expansion, we create two linear models described as LM1 and LM2 by fuzzy clustering, and then one can see that the error is decreased from 0.8468 to 0.5098. Fig. 2(a) shows the predicted value in each model and (b) shows actual and predicted values according to the input value.

With LM2 among the two linear models (After partitioning for LM1, a child node is not satisfied with the splitting criterion, thus we do not consider the LM1 for growth procedure), we repeated the growing procedure and obtained better performance than one before splitting. More specifically, the error is decreased from 0.5452 to 0.5275 after splitting for LM2. Fig. 3(a) shows the predicted value in each model and Fig. 3(b) shows the actual and predicted values after the second expansion. Fig 4 presents overall structure of the constructed c-fuzzy model tree. As shown in these figures, performances are dramatically improved according to the splitting model.

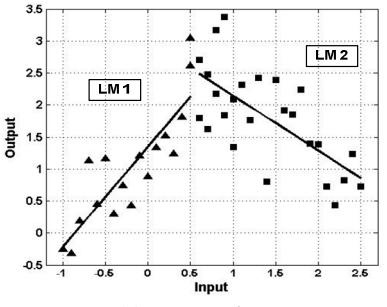



(a) Input and output

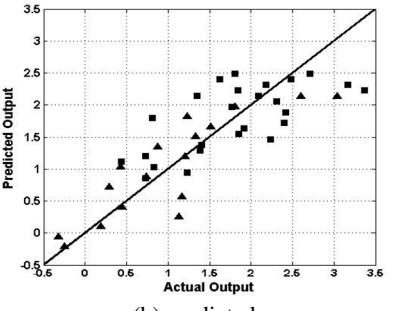



(b) predicted error

**Fig. 1.** Linear model at a root node

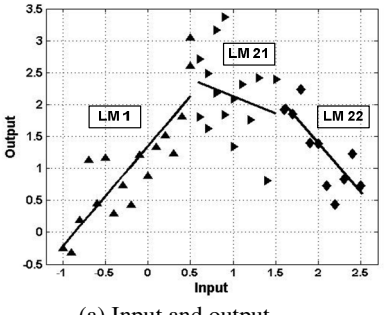



(a) Input and output

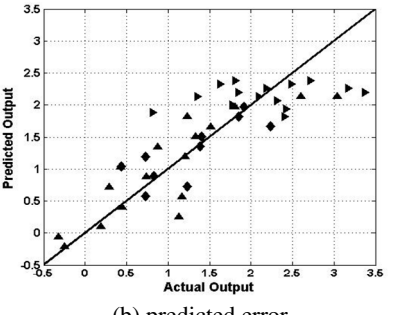



(b) predicted error

**Fig. 2.** Linear models after the first expansion




(a) Input and output




(b) predicted error

**Fig. 3.** Linear models after the second expansion

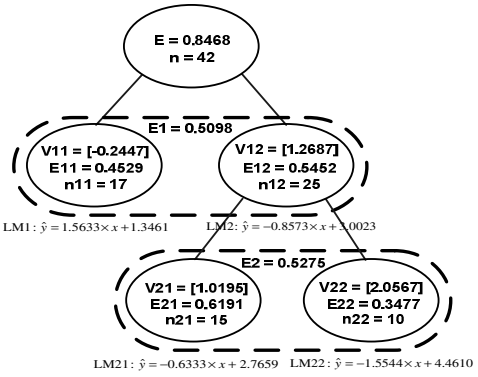



**Fig. 4.** c-fuzzy model tree after the second expansion (E: RMS error, n: instance number, V: cluster center)

### 3.2 Real World Regression Problems

The performance of the proposed c-fuzzy model tree algorithm is compared with the well-known algorithms such as M5' and neural networks. For building the M5' models, a popular date mining software WEKA was used [11]. For the neural networks modeling, related experiments were executed by the BP algorithm called Levenberg-Marquardt algorithm which shows faster running speed than the other methods [12]. On the other hand, all simulations for c-fuzzy model tree algorithm are performed in the MATLAB environment. The performance of C-fuzzy model tree, M5' and BP are compared with 5 real world benchmark data set taken from the site of system HTL(http://www. niaad.liacc.up.pt/~ltorgo/Regression/ds_menu.html). They have a continuous variable to be predicted and have been used as benchmarks in various studies on regression tree and model tree. The specifications of the data set are listed in Table 2. For each experiment, the data set is obtained by randomly generated training and testing data set from its whole data set.

The overall experimental results are shown in Table 3. Here the comparing performance index is the root mean square error. The values of the training and testing data were normalized into a range of 0-1. For simulation, the number of nodes was selected as 10 except for Delta elevator having five for BP [12]. For M5', the minimum number of instances to create a node was set to four and smoothed linear model and pruned option were used. On the other hands, the parameters used in splitting nodes are presented in Table 4. As one can see in Table 5, tree based method such as M5' and proposed method shows better performance than BP, especially, for "Machine CPU" data set. For c-fuzzy model tree, we described the predicted errors obtained at root node (before splitting) and leaf node (after splitting), respectively. As seen in Table 5, the performance was improved according to growing procedures. Comparing with proposed method with M5', it shows better performance than M5'. In particular, the proposed method outperformed the M5' for "Computer activity" data set. From the experimental results, we confirmed that the proposed method make it possible to minimize the predicted errors than the other methods for various data sets.

**Table 2.** Specification of benchmark data sets

| Data sets | # Observations | | # Attributes | | Output properties | | | |
|---|---|---|---|---|---|---|---|---|
| | Training | Test | Continuous | Nominal | Max | Min | Mean | Std |
| Machine CPU | 100 | 109 | 6 | 0 | 1150 | 6 | 105.6 | 160.8 |
| Abalone | 2000 | 2177 | 7 | 1 | 29 | 1 | 9.933 | 3.224 |
| Delta ailerons | 3000 | 4129 | 6 | 0 | 0.002 | -0.002 | -7e-6 | 3.0e-4 |
| Delta elevator | 4000 | 5517 | 6 | 0 | 0.013 | -0.014 | -1e-4 | 0.002 |

**Table 3.** Experimental results for benchmark data sets

| Data sets | BP | | M5' | | Proposed method | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | Root node | | Leaf node | |
| | Training | Testing | Training | Testing | Training | Testing | Training | Testing |
| Machine CPU | 0.0030 | 0.0919 | 0.0245 | 0.0441 | 0.0340 | 0.0546 | 0.0152 | 0.0333 |
| Abalone | 0.0715 | 0.0779 | 0.0774 | 0.0764 | 0.0811 | 0.0781 | 0.0769 | 0.0762 |
| Delta ailerons | 0.0340 | 0.0411 | 0.0367 | 0.0402 | 0.0386 | 0.0411 | 0.0374 | 0.0399 |
| Delta elevators | 0.0506 | 0.0545 | 0.0536 | 0.0528 | 0.0544 | 0.0531 | 0.0537 | 0.0528 |
| Computer activity | 0.1565 | 0.1742 | 0.1728 | 0.1566 | 0.0434 | 0.0445 | 0.0410 | 0.0428 |

**Table 4.** Parameters used at splitting nodes for c-fuzzy model tree

| Data sets | FCM | Node spitting criterion presented in Table 1 | | | |
|---|---|---|---|---|---|
| | Max iteration | $S_1$ | $S_2$ | $S_3$ | $S_4$ |
| Machine CPU | 15 | 0.01 | 20 | 0.001 | 2 |
| Abalone | 10 | 0.01 | 300 | 0.001 | 2 |
| Delta ailerons | 30 | 0.01 | 300 | 0.001 | 3 |
| Delta elevator | 10 | 0.01 | 1000 | 0.001 | 2 |

## 4 Concluding Remarks

This paper proposed a c-fuzzy model tree which makes it possible to split input space into several subspaces by taking all input and output attributes rather than an input attribute as decision tree. In particular, the subspace was determined by the cluster centers calculated by the fuzzy clustering method, and then a linear model was

constructed in each partitioning space at the internal nodes having fuzzy membership values between the predefined centers and input attributes. The expansion of internal node was determined by comparing errors calculated at the parent node with the sum of ones at the child nodes. Finally, data prediction was performed with the linear model having the highest fuzzy membership value between input attributes and cluster centers at the leaf nodes. To show the effectiveness of the proposed method, we have applied our method to various fields observed data sets. Under various experiments, the proposed method showed better performance than the widely used method, such as model tree and neural networks. From these, we confirm that the proposed method can be applied to dramatically reduce the prediction errors than the other methods for various real world data sets.

## References


1. Setiono, R., Thong, J. Y. L.: An approach to generate rules from neural networks for regression problems, European Journal of Operational Research, Vol. 155. (2004) 239-250.
2. Pedrycz, W., Sosnowski, Z.A.: The design of decision trees in the framework of granular data and their application to software quality models, Fuzzy Sets and Systems, Vol. 123. (2001), 271-290
3. Quinlan, J. R.: Learning with continuous classes, in Proceedings AI'92, Adams & Sterling (Eds.), World Scientific, (1992) 343-348
4. Wang, Y., Witten, I. H.: Inducing Model Trees for Continuous Classes, in Poster Paper of the 9th European Conference on Machine Learning (ECML 97)., M. van Someren and G. Widmer (Eds.), Prague, Czech Republic, (1997) 128-137.
5. Malerba, D., Appice, A., Bellino, A., Ceci, M., Pallota, D.: Stepwise Induction of Model Trees, F. ERsposito (Ed): AI*IA 2001, LNAI 2175 (2001) 20-32
6. Pedrycz, W., Sosnowski, Z.A.: C-Fuzzy Decision Tress, IEEE Trans. Systems, Man, and Cybernetics, C: Applications and Reviews, Vol. 35, No. 4. (2005) 498-511
7. Bhattacharya, B., Solomatine, D.P.: Machine learning in sedimentation modeling, Neural Networks, Vol. 19. (2006) 208-214.
8. Bhattacharya, B., Solomatine, D.P.: Neural networks and M5 model trees in modelling water level-discharge relationship, Neurocomputing, Vol. 63. (2005) 381-396
9. Slolmatine, D. P., Siek, M.B.: Modular learning models in forecasting natural phenomena, Neural Networks, Vol. 19. (2006) 215-224
10. Mendonca, L. F., et al.: Decision tree search methods in fuzzy modeling and classification, International Journal of Approximate Reasoning 44 (2007) 106-123
11. Witten, I. H., Frank, E.: DATA MININING-Practical machine learning tools and techniques, Morgan Kaufmann publisher, (2005)
12. Huang, G. B., Zhu, Q. Y., Siew, C. K.: Extreme learning machine: Theory and applications, Neurocomputing 70 (2006) 489-501

# Parameter Tuning for Disjoint Clusters Based on Concept Lattices with Application to Location Learning

Brandon M. Hauff and Jitender S. Deogun

University of Nebraska - Lincoln
Lincoln, NE 68588-0115
bhauf@cse.unl.edu, deogun@cse.unl.edu

**Abstract.** Clustering is a technique for grouping items in a dataset that are similar, while separating those items that are dissimilar. The use of concept lattices, from Formal Concept Analysis, for disjoint clustering is a recently studied problem. We develop an algorithm for disjoint clustering of transactional databases using concept lattices. Several heuristics are developed for tuning the support parameters used in this algorithm. Additionally, we discuss the application of this algorithm to Location Learning. In location learning, an object (for example an employee) to be tracked and localized carries an electronic tag, such as an RFID, capable of communicating with some access points that are in the range of the tag. Clustering can then be used to estimate the location of the tag given the signal strengths that can be heard.

**Keywords:** Clustering, Concept Lattice, Data Mining, Formal Concept Analysis, Frequent Itemsets, Location Learning, Parameter Tuning.

## 1 Introduction

Data Mining is a discipline concerned with extracting information from large sets of data. This research in data mining deals specifically with *clustering* of the data in a transactional database and seeks to improve the clustering process on transactional databases through parameter learning for a specialized clustering algorithm that uses concept lattices found in formal concept analysis. In clustering, objects are divided into groups based on some properties. Thus, these groups represent a relationship that the objects within it hold. Objects within a group are more closely related to one another than to objects in other groups.

Formal Concept Analysis (FCA) provides several tools for data mining and clustering. FCA is primarily concerned with contexts, concepts and conceptual hierarchies [1]. A context is essentially a table of items and item values. A concept is a group of items that share common attribute values. The concept lattice is a lattice structure where each node represents a concept in a hierarchical structure. Clusters can easily be formed from a fully constructed concept lattice. The use of concept lattices for disjoint clustering is a problem that has recently been studied by Saquer [2].

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 232–239, 2007.

The approach to disjoint clustering with concept lattices in [2] is based on finding Frequent Closed Feature Sets (FCFS) in the transactional database being clustered. Tools such as [3] generate FCFSs from transactional databases. Each FCFS becomes an initial cluster candidate and each transaction is assigned to one or more of these cluster candidates. The clusters are then made disjoint through a scoring function. In this research, we develop an algorithm for disjoint clustering based on the approach given in [2]. We study how the algorithms parameters can be tuned to facilitate finding "better clusters" for a given transactional database.

Our clustering technique can be extended to additional problem domains. Specifically, the problem of location learning. In the location learning problem an area, say an office building, has a grid of beacons deployed that transmit data such as location using radio frequency (RF). An object (for example an employee) to be localized and tracked carries a reader capable of measuring signal strengths of some or all of these beacons that are in the range of the reader. Clustering can then be used to identify the location of a reader given the signal strengths that can be heard.

## 2 Related Works

Frequent itemsets provide an attractive solution to document clustering because of their ability to reduce the dimensionality of the vocabulary. Fung *et al.* present a hierarchical document clustering algorithm using frequent itemsets in [4]. Their approach differs from our approach in a couple ways. First, [4] utilize frequent itemsets, while we utilize frequent closed itemsets, a subset of frequent itemsets. This is important, as frequent closed itemsets contain all dataset information necessary for clustering. Second, in our approach a document would only be assigned to a cluster that represents the maximal frequent closed itemset in that document.

Frequent term-based text clustering is studied by Beil *et al.* in [5]. This clustering approach accomplishes the goal of providing interactive exploration of document spaces, such as the world wide web. Like [4] Beil utilizes frequent itemsets to minimize the dimensionality of large sets of text and hypertext documents. Conceptual clustering utilizes concept lattices to discover additional insight about datasets. The combination of Formal Concept Analysis and Conceptual Clustering is studied in [6,7,8].

## 3 Methods

Formal Concept Analysis (FCA) [1] provides mathematical views of *contexts*, *concepts* and *conceptual hierarchies*. A context is a view of a set of objects and their attributes. A formal context is the triple $T = (G, M, I)$ where $G$ and $M$ are two sets and $I$ is a binary relation between them. The set $G$ represents the objects in a context. The set $M$ represents the attributes of the objects in $G$.

The attributes that are common to a set of objects $A \subseteq G$ are defined as $\beta(A) = \{m \in M \mid gIm \ \forall \ g \in A\}$. The set of objects that all contain the set

of attributes $B \in M$ are defined as $\alpha(B) = \{g \in G \mid gIm \ \forall \ m \in B\}$. A set of objects that share a set of attributes $B$ can be described as a *formal concept* $(A, B)$ where $A \subseteq G$, $B \subseteq M$, $\beta(A) = B$ and $\alpha(B) = A$. The set $A$ is the *extent* of the concept and $B$ is the *intent* of the concept. We can define $\mathcal{K}$ as the set of all concepts in the context $(G, M, I)$.

Concepts of a context have a hierarchical relationship, that is, if $(A_1, B_1)$ and $(A_2, B_2)$ are concepts of a context and $A_1 \subseteq A_2$ (equivalently $B_2 \subseteq B1$), we say $(A_1, B_1)$ is a *subconcept* of $(A_2, B_2)$, and $(A_2, B_2)$ is a *superconcept* of $(A_1, B_1)$. We then write $(A_1, B_1) \leq (A_2, B_2)$. The $\leq$ symbols represents the *order* of the concepts. This order of the concepts is used to form the concept lattice.

The concept lattice is a *Hasse* diagram where each concept is a node. An edge exists between nodes representing concept $C_1$ and $C_2$ if and only if $C_1 \leq C_2$ and there is no other concept $C_3$ such that $C_1 \leq C_3 \leq C_2$. The intent and extent of a concept representing a node in the graph are used as the labels of the node.

A *closed feature set* has the property $\beta(\alpha(B)) = B$ for a set of features $B \subseteq M$. A closed feature set is thus the maximal set of features shared by a set of objects. As mentioned, each node of a concept lattice is a concept, and we observe for a concept $(A, B)$ that $\beta(A) = B$ and $\alpha(B) = A$, thus by substitution $\beta(\alpha(B)) = B$. Following this observation it is clear that the intents of the concept lattice are closed feature sets.

The *support* of an individual feature in a transactional database is calculated as the percent of objects that contain that feature. Similarly, the support of a set of features $B$ is the percentage of objects that contain all the features in $B$. Formally, $support(B) = \frac{|\alpha(B)|}{|G|}$. In clustering transactions we are only interested in sets of features that meet a user-specified minimum support level, $minSupport$. Thus, a feature set $B$ is only frequent if $support(B) \geq minSupport$ holds for the transactional database. A closed feature set that is frequent is called a *frequent closed feature set* (FCFS).

### 3.1 Generating Initial Clusters

The clustering of the transactional database is based on FCFSs from FCA. For clustering we consider each transaction of the database to be an object. The transactional database $T$, that is to be clustered, contains a set of objects $G = \{g_1, g_2, g_3, ..., g_n\}$. A clustering of $G$ results in a set $C = \{C_1, C_2, C_3, ..., C_k\}$ where each $C_i \subseteq G$ and the union of all clusters is $\bigcup_{i=1}^{k} C_i = G$. We add an additional restriction that $C_i \cap C_j \ \forall i \neq j$ in order to guarantee that clusters are disjoint, that is to say each object is assigned to only one cluster.

The set of transactions in $T$ are processed by [3] in order to generate a set $S$ of FCFSs ($fcfs$), given a $minSupport$ parameter. The number and quality of FCFSs is controlled by this parameter.

For the purpose of initial cluster generation we assign each $fcfs \in S$ as an initial cluster. Thus, the number of initial clusters $|C| = |S|$. The label for each cluster $C_i \in C$ is the label of the $fcfs$, which is a listing of the contained features providing a clear description of the cluster.

### 3.2 Assigning Transactions to Initial Clusters

Each object $g_i \in G$ must be assigned to initial clusters based on the *Maximal Frequent Closed Feature Sets* (MFCFS) it contains. There are several important characteristics of this process to examine. First, objects may be assigned to multiple clusters in this initial phase, thus, clusters are not disjoint. Second, objects contain at least all features described by their cluster, thus, the cluster labels are descriptive of their objects, allowing simplified interpretation.

### 3.3 Making Clusters Disjoint

In order to make the initial cluster assignments disjoint, we must find the "best" cluster for each object $g_i \in G$, and remove $g_i$ from all other clusters. This process is accomplished using a scoring function, that measures the goodness of an object cluster pair, $(g, C_i)$. One metric for determining the goodness of a cluster for an object is by evaluating how many frequent features the object and cluster share. A high number of shared frequent features implies a close relationship between an object and cluster. The function *global-support(f)* calculates the percentage of objects in the database containing the feature $f$. The feature $f$ is then said to be globally frequent if the percentage exceeds a user specified minimum threshold *globalSupport*. Similarly, the function *cluster-support(f)* calculates the percentage of objects in cluster $C_i$ containing feature $f$. The feature $f$ is then said to be cluster frequent if the percentage exceeds a user specified minimum threshold *clusterSupport*.

We assign all global-frequent features from object $g$ to one of two sets: *pos* or *neg*. A feature $f \in \beta(g)$ that is globally-frequent and cluster-frequent is assigned to the set *pos*, while a feature $f \in \beta(g)$ that is globally-frequent but not cluster-frequent is assigned to the set *neg*. Thus, for cluster $C_i$ we have the sets $pos(g, C_i)$ and $neg(g, C_i)$. The score function is then defined as:

$$score(g, C_i) = \sum_{f \in pos(g, C_i)} cluster - support(f) - \sum_{f \in neg(g, C_i)} global - support(f).$$

The sum of the *cluster-support*$(f)$ values for all features in $pos(g, C_i)$ puts emphasis on intra-cluster similarity and gives a boost to goodness of cluster $C_i$ for object $g$. On the other hand, the sum of the *global-support*$(f)$ values for all features in $neg(g, C_i)$ penalizes $C_i$ as these features contribute to inter-cluster similarity.

The score of object $g$ is calculated for each cluster $C_i$ it is a member of. The object $g$ is permanently assigned to the cluster with the highest score, and removed from all other clusters. The ties are broken by assigning $g$ to the cluster with the longest label. Further ties can be broken randomly.

### 3.4 Disjoint Clustering with Concept Lattices Algorithm

In the previous sections we have presented algorithms related to components, initially described in [2], of the disjoint clustering with concept lattices algorithm. The algorithm developed in this paper is shown in Fig. 1.

**Disjoint Clustering with Concept Lattices Algorithm**

1. Input: Database $T$, *minSupport* for generating the FCFS.
2. Output: The set of disjoint clusters $C$.
3. Run [3] on $T$ with *minSupport* to generate the FCFS, store in cluster set $C$.
4. Extract all objects from $T$ and store in the set $G$.
5. For each $g \in G$ assign each initial cluster $C_i$ to $g$ if $C_i$ is a MFCFS for $g$.
6. For each $g \in G$ calculate $score(g, C_i)$ for each $C_i$ that $g$ is assigned to. Assign $g$ to best $C_i$ according to MFCFS, remove $g$ from all others.
7. Return disjoint clusters $C$.

**Fig. 1.** Disjoint Clustering with Concept Lattices Algorithm

## 4 Parameter Learning

Automatic parameter tuning in the Disjoint Clustering with Concept Lattices Algorithm with Parameter Tuning (CLAPT) is an exploratative process, as the best combination of *globalSupport* and *clusterSupport* is unknown. It is not possible to use traditional Artificial Intelligence search algorithms, as they require distance estimates that approximate the distance traveled through the search space, and the distance still to be traveled to reach the goal node. It is not possible to estimate these metrics in CLAPT, nor is the goal node known. We propose two heuristics, that use a Cluster Evaluation Rating (CER) which combines three metrics: *density*, *distance*, and *class purity*, to perform the exploratative process of CLAPT.

The most computationally intensive heuristic is **k-Granularity**. In the k-Granularity heuristic an exhaustive search of the support space is performed to $k$ decimal places of support accuracy. For example, with $k = 2$ there will be 100 tests. The first run has $globalSupport, clusterSupport$ pair $(0.1, 0.1)$, the second run is $(0.1, 0.2)$, the 100th run is $(1.0, 1.0)$. The choice of $k$ is influenced by the size of the dataset. The results of 2-Granularity test show a landscape of the dataset which can then be used for determing appropriate further tests.

The **k-DepthFirstSearch** (k-DFS) heuristic combines the k-Granularity heuristic with search space pruning techniques. The k-DFS heuristic begins by invoking a 2-Granularity search of the support space. Each CLAPT result from the 2-Granularity search is set as the child of a root node in a tree $T$. The k-DFS heuristic then selects the child with the best CER. The selected node then becomes the current node and a set of child nodes with finer support granularity are computed. This continues for k levels of the tree.

In order to limit the search space at each level of the tree a 2-Granularity search is only performed at the root of the tree. At height 1 in the tree a 1-Granularity search is performed on the *clusterSupport* parameter. At height 2 in the tree a 1-Granularity search is performed on the *globalSupport* parameter. Thus, odd levels of the tree expand the *clusterSupport* parameter and even levels of the tree expand the *globalSupport* parameter.

### 4.1 Experiments

To test the CLAPT algorithm we choose several datasets from the UCI Machine Learning Repository [9]. The experimental results for the datasets of most interest, Congressional Voting and Breast-Cancer, are presented below.

The k-Granularity heuristic provides a wide view of the dataset being evaluated. We first evaluate the k-Granularity heuristic on the Congressional Voting dataset. Figure 2 shows that in order to maximize the CER for the Congressional Voting dataset, small *minSupport* support values should be used. The parameter space for a *minSupport* support of 0.3 is primarily flat. This means that transactions tend to be assigned to few initial clusters, thus, there is little volatility in the CER for most parameter combinations. The highest CER, 0.766, was achieved with $globalSupport = 0.5$, and $clusterSupport = 0.5$. CLAPT generated 51 clusters using this parameter tuning. The average cluster density was 1.878, while the average cluster distance was 7.24. In other words, on average there were fewer than two features that differed between each pair of transactions in a cluster. Additionally, the clusters were separated by over seven features on the average.

The evaluation of k-Granularity on the Congressional Voting dataset for larger *minSupport* values yields some interesting results. First, the CER volatility increases as *minSupport* increases, indicating transactions are being assigned to many clusters initially.

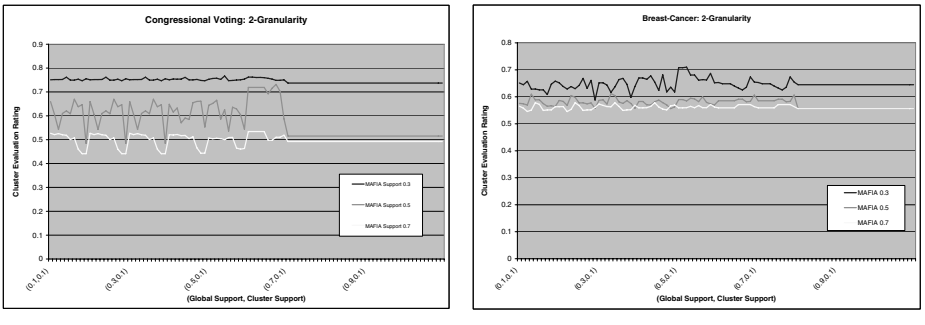



**Fig. 2.** 2-Granularity results

The Breast-Cancer dataset yields similar results as those seen in the Congressional Voting dataset. Lower *minSupport* values consistently perform better in the k-Granularity evaluation of this dataset. The max CER is achieved with $globalSupport = 0.5$ and $clusterSupport = 0.3$. The number of clusters generated for this parameter configuration was 13. The k-Granularity test had a more difficult time finding high CER values for the Breast-Cancer dataset than it did for the Congressional Voting dataset. This suggests that the k-DFS heuristic may be able to expand the parameter space further and thus be able to find a more accurate CER.

**Table 1.** The k-DFS heuristic was used to search to a depth of $k = 5$ in the datasets

| Congressional Voting | | | | Breast-Cancer | | | |
|---|---|---|---|---|---|---|---|
| Global Support | Cluster Support | Number of Clusters | CER | Global Support | Cluster Support | Number of Clusters | CER |
| 0.5 | 0.5 | 51 | 0.76681 | 0.5 | 0.3 | 13 | 0.70976 |
| 0.5 | 0.53 | 51 | 0.76486 | 0.5 | 0.36 | 13 | 0.91653 |
| 0.56 | 0.53 | 50 | 0.76583 | 0.51 | 0.36 | 13 | 0.91653 |
| 0.56 | 0.531 | 50 | 0.76583 | 0.51 | 0.361 | 13 | 0.89481 |
| 0.561 | 0.531 | 50 | 0.76583 | 0.511 | 0.361 | 13 | 0.89481 |

We test k-DFS according to the best *minSupport* rating for each dataset. Table 1 details a 5-DFS test on the Congressional Voting dataset with *minSupport* 0.3. The best CER found using k-Granularity is detailed in row one. The k-DFS heuristic searches the parameter space for a better parameter configuration. Each step through the tree that was made is detailed in Table 1. The k-DFS heuristic was unable to find a better parameter configuration than $globalSupport = 0.5$ and $clusterSupport = 0.5$ for the Congressional Voting dataset. This suggests that the dataset is not sensitive to increased parameter precision.

The k-DFS heuristic was very successful in finding better parameter configurations for the Breast-Cancer dataset, as shown in Table 1. Notice that a change of .06 to the *clusterSupport* parameter increased the CER from 0.70976 to 0.91653. The number of clusters does not change, but the placement of the transactions yields clusters that are more dense, and more distant from each other.

## 5 Location Learning

We apply a relaxed version of the Disjoint Clustering with Concept Lattices algorithm to the tracking and localization problem. We predict the location of a reading $r$ by first assigning it to a cluster $C_i$ if $C_i$ is a MFCFS for $r$. After $r$ is assigned to all of its initial clusters, $score(r, C_i)$ is calculated for each $C_i$. The computed score values are then used as weights in a linear combination of the clusters $C$ to determine the location of the new reading $r$. We were able to predict locations of new readings within a multiplicative constant of 2 or 3 compared to competing methods. Our decreased accuracy is a side effect of using a reduced set of information available to other learners in order to accomodate our model. Details of the location learning experiments are omitted because of space limitations and we are in the process of developing improvements to the location learning approach. This is the first known application of clustering with concept lattices applied to the location learning problem.

## 6 Conclusions and Future Works

We have developed an algorithm CLAPT that finds disjoint clusters for a transactional database using concept lattices, based on results reported in [2]. The

quality of the clusters generated by CLAPT are dependent on three parameters: *minSupport*, *globalSupport*, and *clusterSupport*. CLAPT uses the FCFSs generated by [3] as the initial clusters, and we empirically determine what range of *minSupport* values are optimal for a given dataset. We find that support levels above 0.5 always degrade the quality of disjoint clusters generated by CLAPT. In the testing of the datasets we find that a *minSupport* value between 0.1 and 0.3 provides the highest quality frequent closed feature sets.

The tuning mechanism receives the CER through a feedback system, which allows a search heuristic to further explore the parameter space in order to optimize the generated clusters. Two search heuristics were developed and tested in this paper: k-Granularity and k-DFS. Each dataset was tested with each search heuristic and disjoint clusters found were compared. In some case the k-Granularity heuristic was able to find parameter combinations at least as good as those found by k-DFS. The k-DFS heuristic further explores the parameter space after performing a 2-Granularity search. Thus, k-DFS is able to optimize clusterings in datasets that are sensitive to increased parameter precision.

This paper, thus, has several main contributions. We contribute an algorithm for disjoint clustering using concept lattices. A method of evaluating the disjoint clusters is adopted, and utilized to automatically tune parameters. Finally, we apply a relaxed version of CLAPT to the location learning problem.

## References

1. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1999) Translator-C. Franzke.
2. Saquer, J.M.: Using concept lattices for disjoint clustering. In: Proceedings of Information Knowledge and Sharing IKS, Scottsdale, AZ, USA (2003)
3. Burdick, D., Calimlim, M., Gehrke, J.: MAFIA: A maximal frequent itemset algorithm for transactional databases. In: Proceedings of the 17th International Conference on Data Engineering. (2001) 443–452
4. Fung, B.C.M., Wang, K., Ester, M.: Hierarchical document clustering using frequent itemsets. In: SDM. (2003)
5. Beil, F., Ester, M., Xu, X.: Frequent term-based text clustering. In: KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Press (2002) 436–442
6. Carpineto, C., Romano, G.: GALOIS: An order-theoretic approach to conceptual clustering. In: ICML Proceedings of International Conference on Machine Learning. (1993) 33–40
7. Mineau, G.W., Godin, R.: Automatic structuring of knowledge bases by conceptual clustering. Knowledge and Data Engineering **7** (1995) 824–828
8. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with TITANIC. Data Knowledge Engineering **42** (2002) 189–222
9. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI repository of machine learning databases (1998)

# Web Document Classification Based on Rough Set

Qiguo Duan, Duoqian Miao, and Min Chen

Department of Computer Science and Technology, Tongji University, Shanghai 201804, China
The Key Laboratory of "Embedded System and Service Computing", Ministry of Education, Shanghai 201804, China
dqgcn@126.com, miaoduoqian@163.com, tomatocm@163.com

**Abstract.** For traditional way of Web document representation in Vector Space Model, zero-valued similarity problem between vectors occurs frequently, which decreases classificatory quality when defining the relation between Web documents. In this paper, a novel Web document representation and classification approach based on rough set is proposed. Firstly, TF*IDF weighting scheme is used to assign weight values for Web document's vector. The weights of those terms which do not occur in a Web document are considered missing information. Then rough set for incomplete information is introduced to supplement loss and expand Web document representation. Through generating tolerance classes in both term space and Web document space, the missing information of Web document can be complemented by incorporating the corresponding weights of terms in tolerance classes, which extends the essential information to Web document. Finally, Web document classification algorithm is implemented. Experimental results show that the performance of the classification is greatly improved.

**Keywords:** Rough sets, Web document classification, Web mining.

## 1 Introduction

With the rapid growth of information on the World Wide Web, automatic classification of Web documents has become important for effective retrieval. As one of the essential techniques for Web mining, Web document classification has been studied extensively [1], [2], [3]. Nowadays, many Web document classification methods are based on the Vector Space Model (VSM), which is a widely used data model for text mining. In VSM, a Web document is represented as a term vector. Term weights, contained in each term vector, are assigned by weighting schemes. Traditionally, the weights of those terms which do not occur in the Web document are assigned zero value. A single Web document is usually represented by relatively few terms, thereby the Web document vector is characteristic of high dimension and sparseness, which results in zero-valued similarity between vectors. This problem would decrease classificatory quality because the relation

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 240–247, 2007.

between Web documents is defined by measuring distance of the corresponding vectors.

In this paper, a rough set approach to Web document representation and classification is proposed. Instead of assigning zero to the weights of those terms are absent in a Web document, these weights are considered missing information. Thus Web document is represented as an incomplete term vector firstly. Then through generating tolerance classes in both term space and Web document space, the missing information of Web page can be complemented by incorporating the corresponding weights of terms in tolerance classes. Only using a little heuristics knowledge, the zero-valued similarity problem can be avoided through complementing the potential missing information and therefore the classification performance can be improved.

The rest of the paper is organized as follows. Section 2 describes the weighting scheme briefly. Section 3 introduces the extended rough set for incomplete information. Section 4 presents the novel approach to Web document representation and classification in detail. Section 5 reports and discusses the experimental results and section 6 concludes the paper.

## 2 Weighting Scheme

In VSM, each Web document is viewed as a bag of terms and represented by a term vector. In this paper, we apply the popular TF*IDF (Term Frequency times Inverse Document Frequency) weighting scheme to assign weight values for Web document's vector. The standard TF*IDF is defined as follows:

$$w_{ij} = tf_{ij} \times \log(N/df_i) \ . \tag{1}$$

where $tf_{ij}$ is the frequency of the term $t_i$ in Web document $d_j$; $df_i$ is number of Web documents in which term $t_i$ occurs; $N$ is the total number of Web documents. Normalization by vector's length is applied to all vectors:

$$w^*_{ij} = w_{ij} / \sqrt{\sum_{t_k \in d_i} (w_{ik})^2} \ . \tag{2}$$

Assume that there are $N$ Web documents and $n$ different terms in a set of Web document. Using TF*IDF, each Web document is represented by an $n$-dimensional term vector. The $N$ Web documents in the set can be represented by an $N \times n$ matrix, $DW = [w^{'}_{ij}]$, where $w^{'}_{ij} = w^*_{ij}$ , if the term $t_j$ occurs in the Web document $d_j$; otherwise, $w^{'}_{ij} = 0$. Together with decision attributes, i.e., the class label of Web documents, the matrix can be considered as a decision table. According to the weight computation, if the term $t_j$ is absent in the Web document $d_i$, $w^{'}_{ij}$ is equal to zero. This way of assigning the weights to absent terms brings zero-valued similarity problem between vectors. In this paper, as an extended rough set, tolerance rough set is preferred to avoid zero-valued similarity through complementing the incomplete information of Web documents.

## 3 Extended Rough Set for Incomplete Information

Rough set theory, introduced by Pawlak, is a formal mathematical tool to deal with incomplete or imprecise information [4]. It has been successful in many applications [9] [10]. The classical rough set theory is based on equivalence relation that divides the universe of objects into disjoint classes [4]. By relaxing the equivalence relation to a tolerance relation, where transitivity property is not required, a generalized tolerance space is introduced below [5], [6], [7], [8].

Let $S = (U, A, V, f)$ be an information system, where $U$ is a nonempty finite set of objects called universe of discourse, $A$ is a nonempty finite set of conditional attributes; and for every a$\in A$, such that $f: U \rightarrow V_a$ , where $V_a$ is called the value set of attribute $a$.

**Definition 1.** *If some of the precise attribute values in an information system are not known, i.e., missing or known partially, then such a system is called an incomplete information system. Otherwise the system is called a complete information system.*

**Definition 2.** *Let $S = (U, A, V, f)$ be an incomplete information system and the sign * denote null value, a tolerance relation $T$ is defined as:*

$$T(B) = \{(x, y) \in U \times U | \forall b \in B, b(x) = b(y) \vee b(x) = * \vee b(y) = *\} . \quad (3)$$

where $B \subseteq A$. Obviously, $T$ is reflexive and symmetric, but not transferable. Let $I_B(x) = \{y \in U | (x, y) \in T(B)\}$, and then $I_B(x)$ is called the tolerance class of the object $x$ with respect to the set $B \subseteq A$.

**Definition 3.** *Let $S = (U, A, V, f)$ be an incomplete information system, $X \subseteq U$, $B \subseteq A$, the upper approximation and lower approximation of $X$ with regard to attribute set $B$ under the tolerance relation $T$ can be defined as:*

$$U_B(X) = \{x \in U \,|I_B(x) \cap X \neq \emptyset\} . \quad (4)$$

$$L_B(X) = \{x \in U \,|I_B(x) \subseteq X\} . \quad (5)$$

## 4 Web Document Representation and Classification

### 4.1 Web Document Representation

According to Section 3, here we introduce the corresponding concepts in the Web document classification domain.

An incomplete information system for a web page set is represented as $WS = (U, TS \cup \{class\}, f)$, where $U$ is the set of Web documents, each Web document is an object $d \in U$; $TS$ is the set of total terms which occur in the Web document set, *class* is the decision attribute, i.e., the class label of the Web documents. The weights of those terms which do not occur in a Web document are considered missing information and denoted by sign * instead of zero.

In Web document space, the tolerance relation and tolerance class of Web document are defined as:

**Definition 4.** *For a subset of TS, $B \subseteq TS$, a tolerance relation T(B) on U is defined as:*

$$T(B) = \{(d_x, d_y) \in U \times U | \forall b \in B, |b(d_x) - b(d_y)| \leq \delta \vee b(d_x) = * \vee b(d_y) = *\} . \tag{6}$$

Because weights are real values, the requirement $b(d_x) = b(d_y)$ is too strict. Here it is replaced with $|b(d_x) - b(d_y)| \leq \delta$, where $\delta \in [0, 1]$. Consequently, tolerance class of a Web document $d_x$ with respect to $B \subseteq TS$, $I_B(d_x)$, is the set of Web documents which are indiscernible to $d_x$, i.e., $I_B(d_x) = \{d_y \in U | (d_x, d_y) \in T(B)\}$.

On the other hand, correlation between terms is valuable for complementing missing information. Thus, the tolerance class of term is also defined in term space. Let $U = \{d_1, \ldots, d_M\}$ be a set of Web documents and $TS = \{t_1, ..., t_N\}$ set of terms for $U$. The tolerance space of term is defined over a universe of all terms for $U$.

**Definition 5.** *Let $f_U(t_i, t_j)$ denotes the number of Web documents in U in which both terms $t_i$ and $t_j$ occurs. The uncertainty function I with regards to co-occurrence threshold $\theta$ defined as:*

$$I_\theta(t_i) = \{t_j | f_U(t_i, t_j) \geqslant \theta\} \cup \{t_i\} . \tag{7}$$

Clearly, the above function satisfies conditions of being reflexive: $t_i \in I_\theta(t_j)$ and symmetric: $t_j \in I_\theta(t_i) \Longleftrightarrow t_i \in I_\theta(t_j)$ for any $t_i, t_j \in T$. Thus, $I_\theta(t_i)$ is the tolerance class of term $t_i$. Tolerance class of terms is generated to capture conceptually related terms into classes. The degree of correlation of terms in tolerance classes can be controlled by varying the threshold $\theta$.

In tolerance space of term, an expanded representation of Web document can be acquired by representing Web document as set of tolerance classes of terms it contains. This can be achieved by simply representing Web document with its upper approximation, e.g., the Web document $d_i \in U$ is represented by:

$$U_R(d_i) = \{t_i \in T | I_\theta(t_i) \cap d_i \neq \emptyset\} . \tag{8}$$

This approach to Web document representation expands Web document because it takes into consideration not only terms actually occurring Web document but also other related terms with similar meanings.

### 4.2 Missing Weights Complement

The best values of these missing weights are determined by incorporating two parts, i.e., weights of terms in term's tolerance class and corresponding term weight of the most similar vector, which has the same class label in tolerance class of the Web document. Here, the similarity measure between vectors is computed based on the distance:

$$Sim(d_x, d_y) = \frac{1}{1 + \sum_{k=1}^{M} |w_{ik} - w_{jk}|} . \tag{9}$$

After the tolerance classes for both term and Web document are generated, the essential information (i.e., the similarity between Web documents and the correlation between terms) is identified. To complement missing weights of terms in the Web document's vectors, we produce an improved TF*IDF weighting scheme based on the traditional TF*IDF. The improved weighting scheme is defined as below.

$$w_{ij} = \begin{cases} 1 + \log(f_{d_i}(t_j)) \times \log \frac{N}{f_D(t_j)} & \texttt{if } t_j \in d_i, \\ \alpha \times w_{kj} & \texttt{if } t_j \notin U_R(d_i), \\ \alpha \times w_{kj} + \beta \times (\min_{t_n \in d_i \wedge t_n \in I_\theta(t_j)} w_{in}) & \texttt{if } t_j \in U_R(d_i) \wedge t_j \notin d_i. \end{cases} \tag{10}$$

In above formula, $w_{kj}$ is the weight value of corresponding term of the most similarity vector with the same class label in Web document tolerance class; $\alpha$, $\beta \in [0, 1]$, they adjust the relative impact of relevant terms and Web documents respectively. Here, let parameters $\alpha$ and $\beta$ be 0.2.

To demonstrate the use of the improved TF*IDF weighting scheme, we detail an example as follows.

*Example*: Let Web document set be $U=\{d_1,d_2,d_3,d_4,d_5,d_6,d_7\}$, term set be $TS=\{t_1,t_2,t_3,t_4,t_5\}$, $B=TS$, the class label set be *class*$=\{C1,C2\}$, the frequency data is listed in Table 1.

**Table 1.** Sample Web document-term frequency array

| | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ | Class |
|---|---|---|---|---|---|---|
| $d_1$ | 0 | 0 | 6 | 8 | 0 | *C1* |
| $d_2$ | 1 | 3 | 12 | 0 | 9 | *C1* |
| $d_3$ | 2 | 3 | 0 | 12 | 14 | *C1* |
| $d_4$ | 0 | 0 | 5 | 4 | 2 | *C2* |
| $d_5$ | 10 | 9 | 4 | 0 | 3 | *C2* |
| $d_6$ | 12 | 14 | 2 | 2 | 0 | *C2* |
| $d_7$ | 11 | 12 | 0 | 4 | 2 | *C2* |

Let co-occurrence threshold $\theta$ equal 4, tolerance class of each term $t_i$ ($i$=1,2, ...,5) and upper approximations of the Web document $d_j$ ($j$=1,2,...,7) can be computed as below:

$I_\theta(t_1) = I_\theta(t_2) = \{t_1, t_2\}$; $I_\theta(t_3) = \{t_3\}$; $I_\theta(t_4) = I_\theta(t_4) = \{t_4, t_5\}$.

$U_B(d_1) = U_B(d_4) = \{t_3, t_4, t_5\}$; $U_B(d_2) = \{t_1, t_2, t_3\}$; $U_B(d_3) = U_B(d_7) = \{t_1, t_2, t_4, t_5\}$; $U_B(d_5) = U_B(d_6) = \{t_1, t_2, t_3, t_4, t_5\}$.

Note that the Web document $d_1$ and $d_4$ have different class label. We weigh them with traditional TF*IDF and improved TF*IDF respectively, result is listed in Table 2.

### 4.3 Web Document Classification

Firstly, terms are extracted from training set of Web documents, and then tolerance classes of Web documents and terms are computed. Secondly, the missing

**Table 2.** Weight of normal TF*IDF versus of improved TF*IDF

| Traditional TF*IDF | | | Improved TF*IDF | | |
|---|---|---|---|---|---|
| term | $d_1$ | $d_4$ | term | $d_1$ | $d_4$ |
| $t_1$ | 0 | 0 | $t_1$ | 0.067 | 0.115 |
| $t_2$ | 0 | 0 | $t_2$ | 0.072 | 0.118 |
| $t_3$ | 0.684 | 0.636 | $t_3$ | 0.684 | 0.636 |
| $t_4$ | 0.731 | 0.600 | $t_4$ | 0.731 | 0.600 |
| $t_5$ | 0 | 0.487 | $t_5$ | 0.091 | 0.487 |

weights of incomplete vectors are complemented. Thirdly, the classifier is constructed. Finally, the new Web document is classified into the category where the similarity measure is the highest among all other categories.

The similarities are computed between the new Web document and each category centroid, in which the similarity formula is defined as follows:

$$Dis(d_i, c_j) = \frac{\sum_{k=1}^{M} w_{ik} \times w_{jk}}{\sqrt{(\sum_{k=1}^{M} w_{ik}^2) \times (\sum_{k=1}^{M} w_{jk}^2)}} . \tag{11}$$

where $d_i$ is the new Web document, $c_j$ is the $j$th category centroid, $M$ is the term dimension.

## 5 Experimental Evaluation

### 5.1 Experimental Data Sets

To evaluate the proposed approach, we use two popular data collections in our experiments. The first one is the WebKB data set [1], which contains 8282 Web documents collected from computer science departments of various universities. The pages were manually classified into the following categories: student, faculty, staff, department, course, project, other (respectively abbreviated here as St, Fa, Sta, De, Co, Pr, Ot). In our experiments, each category is employed.The second collection is the Reuters-21578 [2], which has 21578 documents collected from the Reuters newswire. Of the 135 categories, only the most populous eight categories are used, i.e, acq, corn, crude, earn, grain, interest, money and trade (respectively abbreviated here as Ac, Co, Cr, Ea, Gr, In, Mo, Tr). The construction of each data set for our experiments is done as follows: Firstly, we randomly select 10% of the Web documents from the each category, and put them into test set to evaluate the performance of classifier. Then, the rest are used to create training sets. We extract and select the 100 most frequently occurred keywords from each category. For WebKB data set and Reuters-21578, the total numbers of all distinct keywords are 463 and 689 respectively.

[1] http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/
[2] http://www.research.att.com/ lewis/reuters21578.html

### 5.2 Performance Measures

To analyze the performance of classification, we adopt the popular F1 measure. F1 measure is combination of *recall* ($re$) and *precision* ($pr$), F1=$2.re.pr/(re+pr)$. *Precision* means the rate of documents classified correctly among the result of classifier and *recall* signifies the rate of correct classified documents among them to be classified correctly. The F1 measure which is the harmonic mean of precision and recall is used in this study since it takes into account effects of both quantities.

### 5.3 Experimental Results and Discussion

The results on WebKB data set are summarized in Table 3. Our approach yields a higher performance compared to the normal VSM for all categories. For example, in student category, our approach yields the F1 values of 75.6%, whereas the normal VSM yields the F1 values of 67.1%.

**Table 3.** Comparison of classification performance on WebKB

| | St | Fa | Sta | De | Co | Pr | Ot | Avg |
|---|---|---|---|---|---|---|---|---|
| VSM | 0.671 | 0.613 | 0.437 | 0.468 | 0.635 | 0.554 | 0.725 | 0.586 |
| RS | 0.756 | 0.734 | 0.633 | 0.630 | 0.691 | 0.712 | 0.787 | 0.710 |

**Table 4.** Comparison of classification performance on Reuters-21578

| | Ac | Co | Cr | Ea | Gr | In | Mo | Tr | Avg |
|---|---|---|---|---|---|---|---|---|---|
| VSM | 0.710 | 0.575 | 0.644 | 0.723 | 0.681 | 0.637 | 0.625 | 0.612 | 0.651 |
| RS | 0.736 | 0.673 | 0.727 | 0.780 | 0.769 | 0.740 | 0.768 | 0.694 | 0.736 |

In Table 3, *avg* shows summarized result which is calculated by averaging the F1 values over all categories. Our approach yields higher average classification performance of 12.4% over the normal VSM.We perform the same experiments on the Reuters-21578. The results are shown in Table 4, in which *avg* also shows summarized result. Our approach yields higher average classification performance of 8.5% over the normal VSM for Reuters-21578.

## 6 Conclusion

In this paper, a novel approach to Web document representation and classification based on rough set is proposed. For traditional way of Web document representation in the VSM, zero-valued similarity between vectors would decrease classificatory quality. Instead of assigning zero to the weights of those terms are absent in a Web page, these weights are considered missing information. Rough set for incomplete information is applied to discover valuable information, i.e., indiscernibility between Web documents and correlation between terms. Then,

the information is used for expanding representation of Web document to avoid zero-valued similarity. To validate the proposed approach, we compared our approach with the VSM. The experimental results show that the proposed approach yields a considerable improvement of classification performance.

**Acknowledgments.** This work was supported by the National Natural Science Foundation of China (No.60475019) and the Ph.D. programs Foundation of Ministry of Education of China (No.20060247039).

## References

1. Michelangelo Ceci, Donato Malerba: Hierarchical Classification of HTML Documents with WebClassII. F. Sebastiani (Ed.): ECIR 2003, LNCS 2633, pages 57-72, 2003.
2. Lawrence Kai Shih, David R. Karger: Using URLs and Table Layout for Web Classification Tasks. WWW2004, pages 193-202, 2004.
3. Sebastiani,F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34(1):1-47, March 2002.
4. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic, Dordrecht (1991)
5. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, pages 245-253, 1996.
6. Kryszkiewicz, M.: Rough set approach to incomplete information system. Information Sciences, 112:39-49, 1998.
7. Tu Bao Ho, Ngoc Binh Nguyen: Nonhierarchical Document Clustering based on A Tolerance Tough Set Model. International Journal of Intelligent Systems, Vol. 17, pages 199-212, 2002.
8. Chi Lang Ngo, Hung Son Nguyen: A Tolerance Rough Set Approach to Clustering Web Search Results. In: J.-F. Boulicaut et al. (eds.): PKDD 2004. Springer-Verlag, Berlin Heidelberg, pages 515-517, 2004.
9. D.Q. Miao, L.S. Hou: A comparison of rough set methods and representative inductive learning algorithms, Fundamenta Informaticae, v 59, n 2-3, pages 203-219, 2004.
10. Y.Y.Yao, C.-J.Liau, N.Zhong: Granular computing based on rough sets, quotient space theory, and belief functions. Proceedings of ISMIS03, pages 152-159, 2003.

# Transformation of Suffix Arrays into Suffix Trees on the MPI Environment

Inbok Lee[1], Costas S. Iliopoulos[2], and Syng-Yup Ohn[1]

[1] School of Electronics, Telecommunication, and Computer Engineering,
Hankuk Aviation University
{syohn,inboklee}@hau.ac.kr
[2] King's College London, Department of Computer Science
London WC2R 2LS, UK
csi@dcs.kcl.ac.uk

**Abstract.** Suffix trees and suffix arrays are two well-known index data structures for strings. It is known that the latter can be easily transformed into the former: Iliopoulos and Rytter [5] showed two simple transformation algorithms on the CREW PRAM model. However, the PRAM model is a theoretical one and we need a practical parallel model. The Message Passing Interface (MPI) is a standard widely used on both massively parallel machines and on clusters.

In this paper, we show how to implement the algorithms of Iliopoulos and Rytter on the MPI environment. Our contribution includes the modification of algorithms due to the lack of shared memory, small number of processors, communication costs between processors.

## 1 Introduction

Index data structures play an important role in various applications related to text processing. Suffix trees [14,16] and suffix arrays [8,10,11] are an example of index data structures and they are widely used in theoretical and practical problems. A lot of algorithms were developed using them to solve problems which arise from various applications including text processing, data compression, and Bioinformatics [4].

Suffix arrays are replacing suffix trees in the text processing due to the large memory requirement of suffix trees. There are some works on enhancing the power of suffix arrays similar to that of suffix trees [1,2,6,7]. However, suffix trees are the basic index data structure for solving problems in text processing and in some cases we need the suffix tree because no algorithm with suffix arrays is available. Therefore we need the transformation between suffix trees and suffix arrays. Converting suffix trees into suffix arrays is straightforward: we have only to traverse the suffix tree from left to right. But converting suffix arrays into suffix trees is not straightforward. Iliopoulos and Rytter [5] proposed two algorithms for this problem.

The merits of parallel index data structures are as follows.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 248–255, 2007.

– As the size of text data grows bigger, it is better to distribute the data into several locations than to use one huge storage.
– Parallel index data structures can reduce query processing time when there are a lot of queries.

Several algorithms were proposed for building suffix arrays in the parallel environment [3,12,13]. So once we have a parallel transformation algorithm for converting suffix arrays into suffix trees, we can build parallel suffix trees. Iliopoulos and Rytter's algorithms [5] are motivated by these observations and their algorithms are based on the Parallel Random Access Machine (PRAM). Although the PRAM is the basic model for parallel computation, it is a theoretical one and suffers from the followings.

– The PRAM model assumes that all the data is stored in shared memory which can be accessed by processors simultaneously. In real world it is hard, or impossible to build such an architecture.
– Also, it does not consider the communication cost between processors. Therefore, theoretically-efficient algorithms on the PRAM can work poorly in practice, due to the communication overhead.
– We cannot use as many processors as we want. The number of processors is restricted.

Therefore we need a realistic model of parallel computation. The Message Passing Interface (MPI) [17] is a *de facto* standard for writing parallel programs on the distributed memory system. It has several merits over the traditional PRAM model:

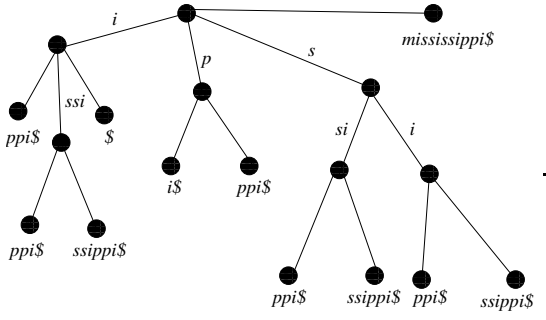
– MPI is language-independent. APIs for Fortran, C, C++, and Python are freely available.
– Although the MPI supports only low-level calls for parallel computation, it is easy to write parallel programs on the MPI. For example, a simple parallel program can be written using only five MPI function calls.
– There are a few implementations of the MPI standard specification which can run a single computer as well as clusters of processors. For example, MPICH2 [18] can simulate parallel computation on a single machine.

In this paper, we show how to transform suffix arrays to suffix trees on MPI. Especially we will focus on the differences between PRAM and MPI and how to handle them.

## 2 Preliminaries

Let $T$ be a string over an alphabet $\Sigma$ and $n = |T|$, the length of the string. $T[i]$ denotes the $i$-th character of $T$. $T[i..j]$ is the substring $T[i]T[i+1]\cdots T[j]$ of $T$. For any integer $1 \leq j \leq n$, we say that $T[1..j]$ is a *prefix* of $T$. Also for any integer $1 \leq j \leq n$, we say that $T[j..n]$ is a *suffix* of $T$.

The *suffix tree* of a text $T$ is a tree which stores all the suffixes of the string. A path from the root to one leaf represents a suffix. If two suffix share a common




| | $suf$ | $lcp$ | |
|---|---|---|---|
| 1 | 11 | - | i |
| 2 | 8 | 1 | ippi |
| 3 | 5 | 1 | issippi |
| 4 | 2 | 4 | ississippi |
| 5 | 1 | 0 | mississippi |
| 6 | 10 | 0 | pi |
| 7 | 9 | 1 | ppi |
| 8 | 7 | 0 | sippi |
| 9 | 4 | 2 | sissippi |
| 10 | 6 | 1 | ssippi |
| 11 | 3 | 3 | ssissippi |

**Fig. 1.** The suffix tree and the suffix array for `mississippi`

prefix, we create an internal node. For example, in Figure 1, two suffixes $pi$ and $ppi$ share a common prefix $p$. We create an internal node and the edge between it and the root represents the common prefix $p$. We store $i$ and $pi$ at two leaves. The time and space complexity of building the suffix tree for a string $T[1..n]$ is $O(n)$, but the constant hidden in $O(n)$ is quite large. More details on the suffix tree can be found in [14,16].

The *suffix array* of a text $T$ is a well-known indexed structure for the string. Basically it is the sorted array $suf[1..n]$ of all the suffixes of $T$ in the lexicographical order. It means that $suf[k] = i$ if and only if $T[i..n]$ is the $k$-th suffix of $T$. The suffix array can be built in $O(n)$ time and space [8,10,11]. We also define the auxiliary *LCP array* as an array of the length of the longest common prefix between each substring in the suffix array and its predecessor: $lcp[i+1]$ is the length of the common prefix between $T[suf[i]..n]$ and $T[suf[i+1]..n]$. Given $suf$ array, $lcp$ array can be calculated in $O(n)$ time [9].

Figure 1 shows an example of the suffix tree and the suffix array for a string `mississippi`. Note that \$ character is added to each suffix to denote the end of suffixes in the suffix tree. Otherwise it would be hard to represent the suffix $i$. As we said before, suffix trees and suffix arrays are closely related since they have the same information in different structures.

We assume that we have $p$ processors which have the same computing power. On the MPI environment, all the communication between processors are message-passing: if processor $i$ wants to access processor $j$'s memory, it has to send a request to $j$ explicitly and wait for the answer from $j$. Since the communication is over TCP/IP, it takes time and is not free. Additionally, we assume that $n \gg p$. Usually the text is quite large and the number of processors is small, so this assumption makes sense. Our aim is as follows: (a) allocate an equal amount of

works to each processor, and (b) reduce the communication between processors as small as possible.

## 3 Algorithm 1

Algorithm 1 is a very simple recursive algorithm. It is based on the divide-and-conquer approach. Roughly, we first begin with two neighboring suffixes, $T[suf[i]..n]$ and $T[suf[i+1]..n]$. $lcp[suf[i+1]]$ tells the common prefix which two suffixes share. We create an internal node and two leaves as in Figure 3 (b). If we have only one suffix, then we can create a root and a leaf. Next we merge two neighboring trees into one. We divide the suffix array so that two neighboring trees share one path in common. More precisely, let us assume that we have two trees $T_1$ and $T_2$. The suffix obtained by the path from the rightmost leaf of $T_1$ to the root of $T_1$ is the same as the one obtained by the path from the leftmost leaf of $T_2$ to the root of $T_2$. We follow these two paths and merge them into one, as in Figure 3 (c). We obtain one bigger tree and do the same again. Finally we have one suffix array. The time complexity of Algorithm 1 is $O(\log^2 n)$ and the total work is $O(n \log n)$ [5].

```
Algorithm1(i, j)
if (j − i) ≤ 2 then
    Compute the partial suffix tree directly;
else
    parallel do
        T1 =Algorithm1(i, (i + j)/2);
        T2 =Algorithm1((i + j)/2, j);
        Create T by merging T1 and T2, by joining the rightmost path of T1
         and the leftmost path of T2;
    od
return T;
```

**Fig. 2.** Algorithm 1: A simple recursive algorithm

The above algorithm does not work directly on the MPI environment. First of all, there is no notion of shared memory in the MPI standard. Therefore we need to decide how to store the suffix array among the processors. We divide the suffix array into $p$ blocks (for simplicity, we assume that $n = pk$ for some integer $k$). The first block consists of $T[suf[1]..n], T[suf[2]..n], \cdots, T[suf[n/p]..n]$. The $i$-th block ($2 \leq i \leq p$) block consists of $T[suf[in/p]..n], T[suf[in/p + 1]..n], \cdots, T[suf[(i+1)n/p]..n]$ so that two neighboring blocks share one suffix. We allocate the $i$-th block to the $i$-th processor. A naïve allocation algorithm will need $O(p)$ rounds to distribute the suffix arrays to the processors but we can reduce it to $O(\log p)$ rounds: at first processor 1 sends $n/2$ elements to processor $p/2$, then they sends $n/4$ to processor 1, $p/4$, $p/2$, $3p/4$ and so on. Note that $p - 1$ elements in the suffix array are duplicated in two processors. However, as we assume that $p \ll n$, this overhead is not great.

For each processor, it first performs the same algorithm we did on the PRAM model. Then the $i$-th processor has a partial suffix tree for suffixes $T[suf[in/p]..n], T[suf[in/p+1]..n], \cdots, T[suf[(i+1)n/p]..n]$. Finally, we merge these partial suffix trees into one suffix tree. The idea is the same. The first processor sends the information on the rightmost path of the first partial tree to the second processor. The last processor sends the information on the leftmost path of the last partial tree to the $(p-1)$-th processor. The $i$-th processor $(2 \leq i \leq p-1)$ sends the information on the leftmost path of the $i$-th tree to the $(i-1)$-th and the rightmost path of it to $(i+1)$-th processors. Since a path from the root to one of leaves may include two or more processors, the pointer of an internal node can be represented by $(p, addr)$: the next node is stored at processor $p$'s memory, at the address of $addr$.

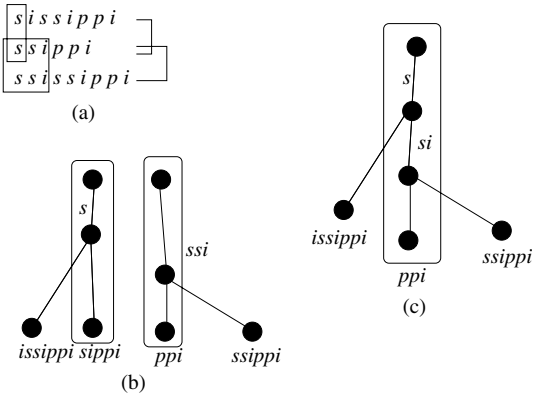



**Fig. 3.** Transforming the suffix array of $T =$`mississippi` into the suffix tree

Now we analyze our algorithm. Each processor performs the same algorithm we did with the PRAM algorithm, but the data size is $n/p$. Each processor sends and receives at most $2n/p$ nodes since we make at most one internal node each time we merge two suffixes and each processor has $n/p$ suffixes. Merging tree requires $O(n/p)$ time. Therefore, the total time complexity is $O(\log^2(n/p)+n/p)$. The total communication cost is $p \times O(n/p) = O(n)$.

Note that our resulting tree has additional $p-1$ duplicated paths. We should take care of this property. For example, when we are finding a pattern from the text, we need to traverse every possible path if there are two or more paths.

## 4 Algorithm 2

Algorithm 2 is a non-recursive one. For simplicity we assume that $n = 2^k p$ for some integer $k$. Due to space restriction, we briefly sketch Iliopulos and Rytter's algorithm. More details can be found in [5].

We use a few arrays for the transformation. First we set $lcp[1] = 0$. For $1 \leq i \leq n$, $L[i] = \max\{j < i : lcp[j] < lcp[i]\}$ and $R[i] = \min\{j > i : lcp[j] < lcp[i]\}$.

| index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *lcp* | 0 | 1 | 1 | 4 | 0 | 0 | 1 | 0 | 2 | 1 | 3 |
| *SN* | - | 1 | 1 | 3 | - | - | 6 | - | 8 | 9 | 10 |
| *Leftmost* | 1 | 2 | 2 | 4 | 1 | 5 | 7 | 8 | 9 | 10 | 11 |

**Fig. 4.** An example of $SN$ and *Leftmost* array for the suffix arrays in Figure 1

And we define the *nearest smaller neighbor* $SN$. If $lcp[L[i]] \geq lcp[R[i]]$, then $SN[i] = L[i]$. Otherwise, $SN[i] = R[i]$. We also define $Leftmost[i] = \min\{i \leq j \leq i : lcp[j] = lcp[i]$ and $lcp[k] \geq lcp[i]$ for each $j \leq k \leq i\}$. Put it another way, $SN[i]$ is the nearest location $j$ where $lcp[j] < lcp[i]$ and $Leftmost[i]$ is the leftmost location $j$ where $lcp[j] = lcp[i]$.

We explain the key idea of Algorithm 2. If $Leftmost[i] = j$ and $i \neq j$, then $T[suf[j]..n], T[suf[j]+1..n], \cdots, T[i..n]$ shares a common prefix of length $lcp[i]$. We can create one internal node at this point, but there may be another internal node in the path from this internal node to the root. We use $SN$ array to find this internal node. Since $lcp[SN[j]$ is smaller than $lcp[j]$ by definition, there are other suffixes sharing a shorter common prefix.

To represent the suffix tree, we use two different types of nodes. Each suffix $suf[i]$ makes a leaf node $leaf[i]$. Additionally we create an array $int[1..n]$ which stores an internal node $int[k]$ if $Leftmost[k] = k$.

We use two lemmas here without proof. Proofs can be found in [5].

**Lemma 1.** *If $int[s]$ is the father of an internal node $int[k]$ and $r = lcp[k]$, then the edge from $int[s]$ to $int[k]$ is $T[suf[k]+p..suf[k]+r-1]$ where $p = lcp[s]$. If $int[s]$ is the father of $leaf[k]$, then the edge represents $T[suf[k]+r..n]$.*

**Lemma 2.** *The father of $int[k] = int[j]$ where j=Leftmost[SN[k]]. The father of $leaf[t] = int[j]$ if $lcp[t] \leq lcp[t+1]$ or $t = n$. Otherwise the father of $leaf[t] = int[t+1]$.*

By Lemma 2 we can find the father of leaves and internal nodes. By Lemma 1 we can find the strings represented by edges. The time complexity is $O(logn)$ and it takes $O(n)$ space.

The implementation on the MPI environment is straightforward. First we divide the suffix array into $p$ blocks again, but now no suffix is stored in two blocks. The $i$-th block ($1 \leq i \leq p$) block consists of $T[suf[in/p]..n], T[suf[in/p+1]..n], \cdots, T[suf[(i+1)n/p]-1..n]$. Again we allocate the $i$-th block to the $i$-th processor. Then we first compute $SN$ and *Leftmost* arrays. They can be computed efficiently using the classical prefix computation algorithm. The remaining part is to apply Lemma 2 and 3. Processor $i$ tries to compute $leaf[in/p]$, $leaf[in/p+1], \cdots, leaf[(i+1)n/p-1]$ and $int[in/p], int[in/p+1], \cdots, int[(i+1)n/p-1]$, with strings represented by edges linking them. To do so, we need the values in $SN$ and *Leftmost* arrays. If they are stored in processor $i$'s memory then there is no problem. Processor $i$ can access them directly. But if they are stored in processor $j$'s memory, then we need to request them explicitly. First processor $i$ computes which processor to request, and the number of element

to ask that processor. Then processor $j$ send the requested data to processor $i$ back. Our algorithm is rather sequential here: the difference is that the data is distributed among processors.

Once each processor receives the necessary data, it creates internal nodes and edges. Since a processor can handle one leaf or internal node at each step, the time complexity is $O(n/p)$. The space complexity is the same, $O(n)$.

## 5 Implementation

We used MPICH2 [18] and C++ language to implement these algorithms. The suffix arrays were generated by Puglisi et al.'s excellent implementation [15]. We were able to find that these algorithms work correctly.

Since we were unable to find a cluster of processors, the experiment was done by a DELL Optiplex GX520 machine with one 3.0 GHz Pentium 4 processor and 1GB RAM, running Windows XP. As we mentioned before, MPICH2 provides the simulation of parallel computing environment. The same implementation can run on the PC cluster.

## 6 Conclusion

We showed how to transform suffix arrays into suffix trees on the MPI environment. Unlike the PRAM model, we were able to implement parallel algorithms on real machines.

## References

1. M. I. Abouelhoda, S. Kurtz, and E. Ohlebusch. Replacing suffix trees with enhanced suffix arrays. *Journal of Discrete Algorithms*, 2(1):53–86, 2004.
2. M. I. Abouelhoda, E. Ohlebusch, and S. Kurtz. The Enhanced Suffix Array and Its Applications to Genome Analysis. In *Proceedings of Second International Workshop on Algorithms in Bioinformatics (WABI '02)*, pages 449–463, 2002.
3. R. Dementiev, J. Karkkainen, J. Mehnert, and P. Sanders. Better External Memory Suffix Array Construction. *ACM Journal of Experimental Algorithmics*, to appear.
4. D. Gusfield. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, 1997.
5. C. S. Iliopoulos and W. Rytter. On parallel transformations of suffix arrays into suffix trees. In *Proceedings of the 15th Australasian Workshop on Combinatorial Algorithms (AWOCA04)*, 2004.
6. D. K. Kim, J. E. Jeon, and H. Park. An Efficient Index Data Structure with the Capabilities of Suffix Trees and Suffix Arrays for Alphabets of Non-negligible Size. In *Proceedings of 11th International Conference on String Processing and Information Retrieval (SPIRE 2004)*, pages 138–149, 2004.
7. D. K. Kim and H. Park. A New Compressed Suffix Tree Supporting Fast Search and Its Construction Algorithm Using Optimal Working Space. In *Proceedings of 16th Annual Symposium on Combinatorial Pattern Matching (CPM 2005)*, pages 33–44, 2005.

8. J. Kärkkäinen and P. Sanders. Simpler linear work suffix array construction. In *Proceedings of the 13th International Colloquium on Automata, Languages and Programming (ICALP 2003)*, pages 943–945, 2003
9. T. Kasai, G. Lee, H. Arimura, S. Arikawa, and K. Park. Linear-time longest-common-prefix computation in suffix arrays and its applications. In *Proceedings of the 12th Annual Symposium on Combinatorial Pattern Matching (CPM 2001)*, pages 181–192, 2001.
10. D. K. Kim, J. S. Sim, H. Park, and K. Park. Linear-time construction of suffix arrays. In *Proceedings of the 14th Annual Symposium on Combinatorial Pattern Matching (CPM 2003)*, pages 186–199, 2003
11. P. Ko and S. Aluru. Space-efficient linear time construction of suffix arrays. In *Proceedings of the 14th Annual Symposium on Combinatorial Pattern Matching (CPM 2003)*, pages 200–210, 2003
12. M. Marín and G. Navarro. Suffix Arrays in Parallel. In *Proceedings of 9th International Conference on Parallel and Distributed Computing (EuroPar 2003)*, pages 338–441, 2003.
13. M. Marín and G. Navarro. Distributed Query Processing using Suffix Arrays. In *Proceedings of the 10th International Symposium on String Processing and Information Retrieval (SPIRE 2003)*, pages 311–325, 2003.
14. E. M. McCreight. A Space-Economical Suffix Tree Construction Algorithm. *Journal of the ACM*, 23(2):262–272, 1976.
15. S. J. Puglisi, W. F. Smyth, and A. Turpin. A taxonomy of suffix array construction algorithms. *ACM Computing Surveys*, to appear.
16. E. Ukkonen. On-line Construction of Suffix Trees. *Algorithmica*, 14:249–260, 1995.
17. `http://www-unix.mcs.anl.gov/mpi/`
18. `http://www-unix.mcs.anl.gov/mpi/mpich2/`.
19. `http://datamining.anu.edu.au/~ole/pypar/`.

# Clustering High Dimensional Data Using SVM

Tsau Young Lin and Tam Ngo

Department of Computer Science, San José State University,
San Jose, CA 95192, USA
tylin@cs.sjsu.edu, tam.p.ngo@gmail.com

**Abstract.** The Web contains massive amount of documents to the point where it has become impossible to classify them manually. This project's goal is to find a new method for clustering documents that is as close to humans' classification as possible and at the same time to reduce the size of the documents. This project uses a combination of Latent Semantic Indexing (LSI) with Singular Value Decomposition (SVD) calculation and Support Vector Machine (SVM) classification. Using SVD, data is decomposed and truncated to reduce the data size. The reduced data will be clustered into different categories. Using SVM, clustered data from SVD calculation is used for training to allow new data to be classified based on SVM's prediction. The project's result show that the method of combining SVD and SVM is able to reduce data size and classifies documents reasonably compared to humans' classification.

**Keywords:** SVM, SVD, LSI, clustering, text classification, unsupervised.

## 1 Introduction

Ever since the World Wide Web has become popular, document clustering has become increasingly more important. With billions of documents on the Web, it is impossible to classify them by humans. The challenge is to find a way to organize this massive data in some meaningful structure. This project proposes a method that can cluster documents reasonably.

The project deals with clustering high dimensional data. The data used are processed documents organized in a text file that contains category labels and term frequency–inverse document frequency (tf–idf) values. Data sets used in the research are classified by humans and have been processed into tf-idf values. By using human-classified data set, we are able to compare our clustering method with humans' classification.

The first few sections of the paper discuss and analyze Support Vector Machine (SVM) and Singular Value Decomposition (SVD). This will allow the reader to understand how these methods are applied to the project. The last few sections discuss the algorithms used and the analysis of the results after applying methods from previous sections.

## 2 Support Vector Machine

Vladimir Vapnik and his colleagues first introduced SVM in 1963. Support Vector Machine (SVM) is a learning machine that uses supervised learning to perform data

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 256–262, 2007.

classification and regression [10]. In SVM, each line within the data set is given a label and SVM learns the data and puts the new data in the group/category that is closest to the learned data.

Beside SVM, there are many other methods for text classification such as Bayes and k-Nearest Neighbor. Based on many research papers [7], SVM outperforms many, if not all, popular methods for text classification. The studies also show that SVM is effective, accurate, and can work well with small amount of training data.

### 2.1 Understanding SVM

The concept of SVM is quite intriguing once the reader understands the math behind it. The idea for SVM is to find a boundary (known as a hyperplane) or boundaries that separate clusters of data. SVM does this by taking a set of points and separating those points using mathematical formulas. The process of SVM starts with data that are in an input space and can or cannot be separated with a linear hyperplane. To separate the data linearly, points are map to a feature space using a kernel method. Once the data in the feature space are separated, the linear hyperplane gets map back to the input space and is shown as a curvy non-linear hyperplane.

The SVM's algorithm first learns from data that has already been classified, which is represented in numerical labels (e.g. 1, 2, 3, etc.) with each number representing a category. SVM then groups the data with the same label in each convex hull. From there, it determines where the hyperplane(s) is by calculating the closest points between the convex hulls [1]. Once SVM determines where the hyperplane(s) is, it creates a model file that is used to classify new data. For example, any new data that lies on the side of the positive plane is classified with a positive label and any new data that lies on the side of the negative plane is classified with a negative label.

## 3 Data Preparation Using SVD

In order to separate the data, SVM requires training data to be in categories. This project's aim is to cluster data, however, as mentioned above, SVM is a supervised learning machine so it does not cluster data. From our research, we found that there is not a proven working method for an unsupervised SVM. Thus, we use a different approach; clustering data using Singular Value Decomposition (SVD) and then using SVM, we can predict the category/label of the new data.

### 3.1 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a method that separates a matrix into three parts; left eigenvectors, singular values, and right eigenvectors [5]. It can be used to decompose data such as images and text. Since SVM requires supervised learning, SVD is chosen to cluster the data and give the data its label.

Given a matrix **A**, we can factor it into three parts: **U**, **S**, and $\mathbf{V}^{\mathbf{T}}$ (**V** transpose). If the matrices, **U**, **S**, and $\mathbf{V}^{\mathbf{T}}$ are multiplied together, the original matrix **A** is reconstructed. One of the nice properties of SVD is that after the matrix is decomposed its dimension can be reduced by choosing to keep only the largest singular values in the **S** matrix. For example, to find rank 2 approximation, the first 2

columns of **U** and **V**, and the first 2 columns and rows of **S** are kept. The rest of the columns and rows can be dropped. Using the truncated **V** matrix, **V'**, documents can be clustered by calculating cosine similarities between each document. This will return the distance between the documents.

$$\text{sim}(D_s', D') = (D_s' \bullet D') / (|D_s'| |D'|)$$

Here $D_s'$, represents the selected row within the truncated matrix **V'** and D' represents another row, also in truncated matrix **V'**, in which $D_s'$ is calculated with. We calculate cosine similarities between $D_s'$ and all other rows. From the result, the highest value is most similar/closest to the selected document. Doing this procedure for each document, we can pair up documents that are most similar to each other to form small clusters of documents.

## 4 The Project

This project will use the same methods that were discussed to cluster a good-sized data set. The purpose is to see how well the data clusters using SVD and running the clustered data (data with labels) using SVM to predict labels for new data. One might wonder why use SVM when SVD can do the same job. Based on the algorithms, SVM is faster and has the ability to separate the data nicely. With SVM, new data is classified without having to process cosine similarities.

In order to use SVM for this project, the documents need to be represented in numerical values. A way to do this is to calculate the term frequency–inverse document frequency (tf–idf) values.

$$\text{tf} = \frac{n_i}{\sum_k n_k}$$

$$\text{tfidf} = \text{tf} \cdot \log\left(\frac{|D|}{|(d_j \supset t_i)|}\right)$$

The above equations show one way of calculating tf-idf. Tf stands for term frequency with $n_i$ as the number of occurrences of a term in a document and $\sum_k n_k$ as the number of occurrences of all terms in the same document. The tf equation is then multiplied by the inverse document frequency (idf) equation. Idf is the log of |D|, which is the total number of all considered documents, divided by $| d_j \supset t_i |$, which is the number of documents that a term appears [11].

Fortunately, there are many data sets in tf-idf format that have already been human-classified for the public to use to compare their results [4], [9]. Therefore, it is not necessary to compute the tf-idf values for this project.

Once a matrix of tf-idf values are obtained, it needs to be decomposed using SVD. Both data sets that are used for SVM training and predicting need to be truncated with SVD by the same ranking approximation value. There are two ways to do this. One way is perform SVD calculation and then truncate the new data (the data that we want to find the labels for using SVM) with the same ranking value as the training data.

Another way is to multiply the new data with the **U'** and **S'**$^{-1}$ matrix of the training data as shown below. This method is used for the project.

**SVM Prediction Data = NewDataMatrix * trainingU' * trainingS'**$^{-1}$

### 4.1 Implementation

The implementation of this project requires two libraries: JAMA [9] and LIBSVM [2]. Given a matrix, JAMA can calculate SVD, which gives the matrix **U**, **S**, and **V**. The application truncates **U**, **S**, and **V** based on a user-inputted value. This value is the ranking approximation. With the truncated matrix, **V'**, the program selects the first row of the matrix and does cosine similarities calculation, which was discussed above, with the second row , then the third row, then the forth row, and so forth, until it reaches the last row. The row that returns the highest cosine similarities value is clustered with the first row. Then the application selects the next row and does cosine similarities calculation with all other rows. Doing this for each row will return small clusters of 2-4 documents. These are naturally formed clusters.

Since having many small clusters is not practical, the application allows user to input the desired number of clusters. With the given number of clusters, the program does cosine similarities calculation on each group of clusters and combining clusters that are closest together until it reaches to the user-inputted cluster value. In the final step, each document will be assigned to a category label based on which cluster it belongs. Using this information, we can compare our result to what humans' have classified. We then use this result as training data for SVM. The library, LIBSVM, will use the training data to separate the documents with hyperplane(s). From here, SVM can predict the labels for new data by using the given training data. By using SVD to cluster the documents, we are able to achieve clustering documents without the need of humans' classification. In addition, SVD can reduced the document size as well as document noise.

### 4.2 Using Larger Data Set

The previous sections give background research on the approach used to cluster a data set. Now we would like to use a good-sized data set to test our method. The data set used is Reuters-21578, which is the most widely used data set for text categorization [9]. Reuters-21578 is a collection of newswire articles that have been human-classified by Carnegie Group, Inc. and Reuters, Ltd. The data used for this project is part of the already processed Reuters-21578 by Joachims [7]. Due to the expensive calculation of SVD, the data is further separated into 200 lines (rows) and 9928 terms (columns) per data set. In Table 1 and 2, "SVD Cluster Accuracy" will measure how close our SVD clustering method compares to humans classification and "SVM Prediction Accuracy" will measure how accurate it is to use the SVD clustered data for training and then use it to predict labels on new data. Please note that a different set of Reuters-21578 that is 200 lines by 9928 terms is used as new data for SVM prediction.

### 4.3 Result Analysis

**Table 1.** Results: Clustering with SVD vs. Humans Classification, First Data Set

| | First Data Set from Reuters-21578 (200 x 9928) | | |
|---|---|---|---|
| | # of Naturally Formed Cluster using SVD | SVD Cluster Accuracy | SVM Prediction Accuracy |
| Rank 002 | 80 | 75.0% | 65.0% |
| Rank 005 | 66 | 81.5% | 82.0% |
| Rank 010 | 66 | 60.5% | 54.0% |
| Rank 015 | 64 | 52.0% | 51.5% |
| Rank 020 | 67 | 38.0% | 46.5% |
| Rank 030 | 72 | 60.0% | 54.0% |
| Rank 040 | 72 | 62.5% | 58.5% |
| Rank 050 | 73 | 54.5% | 51.5% |
| Rank 100 | 75 | 45.5% | 58.5% |

**Table 2.** Results: Clustering with SVD vs. Humans Classification, Second Data Set

| | Second Data Set from Reuters-21578 (200 x 9928) | | |
|---|---|---|---|
| | # of Naturally Formed Cluster using SVD | SVD Cluster Accuracy | SVM Prediction Accuracy |
| Rank 002 | 76 | 67.0% | 84.5% |
| Rank 005 | 73 | 67.0% | 84.5% |
| Rank 010 | 64 | 70.0% | 85.5% |
| Rank 015 | 64 | 63.0% | 81.0% |
| Rank 020 | 67 | 59.5% | 50.0% |
| Rank 030 | 69 | 68.5% | 83.5% |
| Rank 040 | 69 | 59.0% | 79.0% |
| Rank 050 | 76 | 44.5% | 25.5% |
| Rank 100 | 71 | 52.0% | 47.0% |

Based on the results, the highest percentage accuracy for SVD clustering is 81.5% for rank 5 approximation (Table 1). This accuracy percentage is reasonably good. Based on observation, lower ranking approximation values do better than the higher approximation values. This supports many researchers' claim that truncated SVD gives better results. As for SVM prediction, the results are not surprising, since SVM can only predict as well as what is given it to train. Therefore, its prediction accuracy is about the same as SVD's.

There are several reasons why the highest accuracy is 81.5% and not higher. When calculating SVD and using cosine similarities calculation to cluster, the documents form small clusters naturally. Having too many small clusters is not practical; therefore, a new algorithm is needed on top of the clustering algorithm to reduce the

clusters to a desirable number. What the algorithm does is for each small cluster, it calculates the average of the vector documents within that cluster and compares it, using cosine similarities, to another cluster. The cluster that yields the highest value will be combined with the selected cluster. As the reader can see, reducing the number of clusters from 64-80 to just two clusters will reduce the accuracy. Because the data used to test in Table 1 and 2 are classified in only 2 categories, the algorithm also needs to reduce the clusters to 2 clusters so that it is possible to compare the results. Also, humans' classification is more subjective than a program so the methods used to classify are different from each other.

# 5 Conclusion

The project's goal is to find a method that can cluster high dimensional data. After our research, we choose to use a combination of SVD and SVM. In section 2, the concept of SVM is discussed. With the use of kernel methods, SVM can classify data in high dimensional space. Although SVM is an excellent method for data classification, it cannot cluster the data. Because of this, the project goes further into researching a method that can cluster and reduce the data. In section 3, SVD is used to accomplish this task. In section 4, SVD is used with SVM on data sets. The method is then compared with data that are classified by humans. From the experiment and analysis, the results show that the method proposed is able to cluster documents reasonably. However, there are plenty of rooms to improve this method. Overall, the result of the project is satisfactory.

## 5.1 Future Work

As mentioned previously, there are still a lot more work that could be done to improve this project. One way is to create a method that stores the data sets into a database. This way accessing the data each time will be much faster. In addition, a database can store a lot more data. Another way is when calculating the distance between vectors using cosine similarities, parallel processing can be used to speed up the time. Also, the libraries, LIBSVM and JAMA, used in this project is excellent for small size data set, however, these libraries need modification to accommodate larger data processing. For example, JAMA cannot process matrices that have m rows less than n columns ($m < n$) and it uses a double matrix array, which limits the size one can store. Lastly, we can look for more efficient kernels to use on SVM.

# References

1. Bennett, K.P., Campbell, C.: Support Vector Machines: Hype or Hellelujah?, ACM SIGKDD Explorations 2(2) (2000) 1-13.
2. Chang, C., Lin, C.: LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm, November 29 (2006)
3. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines, Cambridge University Press (2000)

4. Fan, R.: LIBSVM Data: Classification, Regression, and Multi-label. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/ , November 28 (2006)
5. Garcia, E.: SVD and LSI Tutorial 4: Latent Semantic Indexing (LSI) How-to Calculations. http://www.miislita.com/information-retrieval-tutorial/svd-lsi-tutorial-4-lsi-how-to-calculations.html, November 28 (2006)
6. Hicklin, J., Moler, C., Webb, P.: JAMA : A Java Matrix Package. http://math.nist.gov/javanumerics/jama/, November 28 (2006)
7. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. http://www.cs.cornell.edu/People/tj/publications/joachims_98a.pdf November 28 (1998)
8. Joachims, T.: Support Vector Machines. Available: http://svmlight.joachims.org/ November 28 (2006)
9. Reuters-21578 Text Categorization Test Collection. http://www.daviddlewis.com/resources/testcollections/reuters21578/, November 28 (2006)
10. Support vector machine. http://en.wikipedia.org/wiki/Support_vector_machine, December 28 (2005)
11. Wikipedia. http://en.wikipedia.org/wiki/Tf, December 8 ( 2005)
12. Vapnik, V. N., The Nature of Statistical Learning Theory, Springer-Verlag (2000).

# Constructing Associative Classifier Using Rough Sets and Evidence Theory

Yuan-Chun Jiang, Ye-Zheng Liu, Xiao Liu, and Jie-Kui Zhang

Institute of Electronic Commerce, Hefei University of Technology,
230009 Hefei, China
{yuanchunjiang,liuyezheng,jychmail,zhangjiekui}@gmail.com

**Abstract.** Constructing accurate classifier based on association rule is an important and challenging task in data mining. In this paper, a novel combination strategy based on rough sets (RST) and evidence theory (DST) for associative classification (RSETAC) is proposed. In RSETAC, rules are regarded as classification experts, after the calculation of the basic probability assignments (bpa) according to rule confidences and evidence weights employing RST, Yang's rule of combination is employed to combine the distinct evidences to realize an aggregate classification. A numerical example is shown to highlight the procedure of the proposed method. The comparison with popular methods like CBA, C4.5, RIPPER and MCAR indicates that RSETAC is a competitive method for classification based on association rule.

**Keywords:** Combination strategy, Associative classification, Evidence theory, Rough sets, Evidence weight.

## 1 Introduction

Associative classification (AC) is one of the most important tasks in data mining and knowledge discovery. Previous studies have shown that AC is a powerful tool to handle unstructured data and often has more accurate classification result than conventional techniques [1,2].

The first algorithm using association rule for classification is named CBA [2], which applies the popular Apriori algorithm to extract association rules with their consequents limited to class labels. In the procedure of AC, rule discovery and classification are two crucial tasks. In the last few years, many investigations have been contributed to discover rules from data, and proposed many methods such as CBA, C4.5 [3] and MCAR [4]. However, they may suffer the following weakness.

On one hand, the majority of research work is to solve single label classification problem [5]. In this paper, we named the rules with the same condition but different classes conflicting rules. Actually, the ignored rules might play an important role in some cases and make their contributions to improve classification accuracy.

The other intractable problem is that they can not easily identify the most effective rule during the process of classifying new cases [6]. In this paper, we named these methods selection strategy of rules. The problem of selection strategy lies in the difficulty of establishing the optimal rule. To solve this problem, several approaches

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 263–271, 2007.

have been proposed, such as CMAR [1] and CAEP [6]. They perform classification based on a set of rules for predicting new cases. Unfortunately, those methods can just only solve the single label classification problem.

The limitation of the selection strategy and the creation of single class rules is that some valuable information is ignored, thus leading to the knowledge hidden in data or the information mined through rule discovery methods can not be used effectively.

In this paper, a novel combination strategy based on rough sets and evidence theory (RSETAC) is proposed for multi-class classification based on multiple association rules. RSETAC makes use of all the conflicting rules and determines the class label of a new case based on a set of matching rules. First, rules are transformed to be classification experts, and then evidences given by these experts are calculated based on rule confidences. Second, evidence weights are calculated based on rule supports and attribute significances from the viewpoint of RST [7]. Finally, distinct evidences are combined employing Yang's rule of combination in DST [8,9] to give an aggregated classification.

The remainder of this paper is organized as follows. The next section gives details of the RSETAC algorithm. A numerical example to highlight the procedure of RSETAC is included in Section 3. Conclusions are given in Section 4.

## 2 RSETAC Algorithm

### 2.1 Presentation of the Problem

Suppose a data set $I=(U, A\cup\{C\})$, where $U$ is a finite set of objects, $A=\{A_1,A_2,\cdots,A_n\}$ is a finite set of attributes, $C$ is the class attribute, and $VC=\{c_1,c_2,\cdots,c_{|C|}\}$ is the finite set of class labels. RSETAC treats attributes in $A$ as discrete ones because many discretization methods can be used to map continuous values to categorical ones. For any objects in $I$, there exists a class label $c_i\in VC$ associated with it. The aim of AC is to associate a new case with a class label. The AC is characterized by the following components [4,10].

(1) A condition $P_i$ is defined as a set of attributes together with specific attribute values for each attribute, denoted $P_i=(<A_{i1},a_{i1}>,<A_{i2},a_{i2}>,\cdots,<A_{im},a_{im}>)$.

(2) A rule $r_i$ maps a condition to a specific class label, denoted $P_i\rightarrow c_i$.

(3) The actual occurrence $Count(P_i)$ of a condition $P_i$ is the number of rows in $I$ that match $P_i$. The support count $SupCount(r_i)$ of rule $r_i$ is the number of rows in $I$ that match $P_i$ and have class label $c_i$.

(4) The rule support and confidence of $r_i$ is defined as

$$sup(r_i)=SupCount(r_i)/|I|, \quad conf(r_i)=SupCount(r_i)/Count(P_i), \tag{1}$$

where $|I|$ is the number of objects in $I$.

As mentioned in Section 1, to make a credible and accurate classification, we should discover more information from data, retain more useful rules in the procedure of rule discovery and handle them properly. Furthermore, a detailed and aggregated analysis based on multiple rules may lead to more accurate classifications.

RSETAC is a novel method to treat such problems. RSETAC retains conflicting rules, classifies new cases based on multiple correlative rules, and takes rule supports and the significances of criteria into account to enhance classification accuracy.

### 2.2 Discovery of Conflicting Rules

Discovery of association rule is a crucial procedure in the AC. In the past few years, many popular methods have been proposed to mine rules for classification, e.g. CAEP and CMAR. However, the majority of them discover only rules with the maximal confidences when rules are conflicting. This strategy decreases the complexity of classification while pruning some valuable information.

Actually, we can make some improvement of the conventional methods to retain conflicting rules when inconsistency exists. In RSETAC, unlike the conventional methods which prune the rules without the maximal confidences, conflicting rules that pass a specified confidence threshold are all retained. The advantage of this strategy is that it can reduce some confusion caused by noisy and retain adequate information especially when the data set is inconsistent.

To retain conflicting rules introduces another problem into the analysis, that is, how to deal with these conflicting rules when employing them to classify new cases. In the following section, we employ a DST-based method to transform the conflicting rules into classification experts and combine them to give an aggregated classification.

### 2.3 Prediction of New Cases

In the classification procedure, RSETAC employs DST to combine the classification results of all the matching rules. The following three questions have to be solved if DST is employed to build a classifier. The first question is how to get evidences from rules. The second issue is whether the weights of evidence sources are equal and how to measure them if not. The last one is to choose a proper rule of combination in DST to combine distinct evidences in order to get an aggregated classification.

#### 2.3.1 Evidence Acquisition from Conflicting Rules

Suppose $RS=\{r_1,\cdots,r_i,\cdots,r_{|RS|}\}$ is the rule set based on which classification is made. For any subset $R_i$ of $RS$, $R_i$ is regarded as an expert if the three constraints are satisfied:

(1) $R_i=\{P_i \to c_{i_1},\cdots,P_i \to c_{i_j},\cdots,P_i \to c_{i_m}\}$, where $1\le i_1,\cdots,i_j,\cdots,i_m\le|C|$.

(2) To any other rules $(P_j\to c)\in\{RS\backslash R_i\}$, $P_j\ne P_i$.

(3) $c_{i_1}\ne\cdots\ne c_{i_j}\ne\cdots\ne c_{i_m}$.

That is to say, $R_i$ includes all of the rules which have the same condition but distinct class labels. For convenience, we rewrite $R_i$ as follows:

$$R_i: P_i \to (c_{i_1}, conf_{i_1})\vee\cdots\vee(c_{i_j}, conf_{i_j})\vee\cdots\vee(c_{i_m}, conf_{i_m}). \tag{2}$$

where $conf_{i_j}$ is the rule confidence of $P_i \to c_{i_j}$.

Therefore, we can get the basic probability assignment given by expert $R_i$:

$$m_i = \{conf_{i_1}, \cdots, conf_{i_j}, \cdots, conf_{i_m}, conf_\theta\}. \tag{3}$$

where $conf_\theta$ is the belief degree assigned to the ignorance by expert $R_i$:

$$conf_\theta = conf(P_i \to C \setminus \{c_{i_1}, \cdots, c_{i_m}\}) = 1 - \sum_{j=1}^{m} conf_{i_j} \tag{4}$$

In this way, a set of classification expert $\{R_1,\cdots,R_i,\cdots,R_j,\cdots,R_t\}$ can be obtained from data set $I$, which satisfies the following constraints:

(1) $\bigcup_{i=1}^{t} R_i = RS$.

(2) $R_i \cap R_j = \varnothing$, for any $i$ and $j$, $i \neq j$.

That is to say, we can not only utilize all the rules obtained by rule discovery methods but also transform them into $t$ independent evidence sources.

### 2.3.2 Evidence Weight

In RST, different attributes have different significances and support our decision distinctly. Obviously, decisions based on the observation of crucial attributes are more reliable and accurate than those based on the observation of nonsignificant criteria.

Suppose that the condition of rule $R_i$ consists of $B_i$, $B_i=\{A_{i1},A_{i2},\cdots,A_{im}\}$, RSETAC defines the weight of $R_i$ associated with attribute significance from the viewpoint of VPRS model [11] as follows:

$$impa_i = \sigma^\beta(B_i, C) = \frac{\gamma_A^\beta(C) - \gamma_{A-B_i}^\beta(C)}{\gamma_A^\beta(C)} = 1 - \frac{\gamma_{A-B_i}^\beta(C)}{\gamma_A^\beta(C)} \tag{5}$$

where $\gamma_A^\beta(C)$ is the dependency of $C$ on $A$ and $\gamma_{A-B_i}^\beta(C)$ is the dependency of $C$ on $A$ without $B_i$ and $\beta$ is the admissible classification error in the VPRS model[11].

In addition, RSETAC also takes a horizontal measure, rule support, into account to obtain a more credible measure of evidence weight. Intuitively, if a rule has a high frequency in data, it would be assigned a large weight and the rule has a high possibility to be useful.

Suppose $sup_i$ is the set of supports associated with rules in $R_i$,

$$sup_i = \{sup_{i1}, sup_{i2}, \cdots, sup_{ip}\}. \tag{6}$$

To obtain the maximal support of expert $R_i$, RSETAC defines the maximal number in $sup_i$ as the weight of $R_i$ associated with rule support:

$$imps_i = \max\{sup_{i1}, sup_{i2}, \cdots, sup_{ip}\}. \tag{7}$$

If a body of evidence is supported by the majority of cases in data and its condition consists of important attributes, it should be considered of more importance and the resulting classification should be given more credibility. Therefore, RSETAC defines the integrated weight of evidence as follows:

$$\omega_i = \frac{\alpha * impa_i + \gamma * imps_i}{\sum_{j=1}^{t} (\alpha * impa_j + \gamma * imps_j)} \tag{8}$$

where $i$=1,2,⋯,$t$, $\alpha$ and $\gamma$ are two adjustable parameters that control the relative influence of $impa_i$ and $imps_i$.

### 2.3.3 Evidence Combination and Classification

In this section, Yang's rule of combination [12] in DST is used to combine the matching evidences of a new case. The final classification will be made according to the aggregated bpas.

In the framework of DST, a crucial role is played by Dempster's rule which has several interesting properties such as commutativity and associativity [9]. However, counterintuitive results may be obtained by Dempster's rule when evidences conflict [13]. To avoid counterintuitive results and get more accurate classification, RSETAC employs Yang's rule of combination to combine evidences obtained from rules.

For a new case $o$, suppose its matching expert set is $RS_o$={$R_{o1}$,⋯,$R_{oi}$,⋯,$R_{os}$}, the corresponding evidence weights and bpas are $W_o$={$\omega_{o1}$,⋯,$\omega_{oi}$,⋯,$\omega_{os}$} and $M_o$={$m_{o1}$,⋯,$m_{oi}$,⋯,$m_{os}$}, respectively.

Trivially, if $RS_o$ is not conflicting, i.e. all the rules matching $o$ have the same class label, RSETAC just assigns that class label to $o$.

If $s$=1, there is only one expert $R_{o1}$ can be employed to classify $o$, RSETAC assigns the class label with the maximal confidence to $o$.

If $s$≥2, i.e. $RS_o$ consists of more than one conflicting experts, RSETAC combines two evidences every time to get an aggregated classification. The steps of the pairwise combination, similar to the Yang's rule of combination [12], are as follows.

First, the set of evidence weights $W_o$ is transformed to be $W'$={$\omega'_1$,⋯,$\omega'_i$,⋯,$\omega'_s$}, by the following unitary function:

$$\omega'_i = \frac{\omega_{oi}}{\sum_{j=1}^{s} \omega_{oj}} \quad i = 1,2,\cdots,s \tag{9}$$

Second, with the new weights of evidence $W'$ and multiple evidences in $M_o$, we can get the integrative belief assignment, denoted $m'_s$, employing the combination steps of Yang's rule. The bpa $m'_s$ is the aggregated belief assignment of all the matching experts transformed from $RS_o$.

Third, RSETAC assigns the class label associated with the maximal belief in $m'_s$ as the final classification result.

## 3 Numerical Example

This numerical example employs a data set, named balance scale database, provided by Tim Hume to show the procedure of the proposed method [14]. The database, containing 625 examples, is initially generated to model psychological experiments. Each example is classified as one of the three class labels, having the balance scale tip

to the right ($R$), tip to the left ($L$), or be balanced ($B$). The attributes are the left weight ($A_1$), the left distance ($A_2$), the right weight ($A_3$), and the right distance ($A_4$). Of the 625 cases, 500 cases are random selected to create the training set and the 125 remained to form the testing set. Only the training set is submitted to rule generation.

In the procedure of rule generation, the MCAR algorithm is employed to mine rules from the training set. To retain other classes and more useful rules, we make two aspects of improvement to the MCAR algorithm. First, the conflicting rules that pass the minimal confidence threshold are retained. Second, we use a coverage threshold to select training set coverage [6] in the procedure of pruning rules based on training

**Table 1.** Experts generated by the improved MCAR algorithm. The symbol $*$ represents that the attribute value is not considered and the $\theta$ represents the ignorance of each expert.

| *Expert* | *Expert* | | | | | *sup* | *bpa* |
|---|---|---|---|---|---|---|---|
| *ID* | $A_1$ | $A_2$ | $A_3$ | $A_4$ | C | | |
| 1 | * | * | 1 | * | $\{L,\theta\}$ | 0.156 | {0.7959, 0.2041} |
| 2 | * | * | * | 1 | $\{L,\theta\}$ | 0.164 | {0.781, 0.219} |
| 3 | * | 1 | * | * | $\{L,\theta\}$ | 0.156 | {0.78, 0.22} |
| 4 | 1 | * | * | * | $\{L,\theta\}$ | 0.146 | {0.7604, 0.2396} |
| 5 | * | * | 5 | * | $\{L,\theta\}$ | 0.144 | {0.7273, 0.2727} |
| 6 | 5 | * | * | * | $\{L,\theta\}$ | 0.142 | {0.71, 0.29} |
| 7 | * | 5 | * | * | $\{L,\theta\}$ | 0.142 | {0.703, 0.297} |
| 8 | * | * | * | 5 | $\{L,\theta\}$ | 0.142 | {0.703, 0.297} |
| 9 | 4 | * | * | * | $\{L,\theta\}$ | 0.128 | {0.6214, 0.3786} |
| 10 | * | * | 2 | * | $\{L,\theta\}$ | 0.118 | {0.6146, 0.3854} |
| 11 | * | 4 | * | * | $\{L,\theta\}$ | 0.122 | {0.61, 0.39} |
| 12 | * | * | 4 | * | $\{L,\theta\}$ | 0.13 | {0.6075, 0.3925} |
| 13 | * | * | * | 4 | $\{L,\theta\}$ | 0.118 | {0.118, 0.882} |
| 14 | * | * | * | 2 | $\{L,\theta\}$ | 0.112 | {0.112, 0.888} |
| 15 | 2 | * | * | * | $\{L,\theta\}$ | 0.112 | {0.112, 0.888} |
| 16 | * | 2 | * | * | $\{L,\theta\}$ | 0.11 | {0.11, 0.89} |
| 17 | * | 3 | * | * | $\{L,\theta\}$ | 0.1 | {0.5102, 0.4898} |
| 18 | * | * | * | 3 | $\{L,R,\theta\}$ | 0.1 | {0.4158, 0.495, 0.0892} |
| 19 | 3 | * | * | * | $\{L,R,\theta\}$ | 0.096 | {0.48, 0.43, 0.09} |
| 20 | * | * | 3 | * | $\{L,R,\theta\}$ | 0.096 | {0.43, 0.48, 0.09} |

set coverage. In our experiment, the coverage threshold is set to be 2, the minimal support and confidence threshold is set to be the same as MCAR. From the training set, twenty classification experts are obtained using the method proposed in Section 2.3.1. The experts transformed from the twenty three rules together with their supports and bpas are presented in Table 1.

As can be seen in Table 1, there are three pairs of conflicting rules. The differences of confidences between the conflicting rules are not obvious enough to prune any of them. However, it is a pity that the rules like $<A_4, 3>\rightarrow L$, $<A_1, 3>\rightarrow R$ and $<A_3, 3>\rightarrow L$ are pruned by the conventional rule discovery methods. The degrees of belief assigned to the class label $L$ and $R$ are smaller than those of other experts, but they play an important role in the combination in many problems.

The parameters that control the relative weight associated with attribute significance (*impa*) and rule support (*imps*) should be different with respect to different data sets. For the balance scale database, the relative influence of the two parameters (*impa*/*imps*) to the classification accuracy is shown in Fig. 1. As shown in Fig. 1, the larger the value of *impa*/*imps* is, the better the classification accuracy is. And the best classification accuracy is 0.8880 when *impa*/*imps* is larger than 5/1. This means that the attribute significance is more important in the measurement of evidence weight for the balance scale database. For the other databases, the influence of the relative weight may be contrary and the rule support may play a crucial role.

For each case in the testing set, our experiment predicts the class label based on multiple matching experts in Table 1 employing the combination strategy proposed in Section 2.3.3. The best classification accuracy of our method in comparison with CBA, C4.5, RIPPER and MCAR is shown in Table 2.

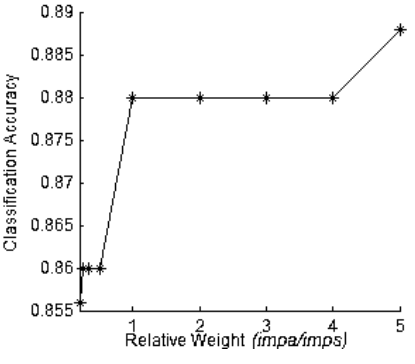



**Fig. 1.** Influence of the attribute significance and rule support to the classification accuracy

**Table 2.** Comparison of CBA, C4.5, RIPPER, MCAR and our method. The classification accuracy of CBA, C4.5, RIPPER and MCAR come from [10].

| Method | CBA | C4.5 | RIPPER | MCAR | Our method |
|---|---|---|---|---|---|
| Accuracy | 0.6566 | 0.6432 | 0.7456 | 0.7754 | 0.8880 |

The results in Table 2 indicate that RSETAC outperforms the four conventional classification methods in terms of accuracy. There are two main reasons result in the high accuracy. First, RSETAC can integrate the effects of all the matching rules and yield more accurate classification. Second, the definition of evidence weight takes not only rule confidence but also rule support and attribute significance into account when determining the class labels of new cases. In addition, the conflicting rules, i.e. the last three experts in Table 1 are also important to the high classification accuracy.

## 4 Conclusions

In this paper, two challenges in associative classification were investigated: (1) extending to multi-class rules classification, and (2) classification based on multiple rules. The outcome is a new combination strategy, RSETAC, that has several distinguished features over other existing techniques: (1) RSETAC develops a strategy to deal with multi-class rules classification and transforms the conflicting rules to be classification experts. (2) Distinguished from the conventional methods, RSETAC employs all of the matching rules to determine the class label of a new case. The evidence theory-based method combines the classification results of all the matching rules and provides a collective class label. (3) RSETAC presents a new method to measure the weights of evidences.

To explicate the procedure of RSETAC, a numerical example based on the balance scale database from UCI database was presented. The example shows that our method can mine and deal with multi-class rules effectively. Particularly, the combination method based on multiple rules gets much better classification accuracy in comparison with CBA, C4.5, RIPPER and MCAR.

**Acknowledgments.** This work was supported by the NSFC (Grant No.70672097) and the State Key Program of NSFC (Grant No.70631003).

## References

1. Dong, G., Zhang, X., Wong, L., etc.: CAEP: Classification by Aggregating Emerging Patterns. Pro. of 2nd Int. Conf. on Discovery Science. Springer-Verlag, Tokyo, Japan (1999)
2. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. Pro. of the 4th Int. Conf. on KDD and DM (KDD-98). ACM press, N.Y., USA (1998)
3. Quinlan, J.: C4.5: Programs for machine learning. Morgan Kaufmann, S.M., Canada (1993)
4. Thabtah, F., Cowling, P., Peng, Y.: MCAR: Multi-class Classification based on Association Rule Approach. Pro. of the 3rd IEEE Int. Conf. on Computer Systems and Applications. Cairo, Egypt (2005)
5. Thabtah, F., Cowling, P., Peng, Y.H.: A study of Predictive Accuracy for Four Associative Classifiers. Journal of Digital Information Management 3 (2005) 202-205
6. Li, W.M., Han, J.W., Pei, J.: CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. Proceedings of the 2001 IEEE International Conference on Data Mining. California, USA (2001)

7. Pawlak, Z.: Rough sets. Int. J. of Computer and Information Sciences **11**(5) (1982) 341-356
8. Shafer, G.: A Mathematical Theory of Evidence. Princeton Uni. Press, Princeton (1976)
9. Dempster, A.P.: Upper and Lower Probabilities Induced by a Multivalued Mapping. Annals of Mathematical Statistics, 38(2) (1967) 325-339
10. Thabtah, F., Cowling, P., Hammoud, S.: Improving rule sorting, predictive accuracy and training time in associative classification. Expert Systems with Appl. 31 (2006) 414-426
11. Ziarko, W.: Variable Precision Rough Set Model. Journal of Computer and System Sciences 40 (1993) 39-59
12. Yang, J.B., Wang, Y.M., Xu, D.L., etc.: The evidential reasoning approach for MADA under both probabilistic and fuzzy uncertainties. Euro. J. of OR. **171**(1) (2006) 309-343
13. Murphy, C.K.: Combining belief functions when evidence conflicts. Decision Support Systems 29 (2000) 1-9
14. Murphy, P.M., Aha, D.W.: UCI Repository machine learning databases Irvine. University of California, Canada (1996)

# Evaluation Method for Decision Rule Sets

Yuhua Qian and Jiye Liang

Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education
School of Computer and Information Technology, Shanxi University
Taiyuan, 030006, People's Republic of China
jinchengqyh@126.com, ljy@sxu.edu.cn

**Abstract.** In this paper, a decision table in rough set theory is classified into three types according to its consistency. Three parameters $\alpha$ (whole certainty measure), $\beta$ (whole consistency measure) and $\gamma$ (whole support measure) are introduced to evaluate the performance of a decision rule set induced from a decision table. For three types of decision tables, the dependency of the parameters upon condition/decision granulation is analyzed. The parameters can be used to construct an evaluation function in favor of selecting a better one from some different rule acquiring methods for real decision problems.

**Keywords:** Rough set theory, decision table, decision rule, knowledge granulation, decision evaluation.

## 1 Introduction

Recently, rough set theory proposed by Pawlak in [1] has become a popular mathematical framework for pattern recognition, image processing, feature selection, neuro computing, conflict analysis, decision support, data mining and knowledge discovery process from large data sets [2-7]. For decision problems, by various kinds of reduct techniques, a set of decision rules can be generated from a decision table for classification or prediction [8-10].

In recent years, how to evaluate the performance of a decision rule has been becoming a very important issue in rough set theory[11-16]. In fact, a set of decision rules can be generated from a decision table by adopting any kind of reduction methods. In [11], Yao proposed several evaluation criterions for decision rules such as the generality, the absolute support, the change of support and the change of support, and so on. In [13], based on information entropy, Düntsch suggested some uncertainty measures of a decision rule, and proposed three criterions for model selection as well. In additional, several other measures such as certainty measure and support measure are often used to evaluate a decision rule [3, 7, 15]. However, because all of these measures are defined only for a single decision rule, they are unsuitable for measuring the whole performance of a rule set. Another two kinds of measures, the approximation accuracy for decision classes and the consistency degree for a decision table [1, 16], in some

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 272–279, 2007.

sense, could be regarded as measures for whole performance of all decision rules generated from a decision table. Nevertheless, the approximation accuracy and consistency degree have some limitations. For instance, the certainty and consistency of a rule set could not be well depicted by the approximation accuracy and consistency degree when their values achieve 0. As we know, the fact that approximation accuracy/consistency degree is equal to 0 only implies that there is no decision rule with the certainty 1 in the decision table. So the approximation accuracy and consistency degree of a decision table cannot give elaborate depictions of the certainty and consistency to a rule set.

This paper aims to find some criterions for evaluating the whole performance of a set of decision rules. In Section 2, some preliminary concepts such as indiscernibility relation, partition, partial relation of knowledge and decision table are briefly recalled. In Section 3, three parameters $\alpha$, $\beta$ and $\gamma$ for evaluating a set of rules are introduced. The dependency of the parameters upon condition/decision granulation is analyzed. Section 4 concludes the paper.

## 2 Some Basic Concepts

An information system $S$ is a pair $(U, A)$, where $U$ is a non-empty, finite set of objects called the universe and $A$ is a non-empty, finite set of attributes, such that $a : U \to V_a$ for any $a \in A$, where $V_a$ is called the domain of $a$.

Each non-empty subset $B \subseteq A$ determines an indiscernibility relation $R_B = \{(x, y) \in U \times U \mid a(x) = a(y), \forall a \in B\}$. The relation $R_B$ partitions $U$ into some equivalence classes $U/R_B = \{[x]_B \mid x \in U\}$, where $[x]_B = \{y \in U \mid (x, y) \in R_B\}$.

We define a partial relation $\preceq$ on the family $\{U/B \mid B \subseteq A\}$ as follows[17]: $U/P \preceq U/Q$ (or $U/Q \succeq U/P$), if and only if, for every $P_i \in U/P$, there exists $Q_j \in U/Q$ such that $P_i \subseteq Q_j$, where $U/P = \{P_1, P_2, \cdots, P_m\}$ and $U/Q = \{Q_1, Q_2, \cdots, Q_n\}$ are partitions induced by $P, Q \subseteq A$, respectively. In this case, we say that $Q$ is coarser than $P$, or $P$ is finer than $Q$. If $U/P \preceq U/Q$ and $U/P \neq U/Q$, we say $Q$ is strictly coarser than $P$ (or $P$ is strictly finer than $Q$), denoted by $U/P \prec U/Q$ (or $U/Q \succ U/P$). It is clear that $U/P \prec U/Q$, if and only if, for every $X \in U/P$, there exists $Y \in U/Q$ such that $X \subseteq Y$, and there exist $X_0 \in U/P$, $Y_0 \in U/Q$ such that $X_0 \subset Y_0$.

A decision table is an information system $S = (U, C \cup D)$ with $C \cap D = \emptyset$, where $C$ is called condition attribute set, and $D$ is called decision attribute set. If $U/C \preceq U/D$, then $S = (U, C \cup D)$ is said to be consistent, otherwise it is inconsistent.

**Definition 1.** [1,16] *Let $S = (U, C \cup D)$ be a decision table, $X_i \in U/C$, $Y_j \in U/D$ and $X_i \cap Y_j \neq \emptyset$. By $des(X_i)$ and $des(Y_j)$, we denote the descriptions of the equivalence classes $X_i$ and $Y_j$ in the decision table $S$. A decision rule is formally defined as $Z_{ij} : des(X_i) \to des(Y_j)$.*

The certainty measure and support measure of a decision rule $Z_{ij}$ are defined as $\mu(Z_{ij}) = |X_i \cap Y_j|/|X_i|$, $s(Z_{ij}) = |X_i \cap Y_j|/|U|$, where, by $|\cdot|$, we denote the

cardinality of a set. It is clear that the values of $\mu(Z_{ij})$ and $s(Z_{ij})$ of a decision rule $Z_{ij}$ fall into the interval $[\frac{1}{|U|}, 1]$.

By $|Z_{ij}|$, we denote the cardinality of the set $X_i \cap Y_j$, which is called the support number of the rule $Z_{ij}$. For convenience, by $a(x)$ $(a \in C)$ and $d(x)$ $(d \in D)$, we denote the values of the object $x$ under the condition attribute $a$ and the decision attribute $d$, respectively.

**Definition 2.** *Let $S = (U, C \cup D)$ be a decision table, $U/C = \{X_1, X_2, \cdots, X_m\}$, $U/D = \{Y_1, Y_2, \cdots, Y_n\}$. A condition class $X_i \in U/C$ is said to be consistent if $d(x) = d(y)$ for $\forall x, y \in X_i$ and $\forall d \in D$; a decision class $Y_j \in U/D$ is said to be converse consistent if $a(x) = a(y)$ for $\forall x, y \in Y_j$ and $\forall a \in C$.*

It is easy to see that a decision table $S = (U, C \cup D)$ is consistent if every condition class $X_i \in U/C$ is consistent.

**Definition 3.** *Let $S = (U, C \cup D)$ be a decision table, $U/C = \{X_1, X_2, \cdots, X_m\}$, $U/D = \{Y_1, Y_2, \cdots, Y_n\}$. $S$ is said to be converse consistent, if every decision class $Y_j \in U/D$ is converse consistent, i.e., $U/D \preceq U/C$.*

A decision table is called a mixed decision table if it is neither consistent nor converse consistent.

$S = (U, C \cup D)$ is called to be restrict consistent (restrict converse consistent) if $U/C \prec U/D$ $(U/D \prec U/C)$.

**Definition 4.** [15,18] *Let $S = (U, A)$ be an information system, $U/A = \{R_1, R_2, \cdots, R_m\}$. The knowledge granulation of $A$ is defined as*

$$G(A) = \frac{1}{|U|^2} \sum_{i=1}^{m} |R_i|^2. \tag{1}$$

Consequently, $G(C)$, $G(D)$ and $G(C \cup D)$ are called as the condition granulation, decision granulation and granulation of $S$, respectively.

## 3 Whole Performance Evaluation for a Rule Set

In rough set theory, several measures for a decision rule $Z_{ij} : des(X_i) \to des(Y_j)$ have been introduced in [1], such as certainty measure $\mu(X_i, Y_j) = |X_i \cap Y_j|/|X_i|$, support measure $s(X_i, Y_j) = |X_i \cap Y_j|/|U|$. However, because $\mu(X_i, Y_j)$ and $s(X_i, Y_j)$ are defined only for a single decision rule, they are unsuitable for measuring the whole performance of a rule set.

In [1], the approximation accuracy of a classification is introduced by Pawlak. Let $F = \{Y_1, Y_2, \cdots, Y_n\}$ be a classification of the universe $U$, and $C$ a condition attribute set. $\underline{C}F = \{\underline{C}Y_1, \underline{C}Y_2, \cdots, \underline{C}Y_n\}$ and $\overline{C}F = \{\overline{C}Y_1, \overline{C}Y_2, \cdots, \overline{C}Y_n\}$ are called $C$-lower and $C$-upper approximations of $F$, where $\underline{C}Y_i = \bigcup\{x \in U \mid [x]_C \subseteq Y_i \in F\}(1 \le i \le n)$, $\overline{C}Y_i = \bigcup\{x \in U \mid [x]_C \cap Y_i \neq \emptyset, Y_i \in F\}(1 \le i \le n)$. The approximation accuracy of $F$ by $C$ is defined as $a_C(F) = \frac{\sum_{Y_i \in U/D} |\underline{C}Y_i|}{\sum_{Y_i \in U/D} |\overline{C}Y_i|}$. The

approximation accuracy expresses the percentage of possible correct decisions when classifying objects employing the attribute set $C$. In a sense, $a_C(F)$ can be used to measure certainty of a decision table. The consistency degree of a decision table $S = (U, C \cup D)$, another measure in rough set theory, is defined as $c_C(D) = \frac{1}{|U|}\sum_{i=1}^{n} |\underline{C}Y_i|$. The consistency degree expresses the percentage of objects which can be correctly classified to decision classes of $U/D$ by condition attribute set $C$. In a sense, $c_C(D)$ can be used to measure the consistency of a decision table.

Nevertheless, the certainty and consistency of a rule set could not be well depicted by approximation accuracy and consistency degree when their values achieve 0. Here, three new evaluation parameters $\alpha$, $\beta$ and $\gamma$ are introduced to solve the problem.

**Definition 5.** *Let $S = (U, C \cup D)$ be a decision table, $RULE = \{Z_{ij} | Z_{ij} : des(X_i) \rightarrow des(Y_j), X_i \in U/C, Y_j \in U/D\}$. The certainty measure $\alpha$ of $S$ is defined as*

$$\alpha(S) = \sum_{i=1}^{m}\sum_{j=1}^{n} s(Z_{ij})\mu(Z_{ij}) = \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{|X_i \cap Y_j|^2}{|U||X_i|}, \tag{2}$$

*where $s(Z_{ij})$ and $\mu(Z_{ij})$ are the certainty measure and support measure of the rule $Z_{ij}$, respectively.*

Although the parameter $\alpha$ is defined in the context of all decision rules from a decision table, it is also suitable to an arbitrary decision rule set as well.

**Theorem 1 (Extremum).** *Let $S = (U, C \cup D)$ be a decision table, $RULE = \{Z_{ij} | Z_{ij} : des(X_i) \rightarrow des(Y_j), X_i \in U/C, Y_j \in U/D\}$.*

*(1) For every $Z_{ij} \in RULE$, if $\mu(Z_{ij}) = 1$, then the parameter $\alpha$ achieves its maximum value* 1*;*

*(2) If $m = 1$ and $n = |U|$, then parameter $\alpha$ achieves its minimum value $\frac{1}{|U|}$.*

**Remark.** In fact, a decision table $S = (U, C \cup D)$ is consistent if and only if every decision rule from $S$ is certain, i.e., its certainty measure is equal to 1. So, (1) of Theorem 1 shows that the parameter $\alpha$ achieves its maximum value 1 when $S$ is consistent. (2) of Theorem 1 shows that $\alpha$ achieves its minimum value $\frac{1}{|U|}$ when we want to distinguish any two objects of $U$ without any condition information.

**Theorem 2.** *Let $S_1 = (U, C_1 \cup D_1)$ and $S_2 = (U, C_2 \cup D_2)$ be two converse consistent decision tables. If $U/C_1 = U/C_2$, $U/D_2 \prec U/D_1$, then $\alpha(S_1) > \alpha(S_2)$.*

**Proof.** From $U/C_1 = U/C_2$ and the converse consistency of $S_1$ and $S_2$, it follows that there exist $X_p \in U/C_1$ and $Y_q \in U/D_1$ such that $Y_q \subseteq X_p$. By $U/D_2 \prec U/D_1$, there exist $Y_q^1, Y_q^2, \cdots, Y_q^s \in U/D_2$ $(s > 1)$ such that $Y_q = \bigcup_{k=1}^{s} Y_q^k$. In other words, the rule $Z_{pq}$ in $S_1$ can be decomposed into a family of rules $Z_{pq}^1, Z_{pq}^2, \cdots, Z_{pq}^s$ in $S_2$. It is clear that $|Z_{pq}| = \sum_{k=1}^{s} |Z_{pq}^k|$. Therefore, $|Z_{pq}|^2 > \sum_{k=1}^{s} |Z_{pq}^k|^2$. Hence, by the definition of $\alpha(S)$, $\alpha(S_1) > \alpha(S_2)$.

Theorem 2 states that the certainty measure $\alpha$ of a converse consistent decision table decreases with its decision classes becoming finer.

**Theorem 3.** *Let $S_1 = (U, C_1 \cup D_1)$ and $S_2 = (U, C_2 \cup D_2)$ be two converse consistent decision tables. If $U/D_1 = U/D_2$, $U/C_2 \prec U/C_1$, then $\alpha(S_1) < \alpha(S_2)$.*

**Proof.** From $U/C_2 \prec U/C_1$, there exists $X_l \in U/C_1$ and an integer $s > 1$ such that $X_l = \bigcup_{k=1}^{s} X_l^k$, where $X_l^k \in U/C_2$. It is clear that $|X_l| = \sum_{k=1}^{s} |X_l^k|$, and therefore, $\frac{1}{|X_l|} < \frac{1}{|X_l^1|} + \frac{1}{|X_l^2|} + \cdots + \frac{1}{|X_l^s|}$.

Noticing that both $S_1$ and $S_2$ are converse consistent, we have $|Z_{lq}| = |Z_{lq}^k|$ $(k = 1, 2, \cdots, s)$. Hence, we have that

$$\begin{aligned}
\alpha(S_1) &= \sum_{i=1}^{m}\sum_{j=1}^{n} s(Z_{ij})\mu(Z_{ij}) \\
&= \frac{1}{|U|}\sum_{i=1}^{l-1}\sum_{j=1}^{n}\frac{|Z_{ij}|^2}{|X_i|} + \frac{1}{|U|}\sum_{j=1}^{n}\frac{|Z_{lj}|^2}{|X_l|} + \frac{1}{|U|}\sum_{i=l+1}^{m}\sum_{j=1}^{n}\frac{|Z_{ij}|^2}{|X_i|} \\
&< \frac{1}{|U|}\sum_{i=1}^{l-1}\sum_{j=1}^{n}\frac{|Z_{ij}|^2}{|X_i|} + \frac{1}{|U|}\sum_{k=1}^{s}\sum_{j=1}^{n}\frac{|Z_{lj}|^2}{|X_l^k|} + \frac{1}{|U|}\sum_{i=l+1}^{m}\sum_{j=1}^{n}\frac{|Z_{ij}|^2}{|X_i|} \\
&= \alpha(S_2).
\end{aligned}$$

Theorem 3 states that the certainty measure $\alpha$ of a converse consistent decision table increases with its condition classes becoming finer.

**Definition 6.** *Let $S = (U, C \cup D)$ be a decision table, $RULE = \{Z_{ij} | Z_{ij} : des(X_i) \to des(Y_j), X_i \in U/C, Y_j \in U/D\}$. The consistency measure $\beta$ of $S$ is defined as*

$$\beta(S) = \sum_{i=1}^{m}\frac{|X_i|}{|U|}[1 - \sum_{j=1}^{N_i}\mu(Z_{ij})(1 - \mu(Z_{ij}))], \tag{3}$$

*where $N_i$ is the number of decision rules determined by the condition class $X_i$, $\mu(Z_{ij})$ is the certainty measure of the rule $Z_{ij}$.*

Although the parameter $\beta$ is defined in the context of all decision rules from a decision table, it is also suitable to an arbitrary decision rule set as well.

**Theorem 4 (Extremum).** *Let $S = (U, C \cup D)$ be a decision table, $RULE = \{Z_{ij} | Z_{ij} : des(X_i) \to des(Y_j), X_i \in U/C, Y_j \in U/D\}$.*

*(1) For every $Z_{ij} \in RULE$, if $\mu(Z_{ij}) = 1$, then the parameter $\beta$ achieves its maximum value 1;*

*(2) For every $Z_{ij} \in RULE$, if $\mu(Z_{ij}) = \frac{1}{|U|}$, then the parameter $\beta$ achieves its minimum value $\frac{1}{|U|}$.*

It should be noted that the parameter $\beta$ achieves its maximum 1 when $S = (U, C \cup D)$ be a consistent decision table.

**Theorem 5.** *Let $S_1 = (U, C_1 \cup D_1)$ and $S_2 = (U, C_2 \cup D_2)$ be two converse consistent decision tables or mixed decision tables. If $U/C_1 = U/C_2$, $U/D_2 \prec U/D_1$, then $\beta(S_1) > \beta(S_2)$.*

**Proof.** A mixed decision table $S$ can be transformed into a converse consistent decision table $S'$ via deleting all certainty decision rules. And it is clear that

$\beta(S) = \beta(S^{'})$. So, we only need to prove this theorem for converse consistent tables.

Since $U/C_1 = U/C_2$ and the converse consistency of $S_1$ and $S_2$, there exist $X_p \in U/C_1$ and $Y_q \in U/D_1$ such that $Y_q \subseteq X_p$. By $U/D_2 \prec U/D_1$, there exist $Y_q^1, Y_q^2, \cdots, Y_q^s \in U/D_2$ $(s > 1)$ such that $Y_q = \bigcup_{k=1}^{s} Y_q^k$. In other words, the rule $Z_{pq}$ in $S_1$ can be decomposed into a family of rules $Z_{pq}^1, Z_{pq}^2, \cdots, Z_{pq}^s$ in $S_2$. It is clear that $|Z_{pq}| = \sum_{k=1}^{s} |Z_{pq}^k|$. Hence, we have that

$$\begin{aligned}
\mu(Z_{pq})(1-\mu(Z_{pq})) &= \frac{|Z_{pq}||X_p|-|Z_{pq}|^2}{|X_p|^2} \\
&= \frac{|Z_{pq}^1+Z_{pq}^2+\cdots+Z_{pq}^s||X_p|-|Z_{pq}^1+Z_{pq}^2+\cdots+Z_{pq}^s|^2}{|X_p|^2} \\
&< \frac{|Z_{pq}^1+Z_{pq}^2+\cdots+Z_{pq}^s||X_p|-(|Z_{pq}^1|^2+|Z_{pq}^2|^2+\cdots+|Z_{pq}^s|^2)}{|X_p|^2} \\
&= \frac{|Z_{pq}^1||X_p|-|Z_{pq}^1|^2}{|X_p|^2} + \frac{|Z_{pq}^2||X_p|-|Z_{pq}^2|^2}{|X_p|^2} + \cdots + \frac{|Z_{pq}^s||X_p|-|Z_{pq}^s|^2}{|X_p|^2} \\
&= \sum_{k=1}^{s} \mu(Z_{pq}^k)(1-\mu(Z_{pq}^k)).
\end{aligned}$$

Then, we can obtain that

$$\begin{aligned}
\beta(S_1) &= \sum_{i=1}^{m} \frac{|X_i|}{|U|}[1 - \sum_{j=1}^{N_i} \mu(Z_{ij})(1-\mu(Z_{ij}))] \\
&= \sum_{i=1}^{p-1} \frac{|X_i|}{|U|}[1 - \sum_{j=1}^{N_i} \mu(Z_{ij})(1-\mu(Z_{ij}))] + \frac{|X_p|}{|U|}[1 - \sum_{j=1}^{N_p} \mu(Z_{pj})(1- \\
&\quad \mu(Z_{pj}))] + \sum_{i=p+1}^{m} \frac{|X_i|}{|U|}[1 - \sum_{j=1}^{N_i} \mu(Z_{ij})(1-\mu(Z_{ij}))] \\
&> \sum_{i=1}^{p-1} \frac{|X_i|}{|U|}[1 - \sum_{j=1}^{N_i} \mu(Z_{ij})(1-\mu(Z_{ij}))] + \sum_{i=p+1}^{m} \frac{|X_i|}{|U|}[1 - \sum_{j=1}^{N_i} \mu(Z_{ij})(1- \\
&\quad \mu(Z_{ij}))] + \frac{|X_p|}{|U|}[1 - \sum_{k=1}^{s} \mu(Z_{pq}^k)(1-\mu(Z_{pq}^k)) - \sum_{j=1, j\neq q}^{N_i} \mu(Z_{pj})(1-\mu(Z_{pj}))] \\
&= \beta(S_2).
\end{aligned}$$

Theorem 5 states that the consistency measure $\beta$ of a mixed (or converse consistent) decision table decreases with its decision classes becoming finer.

**Theorem 6.** *Let $S_1 = (U, C_1 \cup D_1)$ and $S_2 = (U, C_2 \cup D_2)$ be two converse consistent decision tables or mixed decision tables. If $U/D_1 = U/D_2$, $U/C_2 \prec U/C_1$, then $\beta(S_1) < \beta(S_2)$.*

**Proof.** Similar to the proof of Theorem 5, it can be proved.

Theorem 6 states that the consistency measure $\beta$ of a mixed (or converse consistent) decision table increases with its condition classes becoming finer.

**Definition 7.** *Let $S = (U, C \cup D)$ be a decision table, $RULE = \{Z_{ij} | Z_{ij} : des(X_i) \rightarrow des(Y_j), X_i \in U/C, Y_j \in U/D\}$. The support measure $\gamma$ of $S$ is defined as*

$$\gamma(S) = \sum_{i=1}^{m}\sum_{j=1}^{n} s^2(Z_{ij}) = \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{|X_i \cap Y_j|^2}{|U|^2}, \tag{4}$$

*where $s(Z_{ij})$ is the support measure of the rule $Z_{ij}$.*

Although the parameter $\gamma$ is defined in the context of all decision rules from a decision table, it is suitable to an arbitrary decision rule set as well.

**Theorem 7 (Extremum).** *Let $S = (U, C \cup D)$ be a decision table, $RULE = \{Z_{ij} | Z_{ij} : des(X_i) \to des(Y_j), X_i \in U/C, Y_j \in U/D\}$.*

*(1) If $m = n = 1$, then the parameter $\gamma$ achieves its maximum value 1;*

*(2) If $m = |U|$ or $n = |U|$, then the parameter $\gamma$ achieves its minimum value $\frac{1}{|U|}$.*

**Theorem 8.** *Let $S_1 = (U, C_1 \cup D_1)$ and $S_2 = (U, C_2 \cup D_2)$ be two decision tables, then $\gamma(S_1) < \gamma(S_2)$, if and only if, $G(C_1 \cup D_1) < G(C_2 \cup D_2)$.*

**Proof.** Suppose $U/(C \cup D) = \{X_i \cap Y_j \mid X_i \cap Y_j \neq \emptyset, X_i \in U/C_1, Y_j \in U/D\}$, $RULE = \{Z_{ij} | Z_{ij} : X_i \to Y_j, X_i \in U/C, Y_j \in U/D\}$. From Definition 4 and $s(Z_{ij}) = \frac{|X_i \cap Y_j|}{|U|}$, it follows that

$$\begin{aligned} G(C \cup D) &= \tfrac{1}{|U|^2} \sum_{i=1}^{m} \sum_{j=1}^{n} |X_i \cap Y_j|^2 \\ &= \sum_{i=1}^{m} \sum_{j=1}^{n} (\tfrac{|X_i \cap Y_j|}{|U|})^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} s^2(Z_{ij}) \\ &= \gamma(S). \end{aligned}$$

Therefore, $\gamma(S_1) < \gamma(S_2)$ if and only if $G(C_1 \cup D_1) < G(C_2 \cup D_2)$.

Theorem 8 states that the support measure $\gamma$ of a decision table increases with the granulation of the decision table becoming bigger.

**Theorem 9.** *Let $S_1 = (U, C_1 \cup D_1)$ and $S_2 = (U, C_2 \cup D_2)$ be two converse consistent decision tables. If $U/C_1 = U/C_2$, $U/D_1 \prec U/D_2$, then $\gamma(S_1) < \gamma(S_2)$.*

**Proof.** Similar to Theorem 5, it can be proved.

Theorem 9 states that the support measure $\gamma$ of a decision table decreases with its decision classes becoming finer.

## 4 Conclusions

In this paper, the limitations of the traditional measures are exemplified. Three parameters $\alpha$, $\beta$ and $\gamma$ are introduced to measure the certainty, consistency and support of a rule set obtained from a decision table, respectively. For three types of decision tables (consistent, converse consistent and mixed), the dependency of parameters $\alpha$, $\beta$ and $\gamma$ upon condition/decision granulation is analyzed.

**Acknowledgements.** This work was supported by the national natural science foundation of China (No. 70471003, No. 60573074, No. 60275019), the foundation of doctoral program research of ministry of education of China (No. 20050108004), the top scholar foundation of Shanxi, China, key project of science and technology research of the ministry of education of China (No. 206017) and the graduate student innovation foundation of Shanxi.

## References


1. Pawlak, Z.: Rough sets, *International Journal of Computer and Information Science*, 11 (1982) 341-356.
2. Bazan, J., Peters, J.F., Skowron, A., Nguyen, HS., Szczuka, M.: Rough set approach to pattern extraction from classifiers, *Electronic Notes in Theoretical Computer Science*, 82(4) (2003) 1-10.
3. Pawlak, Z., Skowron, A.: Rudiments of rough sets, *Information Sciences*, 177 (2007) 3-27.
4. Pawlak, Z., Skowron, A.: Rough sets: some extensions, *Information Sciences*, 177 (2007) 28-40.
5. Pawlak, Z., Skowron, A.: Rough sets and boolean reasoning, *Information Sciences*, 177 (2007) 41-73.
6. Pal, S. K., Pedrycz, W., Skowron, A., Swiniarski, R.: Presenting the special isssue on rough-neuro computing, *Neurocomputing*, 36 (2001) 1-3.
7. Pawlak, Z.: Some remarks on conflict analysis, *Information Sciences*, 166 (2005) 649-654.
8. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems, In: R. Slowiński(Eds.), Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory. Kluwer Academic, Dordrecht, (1992) 331-362.
9. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: a tutuorial. In: S.K. Pal, A. Skowron(Eds.), Rough Fuzzy Hubridization: A New Trend in Decision Making. Springer, Singapore, (1999) 3-98.
10. Qian, Y. H., Liang, J. Y.: Combination entropy and combination granulation in incomplete information system, *Lecture Notes in Artificial Intelliegence*, 4062 (2006) 184-190.
11. Y. Y. Yao.: Probabilistic approaches to rough sets, *Expert Systems*, 20(5) (2003) 287-297.
12. Slezak, D.: Various approaches to reasoning with frequency based decision reducts: a survey, in: Polkowski, L., Tsumoto, S., Lin, T.Y.: Rough set methods and applications, Physica-verlag, Heidelberg, (2000) 235-285.
13. Düntsch, I., Gediaga, G.: Uncertainty measures of rough set prediction, *Artificial Intelligence*, 106(1) (1998) 109-137.
14. Huynh, V. N., Nakamori, Y.: A roughness measure for fuzzy sets, *Information Sciences*, 173 (2005) 255-275
15. Liang, J.Y., Li, D.Y.: *Uncertainty and Knowledge Acquisition in Information Systems*, Science Press, Beijing (2005).
16. Zhang, W.X., Wu, W.Z., Liang, J.Y., Li, D.Y.: *Theory and Method of Rough Sets*, Science Press, Beijing (2001).
17. Yao, J.T., Yao, Y.Y.: Induction of classification rules by granular computing, *Lecture Notes in Artifical Intelligence*, 2475 (2002) 331-338.
18. Liang, J.Y., Shi, Z.Z., Li, D.Y.: The information entropy, rough entropy and knowledge granulation in rough set theory, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(1) (2004) 37-46.

# On Possible Rules and Apriori Algorithm in Non-deterministic Information Systems: Part 2

Hiroshi Sakai[1], Ryuji Ishibashi[1], Kazuhiro Koba[1], and Michinori Nakata[2]

[1] Department of Mathematics and Computer Aided Science,
Faculty of Engineering, Kyushu Institute of Technology,
Tobata, Kitakyushu 804, Japan
sakai@mns.kyutech.ac.jp
[2] Faculty of Management and Information Science,
Josai International University,
Gumyo, Togane, Chiba 283, Japan
nakatam@ieee.org

**Abstract.** A framework of *Non-deterministic Information Systems* (*NISs*) is known well for handling information incompleteness in *Deterministic Information Systems* (*DISs*). Apriori algorithm for the standard tables or *DISs* is also known as an algorithm to generate rules, which are characterized by criteria, *support* and *accuracy*. This paper extends Apriori algorithm in *DISs* to Apriori algorithm in *NISs*. This extended Apriori algorithm employs criteria, *minimum support* and *minimum accuracy* in *NISs*, and generates rules under the worst condition. A software tool is also implemented.

**Keywords:** Rough sets, Non-deterministic information, Apriori algorithm, Association rules.

## 1 Introduction

Rough set theory is seen as a mathematical foundation of soft computing. This theory usually handles tables with deterministic information. Many applications of this theory to rule generation, machine learning and knowledge discovery have been presented [1,2].

We follow rule generation in *DISs* [1,2], and we cope with rule generation in *NISs*. *NISs* were proposed by Pawlak, Orłowska and Lipski in order to handle information incompleteness in *DISs*, like null values, unknown values, missing values. From the beginning of the research on incomplete information, *NISs* have been recognized to be the most important framework for handling information incompleteness [3,4].

In [3], Lipski showed a question-answering system besides an axiomatization of logic. Orłowska established rough set analysis for incomplete information [4], and Grzymala-Busse developed a system named *LERS*, which depends upon *LEM*1 and *LEM*2 algorithms [5]. Stefanowski and Tsoukias also defined non

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 280–288, 2007.

symmetric similarity relations and valued tolerance relations for analysing incomplete information [6], and recently Kryszkiewicz proposed a framework of rules in incomplete information systems [7]. As far as authors know, these are the most important work for handling incomplete information, especially missing values. We have also coped with several issues related to $NISs$, and proposed a framework *Rough Non-deterministic Information Analysis* ($RNIA$) [8]. Apriori algorithm is also known as an algorithm to generate rules in $DISs$ [9], and we have extended this algorithm to a new algorithm in $NISs$ [10].

This paper enhances the contents in [10], and refers to an implementation of programs. In reality, we give more comprehensive proofs for two propositions, then we refer to the prototype system and the computational complexity.

## 2 Basic Definitions and An Illustrative Example

### 2.1 Basic Definitions

A *Deterministic Information System* ($DIS$) is a quadruplet $(OB, AT, \{VAL_A| A \in AT\}, f)$. Let us consider two sets $CON \subseteq AT$ which we call *condition attributes* and $DEC \subseteq AT$ which we call *decision attributes*. An object $x \in OB$ is *consistent* (with any distinct object $y \in OB$), if $f(x, A)$=$f(y, A)$ for every $A \in CON$ implies $f(x, A)$=$f(y, A)$ for every $A \in DEC$.

A *Non-deterministic Information System* ($NIS$) is also a quadruplet ($OB$, $AT, \{VAL_A|A \in AT\}, g)$, where $g : OB \times AT \rightarrow P(\cup_{A\in AT}VAL_A)$ (a power set of $\cup_{A\in AT}VAL_A$). Every set $g(x, A)$ is interpreted as that there is an actual value in this set but this value is not known. For a $NIS$=$(OB, AT, \{VAL_A|\ A \in AT\}, g)$ and a set $ATR \subseteq AT$, we name a $DIS$=$(OB, ATR, \{VAL_A|A \in ATR\}$, $h)$ satisfying $h(x, A) \in g(x, A)$ a *derived* $DIS$ (*for* $ATR$) *from* $NIS$.

For a set $ATR$=$\{A_1, \cdots, A_n\} \subseteq AT$ and every $x \in OB$, let $PT(x, ATR)$ denote the Cartesian product $g(x, A_1) \times \cdots \times g(x, A_n)$. We name every element a *possible tuple* ($for\ ATR$) *of* $x$. For $\zeta$=$(\zeta_1, \cdots, \zeta_n) \in PT(x, ATR)$, let $[ATR, \zeta]$ denote a formula $\bigwedge_{1\leq i\leq n}[A_i, \zeta_i]$. Let $PI(x, CON, DEC)$ $(x \in OB)$ denote a set $\{[CON, \zeta] \Rightarrow [DEC, \eta]|\zeta \in PT(x, CON), \eta \in PT(x, DEC)\}$. We name an element of $PI(x, CON, DEC)$ a *possible implication* ($from\ CON\ to\ DEC$) *of* $x$. If $PI(x, CON, DEC)$ is a singleton set $\{\tau\}$, we say $\tau$ (from $x$) is $definite$. Otherwise we say $\tau$ (from $x$) is $indefinite$.

### 2.2 An Illustrative Example

Let us consider $NIS_1$ in Table 1. In $NIS_1$, there are four derived $DISs$, which are in Table 2. Let us focus on a possible implication $\tau : [Color, blue] \Rightarrow [Size, big]$ from object 3. This $\tau$ is definite, and it is possible to calculate *support* and *accuracy* values of $\tau$ in every derived $DIS$. In reality, both *support* and *accuracy* values are minimum in the derived $DIS_2$. We name such values *minimum support* ($minsupp(\tau)$) and *minimum accuracy* ($minacc(\tau)$), and $minsupp(\tau)$= 1/3 and $minacc(\tau)$=1/2 hold in $NIS_1$. Both *support* and *accuracy* values are maximum in the derived $DIS_3$. Similarly, $maxsupp(\tau)$=1 and $maxacc(\tau)$=1 hold.

**Table 1.** A table of $NIS_1$

| $OB$ | $Color$ | $Size$ |
|---|---|---|
| 1 | $\{red, blue\}$ | $\{big\}$ |
| 2 | $\{blue\}$ | $\{big, small\}$ |
| 3 | $\{blue\}$ | $\{big\}$ |

**Table 2.** Four derived $DISs$ from $NIS_1$. Tables mean $DIS_1$ to $DIS_4$ sequentially.

| $OB$ | $Color$ | $Size$ |
|---|---|---|
| 1 | $red$ | $big$ |
| 2 | $blue$ | $big$ |
| 3 | $blue$ | $big$ |

| $OB$ | $Color$ | $Size$ |
|---|---|---|
| 1 | $red$ | $big$ |
| 2 | $blue$ | $small$ |
| 3 | $blue$ | $big$ |

| $OB$ | $Color$ | $Size$ |
|---|---|---|
| 1 | $blue$ | $big$ |
| 2 | $blue$ | $big$ |
| 3 | $blue$ | $big$ |

| $OB$ | $Color$ | $Size$ |
|---|---|---|
| 1 | $blue$ | $big$ |
| 2 | $blue$ | $small$ |
| 3 | $blue$ | $big$ |

## 3 Calculation of Minimum Support and Minimum Accuracy for Possible Implications

Let us consider how to calculate $minsupp(\tau)$ and $minacc(\tau)$ for $\tau : [CON, \zeta] \Rightarrow [DEC, \eta]$ from object $x$. Every object $y$, which has descriptors $[CON, \zeta]$ or $[DEC, \eta]$, influences the values $minsupp(\tau)$ and $minacc(\tau)$. Table 3 shows all possible implications with descriptors $[CON, \zeta]$ or $[DEC, \eta]$. For example in CASE 1, we can obtain just $\tau$ from $y$. However in CASE 2, we can obtain two kinds of possible implications (C2.1) and (C2.2), which depend upon the selection of a value in $g(y, DEC)$. This selection specifies some derived $DISs$ from a $NIS$.

**Table 3.** Seven cases of possible implications (related to $[CON, \zeta] \Rightarrow [DEC, \eta]$ from object $x$, $\eta \neq \eta'$, $\zeta \neq \zeta'$) in NISs

| | $Condition : CON$ | $Decision : DEC$ | $Possible\_Implications$ |
|---|---|---|---|
| $CASE1$ | $g(y, CON) = \{\zeta\}$ | $g(y, DEC) = \{\eta\}$ | $[CON, \zeta] \Rightarrow [DEC, \eta](C1.1)$ |
| $CASE2$ | $g(y, CON) = \{\zeta\}$ | $\eta \in g(y, DEC)$ | $[CON, \zeta] \Rightarrow [DEC, \eta](C2.1)$<br>$[CON, \zeta] \Rightarrow [DEC, \eta'](C2.2)$ |
| $CASE3$ | $g(y, CON) = \{\zeta\}$ | $\eta \notin g(y, DEC)$ | $[CON, \zeta] \Rightarrow [DEC, \eta'](C3.1)$ |
| $CASE4$ | $\zeta \in g(y, CON)$ | $g(y, DEC) = \{\eta\}$ | $[CON, \zeta] \Rightarrow [DEC, \eta](C4.1)$<br>$[CON, \zeta'] \Rightarrow [DEC, \eta](C4.2)$ |
| $CASE5$ | $\zeta \in g(y, CON)$ | $\eta \in g(y, DEC)$ | $[CON, \zeta] \Rightarrow [DEC, \eta](C5.1)$<br>$[CON, \zeta] \Rightarrow [DEC, \eta'](C5.2)$<br>$[CON, \zeta'] \Rightarrow [DEC, \eta](C5.3)$<br>$[CON, \zeta'] \Rightarrow [DEC, \eta'](C5.4)$ |
| $CASE6$ | $\zeta \in g(y, CON)$ | $\eta \notin g(y, DEC)$ | $[CON, \zeta] \Rightarrow [DEC, \eta'](C6.1)$<br>$[CON, \zeta'] \Rightarrow [DEC, \eta'](C6.2)$ |
| $CASE7$ | $\zeta \notin g(y, CON)$ | $Any$ | $[CON, \zeta'] \Rightarrow Decision(C7.1)$ |

**Definition 1.** For every descriptor $[A, \zeta]$ ($A \in AT$, $\zeta \in VAL_A$) and every $[ATR, val](=[\{A_1, \cdots, A_k\}, (\zeta_1, \cdots, \zeta_k)])$ in a $NIS$, we define the following.
(1) $Descinf([A, \zeta])=\{x \in OB|\ g(x, A)=\{\zeta\}\}$.
(2) $Descinf([ATR, val])=Descinf(\wedge_i[A_i, \zeta_i])=\cap_i Descinf([A_i, \zeta_i])$.
(3) $Descsup([A, \zeta])=\{x \in OB|\ \zeta \in g(x, A)\}$.
(4) $Descsup([ATR, val])=Descsup(\wedge_i[A_i, \zeta_i])=\cap_i Descsup([A_i, \zeta_i])$.

Clearly, $Descinf([CON, \zeta])$ is a set of objects belonging to either CASE 1, 2 or 3 in Table 3, and $Descsup([CON, \zeta])$ is a set belonging to either CASE 1 to CASE 6. $Descsup([CON, \zeta]) - Descinf([CON, \zeta])$ is a set belonging to either CASE 4, 5 or 6.

**Proposition 1.** Let us employ possible implications (C2.2), (C4.2), either (C5.2), (C5.3) or (C5.4) in Table 3. In derived $DISs$ with this selection, the *support* value of $\tau : [CON, \zeta] \Rightarrow [DEC, \eta]$ from $x$ is minimum. If $\tau$ is definite, namely $\tau$ belongs to CASE 1,

$minsupp(\tau)=|Descinf([CON, \zeta]) \cap Descinf([DEC, \eta])|/|OB|$.

If $\tau$ is indefinite, namely $\tau$ does not belong to CASE 1,

$minsupp(\tau)=(|Descinf([CON, \zeta]) \cap Descinf([DEC, \eta])| + 1)/|OB|$.

**Proof.** This selection of attribute values in a $NIS$ excludes every $[CON, \zeta] \Rightarrow [DEC, \eta]$ from object $y$ except CASE 1. In reality, we remove (C2.1), (C4.1) and (C5.1) from Table 3. Therefore, the *support* value of $\tau$ is minimum in a derived $DIS$ with such selection of attribute values. If $\tau$ is definite, object $x$ is in a set $Descinf([CON, \zeta]) \cap Descinf([DEC, \eta])$. Otherwise, $\tau$ belongs to either (C2.1), (C4.1) or (C5.1). It is necessary to obtain one $\tau$ from either (C2.1), (C4.1) or (C5.1), thus it is necessary to add 1 to the numerator.

**Proposition 2.** Let us employ possible implications (C2.2), (C4.2), (C5.2), (C6.1) in Table 3. In derived $DISs$ with this selection, the *accuracy* value of $\tau : [CON, \zeta] \Rightarrow [DEC, \eta]$ from $x$ is minimum. Let $OUTACC$ be $[Descsup([CON, \zeta]) - Descinf([CON, \zeta])] - Descinf([DEC, \eta])$. If $\tau$ is definite, namely $\tau$ belongs to CASE 1,

$minacc(\tau)=\frac{|Descinf([CON,\zeta]) \cap Descinf([DEC,\eta])|}{|Descinf([CON,\zeta])|+|OUTACC|}$.

If $\tau$ is indefinite, namely $\tau$ does not belong to CASE 1,

$minacc(\tau)=\frac{|Descinf([CON,\zeta]) \cap Descinf([DEC,\eta])|+1}{|Descinf([CON,\zeta]) \cup \{x\}|+|OUTACC-\{x\}|}$.

**Proof.** Since $m/n \leq (m + k)/(n + k)$ ($0 \leq m \leq n$, $n \neq 0$, $k > 0$) holds, we excludes every $[CON, \zeta] \Rightarrow [DEC, \eta]$ from object $y$ except CASE 1, and we include possible implications $[CON, \zeta] \Rightarrow [DEC, \eta']$, which increase the denominator. The *accuracy* value of $\tau$ is minimum in a derived $DIS$ with such selection of attribute values. The set $OUTACC$ defines objects in either CASE 5 or CASE 6. As for CASE 4 and CASE 7, we can omit them for $minacc(\tau)$. If $\tau$ is definite, namely $\tau$ belongs to CASE 1, the numerator is $|Descinf([CON, \zeta]) \cap Descinf([DEC, \eta])|$ and the denominator is $|Descinf([CON, \zeta])|+|OUTACC|$. If $\tau$ is indefinite, $\tau$ belongs to either (C2.1), (C4.1) or (C5.1). In every cases,

the denominator is $|Descinf([CON,\zeta]) \cup \{x\}| + |OUTACC - \{x\}|$, and the numerator is $|Descinf([CON,\zeta]) \cap Descinf([DEC,\eta])|+1$.

**Theorem 3.** For a $NIS$, let us consider a possible implication $\tau$:$[CON,\zeta] \Rightarrow [DEC,\eta] \in PI(x,CON,DEC)$. Let $M$=$\{\psi|\psi$ is a derived $DIS$ from $NIS$, and $support(\tau)$ is minimum in $\psi\}$. Then, $accuracy(\tau)$ is minimum in some $\psi \in M$.

**Proof.** The selection of possible implications in Proposition 2 is a special case of the selection of possible implications in Proposition 1.

## 4 A Definition of Rule Generation in NISs

The following problem was solved by employing Apriori algorithm in [9].

**Problem 1. [9]** In every standard table or every $DIS$, find every implication $\tau$, whose $accuracy(\tau)$ is maximum under the condition $support(\tau) \geq \alpha$ for a fixed value $\alpha$ $(0 < \alpha \leq 1)$.

This paper extends Problem 1 to Problem 2 in $NISs$.

**Problem 2. (Rule generation based on Min-Max Strategy).** In every $NIS$, find every possible implication $\tau$, whose $minacc(\tau)$ is maximum under the condition $minsupp(\tau) \geq \alpha$ for a fixed value $\alpha$ $(0 < \alpha \leq 1)$.

According to Theorem 3, there exist at least one $\psi \in \{\psi|\psi$ is a derived $DIS$ from $NIS$, and $support(\tau)$ is minimum in $\psi\}$, which makes $minacc(\tau)$ minimum. Therefore, Problem 2 is well-defined. Intuitively, Problem 2 specifies rules in the worst condition.

Generally, $\tau$ depends upon $\prod_{x \in OB, A \in AT} |g(x,A)|$ number of derived $DISs$ and condition attributes $CON$ $(CON \in 2^{AT-DEC})$. Therefore, it will be hard to pick up every possible implication sequentially. For solving this computational issue, we focus on descriptors $[A,\zeta]$ $(A \in AT,\ \zeta \in VAL_A)$. The number of all descriptors is usually much smaller than the number of all possible implications.

## 5 A Real Execution by Implemented Programs in NISs

This section gives real execution of rule generation in $NIS_2$, and we show the overview of the rule generation.

```
% ./nis_apriori
CAN(1)={[1,2],[1,4],[2,1],[2,3],[3,3],[3,4],[3,5],[4,3],[4,5],
   [5,2],[5,5],[6,5],[7,2],[7,4],[8,1],[8,3]} (16)
CAN(2)={[8,1][1,4](0.600),[8,1][4,5](0.750),[8,1][5,2](0.750)} (3)
CAN(3)={[8,1][1,4][4,5](0.750),[8,1][4,5][5,2](1.000)} (2)
   [4,5]&[5,2]=>[8,1](minsupp=0.300,minacc=1.000)(INDEF)(from 5)
EXEC_TIME=0.000(sec)
```

The above is the real rule generation(a set of decision attribute:$\{H\}$, threshold value $\alpha$=0.3) by *nis_apriori* command. In reality, we obtained a possible

**Table 4.** A Table of $NIS_2$, which we generated by using a random number program

| $OB$ | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ | $G$ | $H$ |
|---|---|---|---|---|---|---|---|---|
| 1 | $\{3\}$ | $\{1,3,4\}$ | $\{3\}$ | $\{2\}$ | $\{5\}$ | $\{5\}$ | $\{2,4\}$ | $\{3\}$ |
| 2 | $\{2\}$ | $\{3,4\}$ | $\{1,3,4\}$ | $\{4\}$ | $\{1,2\}$ | $\{2,4,5\}$ | $\{2\}$ | $\{2\}$ |
| 3 | $\{4,5\}$ | $\{5\}$ | $\{1,5\}$ | $\{5\}$ | $\{2\}$ | $\{5\}$ | $\{1,2,5\}$ | $\{1\}$ |
| 4 | $\{1\}$ | $\{3\}$ | $\{4\}$ | $\{3\}$ | $\{1,2,3\}$ | $\{1\}$ | $\{2,5\}$ | $\{1,2\}$ |
| 5 | $\{4\}$ | $\{1\}$ | $\{2,3,5\}$ | $\{5\}$ | $\{2,3,4\}$ | $\{1,5\}$ | $\{4\}$ | $\{1\}$ |
| 6 | $\{4\}$ | $\{1\}$ | $\{5\}$ | $\{1\}$ | $\{4\}$ | $\{2,4,5\}$ | $\{2\}$ | $\{1,2,3\}$ |
| 7 | $\{2\}$ | $\{4\}$ | $\{3\}$ | $\{4\}$ | $\{3\}$ | $\{2,4,5\}$ | $\{4\}$ | $\{1,2,3\}$ |
| 8 | $\{4\}$ | $\{5\}$ | $\{4\}$ | $\{2,3,5\}$ | $\{5\}$ | $\{3\}$ | $\{1,2,3\}$ | $\{1,2,3\}$ |
| 9 | $\{2\}$ | $\{3\}$ | $\{5\}$ | $\{3\}$ | $\{1,3,5\}$ | $\{4\}$ | $\{2\}$ | $\{3\}$ |
| 10 | $\{4\}$ | $\{2\}$ | $\{1\}$ | $\{5\}$ | $\{2\}$ | $\{4,5\}$ | $\{3\}$ | $\{1\}$ |

implication $[D,5] \wedge [E,2] \Rightarrow [H,1]$ ($minsupp$=0.300,$minacc$=1.000) from object 5. We identify every attribute with its ordinal number in the programs. We show each procedure in every step.

**An Overview of Apriori Algorithm in $NISs$**

**(STEP 1: Analysis of the condition)**
Since $\alpha$=0.3, an implication $\tau$ must occur more than 3 ($=|OB| \times 0.3$) times in $NIS_2$. According to Proposition 1, if $\tau : [CON,\zeta] \Rightarrow [DEC,\eta]$ is definite, $|Descinf([CON,\zeta]) \cap Descinf([DEC,\eta])| \geq 3$. If $\tau$ is indefinite, $|Descinf([CON,\zeta]) \cap Descinf([DEC,\eta])| \geq 2$.

**(STEP 2: Generation of Meaningful descriptors, CAN(1))**
For every $Descinf([A,\zeta])$ and $Descsup([A,\zeta])$, we pick up descriptors satisfying either (1) or (2) in the following;
(1) $|Descinf([A,\zeta])| \geq 3$,
(2) $|Descinf([A,\zeta])| \geq 2$ and $Descsup([A,\zeta]) - Descinf([A,\zeta]) \neq \emptyset$.
In $NIS_2$, there are 38 descriptors, and 16 descriptors are picked up ($CAN(1)$ in the real execution). As for a descriptor $[1,3]$, $Descsup([1,3])$=$Descinf([1,3])$ $=\{1\}$ holds. Therefore, $\tau$ with $[1,3]$ does not occur more than 3 times, and we omit such descriptors.

**(STEP 3: Combinations of Meaningful descriptors, CAN(2))**
Since condition attributes are from $A$ to $G$ and the decision attribute is $H$, we examine $Descinf([CON,val_{CON}] \wedge [H,val_H])$ and $Descsup([CON,val_{CON}] \wedge [H,val_H])$ ($CON \in \{A,B,C,D,E,F,G\}$). We first pick up the meaningful combinations $[CON,val_{CON}] \wedge [H,val_H]$ of descriptors in $CAN(1)$ according to the same way in STEP 2. Then, we apply Proposition 2 to the meaningful combinations, and we obtain $minacc(\tau')$ ($\tau' : [CON,val_{CON}] \Rightarrow [H,val_H]$). The value of $minacc(\tau')$ is also displayed in $CAN(2)$ in the real execution.

**(STEP 4: Recursive Steps for Meaningful descriptors, CAN(3))**
In order to increase *minacc*, we recursively employ STEP 3. Since $minacc([8,1] \wedge [4,5] \wedge [5,2])=1$, we know that an implication $[D,5] \wedge [E,2] \Rightarrow [H,1]$ from object 5 satisfies the condition for the rule generation.

**(STEP 5: Closing the Steps)**
The number of applying recursive steps is less than the number of condition attributes, becasue just a descriptor is employed for $\tau$ in every attribute. Therefore, the above steps certainly terminate. In the real execution, $CAN(4)=\emptyset$ is derived.

The following is other real execution (a set of decision attribute:$\{H\}$, threshold value $\alpha$=0.2).

```
% ./nis_apriori
CAN(1)={[1,2],[1,4],[2,1],[2,3],[2,4],[2,5],[3,1],[3,3],[3,4],[3,5],
   [4,2],[4,3],[4,4],[4,5],[5,2],[5,3],[5,4],[5,5],[6,1],[6,4],[6,5],
   [7,2],[7,3],[7,4],[8,1],[8,2],[8,3]} (27)
CAN(2)={[8,1][1,4](0.500),[8,1][2,1](0.667),[8,1][2,5](1.000),
   [8,1][3,1](0.667),[8,1][4,5](0.750),[8,1][5,2](0.500),[8,1][6,5](0.333),
   [8,1][7,3](1.000),[8,1][7,4](0.667),[8,2][1,2](0.667),[8,2][4,4](1.000),
                  :                    :                   :
   [5,5]=>[8,3](minsupp=0.200,minacc=1.000)(INDEF)(from 8,9)
CAN(3)={[8,1][1,4][2,1](1.000),[8,1][1,4][3,1](1.000),[8,1][1,4][4,5](0.667),
   [8,1][1,4][5,2](0.667),[8,1][3,1][4,5](1.000),[8,1][3,1][5,2](0.667),
   [8,1][4,5][5,2](0.667),[8,1][4,5][6,5](0.667),[8,1][5,2][6,5](0.500),
   [8,3][1,2][6,4](0.667),[8,3][2,3][7,2](0.500),[8,3][3,3][6,5](0.500),
   [8,3][3,5][6,4](1.000),[8,3][3,5][7,2](0.667),[8,3][4,3][7,2](0.667),
   [8,3][6,4][7,2](0.667)} (16)
   [1,4]&[2,1]=>[8,1](minsupp=0.200,minacc=1.000)(INDEF)(from 6)
   [1,4]&[3,1]=>[8,1](minsupp=0.200,minacc=1.000)(INDEF)(from 3)
   [3,1]&[4,5]=>[8,1](minsupp=0.200,minacc=1.000)(INDEF)(from 3)
   [3,5]&[6,4]=>[8,3](minsupp=0.200,minacc=1.000)(INDEF)(from 6)
CAN(4)={[8,1][1,4][4,5][5,2](0.667),[8,1][4,5][5,2][6,5](0.667)} (2)
EXEC_TIME=0.016(sec)
```

## 6 Computational Issues on Apriori Algorithm in NISs

Let us consider some computational issues for every STEP 2 to STEP 4.

**(STEP 2)**
We first prepare two arrays $Descinf_{A,val}[i]$ and $Descsup_{A,val}[i]$ for every $val \in VAL_A$ ($A \in AT$). For every object in $OB$, we apply (A) or (B) in the following;
(A) If $|g(x,A)|=\{val\}$, we assign $x$ to $Descinf_{A,val}[i]$.
(B) If $val \in g(x,A)$, we assign $x$ to $Descsup_{A,val}[i]$.
Then, we examine conditions (1) and (2) in STEP 2. We include every descriptor satisfying (1) or (2) into $CAN(1)$. For every $A \in AT$, this procedure is applied,

and we can obtain meaningful descriptors with $Descinf$ and $Descsup$ information. The complexity depends upon $|OB| \times |AT|$.

**(STEP 3)**
Let us suppose $CAN(1)$ be $\cup_{A \in AT}\{[A, val_A] | A \in AT\} = \cup_{A \in AT} CAN(1, A)$. For a decision attribute $DEC \in AT$, we produce $[A, val_A] \wedge [DEC, val_{DEC}]$ ($[A, val_A] \in CAN(1)$, $[DEC, val_{DEC}] \in CAN(1)$). The number of such combinations is $\sum_{A \in AT - \{DEC\}} |CAN(1, A)| \times |CAN(1, DEC)|$ ($*1$). For every combination, we examine conditions (1) and (2) in STEP 2. Since $Descinf([A, val_A] \wedge [DEC, val_{DEC}]) = Descinf([A, val_A]) \cap Descinf([DEC, val_{DEC}])$ holds, and both $Descinf([A, val_A])$ and $Descinf([DEC, val_{DEC}])$ are obtained in STEP2, it is possible to obtain $Descinf([A, val_A] \wedge [DEC, val_{DEC}])$ by checking $|Descinf([A, val_A])| \times |Descinf([DEC, val_{DEC}])|$ ($*2$) cases. As for $Descsup([A, val_A] \wedge [DEC, val_{DEC}])$, we also check $|Descsup([A, val_A])| \times |Descsup([DEC, val_{DEC}])|$ ($*3$) cases. In this way, we generate $CAN(2)$, and the complexity to generate $CAN(2)$ depends upon $(*1) \times ((*2) + (*3))$. According to Proposition 2, we can easily calculate $minacc([A, val_A] \Rightarrow [DEC, val_{DEC}])$ by using $Descinf$ and $Descsup$.

**(STEP 4)**
In STEP 4, we recursively employ STEP 3. Let us suppose $CAN(n) = \{[CON, \zeta] \wedge [DEC, val_{DEC}]\}$. For every element in $CAN(n)$, we generate a new combination $[A, val_A] \wedge [CON, \zeta] \wedge [DEC, val_{DEC}]$ ($[A, val_A] \in CAN(1), A \notin CON$), and we examine conditions (1) and (2) in STEP 2. By repeating this procedure, we generate $CAN(n + 1)$.

In every step, the number of derived $DISs$ does not appear. Therefore, this algorithm does not depend upon the number of derived $DISs$. The most time-consuming part is to generate combinations of descriptors. In our algorithms, two sets $Descinf$ and $Descsup$ are employed, and $minsupp()$ and $minacc()$ are calculated by using $Descinf$ and $Descsup$. Other part is almost the same as the Apriori algorithm [9]. Therefore, the complexity of this algorithm is almost the same as the original Apriori algorithm.

## 7 Concluding Remarks

We proposed a framework of Apriori based rule generation in $NISs$, and clarified some theoretical properties. This algorithm does not depend upon the number of derived $DISs$ from a $NIS$. We also gave comprehensive proofs for Proposition 1, 2 and Theorem 3. The program *nis_apriori* is realized on Windows PC with Pentium 4 (3.4GHz), and consists of about 1300 lines in C.

**Acknowledgment.** The authors would be grateful to anonymous referees for their useful comments. This work is partly supported by the Grant-in-Aid for Scientific Research (C) (No.18500214), Japan Society for the Promotion of Science.

## References

1. Z.Pawlak: Rough Sets - Theoretical Aspects of Reasoning about Data, Kluwer Academic Publisher, 1991.
2. J.Komorowski, Z.Pawlak, L.Polkowski and A.Skowron: Rough Sets: a tutorial, Rough Fuzzy Hybridization, Springer, pp.3-98, 1999.
3. W.Lipski: On Semantic Issues Connected with Incomplete Information Data Base, ACM Trans. DBS, Vol.4, pp.269-296, 1979.
4. E.Orłowska: What You Always Wanted to Know about Rough Sets, Incomplete Information: Rough Set Analysis, Physica-Verlag, pp.1-20, 1998.
5. J.Grzymala-Busse, P.Werbrouck: On the Best Search Method in the LEM1 and LEM2 Algorithms, Incomplete Information: Rough Set Analysis, Physica-Verlag, pp.75-91, 1998.
6. J.Stefanowski and A.Tsoukias: On the Extension of Rough Sets under Incomplete Information, Lecture Notes in AI (RSFDGrC99), Vol.1711, pp.73-81, 1999.
7. M.Kryszkiewicz: Rules in Incomplete Information Systems, Information Sciences, Vol.113, pp.271-292, 1999.
8. H.Sakai and A.Okuma: Basic Algorithms and Tools for Rough Non-deterministic Information Analysis, Transactions on Rough Sets, Vol.1, pp.209-231, 2004.
9. R.Agrawal and R.Srikant: Fast Algorithms for Mining Association Rules, Proc. 20th Very Large Data Base, pp.487-499, 1994.
10. H.Sakai and M.Nakata: On Possible Rules and Apriori Algorithm in Non-deterministic Information Systems, Lecture Notes in AI (RSCTC2006), Vol.4259, pp.264-273, 2006.

# Neonatal Infection Diagnosis Using Constructive Induction in Data Mining

Jerzy W. Grzymala-Busse[1], Zdzislaw S. Hippe[2], Agnieszka Kordek[3], Teresa Mroczek[2], and Wojciech Podraza[4]

[1] Department of Electrical Engineering and Computer Science,
University of Kansas, Lawrence, KS 66045, USA
and
Institute of Computer Science,
Polish Academy of Sciences, 01-237 Warsaw, Poland
jerzy@ku.edu
http://lightning.eecs.ku.edu/index.html
[2] Department of Expert Systems and Artificial Intelligence
University of Information Technology and Management, 35-225 Rzeszow, Poland
{zhippe,tmroczek}@wenus.wsiz.rzeszow.pl
[3] Department of Obstetrics and Perinatology
Pomeranian Medical University, 70-111 Szczecin, Poland
agkordek@sci.pam.szczecin.pl
[4] Department of Medical Physics
Pomeranian Medical University, 70-111 Szczecin, Poland
podrazaw@sci.pam.szczecin.pl

**Abstract.** This paper presents the results of our experiments on a data set describing neonatal infection. We used two main tools: the MLEM2 algorithm of rule induction and BeliefSEEKER system for generation of Bayesian nets and rule sets. Both systems are based on rough set theory. Our main objective was to compare the quality of diagnosis of cases from two testing data sets: with an additional attribute called PCT and without this attribute. The PCT attribute was computed using constructive induction. The best results were associated with the rule set induced by the MLEM2 algorithm and testing data set enhanced by constructive induction.

## 1 Introduction

Severe intrauterine bacterial infection in neonates remains a major diagnostic problem because of non-specific clinical signs and low sensitivity of the routine diagnostic tests such as white blood cell (WBC) count, absolute neutrophil count, thrombocyte count, C-reactive protein (CRP) level, blood cultures and chest X-rays. These tests do not enable the diagnosis decisively. Furthermore, blood cultures (the gold standard for diagnosis) are often negative due to low blood volumes drawn and single cultures as well as the prenatal management of antibiotics [19].

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 289–296, 2007.

The diagnosis of intrauterine bacterial infection plays an important role in prompt introduction of antibiotic on one hand and avoiding an overtreatment and toxicity on the other [12,18].

A prospective study was conducted in 2000–2001 in the Department of Obstetrics and Perinatology, Pomeranian Medical University, Szczecin, Poland. A total 187 newborns participated in the study. The study protocol included an evaluation of the level of procalcitonin (PCT), a propeptide of the hormone calcitonin. PCT is a novel marker of the inflammatory response to infection [13,15]. A neonatal infection was diagnosed on the basis of three or more of the following categories of clinical signs: respiratory, cardiac, neurological, circulatory, systemic, and gastrointestinal [20]. The final diagnosis was frequently delayed because of delayed appearance of clinical symptoms. It is very important to find a method that will allow diagnosis in a short period of time after birth. Rough set theory and/or other mathematical tools may assist in this goal.

Two groups of patients have been identified: I - non-infected (n = 155) including full-term (n = 117) and preterm (n = 38) neonates and II - infected (n = 32) including full-term (n = 8) and preterm (n = 24) ones. Each newborn was characterized by the following 13 attributes:

- procalcitonin (PCT) concentration,
- premature rupture of membranes (PROM),
- way of delivery (WoD), i.e., Caesarian section or natural,
- signs of mother's infection,
- amniotic fluid color,
- Apgar score,
- white blood cell count (WBC),
- presence of respiratory distress syndrome (RDS),
- gender,
- gestational age (Hbd),
- birth weight,
- smoking during pregnancy,
- C-reactive protein (CRP) concentration

The decision was a presence of bacterial infection (group II) or not (group I). The above set of 187 cases was a training set. Additionally, 30 patients born in 2005 and 2006 were included as a testing set. There were 12 infected and 18 non-infected newborns. All of them were characterized by the same attributes except the PCT concentration, which is not a routine test in the neonatal ward of the Pomeranian Medical University.

Our main objective was to study how a testing data set with an additional attribute called PCT compares with the same data set without PCT. This additional attribute was computed for every testing case by rules induced from the training data set, using a technique called constructive induction.

## 2 MLEM2

One of the data mining algorithms used for our experiments was the MLEM2 (Modified Learning from Examples Module, version 2) rule induction module of

the LERS (Learning from Examples based on Rough Sets) data mining system, [2,3,4,5].

In the first step of processing the input data file, the data mining system LERS checks if the input data file is *consistent* (i.e., if the file does not contain conflicting examples). If the input data file is inconsistent, LERS computes lower and upper approximations [16,17] of all concepts. Rules induced from the lower approximation of the concept *certainly* describe the concept, so they are called *certain* [2]. On the other hand, rules induced from the upper approximation of the concept describe the concept only *possibly* (or *plausibly*), so they are called *possible* [2].

The MLEM2 algorithm is based on its predecessor called LEM2 (Learning from Examples Module, version 2). LEM2 learns the smallest set of minimal rules, describing the concept. LEM2 explores the search space of attribute-value pairs. Its input data file is a lower or upper approximation of a concept, so its input data file is always consistent. In general, LEM2 computes a local covering and then converts it into a rule set. We will quote a few definitions to describe main ideas of the LEM2 algorithm.

The main notion of the LEM2 algorithm is an attribute-value pair block. For an attribute-value pair $(a,\ v) = t$, a *block* of $t$, denoted by $[t]$, is a set of all cases from $U$ such that for attribute $a$ have value $v$. For a set $T$ of attribute-value pairs, the intersection of blocks for all $t$ from $T$ will be denoted by $[T]$. Let $B$ be a nonempty lower or upper approximation of a concept represented by a decision-value pair $(d, w)$. Set $B$ *depends* on a set $T$ of attribute-value pairs $t = (a, v)$ if and only if

$$\emptyset \neq [T] = \bigcap_{t \in T} [t] \subseteq B.$$

Set $T$ is a *minimal complex* of $B$ if and only if $B$ depends on $T$ and no proper subset $T'$ of $T$ exists such that $B$ depends on $T'$. Let $\mathcal{T}$ be a nonempty collection of nonempty sets of attribute-value pairs. Then $\mathcal{T}$ is a *local covering* of B if and only if the following conditions are satisfied:

(1) each member $T$ of $\mathcal{T}$ is a minimal complex of $B$,

(2) $\bigcup_{t \in \mathcal{T}} [T] = B$, and

(3) $\mathcal{T}$ is minimal, i.e., $\mathcal{T}$ has the smallest possible number of members.

In selection for an attribute-value pair $t$, a future rule condition, the LEM2 algorithm provides the highest priority to $t$ that is the most relevant to a goal $G$, $G$ being initially equal to $B$. If a tie occurs, LEM2 selects an attribute-value pair $t$ with the smallest cardinality of $[t]$. For details of the LEM2 algorithm see, e.g., [3,4].

MLEM2 is a modified version of the algorithm LEM2. The original algorithm LEM2 needs discretization, a preprocessing, to deal with numerical attributes. MLEM2 recognizes integer and real numbers as values of attributes, and labels such attributes as numerical. For numerical attributes MLEM2 computes blocks in a different way than for symbolic attributes. First, it sorts all values of a numerical attribute. Then it computes cutpoints as averages for any two

consecutive values of the sorted list. For each cutpoint $c$ MLEM2 creates two blocks, the first block contains all cases for which values of the numerical attribute are smaller than $c$, the second block contains remaining cases, i.e., all cases for which values of the numerical attribute are larger than $c$. The search space of MLEM2 is the set of all blocks computed this way, together with blocks defined by symbolic attributes. Then MLEM2 combines attribute-value pairs relevant to a concept and creates rules describing the concept. In addition, MLEM2 handles missing attribute values during rule induction [5]. The previous version of MLEM2, LEM2, induced certain rules from incomplete decision tables with missing attribute values interpreted as lost. Recently, MLEM2 was further extended to induce both certain and possible rules from a decision table with some missing attribute values being lost and some missing attribute values being "do not care" conditions, while attributes may be numerical.

## 3 Classification System

The classification system of LERS is a modification of the *bucket brigade algorithm* [1,10]. The decision to which concept a case belongs is made on the basis of three factors: strength, specificity, and support. They are defined as follows: *strength* is the total number of cases correctly classified by the rule during training. *Specificity* is the total number of attribute-value pairs on the left-hand side of the rule. The matching rules with a larger number of attribute-value pairs are considered more specific. The third factor, *support*, is defined as the sum of scores of all matching rules from the concept, where the score of the rule is the product of its strength and specificity. The concept $C$ for which the support, i.e., the following expression

$$\sum_{matching\ rules\ R\ describing\ C} Strength(R) * Specificity(R)$$

is the largest is the winner and the case is classified as being a member of $C$.

In the classification system of LERS, if complete matching is impossible, all partially matching rules are identified. These are rules with at least one attribute-value pair matching the corresponding attribute-value pair of a case. For any partially matching rule $R$, the additional factor, called $Matching\_factor\ (R)$, is computed. Matching_factor $(R)$ is defined as the ratio of the number of matched attribute-value pairs of $R$ with a case to the total number of attribute-value pairs of $R$. In partial matching, the concept $C$ for which the following expression is the largest

$$\sum_{\substack{partially\ matching \\ rules\ R\ describing\ C}} Matching\_factor(R) * Strength(R) * Specificity(R)$$

is the winner and the case is classified as being a member of $C$.

Every rule induced by LERS is preceded by three numbers: specificity, strength, and the rule domain size (the total number of training cases matching the left-hand side of the rule).

## 4 BeliefSEEKER

In our experiments on the neonatal data set we used also a system called BeliefSEEKER. This system, also based on rough set theory, generates belief networks and rule sets. Initially, BeliefSEEKER, similarly to LERS, computes lower and upper approximations for every concept from the original data set. If the original data set is consistent, the lower approximation of every concept is equal to the upper approximation of the same concept, and BeliefSEEKER generates *certain belief networks*. When the original data set is inconsistent, BeliefSEEKER uses the upper approximations for every concept to generate *possible belief networks*. In the generation of belief networks, BeliefSEEKER uses the Dirichlet process model [8,11], with a scaling parameter $\alpha > 0$. BeliefSEEKER outputs an optimal belief network for any value of the parameter $\alpha$.

BeliefSEEKER generates not only belief networks but also rule sets. A threshold called *certainty factor*, denoted by *CF*, is used in BeliefSEEKER to produce rules with prescribed certainty. The lower CF, the more rules are induced.

Additionally, BeliefSEEKER is equipped with its own classification scheme used for classification of unseen cases as well as validation. For more details on BeliefSEEKER, see, e.g., [7,9,14,21].

## 5 Experiments

First we determined the significance of all 13 attributes using a typical rough-set setup. The original training data set, with all 13 attributes, was inconsistent, with 10 conflicting cases. To determine attribute significance, every attribute, one attribute at a time, was removed from the data set and the number of conflicting cases for a data set with 12 remaining attributes was recorded. The larger number of conflicting cases caused by an attribute removal the greater significance of the attribute. As follows from Table 1, the most significant attribute is PCT.

**Table 1.** Significance of attributes

| Attributes | Number of conflicting cases | Attributes | Number of conflicting cases |
|---|---|---|---|
| All | 10 | All but *WBC* | 11 |
| All but *PCT* | 32 | All but *RDS* | 10 |
| All but *PROM* | 14 | All but *Gender* | 24 |
| All but *WoD* | 20 | All but *Hbd* | 29 |
| All but *Mother's infection* | 10 | All but *Birth weight* | 11 |
| All but *Amniotic fluid color* | 25 | All but *Smoking* | 10 |
| All but *Agpar score* | 10 | All but *CRP* | 12 |

Then rule sets were induced, using MLEM2 and BeliefSEEKER, from two data sets: the original training data set and a data set with all attributes but

PCT. These rule sets were used for classification of 30 cases from the testing data set. Results of our experiments are presented in Tables 2 and 3. Note that the original data set was imbalanced (32 cases of infected neonates and 155 cases of non-infected). Therefore we used a standard technique of changing rule strength [6], i.e., multiplying the rule strength for every rule describing infected neonates (the smaller class) by some number, in our case this number was equal to ten. Results are presented in Tables 2 and 3 as well.

**Table 2.** Results of experiments, testing data without PCT

| | Sensitivity | Specificity | Accuracy |
|---|---|---|---|
| MLEM2 | 58.3% | 44.4% | 50% |
| MLEM2 with strength multiplier | 50% | 33.3% | 56.7% |
| BeliefSEEKER | 100% | 16.7% | 50% |

**Table 3.** Results of experiments, testing data with PCT

| | Sensitivity | Specificity | Accuracy |
|---|---|---|---|
| MLEM2 | 50% | 77.8% | 66.7% |
| MLEM2 with strength multiplier | 100% | 50% | 70% |
| BeliefSEEKER | 75% | 11.1% | 36.7% |

Note that testing cases were characterized by only 12 attributes. Values of the PCT attribute for these 30 cases were not recorded at all. We computed values of PCT for testing cases using an additional, third rule set, induced by MLEM2 from the original training data set by removing the original decision and the inducing a rule set for PCT from a training data set with the remaining 12 attributes. This technique is called *constructive induction*. Then the original testing data set, with 12 attributes, was enhanced by adding the thirteenth attribute (PCT).

Additionally, results of ten-fold cross validation for the training data set and LEM2 algorithm were conducted. Results are: an error rate equal to 14.44% for the data set with PCT and an error rate equal to 16.04% for the data set without PCT.

Induced rules were analyzed by domain experts. Most of the rules were consistent with diagnosticians' expertise, but some rules were surprising. An example of such unexpected rule, in which–relatively–many patients were properly classified by quite general attributes is

3, 9, 9
(Weight, 1..3.5) & (WoD, 1.5..2) & (PROM, 0..1.5) -> (Decision, 1),

with the following interpretation: if a neonate's weight is smaller than 2500 g, there was a sudden Caesarian section and PROM did not place, or took place not more than twelve hours before birth, then the neonate is infected. Note that this rule's specificity is 3, its strength is 9, and this rule matches 9 cases from the data set.

## 6 Conclusions

As follows from Table 1, the attribute PCT is the most significant attribute in our training data set. The best results (100% sensitivity and the best accuracy = 70%) was accomplished using constructive induction, i.e., by adding to the original testing data set a new attribute computed from remaining 12 attributes. Differences in experimental results between MLEM2 and BeliefSEEKER are negligible, both systems may reach the level of 100% sensitivity. MLEM2 better works for data with PCT, on the other hand, BeliefSEEKER was able to produce 100% sensitivity for data without PCT.

Finally, due to our results, diagnostician discovered some unexpected regularities, in the form of rules, hidden in the original training data set.

## References

1. Booker, L. B., Goldberg, D. E. Holland, J. F.: Classifier systems and genetic algorithms. In *Machine Learning. Paradigms and Methods.* Carbonell, J. G. (ed.), The MIT Press, Menlo Park, CA, 235–282, (1990).
2. Grzymala-Busse, J. W.: Knowledge acquisition under uncertainty—A rough set approach. *Journal of Intelligent & Robotic Systems* **1** (1988), 3–16.
3. Grzymala-Busse, J. W.: LERS—A system for learning from examples based on rough sets. In *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory*, Slowinski, R., (ed.), Kluwer Academic Publishers, Dordrecht, Boston, London (1992) 3–18.
4. Grzymala-Busse, J. W.: A new version of the rule induction system LERS. *Fundamenta Informaticae* **31** (1997) 27–39.
5. Grzymala-Busse, J. W.: MLEM2: A new algorithm for rule induction from imperfect data. Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002, Annecy, France, July 1–5, 2002, 243–250.
6. Grzymala-Busse, J. W., Goodwin, L. K., Grzymala-Busse W. J., Zheng X.: An approach to imbalanced data sets based on changing rule strength. Learning from Imbalanced Data Sets, AAAI Workshop at the 17th Conference on AI, AAAI-2000, Austin, TX, July 30–31, 2000, 69–74.
7. Grzymala-Busse, J. W., Hippe, Z. S., Mroczek, T.: Belief rules vs. decision rules: A preliminary appraisal of the problem. Proceedings of the IIPWM'2005, International Conference on Intelligent Information Processing and WEB Mining Systems, Springer-Verlag, Gdansk, Poland, June 13–16, 2005, 431–435.

8. Heckerman, D.: A tutorial on learning Bayesian networks, Microsoft Corporation, Technical Report MSR-TR-95-06, 1996.
9. Hippe, Z. S., Mroczek, T.: Melanoma classification and prediction using belief networks. In: Kurzynski, M., Puchala, E., Wozniak, M. (eds.), Computer Recognition Systems, Wroclaw University of Technology Press, Wroclaw 2003, 337–342.
10. Holland, J. H., Holyoak, K. J., Nisbett, R. E.: *Induction. Processes of Inference, Learning, and Discovery*. The MIT Press, Menlo Park, CA, (1986).
11. Jensen, F. V.: *Bayesian Networks and Decision Graphs*, Springer-Verlag, Heidelberg (2001).
12. Kordek, A., Giedrys-Kalemba, S., Pawlus, B., Podraza, W., Czajka, R.: Umbilical cord blood serum procalcitonin concentration in the diagnosis of early neonatal infection. *J. Perinatol.* **23** (2003) 148–153.
13. Meisner, M.: *Procalcitonin. A New, Innovative Infection Parameter*. Georg Thieme Verlag, Stuttgart, New York (2000).
14. Mroczek, T., Grzymala-Busse, J. W., Hippe, Z. S.: Rules from belief networks: A Rough Set Approach. In: *Rough Sets and Current Trends in Computing*, ed. by Tsumoto, S., Slowinski, R., Komorowski, J., Grzymala-Busse, J. W. (eds), Springer-Verlag, Uppsala, Sweden 2004, 483–487.
15. Nyamande, K. Lalloo, U. G.: Serum procalcitonin distinguishes CAP due to bacteria, mycobacterium tuberculosis and PJP. *Crit. Care Resc.* **3** (2001) 236–243.
16. Pawlak, Z.: Rough Sets. *International Journal of Computer and Information Sciences* **11** (1982) 341–356.
17. Pawlak, Z.: *Rough Sets. Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, Boston, London (1991).
18. Podraza, W., Podraza, R., Domek H., Kordek, A. Gonet B.: Rough set methodology in supporting neonatal infection diagnosis. In *Artificial Intelligence and Soft Computing*. Cader, A., Rutkowski, L., Tadeusiewicz R., Zurada, J. (eds), Exit (2006), 281–287.
19. Stoll, B. J., Gordon, T., Korones, S. B.: *et al.*: Early-onset sepsis in very low birth weight neonates: A report from the National Institute of Child Health and Human Development Neonatal Research Network. *J. Pediatr.* **129** (1996) 72–80.
20. Tollner, U.: Early diagnosis of septicaemia in the newborn: clinical studies and sepsis score. *Eur. J. Pediatr.* **138** (1982) 331-337.
21. Varmuza, K., Grzymala-Busse, J. W., Hippe, Z. S., Mroczek, T.: Comparison of consistent and inconsistent models in biomedical domain: A rough set approach to melanoma data. In *Methods of Artificial Intelligence*, Silesian University of Technology Press, Gliwice, Poland, 2003, 323–328.

# Two Families of Classification Algorithms

Pawel Delimata[1], Mikhail Moshkov[2], Andrzej Skowron[3], and Zbigniew Suraj[4,5]

[1] Chair of Computer Science, University of Rzeszów
Rejtana 16A, 35-310 Rzeszów, Poland
pdelimata@wp.pl
[2] Institute of Computer Science, University of Silesia
Będzińska 39, 41-200 Sosnowiec, Poland
moshkov@us.edu.pl
[3] Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl
[4] Chair of Computer Science, University of Rzeszów
Rejtana 16A, 35-310 Rzeszów, Poland
zsuraj@univ.rzeszow.pl
[5] Institute of Computer Science, State School of Higher Education in Jarosław
Czarnieckiego 16, 37-500 Jarosław, Poland

**Abstract.** In the paper, two families of lazy classification algorithms of polynomial time complexity are considered. These algorithms are based on ordinary and inhibitory rules, but the direct generation of rules is not required. Instead of this, the considered algorithms extract efficiently for a new object some information on the set of rules which is next used by a decision-making procedure.

**Keywords:** rough sets, decision tables, information systems, rules.

## 1 Introduction

In the paper, the following classification problem is considered: for a given decision table $T$ and a new object $v$ generate a value of the decision attribute on $v$ using values of conditional attributes on $v$.

To this end, we divide the decision table $T$ into a number of information systems $S_i$, $i \in D$, where $D$ is the set of values of the decision attribute in $T$. For $i \in D$, the information system $S_i$ contains only objects (rows) of $T$ with the value of the decision attribute equal to $i$.

For each information system $S_i$ and a given object $v$, it is constructed (using polynomial-time algorithm) the so called characteristic table. For any object $u$ from $S_i$ and for any attribute $a$ from $S_i$, the characteristic table contains the entry encoding information if there exist a rule which (i) is true for each object from $S_i$; (ii) is realizable for $u$, (iii) is not true for $v$, and (iv) has the attribute $a$ on the right hand side. Based on the characteristic table the decision on the "degree" to which $v$ belongs to $S_i$ is made for any $i$, and a decision $i$ with the maximal "degree" is selected.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 297–304, 2007.

Note that in [8] for classifying new objects it was proposed to use rules defined by conditional attributes in different decision classes.

In this paper, we consider both ordinary and inhibitory rules of the following form:

$$a_1(x) = b_1 \wedge \ldots \wedge a_t(x) = b_t \Rightarrow a_k(x) = b_k,$$

$$a_1(x) = b_1 \wedge \ldots \wedge a_t(x) = b_t \Rightarrow a_k(x) \neq b_k,$$

respectively.

Using these two kinds of rules and different evaluation functions a "degree" to which $v$ belongs to $S_i$ is computed by two families of classification algorithms.

In the literature, one can find a number of papers which are based on the analogous ideas: instead of construction of huge sets of rules it is possible to extract some information on such sets using algorithms having polynomial time complexity.

In [2,3,4] it is considered an approach based on decision rules (with decision attribute in the right hand side). These rules are obtained from the whole decision table $T$. The considered algorithms find for a new object $v$ and any decision $i$ the number of objects $u$ from the information system $S_i$ such that there exists a decision rule $r$ satisfying the following conditions: (i) $r$ is true for the decision table $T$, (ii) $r$ is realizable for $u$ and $v$, and (iii) $r$ has the equality $d(x) = i$ on the right hand side, where $d$ is the decision attribute.

This approach was generalized by A. Wojna [9] to the case of decision tables with not only nominal but also numerical attributes.

Note that such algorithms can be considered as a kind of lazy learning algorithms [1].

## 2 Characteristic Tables

### 2.1 Information Systems

Let $S = (U, A)$ be an *information system*, where $U = \{u_1, \ldots, u_n\}$ is a finite nonempty set of *objects* and $A = \{a_1, \ldots, a_m\}$ is a finite nonempty set of *attributes* (functions defined on $U$). We assume that for each $u_i \in U$ and each $a_j \in A$ the value $a_j(u_i)$ belongs to $\omega$, where $\omega = \{0, 1, 2, \ldots\}$ is the set of nonnegative integers.

We also assume that the information system $S = (U, A)$ is given by a *tabular representation*, i.e., a table with $m$ columns and $n$ rows. Columns of the table are labeled by attributes $a_1, \ldots, a_m$. At the intersection of $i$-th row and $j$-th column the value $a_j(u_i)$ is included. For $i = 1, \ldots, n$ we identify any object $u_i \in U$ with the tuple $(a_1(u_i), \ldots, a_m(u_i))$, i.e., the $i$-th row of the tabular representation of the information system $S$.

The set $\mathcal{U}(S) = \omega^m$ is called the *universe* for the information system $S$. Besides objects from $U$ we consider also objects from $\mathcal{U}(S) \setminus U$. For any object (tuple) $v \in \mathcal{U}(S)$ and any attribute $a_j \in A$ the value $a_j(v)$ is equal to $j$-th integer in $v$.

### 2.2 Ordinary Characteristic Tables

Let us consider a rule

$$a_{j_1}(x) = b_1 \wedge \ldots \wedge a_{j_t}(x) = b_t \Rightarrow a_k(x) = b_k, \quad (1)$$

where $t \geq 0$, $a_{j_1}, \ldots, a_{j_t}, a_k \in A$, $b_1, , \ldots, b_t, b_k \in \omega$, and numbers $j_1, \ldots, j_t, k$ are pairwise different. Such rules will be called *ordinary* rules. The rule (1) will be called *realizable for an object* $u \in \mathcal{U}(S)$ if $a_{j_1}(u) = b_1, \ldots, a_{j_t}(u) = b_t$. The rule (1) will be called *true for an object* $u \in \mathcal{U}(S)$ if $a_k(u) = b_k$ or (1) is not realizable for $u$. The rule (1) will be called *true for* $S$ if it is true for any object from $U$. The rule (1) will be called *realizable for* $S$ if it is realizable for at least one object from $U$. Denote by $Ord(S)$ the set of all ordinary rules each of which is true for $S$ and realizable for $S$.

Let $u_i \in U$, $v \in \mathcal{U}(S)$, $a_k \in A$ and $a_k(u_i) \neq a_k(v)$. We say that a rule (1) from $Ord(S)$ *contradicts* $v$ *relative to* $u_i$ *and* $a_k$ (or, $(u_i, a_k)$-*contradicts* $v$, for short) if (1) is realizable for $u_i$ but is not true for $v$. Our aim is to recognize for given objects $u_i \in U$ and $v \in \mathcal{U}(S)$, and given attribute $a_k$ such that $a_k(u_i) \neq a_k(v)$ if there exist a rule from $Ord(S)$ which $(u_i, a_k)$-contradicts $v$.

Let

$$M(u_i, v) = \{a_j : a_j \in A, a_j(u_i) = a_j(v)\},$$

and

$$P(u_i, v, a_k) = \{a_k(u) : u \in U, a_j(u) = a_j(v) \text{ for any } a_j \in M(u_i, v)\}.$$

Note that $|P(u_i, v, a_k)| \geq 1$.

**Proposition 1.** *Let* $S = (U, A)$ *be an information system,* $u_i \in U$, $v \in \mathcal{U}(S)$, $a_k \in A$ *and* $a_k(u_i) \neq a_k(v)$. *Then, in* $Ord(S)$ *there exists a rule* $(u_i, a_k)$-*contradicting* $v$ *if and only if* $|P(u_i, v, a_k)| = 1$.

*Proof.* Let $|P(u_i, v, a_k)| = 1$ and $P(u_i, v, a_k) = \{b\}$. In this case, the rule

$$\bigwedge_{a_j \in M(u_i, v)} a_j(x) = a_j(v) \Rightarrow a_k(x) = b, \quad (2)$$

belongs to $Ord(S)$, is realizable for $u_i$, and is not true for $v$, since $a_k(v) \neq a_k(u_i) = b$. Therefore, (2) is a rule from $Ord(S)$, which $(u_i, a_k)$-contradicts $v$.

Let us assume that there exists a rule (1) from $Ord(S)$ $(u_i, a_k)$-contradicting $v$. Since (1) is realizable for $u_i$ and is not true for $v$, we have $a_{j_1}, \ldots, a_{j_t} \in M(u_i, v)$. Also (1) is true for $S$. Hence, the rule

$$\bigwedge_{a_j \in M(u_i, v)} a_j(x) = a_j(v) \Rightarrow a_k(x) = b_k$$

is true for $S$. Therefore, $P(u_i, v, a_k) = \{b_k\}$ and $|P(u_i, v, a_k)| = 1$. □

From Proposition 1 it follows that there exists polynomial algorithm recognizing, for a given information system $S = (U, A)$, given objects $u_i \in U$ and $v \in \mathcal{U}(S)$, and a given attribute $a_k \in A$ such that $a_k(u_i) \neq a_k(v)$, if there exist a rule from $Ord(S)$ $(u_i, a_k)$-contradicting $v$.

This algorithm constructs the set $M(u_i, v)$ and the set $P(u_i, v, a_k)$. The considered rule exists if and only if $|P(u_i, v, a_k)| = 1$.

We also use the notion of *ordinary characteristic table* $O(S, v)$, where $v \in \mathcal{U}(S)$. This is a table with $m$ columns and $n$ rows. The entries of this table are binary (i.e., from $\{0, 1\}$). The number 0 is at the intersection of $i$-th row and $k$-th column if and only if $a_k(u_i) \neq a_k(v)$ and there exists a rule from $Ord(S)$ $(u_i, a_k)$-contradicting $v$.

From Proposition 1 it follows that there exists a polynomial algorithm which for a given information system $S = (U, A)$ and a given object $v \in \mathcal{U}(S)$ constructs the ordinary characteristic table $O(S, v)$.

### 2.3 Inhibitory Characteristic Tables

Let us consider a rule

$$a_{j_1}(x) = b_1 \wedge \ldots \wedge a_{j_t}(x) = b_t \Rightarrow a_k(x) \neq b_k, \tag{3}$$

where $t \geq 0$, $a_{j_1}, \ldots, a_{j_t}, a_k \in A$, $b_1, , \ldots, b_t, b_k \in \omega$, and numbers $j_1, \ldots, j_t, k$ are pairwise different. Such rules are called *inhibitory* rules. The rule (3) will be called *realizable for an object* $u \in \mathcal{U}(S)$ if $a_{j_1}(u) = b_1, \ldots, a_{j_t}(u) = b_t$. The rule (3) will be called *true for an object* $u \in \mathcal{U}(S)$ if $a_k(u) \neq b_k$ or (3) is not realizable for $u$. The rule (3) will be called *true for* $S$ if it is true for any object from $U$. The rule (3) will be called *realizable for* $S$ if it is realizable for at least one object from $U$. Denote by $Inh(S)$ the set of all inhibitory rules each of which is true for $S$ and realizable for $S$.

Let $u_i \in U$, $v \in \mathcal{U}(S)$, $a_k \in A$ and $a_k(u_i) \neq a_k(v)$. We say that a rule (3) from $Inh(S)$ *contradicts* $v$ *relative to the object* $u_i$ *and the attribute* $a_k$ (or $(u_i, a_k)$-contradicts $v$, for short) if (3) is realizable for $u_i$ but is not true for $v$. Our aim is to recognize for given objects $u_i \in U$ and $v \in \mathcal{U}(S)$, and given attribute $a_k$ such that $a_k(u_i) \neq a_k(v)$ if there exist a rule from $Inh(S)$ $(u_i, a_k)$-contradicting $v$.

**Proposition 2.** *Let* $S = (U, A)$ *be an information system,* $u_i \in U$, $v \in \mathcal{U}(S)$, $a_k \in A$ *and* $a_k(u_i) \neq a_k(v)$. *Then in* $Inh(S)$ *there is a rule* $(u_i, a_k)$*-contradicting* $v$ *if and only if* $a_k(v) \notin P(u_i, v, a_k)$.

*Proof.* Let $a_k(v) \notin P(u_i, v, a_k)$. In this case, the rule

$$\bigwedge_{a_j \in M(u_i, v)} a_j(x) = a_j(v) \Rightarrow a_k(x) \neq a_k(v), \tag{4}$$

belongs to $Inh(S)$, is realizable for $u_i$, and is not true for $v$. Therefore, (4) is a rule from $Inh(S)$ $(u_i, a_k)$-contradicting $v$.

Let us assume that there exists a rule (3) from $Inh(S)$, $(u_i, a_k)$-contradicting $v$. In particular, it means that $a_k(v) = b_k$. Since (3) is realizable for $u_i$ and is not true for $v$, we have $a_{j_1}, \ldots, a_{j_t} \in M(u_i, v)$. Since (3) is true for $S$, the rule

$$\bigwedge_{a_j \in M(u_i,v)} a_j(x) = a_j(v) \Rightarrow a_k(x) \neq b_k$$

is true for $S$. Therefore, $a_k(v) \notin P(u_i, v, a_k)$. □

From Proposition 2 it follows that there exists polynomial algorithm recognizing for a given information system $S = (U, A)$, given objects $u_i \in U$ and $v \in \mathcal{U}(S)$, and a given attribute $a_k \in A$ such that $a_k(u_i) \neq a_k(v)$ if there exist a rule from $Inh(S)$ $(u_i, a_k)$-contradicting $v$.

This algorithm constructs the set $M(u_i, v)$ and the set $P(u_i, v, a_k)$. The considered rule exists if and only if $a_k(v) \notin P(u_i, v, a_k)$.

In the sequel, we use the notion of *inhibitory characteristic table* $I(S, v)$, where $v \in \mathcal{U}(S)$. This is a table with $m$ columns and $n$ rows. The entries of this table are binary. The number 0 is at the intersection of $i$-th row and $k$-th column if and only if $a_k(u_i) \neq a_k(v)$ and there exists a rule from $Inh(S)$ $(u_i, a_k)$-contradicting $v$.

From Proposition 2 it follows that there exists a polynomial algorithm which for a given information system $S = (U, A)$ and a given object $v \in \mathcal{U}(S)$ constructs the inhibitory characteristic table $I(S, v)$.

### 2.4 Evaluation Functions

Let us denote by $\mathcal{T}$ the set of binary tables, i.e., tables with entries from $\{0, 1\}$ and let us consider a partial order $\preceq$ on $\mathcal{T}$. Let $Q_1, Q_2 \in \mathcal{T}$. Then $Q_1 \preceq Q_2$ if and only if $Q_1 = Q_2$ or $Q_1$ can be obtained from $Q_2$ by changing some entries from 1 to 0.

An *evaluation function* is an arbitrary function $W : \mathcal{T} \to [0, 1]$ such that $W(Q_1) \leq W(Q_2)$ for any $Q_1, Q_2 \in \mathcal{T}$, $Q_1 \preceq Q_2$. Let us consider three examples of evaluation functions $W_1$, $W_2$ and $W_3^\alpha$, $0 < \alpha \leq 1$. Let $Q$ be a table from $\mathcal{T}$ with $m$ columns and $n$ rows. Let $L_1(Q)$ be equal to the number of 1 in $Q$, $L_2(Q)$ be equal to the number of columns in $Q$ filled by 1 only, and $L_3^\alpha(Q)$ is defined as the number of columns in $Q$ with at least $\alpha \cdot 100\%$ entries equal to 1. Then

$$W_1(Q) = \frac{L_1(Q)}{mn}, \quad W_2(Q) = \frac{L_2(Q)}{m}, \quad \text{and} \quad W_3^\alpha(Q) = \frac{L_3^\alpha(Q)}{m}.$$

It is clear that $W_2 = W_3^1$. Let $S = (U, A)$ be an information system and $v \in \mathcal{U}(S)$. Note that if $v \in U$ then $W_1(O(S, v)) = W_2(O(S, v)) = W_3^\alpha(O(S, v)) = 1$ and $W_1(I(S, v)) = W_2(I(S, v)) = W_3^\alpha(I(S, v)) = 1$ for any $\alpha$ $(0 < \alpha \leq 1)$.

## 3 Algorithms of Classification

A decision table $T$ is a finite table filled by nonnegative integers. Each column of this table is labeled by a conditional attribute. Rows of the table are interpreted

as tuples of values of conditional attributes on some objects. Each row is labeled by a nonnegative integer, which is interpreted as the value of decision attribute. Let $T$ contain $m$ columns labeled by conditional attributes $a_1, \ldots, a_m$. The set $\mathcal{U}(T) = \omega^m$ will be called the *universe* for the decision table $T$. For each object (tuple) $v \in \mathcal{U}(T)$ integers in $v$ are interpreted as values of attributes $a_1, \ldots, a_m$ for this object.

We consider the following classification problem: for any object $v \in \mathcal{U}(T)$ it is required to compute a value of decision attribute on $v$. To this end, we use O-classification algorithms and I-classification algorithms based on the ordinary characteristic table and the inhibitory characteristic table.

Let $D$ be the set of values of decision attribute. For each $i \in D$, let us denote by $S_i$ the information system which tabular representation consists of all rows of $T$, that are labeled by the decision $i$. Let $W$ be an evaluation function.

*O-algorithm.* For a given object $v$ and $i \in D$ we construct the ordinary characteristic table $O(S_i, v)$. Next, for each $i \in D$ we find the value of the evaluation function $W$ for $O(S_i, v)$. For each $i \in D$ the value $W(O(S_i, v))$ is interpreted as the "degree" to which $v$ belongs to $S_i$. As the value of decision attribute for $v$ we choose $i \in D$ such that $W(O(S_i, v))$ has the maximal value. If more than one such $i$ exists then we choose the minimal $i$ for which $W(O(S_i, v))$ has the maximal value.

*I-algorithm.* For a given object $v$ and $i \in D$ we construct the inhibitory characteristic table $I(S_i, v)$. Next, for each $i \in D$ we find the value of the evaluation function $W$ for $I(S_i, v)$. For each $i \in D$ the value $W(I(S_i, v))$ is interpreted as the "degree" to which $v$ belongs to $S_i$. As the value of decision attribute for $v$ we choose $i \in D$ such that $W(I(S_i, v))$ has the maximal value. If more than one such $i$ exists then we choose the minimal $i$ for which $W(I(S_i, v))$ has the maximal value.

## 4 Results of Experiments

We have performed experiments with following algorithms: O-algorithm with the evaluation functions $W_1$, $W_2$ and $W_3^\alpha$, and I-algorithm with the evaluation functions $W_1$, $W_2$ and $W_3^\alpha$. To evaluate error rate of an algorithm on a decision table we use either train-and-test method or cross-validation method.

The following decision tables from [6] were used in our experiments: monk1 (6 conditional attributes, 124 objects in training set, 432 objects in testing set), monk2 (6 conditional attributes, 169 objects in training set, 432 objects in testing set), monk3 (6 conditional attributes, 122 objects in training set, 432 objects in testing set), lymphography (18 conditional attributes, 148 objects, 10-fold cross-validation), diabetes (8 conditional attributes, 768 objects, 12-fold cross-validation, attributes are discretized by an algorithm from RSES2 [7]), breast-cancer (9 conditional attributes, 286 objects, 10-fold cross-validation), primary-tumor (17 conditional attributes, 339 objects, 10-fold cross-validation, missing values are filled by an algorithm from RSES2).

Table 1 contains results of experiments (error rates) for O-algorithm and I-algorithm with the evaluation functions $W_1$ and $W_2$, and for each of the

**Table 1.** Results of experiments with evaluation functions $W_1$ and $W_2$

| Decision table | O-alg., $W_1$ | O-alg., $W_2$ | I-alg., $W_1$ | I-alg., $W_2$ | err. rates [3] |
|---|---|---|---|---|---|
| monk1 | 0.292 | 0.443 | 0.114 | 0.496 | 0.000–0.240 |
| monk2 | 0.260 | 0.311 | 0.255 | 0.341 | 0.000–0.430 |
| monk3 | 0.267 | 0.325 | 0.119 | 0.322 | 0.000–0.160 |
| lymphography | 0.272 | 0.922 | 0.215 | 0.922 | 0.157–0.380 |
| diabetes | 0.348 | 0.421 | 0.320 | 0.455 | 0.224–0.335 |
| breast-cancer | 0.240 | 0.261 | 0.233 | 0.268 | 0.220–0.490 |
| primary-tumor | 0.634 | 0.840 | 0.634 | 0.846 | 0.550–0.790 |
| average err. rate | 0.330 | 0.503 | 0.270 | 0.521 | 0.164–0.404 |

considered tables. The last row contains average error rates. The last column contains some known results – the best and the worst error rates for algorithms compared in the survey [3].

The obtained results show that the evaluation function $W_1$ is noticeably better than the evaluation function $W_2$, and I-algorithm with the evaluation function $W_1$ is better than O-algorithm with the evaluation function $W_1$. The last result follows from the fact that the inhibitory rules have much higher chance to have larger support in the decision tables than the ordinary rules.

The outputs returned by I-algorithm with the evaluation function $W_1$ for each of decision tables are comparable with the results reported in [3], but are worse than the best results mentioned in [3].

Table 2 contains results of experiments (error rates) for two types of algorithms: O-algorithm with the evaluation function $W_3^\alpha$, and I-algorithm with the evaluation function $W_3^\alpha$, where $\alpha \in \{0.50, 0.55, \ldots, 0.95, 1.00\}$. For each decision table and for algorithms of each type the best result (with the minimal error rate) and the corresponding $\alpha$ to this result are presented in the table. The last row contains average error rates. The last column contains some known results – the best and the worst error rates for algorithms discussed in [3].

The obtained results show that the use of the parameterized evaluation functions $W_3^\alpha$, where $\alpha \in \{0.50, 0.55, \ldots, 0.95, 1.00\}$, makes it possible to improve

**Table 2.** Results of experiments with evaluation functions $W_3^\alpha$

| Decision table | O-alg., $W_3^\alpha$ | $\alpha$ | I-alg., $W_3^\alpha$ | $\alpha$ | err. rates [3] |
|---|---|---|---|---|---|
| monk1 | 0.172 | 0.95 | 0.195 | 0.85 | 0.000–0.240 |
| monk2 | 0.301 | 0.95 | 0.283 | 0.95 | 0.000–0.430 |
| monk3 | 0.325 | 1.00 | 0.044 | 0.65 | 0.000–0.160 |
| lymphography | 0.293 | 0.55 | 0.272 | 0.65 | 0.157–0.380 |
| diabetes | 0.421 | 1.00 | 0.351 | 0.95 | 0.224–0.335 |
| breast-cancer | 0.229 | 0.80 | 0.225 | 0.70 | 0.220–0.490 |
| primary-tumor | 0.658 | 0.75 | 0.655 | 0.70 | 0.550–0.790 |
| average err. rate | 0.343 | | 0.289 | | 0.164–0.404 |

the performance of I-algorithm with the evaluation function $W_1$ for tables monk3 and breast-cancer.

In experiments the DMES system [5] was used.

## 5 Conclusions

In the paper, two families of lazy classification algorithms are considered which are based on the evaluation of the number of types of true rules which give us "negative" information about new objects. In the further investigations we are planning to consider also the number of types of true rules which give us "positive" information about new objects. Also we are planning to consider more wide parametric families of evaluation functions which will allow to learn classification algorithms.

## References

1. Aha, D.W. (Ed.): Lazy Learning. Kluwer Academic Publishers, Dordrecht, Boston, London, 1997
2. Bazan, J.G.: Discovery of decision rules by matching new objects against data tables. Proceedings of the First International Conference on Rough Sets and Current Trends in Computing. Warsaw, Poland. Lecture Notes in Artificial Intelligence **1424**, Springer-Verlag, Heidelberg (1998) 521-528
3. Bazan, J.G.: A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision table. Rough Sets in Knowledge Discovery. Edited by L. Polkowski and A. Skowron. Physica-Verlag, Heidelberg (1998) 321–365
4. Bazan, J.G.: Methods of approximate reasoning for synthesis of decision algorithms. Ph.D. Thesis. Warsaw University (1998) (in Polish)
5. Data Mining Exploration System Homepage http://www.univ.rzeszow.pl/rspn (Software)
6. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases http://www.ics.uci.edu/∼mlearn/MLRepository.html. University of California, Irvine, Department of Information and Computer Sciences (1998)
7. Rough Set Exploration System Homepage http://logic.mimuw.edu.pl/∼rses
8. Skowron, A., Suraj, Z.: Discovery of concurrent data models from experimental tables: a rough set approach. Proceedings of the First International Conference on Knowledge Discovery and Data Mining, Montreal, Canada, August, 1995, AAAI Press, Menlo Park CA (1995) 288–293
9. Wojna, A.: Analogy-based reasoning in classifier construction (Ph.D. Thesis), Transactions on Rough Sets IV, Lecture Notes in Computer Science **3700**, Springer-Verlag, Heidelberg (2005) 277–374

# Constructing Associative Classifiers from Decision Tables

Jianchao Han[1], T.Y. Lin[2], Jiye Li[3], and Nick Cercone[4]

[1] Department of Computer Science, California State University Dominguez Hills,
Carson, CA 90747, USA
jhan@csudh.edu
[2] Department of Computer Science, San Jose State University,
San Jose, CA 95192, USA
tylin@cs.sjsu.edu
[3] School of Computer Science, University of Waterloo,
Waterloo, ON N2L 3G1, Canada
j27li@uwaterloo.ca
[4] Faculty of Science and Engineering, York University,
Toronto, ON M3J 1P3, Canada
ncercone@yorku.ca

**Abstract.** Constructing associative classifier is to use the technique of mining association rules to extract attribute-value pairs that are associated with class labels. Since too many such kinds of associations may be generated however, the existing algorithms to finding associations are usually ineffective. It is well known that rough sets theory can be used to select reducts of attributes that represent the original data set. In this paper we present an approach of combining the rough set theory, the association rules mining technique, and the covering method to construct classification rules. With a given decision table, the rough set theory is first used to find all reducts of condition attributes of the decision table. Then an adapted Apriori algorithm to mining association rules is used to find a set of associative classifications from each reduct. And third, all association classification rules are ranked according to their importance, support, and confidence and selected in sequence to build a classifier with high accuracy. An example illustrates how this approach works.

**Keywords:** Rough set, association rule, associative classifier, rule coverage.

## 1 Introduction

Constructing an efficient and accurate classifier from a large data set is an important topic in data mining and machine learning community. The task of building a classifier is to find a small set of classification rules that can well cover the training data set and accurately predict future instances. However, since real-application data sets may contain many attributes to describe data instances, most classifier building algorithms only work well on small data sets. Fortunately, most large data sets contain many redundant or noise attributes that not only degrade the efficiency of classifier building algorithms but also decrease the accuracy of classifiers that are built upon them. In order to reduce the number of attributes, especially removing these

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 305–313, 2007.

redundant and noise attributes, the rough sets theory is commonly used to select reducts of attributes that can represent all original attributes. Rough set theory [15] assumes that the given data set is a decision table that consists of condition attributes and decision attributes. A reduct of the decision table is a subset of condition attributes that can represent the whole data set. Although finding all reducts or a minimum reduce is NP-hard [16], researchers have been devoted to develop efficient approaches of generating approximately minimum or pseudo-minimum reduct [6], [13], [14], [18]. Most reduct generation algorithms are designed to extract important condition attributes from a decision table. Since classification data sets are usually a special format of decision tables where decision attributes are class labels, the rough set theory can also be adapted to find classifiers.

Mining association rules in large databases has been extensively studied since 1993 when the algorithm Apriori was developed [1], and many efficient algorithms have been proposed to deal with various problems encountered in real-life applications [2], [5], [7], [10]. One of the main problems for association rule generation is that the number of rules generated is generally quite large, thus rule templates or constrained rule patterns have been introduced to guide the algorithm execution and integrated into the rule interestingness measures. A special template of constrained association rules is the form of classification rules where the right-side of association rules is a class label. Some efforts have been made to apply algorithms of mining association rules in seeking this kind of classification rules, usually called class association rules or associative classification rules [3], [4], [9], [11]. To build an associative classifier with a small number of classification rules, the associative classification rules generated in such a way should be measured and selected to have a maximum coverage of original data set with high accuracy.

In this paper we present an approach of combining rough set theory [14], [15], techniques of mining class association rules [9], [11]., the rule importance measurement in [8], and the covering method of Michalski [12] to build classifiers. The method consists of four steps with a given decision table. First, the rough set theory is used to find all reducts of condition attributes of the decision table [14]; second, An adapted Apriori algorithm of mining association rules [11] is applied to find a set of associative classification rules from each reduct; third, these rules are measured with the rule importance [8]; and fourth, all association classification rules are ranked according to their importance, support, and confidence and selected in sequence to cover the given data set maximally with the highest accuracy [12].

The rest of the paper is organized as follows. The related work will be summarized and analyzed in the next section. Our approach will be presented and the algorithm will be described in Section 3 and an example is illustrated in Section 4. Section 5 is the discussion and conclusion.

## 2 Related Work

Rough sets theory was first introduced by Pawlak [15] in the 1980's and applied in knowledge discovery systems to identify and remove redundant variables, and to classify imprecise and incomplete information. A *reduct* of a decision table is a subset of condition attributes that suffice to define the decision attributes. More than one

reduct for each decision table may exist. The intersection of all the possible reducts is called the *core*, which represents the most important information of the original data set. Finding all reducts for a decision table is NP-hard [16] unfortunately, therefore approximation algorithms have been proposed and developed to build reducts from a decision table either top-down or bottom-up [18]. Some packages like ROSETTA [14] have been implemented to support data mining, including a variety of reduct generation algorithms.

Association rule mining has been extensively studied in the field since the original Apriori algorithm was proposed by Agrawal *et al.* in 1993 [1], and more and more improved and extended algorithms have been developed [2], [4], [7], [10]. Basically, the task of mining association rules is to find all relationships among items in a transactional database that satisfy some minimum support and minimum confidence constraints. One main drawback of traditional algorithms of mining association rules is that too many rules will be generated where most of them are trivial, spurious, and even useless. This is because there is no mining target predefined, and the rule generation is a blind-search [1]. In order to overcome this problem, some constraints have been enforced on the format of association rules, such as the rule template [7] where a rule pattern is predefined before the algorithm executes and only those rules that match the template will be discovered. A special rule template has been studied to constrain the association rules as classification rules where the right-side of rules must be a class label [3], [9], [11]. This rule template is called class association rule or associative classification rule. With this template, Liu, Hsu, and Ma [11] propose an approach to constructing a classifier, which consists of two steps. First the Apriori algorithm is adapted to find all class association rules with predetermined minimum support and threshold. In the second step, all class association rules are sorted in terms of their confidence and support, and are selected in sequence to find a small set of rules that has the lowest classification error. The small set of class association rules selected in this fashion forms an associative classifier.

Since classification analysis is independent from association mining, one must convert each training instance of classification to a set of items that is represented as an attribute-value pair in order to apply the adapted Apriori algorithm of discovering association rules. Even though the rule template is exploited in generating class association rules, the number of resulting rules is still prohibitive, especially in large databases or decision tables with many different values for each attribute. To reduce the number of possible attribute-value pairs, Szczuka [17] proposes a method to generate reducts from the original data set using rough sets theory, and then construct from the reducts generated classification rules with a rule-based system and neural networks. Li and Cercone [8] introduce another method of using rough set theory to evaluate important association rules. Association rules are generated from a set of reducts of original decision table, and importance of these association rules is defined according to their occurrence in the set of reducts. A new decision table is constructed with association rules being considered as condition attributes and the decision attributes being the same as the original decision attributes. They claim that a reduct of such a decision table (actually a reduct of association rules) represents the essential and the most important rules that fully describe the decision. Both methods in [8] and

[17], however, have a fatal problem that keeping the original decision attributes does not guarantee the new decision table is consistent with the original decision table.

Michalski [12] presents a covering method to construct and select classification rules in terms of their coverage of the training instances. For each class, it finds the best characterization rule for the class and removes those training instances that are covered by the rule. An instance is covered by a rule if it satisfies the conditions of the rule and has the same class label as the rule. The procedure is then recursively applied to the remaining instances in the class until all training instances are covered by at lease one classification rule induced from the class. However, the best rules constructed and selected this way are local to the class from which the rules are induced.

## 3 Our Approach

In this section we present our approach to constructing associative classifiers from a decision table. The framework of this approach is a combination of three strategies that are described in the previous section: rough set theory, association rules mining, and covering method. The four steps of our approach can be described as follows:

**Step 1:** Generating all attribute reducts of the decision table by using existing reduct finding algorithms such as Genetic Reduct generation algorithm in ROSETTA [14] that can find all reducts.

**Step 2:** For each reduct, the adapted Apriori algorithm presented by Liu, Hsu, and Ma [11] is used to mine a set of class association rules with each being attached a support and confidence that are greater than or equal to the predetermined thresholds. The support threshold and confidence threshold must be carefully specified for this special template of class association rules. Some considerations should be made. On one hand, low confidence threshold may degrade performance of the classifier since most classification rules should have very high confidence otherwise the classifier induced will have very low classification accuracy, for example, Bayardo [4] uses 90% as the confidence threshold in the experiment. On the other hand, high support threshold may eliminate important rules especially for unbalanced set of training instances since some rules with low support are necessary to cover the original decision table. This is why Li and Cercone [8] as well as Liu, Hsu, and Ma [11] use 1% as the support threshold in their experiments.

**Step 3:** Class association rules generated by the adapted Apriori algorithm from all reducts of the decision table are ranked in terms of their importance, confidence, and support. We take the importance definition of class association rules defined by Li and Cercone [8], and restate the definition as follows:

**Definition 1:** *(Rule Importance) If a rule is generated more frequently from different reducts of the original decision table, we say this rule is more important than those rules generated less frequently. The rule importance is quantitatively defined as follows: Assume a class association rule R is generated from M reducts of the*

*original decision table, and N is the total number of reducts of the original decision table, then the importance of R is: Importance(R) = M/N.*

A class association rule is generated from a reduct of the original decision table if all the attributes of the attribute-value pairs occurring in the antecedence of the rule are contained by the reduct. The intuition behind the rule importance is that each reduct contains the most representative and important condition attributes of a decision table.

One can easily verify the following properties of the rule importance:

**Property 1:** *For any class association rule R generated in Step 2:*

$$0 < Importance(R) \leq 1.$$

**Property 2:** *If a class association rule R generated in Step 2 only contains core attributes, then Importance(R) = 1, since core attributes are contained by all reducts of condition attributes of the original decision table [15].*

**Step 4:** Our next step is to adapt the covering method presented by Michalski [12] to find a small set of class association rules generated above to induce a classifier. To this end, the precedence relationship on class association rules is defined below.

**Definition 2:** *(Rule Precedence) Given two class association rules $R_1$ and $R_2$ generated in Step 2, $R_1$ precedes $R_2$ ($R_1$ has a higher precedence than $R_2$), denoted $Precedence(R_1) > Precedence(R_1)$, if*

*1) $Importance(R_1) > Importance(R_2)$; or*

*2) $Importance(R_1) = Importance(R_2)$, and $Confidence(R_1) > Confidence(R_2)$; or*

*3) $Importance(R_1) = Importance(R_2)$, $Confidence(R_1) = Confidence(R_2)$, and $Support(R_1) > Support(R_2)$.*

*Otherwise $R_1$ and R2 are considered having the same precedence and denoted $Precedence(R_1) = Precedence(R_2)$.*

One can verify the following property of above precedence relationship:

**Property 3:** *The precedence relationship defined in Definition 2 among class association rules generated in Step 2 is a total order relation. Thus, all these rules can be sorted.*

With this preparation, the associative classifier can be constructed as follows: sort all class association rules; pick the highest precedent rule (if multiple rules have the same highest precedence, then arbitrarily choose one of them); check rows of the original decision table and remove all matching rows (a row of the decision table matches the rule if it satisfies the antecedence of the row); if at least one row is eliminated from the decision table, then move the rule to the classifier; pick the next highest precedent rule and remove all matching rows from the decision table and move the rule to the classifier if at least one row is removed; …; repeat this process until either no more rows remain or no more rules remain.

To summarize, our greedy algorithm of building an associative classifier from a decision table is described in the following procedure.

**Procedure:** Building an associative classifier from a decision table

**Input:** An information system IS = (U, C, D), where U is a decision table with C as the set of condition attributes and D a decision (or class) attribute.

**Output:** An association classifier AC consisting of a set of class association rules.

Step 1: Generate all reducts of condition attributes C from the decision table U with respect to the decision attribute D, and save in ALL_REDUCTS:
    ALL_REDUCTS ← all reducts as the output of Genetic reducer in ROSETTA with each rule attached a support and a confidence measure
Step 2: Generate all class association rules
    CAR ← empty
    For each reduct REDU in ALL_REDUCTS Do
        Generate a set of class association rules and save in CAR
        CAR(REDU) ← Apply the adapted Apriori algorithm
        CAR ← CAR ∪ CAR(REDU)
Step 3: Compute the importance of class association rules
    For each rule R in CAR Do
        M ← Count the number of reducts in ALL_REDUCTS that contain R
        Importance(R) ← M / |ALL_REDUCTS|
Step 4: Construct an associative classifier
    Sort all rules in CAR in terms of their precedence consisting of importance, confidence and support
    AC ← empty
    While CAR is not empty and U is not delete-marked completely Do
        R ← Remove the first rule from CAR
        Delete-flag ← false
        For each row d in U that has not been delete-marked Do
            If d matches R Then
                Delete-mark d
                Delete-flag ← true
        If Delete-flag Then
            AC ← AC ∪ {R}
Step 5: Return AC

## 4 An Illustrative Example

An example is illustrated in this section to demonstrate how our approach works. The decision table on car performance is shown in Table 1, which is taken from Li and Cercone [8]. The decision table is used to decide the mileage of different cars. This data set contains 14 rows, and 8 conditional attributes. There are no inconsistent or incomplete tuples in this data set.

Using the Genetic reducer in ROSETTA [14], one can find 4 reducts of condition attributes with respect to the decision attribute *Mileage*, which are shown in Table 2. Specifying the support threshold = 1% and confidence threshold = 100% and applying the adapted Apriori algorithm, one can generate 13 class association rules, as shown in Table 3, where all rules have been sorted in the middle column and the last column lists the rule importance and absolute support.

**Table 1.** Artificial Car Data Set

| Make-model | cyl | door | displace | compress | power | trans | weight | mileage |
|---|---|---|---|---|---|---|---|---|
| USA | 6 | 2 | Medium | High | High | Auto | Medium | Medium |
| USA | 6 | 4 | Medium | Medium | Medium | Manual | Medium | Medium |
| USA | 4 | 2 | Small | High | Medium | Auto | Medium | Medium |
| USA | 4 | 2 | Medium | Medium | Medium | Manual | Medium | Medium |
| USA | 4 | 2 | Medium | Medium | High | Manual | Medium | Medium |
| USA | 6 | 4 | Medium | Medium | High | Auto | Medium | Medium |
| USA | 4 | 2 | Medium | Medium | High | Auto | Medium | Medium |
| USA | 4 | 2 | Medium | High | High | Manual | Light | High |
| Japan | 4 | 2 | Small | High | Low | Manual | Light | High |
| Japan | 4 | 2 | Medium | Medium | Medium | Manual | Medium | High |
| Japan | 4 | 2 | Small | High | High | Manual | Medium | High |
| Japan | 4 | 2 | Small | Medium | Low | Manual | Medium | High |
| Japan | 4 | 2 | Small | High | Medium | Manual | Medium | High |
| USA | 4 | 2 | Small | High | Medium | Manual | Medium | High |

**Table 2.** Reducts generated by Genetic reducer from Table 1

| Reduct # | Reduct Attributes |
|---|---|
| 1 | Make, compress, power, trans |
| 2 | make, cyl, compress, trans |
| 3 | make, displace, compress, trans |
| 4 | make, cyl, door, displace, trans, weight |

**Table 3.** Class Association Rules generated by Apriori algorithm

| # | Rules | Rule Precedence |
|---|---|---|
| 1 | (make, Japan) → (Mileage, High) | 4/4, 5 |
| 2 | (Trans, Auto) → (Mileage, Medium) | 4/4, 4 |
| 3 | (Compress, High), (Trans, Manual) → (Mileage, High) | 3/4 5 |
| 4 | (make, USA), (Compress, Medium) → (Mileage, Medium) | 3/4, 5 |
| 5 | (Displace, Small), (Trans, Manual) → (Mileage, High) | 2/4, 5 |
| 6 | (Cyl, 6) → (Mileage, Medium) | 2/4, 3 |
| 7 | (USA, Car), (Displace, Medium), (Weight, Medium) → (Mileage, Medium) | 1/4, 6 |
| 8 | (make, USA), (Power, High) → (Mileage, Medium) | 1/4, 4 |
| 9 | (Compress, Medium), (Power, High) → (Mileage, Medium) | 1/4, 3 |
| 10 | (Power, Low) → (Mileage, High) | 1/4, 2 |
| 11 | (Door, 4) → (Mileage, Medium) | 1/4, 2 |
| 12 | (Weight, Light) → (Mileage, High) | 1/4, 2 |
| 13 | (Displace, Small), (Compress, Medium) → (Mileage, High) | 1/4, 1 |

With the covering method, we first choose Rule 1 and match it with rows of the original decision table and find it covers 5 rows 9 through 13. After these 5 rows are deleted and Rule 2 is next chosen to match the decision table and remove two rows 8

and 14. One can verify that Rule 3 covers 4 rows 1, 3, 6, and 7 after Rules 1 and 2, and Rule 4 covers rows 2, 4 and 5 after Rules 1, 2, and 3. Since all rows in the original decision table have been covered by Rules 1 through 4, the final associative classifier contains only these four class association rules.

## 5 Conclusion

An approach of building associative classifiers from decision tables has been described and illustrated with an example. The approach consists of four steps: generating all reducts using rough set theory; extracting class association rules from each reduct using an adapted Apriori algorithm; computing class association rules' importance; and finding a subset of class association rules to form a classifier using the covering method. Compared with the methods presented in [3], [4], [9], [11], extracting class association rules only from reducts can reduce the number of rules prohibitively. While compared with the method proposed in [8] and [17], our method can guarantee the classifier generated containing most important rules and cover the training instances. Our future work will be focusing on more experiments on real-application data sets and the improvement of the covering method.

## References

1. Agrawal, R., Imielinski, T., and Swami, A.: Mining Association Rules between Sets of Items in Large Databases, Proc. of the ACM SIGMOD Conference (1993) 207-216.
2. Agrawal, R. and Srikant, R.: Fast algorithms for mining association rules, Proc. of the 20th International Conference on Very Large Data Bases, Morgan Kaufmann (1994) 487-499.
3. Antonie, M-L., Zaïane, O.: An Associative Classifier based on Positive and Negative Rules, Proc. of ACM Internal. Conf. on Data Mining and Knowledge Discovery (2004) 64-69.
4. Bayardo, R. J.: Brute-force mining of high-confidence classification rules, Proc. of ACM Internal. Conf. on Knowledge Discovery and Data Mining (1997) 123-126.
5. Brin, S., Motwani, R., Ullman, J., Tsur, S.: Dynamic Itemset Counting and Implication Rules for Market BasketData, Proc. of ACM SIGMOD Internal. Conf. on Management of Data (1997) 255-264.
6. Han, J., Hu, X., Lin, T.: Feature Subset Selection Based on Relative Dependency between Attributes, Lecture Notes in Computer Science, Springer (2004) 176-185.
7. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A. I.: Finding interesting rules from large sets of discovered association rules, Proc. of Internal Conf. on Information and Knowledge Management (1994) 401-407.
8. Li, J. and Cercone, N: Discovering and ranking important rules, Proc. of IEEE International Conference on Granular Computing 2 (2005) 506-511.
9. Li, W., Han, J., and Pei, J.: CMAR: Accurate and efficient classification based on multiple class-association rules, Proc. of IEEE Internal. Conf. on Data Mining (2001) 369-376.
10. Lin, T. Y.: Mining Associations by Linear Inequalities, Proc. of IEEE Internal. Conf. on Data Mining (2004) 154-161.
11. Liu, B., Hsu, W., and Ma, Y.: Integrating Classification and Association Rule Mining, Proc. of ACM Internal. Conf. on Knowledge Discovery and Data Mining (1998) 80-86.

12. Michalski, R.: Pattern recognition as rule-guided induction inference, IEEE Trans. On Pattern Analysis and Machine Intelligence 2 (1980) 349-361.
13. Nguyen, H. and Nguyen, S.: Some efficient algorithms for rough set methods, Proc. of IPMU (1996) 1451-1456.
14. Øhrn, A.: *ROSETTA Technical Reference Manual.*, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway, May 2001.
15. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, Dordrecht: Kluwer Academic Publishers (1991).
16. Skowron, C.: The discernibility matrices and functions in information systems, Decision Support by Experience, R. Slowinski (ed.) (1992) 331-362.
17. Szczuka, M. S.: Rules as attributes in classifier construction, Proc. of RSFDGrC (1999) 492-499.
18. Yao, Y., Zhao, Y., and Wang, J.: On reduct construction algorithms, in *Proceedings of RSKT' 06*, Lecture Notes in Artificial Intelligence 4062, Springer (2006) 297-304.

# Evaluating Importance of Conditions in the Set of Discovered Rules

Salvatore Greco[1], Roman Słowiński[2,3], and Jerzy Stefanowski[2]

[1] Faculty of Economics, University of Catania, Corso Italia 55, 95129 Catania, Italy
salgreco@unict.it
[2] Institute of Computing Science, Poznań University of Technology,
60-965 Poznań, Poland
{jerzy.stefanowski,roman.slowinski}@cs.put.poznan.pl
[3] Institute of System Research, Polish Academy of Sciences, 00-477 Warsaw, Poland

**Abstract.** We propose a new method for evaluating the contribution that elementary conditions give to a set of rules. It is based on previous results concerning importance and interaction of elementary conditions for a confidence of a single rule. The usefulness of the method is shown on two rule discovery problems.

**Keywords:** Knowledge Discovery, Rule Evaluation, Fuzzy Measures.

## 1 Introduction

An evaluation of knowledge discovered from data raises quite different issues depending on the application perspective. This paper concerns knowledge represented in a form of "*if* ... *then* ..." rules. If the rules are applied in a *classification perspective* to predict class labels for new objects, then the evaluation refers to a complete set of rules and its prediction ability is estimated usually by a single measure, e.g., the *classification accuracy*. In the other, *descriptive* perspective, each rule is evaluated *individually* as a possible representative of an 'interesting' pattern. This is definitely a more difficult task. Depending on the rule induction algorithm, the user may receive quite a large number of rules to interpret. Selecting some of them is a non-trivial issue and is partly subjective as it generally depends on the context of application and on the interests and expertise of users. To support users, several *quantitative measures* have been proposed and studied, each of them capturing different characteristics of rules. Many of these measures characterize relationships between (condition and decision) parts of the rule and the data set from which the rule is discovered. Generality, support, confidence, logical sufficiency or necessity are examples of popular measures. For their systematic reviews the reader is referred to, e.g., [6,11]. Another class of interestingness measures, called confirmation measures, is also presented in [4].

Let us notice that in this perspective, the evaluation concerns a complete set of elementary conditions in the "*if*" part of each rule. Yet another but interesting issue concerns evaluation of the importance of each single condition or even interactions among conditions in "*if*" part of the rule. The question about the

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 314–321, 2007.

role of subsets of elementary conditions could also be extended from a single rule to a set of rules. We already met an attempt to answering this question in some medical applications, e.g. [9], where the medical experts, first, selected rules and, then, interpreted their syntax to identify combinations of attribute-values being the most characteristic for the patient diagnosis classes.

The above problem has not systematically been studied yet in the literature. To approach it, we focused our attention on special *set functions*, as Shapley and Banzhaf indices or the Möbius representation [2,8]. Originally, they were considered in voting systems, game theory or multiple criteria decision aid. Moreover, Greco *et al.* used them within them rough set theory to study the relative value of information supplied by different attributes to the quality of approximation of object classification [3]. The first attempt to adapt the above set functions for evaluating an importance of rule elementary conditions was presented in [5]. However, it was restricted to a confidence measure and to a single rule only.

The aim of this paper is to introduce a method for studying the importance and the interactions of elementary conditions in a set of decision rules. Moreover, besides the confidence of rules, their support will be taken into account. The second aim is to carry out an experimental evaluation of this method on real life rule discovery problems which were previously analysed by experts.

## 2 Using Set Indices to Evaluate Rule Conditions

### 2.1 The Method for Analysing Confidence of a Single Rule

The considered evaluation measures are based on set functions which originally referred to elements in the finite set $X$ – these elements could be either players in a game, voters in an election, or criteria in a multiple criteria decision problem. Let $P(X)$ denote the power set of $X$, i.e. the set of all subsets of $X$. In the following we consider a *set function* $\mu$: $P(\mathrm{X})\rightarrow$ [0,1]. Function $\mu(\cdot)$ is a *fuzzy measure* (capacity) on $X$ if it satisfies the following requirements: (1) $\mu(\emptyset) = 0$, $\mu(X) = 1$; (2) $A \subseteq B$ implies $\mu(A) \leq \mu(B)$, for all $A, B \in P(X)$. In the following we relax the condition (1) in the part that $\mu(X) = 1$ and condition (2), such that we consider simply a set function $\mu$: $P(\mathrm{X})\rightarrow$ [0,1].

The function $\mu(A)$ has a particular interpretation within the respective theory, e.g. in a multiple criteria decision problem, $\mu(A)$ is interpreted as the *conjoint importance* of criteria from $A \subseteq X$. Some specific indices are defined on the basic functions $\mu$. In previous papers [3,5], it was shown that the most important are the Shapley and Banzhaf values for single elements $i \in X$, their interaction indices for subsets of elements $A \subseteq X$, and the set function $m : P(X) \rightarrow R$, called Möbius representation of $\mu$.

Let us introduce a basic notation. Learning examples are represented in a *decision table* $DT =(U, A\cup\{d\})$, where $U$ is a set of objects, $A$ is a set of condition attributes describing them, and $d \notin A$ is a decision attribute that partitions examples into a set of disjoint classes $\{K_j : j =1,...,k\}$. We assume that decision rule $r$ assigning objects to class $K_j$ is represented in the following form: *if* $P$ *then* $Q$, where $P = p_1 \wedge p_2 \wedge \ldots \wedge p_n$ is a *condition part* of the rule and

$Q$ is a *decision part* of the rule indicating that an object should be assigned to class $K_j$. The *elementary condition* $p_i$ of the rule $r$ is a test on a value of a given attribute, e.g. an attribute value is equal to a constant.

While calculating set indices, for given rule $r$: *if* $(p_1 \wedge p_2 \wedge \ldots \wedge p_n)$ *then* $Q$, one has to consider additional "sub"-rules *if* $(p_{j_1} \wedge p_{j_2} \wedge \ldots \wedge p_{j_m})$ *then* $Q$ constructed by using subsets of its conditions $\{p_{j_1}, p_{j_2}, \ldots, p_{j_m}\} \subseteq \{p_1, p_2, \ldots, p_n\}$. Such rules will be called generalizations of $r$.

The confidence of the rule was chosen in [5] as the basic function $\mu$. It is one of the most frequently used rule evaluation measures, specifying the credibility of the consequence relation represented by the rule [1,11]. The confidence of the rule $r$ is defined as a ratio of a number of examples satisfying both condition $P$ and decision $Q$ to a number of examples satisfying $P$ only. Let $\mu(W, K_j) = \mathit{confidence}(r)$, where $W$ is the set of all $n$ conditions in this rule. Let also $Y \subset W$ be a subset of conditions in the rule being a generalization of $r$. We assume that the function for the rule with an empty condition part is $\mu(\emptyset, K_j) = 0$. Let us first present indices for every elementary condition $p_i \in W$ in a single rule $r$. The Shapley value [8] for it is defined as:

$$\phi_S(p_i, r) = \sum_{Y \subseteq W - \{p_i\}} \frac{(n-|Y|-1)!|Y|!}{n!} \cdot [\mu(Y \cup \{p_i\}, K_j) - \mu(Y, K_j)],$$

where $|\cdot|$ denotes the cardinality of the set. The Banzhaf value [2] is defined as:

$$\phi_B(p_i, r) = \frac{1}{2^{n-1}} \sum_{Y \subseteq W - \{p_i\}} \left[\mu(Y \cup \{p_i\}, K_j) - \mu(Y, K_j)\right].$$

Both indices are calculated using information about an average contribution of condition $p_i$ to all generalized rules of $r$ constructed by using possible subsets of its conditions. They can be interpreted as measures of the contribution of elementary condition $p_i$, $i = 1, \ldots, n$, to the confidence of rule $r$. In the case of $\phi_S(p_i, r)$ the value of $\mu(W)$ is shared among all elements, i.e. $\sum_{i=1}^{n} \phi_S(p_i, r) = \mu(W)$, while an analogous property does not hold for $\phi_B(p_i, r)$.

Other indices, as $I_{MR}(p_i, p_j)$, were introduced to measure the interaction between pairs of conditions [5]. Their values are interpreted in the following way: if $I_{MR}(p_i, p_j) > 0$ both conditions are complementary, i.e. they interact positively by increasing the confidence of the rule, while if $I_{MR}(p_i, p_j) < 0$, then it means that they are interchangeable, i.e. putting them together provide partly redundant information. Below we present their generalized versions for subsets of conditions in rule $r$. The Shapley index for a subset $V \subseteq W$ is defined as:

$$\mathrm{I}_S(V, r) = \sum_{Y \subseteq W - V} \frac{(n-|Y|-|V|)!|Y|!}{(n-|V|+1)!} \sum_{L \subseteq V} (-1)^{|V|-|L|} \mu(Y \cup L, K_j).$$

and the Banzhaf index of conditions from $V \subseteq W$ is defined as:

$$\mathrm{I}_B(V, r) = \frac{1}{2^{n-|V|}} \sum_{Y \subseteq W - V} \sum_{L \subseteq V} (-1)^{|V|-|L|} \mu(Y \cup L, K_j).$$

These indices are interpreted as the average conjoint contribution of the subset of conditions $V \subset W$ to the confidence of all rules generalized from rule $r$ such that $V \cup Z = \emptyset$, where $Z$ is a subset of conditions in the *if* part of any of these generalizations.

Finally, another interpretation of the conjoint contribution of conditions from the subset $V$ to the confidence of the rule $r$ (i.e. with all conditions from $W$) is provided by the *Möbius* representation of set function $\mu$, which is defined as:

$$m(V,r) = \sum_{B \subseteq V} (-1)^{|V-B|} \mu(B, K_j)$$

### 2.2 An Example of Analysing a Single Rule Analysis

To evaluate the usefulness of the described method we chose two real applications of rule induction, where an experts' interpretation of syntax of many rules and even the discussion of characteristic attribute values for decision classes were available. The first problem, described in [10], concerns technical diagnostics of the homogeneous fleet of buses. The buses were described by 8 diagnostic symptoms (attributes) and divided into two classes depending on their technical conditions (good or bad). As described in [10], the continuous valued attributes were discretized and, among other methods, the algorithm *Explore* was applied to induce the set of 28 rules. For each of these rules, we applied our method and interpret the results. Due to page limit we can show few examples only. In the following table, for each generalized rule (its use of conditions is represented in a binary way) we report the appropriate evaluation indices.

Rule no. 4: *if (torque = high)* $\wedge$ *(summer fuel consumption = acceptable) then (technical condition = good)* with confidence = 1 and support = 3 examples.

| $p_1$ | $p_2$ | Mobius | Banzhaf | Shapley | Confidence |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0.1667 | 0.0942 | 0.0942 | 0.1667 |
| 1 | 0 | 0.978 | 0.906 | 0.906 | 0.978 |
| 1 | 1 | -0.1449 | -0.1449 | -0.1449 | 1 |

According to all measures, condition $p_1$ (*torque = high*) is nearly 10 times more important than $p_2$ (*summer fuel consumption = acceptable*). It is consistent with previous results [10] and common sense knowledge saying that a high value of the engine torque is definitely a better symptom of the good technical condition of the bus than acceptable buses fuel consumption. Adding both conditions together results in the highest rule confidence. However, there is a kind of small redundancy provided by putting them together – what is seen by negative values of indices $I_S(\{p_1,p_2\},r), I_B(\{p_1,p_2\},r), m(\{p_1,p_2\},r)$. Let us also consider the other rule:

Rule no. 16: *if (torque = high)* $\wedge$ *(compression pressure = high) then (technical condition = good* with confidence = 1 and support = 46 examples.

| $p_1$ | $p_2$ | Mobius | Banzhaf | Shapley | Confidence |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0.9787 | 0.5384 | 0.5384 | 0.9787 |
| 1 | 0 | 0.9019 | 0.4616 | 0.4616 | 0.9019 |
| 1 | 1 | -0.8807 | -0.8807 | -0.8807 | 1 |

One can notice that both conditions considerably contribute to the confidence of this rule. It is consistent with the discussion from [10] where the compression pressure was identified as the best symptom of the technical state. Moreover, it was previously noticed that both symptoms were highly correlated (Pearson $r$ coefficient = 0.93) - what is directly seen by negative values of all indices.

## 3 Evaluating Conditions in a Set of Rules

Until now, we considered the contribution of elementary conditions to the confidence of the single rule. The choice of the rule could be either an expert's decision or a result of a rule filtering method. However, for many data sets one usually receives a multi-class set of rules. A given condition or a subset of elementary condition may occur in the condition parts of many of these rules. Thus, a challenge is to analyse the importance and interaction of elementary conditions in the entire set of rules. Intuitively, a "good" elementary condition should be highly evaluated in all, or nearly all, rules containing it, rather than in a single rule only. Moreover, we should reward these elementary conditions or their subsets which are *characteristic for a given class*, i.e. they only, or mainly, occur in rules from this class while being nearly absent in rules from other classes.

As rules are not equally important inside the considered set we should take into account the other measures to discriminate them. According to literature (see e.g. [11,1]) the confidence is usually considered together with the rule support. The *rule support*, denoted by $sup(r)$, is calculated as a relative ratio of a number of learning examples satisfying both condition and decision part of a rule to a total number of examples.

Let $R = \{r_1, r_2, \ldots, r_m\}$ be a set of rules induced from the learning examples in $DT$. $R = \bigcup_{j=1}^{k} R(K_j)$, where $R(K_j)$ includes rules assigning objects to class $K_j$, $j = 1, \ldots, k$. Let us assume that we are interested in evaluating a non-empty subset of conditions, denoted by $\Gamma_f$, occurring in at least one rule $r_l \in R$, $l = 1, \ldots, m$. Let $FM_{r_l}(\Gamma_f)$ denote an evaluation of its contribution to the confidence of rule $r_l$, calculated according to one of considered indices: Shapley, Banzhaf or Möbius. The global contribution of $\Gamma_f$ in rule set $R$ with respect to the class $K_j$ is calculated by the following weighted aggregation formula:

$$G_{K_j}(\Gamma_f) = \sum_{r \in R(K_j)} FM_r(\Gamma_f) \cdot sup(r) - \sum_{s \in \neg R(K_j)} FM_s(\Gamma_f) \cdot sup(s),$$

where $s \in \neg R(K_j)$ denotes rules indicating other classes than $K_j$. So, value $G_{K_j}(\Gamma_f)$ means the global importance of the set of conditions $\Gamma_f$ for class $K_j$, which is decreased by the component corresponding to occurrence of $\Gamma_f$ in rules concerning other classes.

For a given set of rules one gets rankings of elementary conditions or their subsets for each class $K_j$, ordered according to the calculated values $G_{K_j}(\Gamma_f)$. The highest positions in these rankings can be interpreted as the characteristic description of the given class by means of the elementary condition subsets being

**Table 1.** Rankings of best conditions according to evaluation measures calculated for "buses" rules

| busses in a good technical condition | | | | | |
|---|---|---|---|---|---|
| Möbius | | Shapley | | Banzhaf | |
| condition | value | condition | value | condition | value |
| comp-press=high | 214.34 | comp-press=high | 116.91 | comp-press=high | 116.91 |
| torque=high | 163.36 | torque=high | 163.36 | torque=high | 163.36 |
| blacking=low | 161.33 | blacking=low | 87.86 | blacking=low | 87.86 |
| oil cons.=low | 132.36 | oil cons.=low | 70.88 | oil cons.=low | 70.80 |
| MaxSpeed=high | 122.66 | MaxSpeed=high | 63.71 | MaxSpeed=high | 63.71 |
| busses in a bad technical condition | | | | | |
| Möbius | | Shapley | | Banzhaf | |
| condition | value | condition | value | condition | value |
| torque=low | 48.33 | torque=low | 29.17 | torque=low | 29.17 |
| blacking=high | 46.70 | comp-press=low | 29.00 | comp-press=low | 29.00 |
| comp-press=low | 29.00 | blacking=high | 27.98 | blacking=high | 28.06 |
| oil-cons.=high | 27.00 | oil-cons.=high | 27.00 | oil-cons.=high | 27.00 |
| summ-cons.=high | 26.67 | horsepower=low | 26.00 | horsepower=low | 26.66 |
| horsepower=low | 26.66 | MaxSpeed=low | 25 | MaxSpeed=low | 25 |

the most important in rules concerning this class while not contributing too much to rules concerning other classes.

From the computational point of view, the crucial issue is generating the list of possible subsets of conditions. They are identified as occurring in rule set $R(K_j)$, however the number of their possible combinations may be high. Thus, in our implementation we used a simple heuristic. First, we consider single conditions only and calculate their values $G_{K_j}(\Gamma_f)$. Then, we combine the single conditions in larger sets (pairs, triples, ...). However increasing the size of the subset in the next phase is allowed only if, evaluations from the previous phase are higher than the user's defined threshold.

## 4 Computational Experiments with Sets of Rules

Let us come back to the diagnostic examination of buses [10]. Calculated values $G_{K_j}$ of all three indices for single elementary conditions in the set of rules are presented in Table 1.

Let us observe that these orders are highly consistent with previous analysis of single attribute significance. According to [10] (both rough sets and statistical analysis) the most valuable was the compression pressure, then torque, maximum speed and engine horsepower. Blacking components in the exhaust gas and oil consumption were found as more important than fuel consumption. Moreover, it was mentioned that experts indicated high compression pressure, high torque, acceptable maximal horsepower to be characteristic for the good technical conditions. Experts also found out that their opposite value were characteristic for bad technical conditions.

The evaluation of condition pairs are close to 0 or negative. For instance, one of the best evaluated pairs for good buses is (*horsepower=average*) $\wedge$ (*oil consumption=low*) with Shapley index -0.1662. In the previous analysis, these

attributes were not correlated, while some others were strongly correlated. This is also detected by our method, e.g. (*blacking=low*) $\wedge$ (*MaxSpeed=average*) with Shapley index -38.71 – these attributes were correlated in degree 0.9. We did not analyse triples of conditions as all pairs were low evaluated, which was also consistent with expert's previous opinion saying that in this problem single conditions or their pairs were sufficient to diagnose the technical condition of buses.

A similar analysis was performed for a medical problem concerning diagnosing an *anterior cruciate ligament* (ACL) *rupture* in a knee on the basis of magnetic resonance images [9]. Due to limited size of this paper we have to skip detailed results and show only the most characteristic conditions for each class (To be short we show the Shapley value only). For the class "*patients with injury*" the best conditions are: (PCL index $< 3.225$) $\Phi_S = 26.0$; (PCL index $\in [3.225,3.71)$) $\Phi_S = 6.04$, (age $\in [16.5,35)$) $\Phi_S = 2.68$; (sex=male) $\Phi_S = 1.22$. While for the class "*patients without injury*" the order is: (PCL index $\geq 4.535$) $\Phi_S = 75.0$; (age $< 16.5$)) $\Phi_S = 14.0$; (sex=female) $\Phi_S = 9.19$.

Again, this result is highly consistent with the previous clinical discussion [9] indicating that two attributes (PCL index, age) and their specific values should support a physician in resigning from performing arthroscopy for some patients. PCL index is a main coefficient constructed from measurements taken from image and is a crucial one for detecting this kind of knee injury. The role of age and sex is also justified as ACL is typical injury of male sportsmen. Moreover, in this problem, the importance of pairs of conditions is higher evaluated than in the buses problem, although it is still slightly lower than single conditions, e.g. for healthy patients the best pair is $(X1 \in [11.75,14.5)) \wedge (Y1 \in [2.75,3.75))$ with Shapley value 1.37, where $X1$ and $Y1$ are two basic distances between some knee parts measured in the image, so they are the basic components inside the PCL index formula.

## 5 Discussion and Final Remarks

The novelty of this study consists in focusing a user attention on the role of subsets of elementary conditions in rules discovered from data, which according to our best knowledge has not been studied sufficiently yet in the literature on the knowledge evaluation. We have described the method for evaluating the contribution that elementary conditions give to a confidence of a single decision rule. It is based on an adaptation of some fuzzy measures introduced in the literature in a different context. Moreover, we extend this method in a kind of aggregation approach to study the importance of conditions in the sets of rules, where the rule support was also considered.

We summarize results of two applications, where we identified the rankings of the importance of elementary conditions in the sets of rules. Although we could show only the main part of obtained results, we claim that they are consistent with published experts' conclusions [10,9]. These rankings are useful for constructing a characteristic description for each decision class. Three indices were studied. We carried out more experiments with other rule sets, and it turned

out that Shapley or Baznhaf values were very similar or the same. Differences rose for larger rule sets containing rules with more conditions. However, the highest positions in rankings were usually the same. Comparing them to Möbius rankings we observed a wider range of their values.

As to disadvantages we should mention the complexity of fuzzy measures calculations. They are burdened by exponential costs as for each subset of conditions $V$ it is necessary to consider all subsets $W - V$ in generalized rules. Shapley and Banzhaf indices require even more additional calculations than Möbius index. Because of these costs, for larger sets of rules one could use the two phase scenario. In the first phase, the user should filter rules and then apply the method to the reduced number of the most valuable rules only.

**Acknowledgement.** We are grateful to Bartosz Jędrzejczak for the cooperation and preparing a software implementation of the method in the framework of the Master Thesis.

## References

1. Bayardo R., Agrawal R., Mining the most interesting rules. In *Proc. 5th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, 1999, 145-154.
2. Banzhaf, J. F., Weighted voting doesn't work: A mathematical analysis, *Rutgers Law Review*, 19, 1965, 317-343.
3. Greco S., Matarazzo B., Slowinski R., Fuzzy measures as a technique for rough set analysis. In *Proc. 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98)*, Aachen, September 7-10, 1998, 99-103.
4. Greco S., Pawlak Z., Slowinski R., Can Bayesian confirmation measures be useful for rough set decision rules? *Engineering Applications of Artificial Intelligence*, **17**, 2004, 345–361
5. Greco S., Matarazzo B., Slowinski R., Stefanowski J., Importance and interactions of conditions in decision rules. In *Proc. 3rd Int. Conference RSCTC 2002*, Springer Verlag, LNAI no. 2475, 2002, 255-262.
6. Hilderman R.J., Hamilton H.J, *Knowledge Discovery and Measures of Interest.* Kluwer Academic Boston, 2002.
7. Jedrzejczak B., *An evaluation of conditions in decision rules.* Master Thesis (supervisor J.Stefanowski), Poznań Univeristy of Technology, June 2006.
8. Shapley L.S., A value for n-person games. In Kuhn H.W., Tucker A.W. (eds), *Contributions to the Theory of Games II*, Princeton University Press, Princeton, 1953, 307-317.
9. Slowinski K., Stefanowski J., Siwinski D., Application of rule induction and rough sets to verification of magnetic resonance diagnosis. *Fundamenta Informaticae*, **53** (no. 3/4), 2002, 345-363.
10. Slowinski R., Stefanowski J., Susmaga R., Rough set analysis of attribute dependencies in technical diagnostics. In Tsumoto S. et al. (eds), *Proc. 4th Int. Workshop on Rough Sets, Fuzzy Sets and Machine Discovery*, Tokyo, 1996, 284-291.
11. Yao Y.Y., Zhong N., An analysis of quantitative measures associated with rules. In *Proc. of the 3rd Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, LNAI no. 1574, 1999, 479-488.

# Constraint Based Action Rule Discovery with Single Classification Rules

Angelina Tzacheva[1] and Zbigniew W. Raś[2,3]

[1] University of South Carolina Upstate, Department of Informatics, Spartanburg, SC 29303
[2] University of North Carolina at Charlotte, Department of Computer Science, Charlotte, N.C. 28223
[3] Polish-Japanese Institute of Information Technology, 02-008 Warsaw, Poland

**Abstract.** Action rules can be seen as an answer to the question: what one can do with results of data mining and knowledge discovery? Some applications include: medical field, e-commerce, market basket analysis, customer satisfaction, and risk analysis. Action rules are logical terms describing knowledge about possible actions associated with objects, which is hidden in a decision system. Classical strategy for discovering them from a database requires prior extraction of classification rules which next are evaluated pair by pair with a goal to suggest an action, based on condition features in order to get a desired effect on a decision feature. An actionable strategy is represented as a term $r = [(\omega) \wedge (\alpha \rightarrow \beta)] \Rightarrow [\phi \rightarrow \psi]$, where $\omega$, $\alpha$, $\beta$, $\phi$, and $\psi$ are descriptions of objects or events. The term $r$ states that when the fixed condition $\omega$ is satisfied and the changeable behavior $(\alpha \rightarrow \beta)$ occurs in objects represented as tuples from a database so does the expectation $(\phi \rightarrow \psi)$. With each object a number of actionable strategies can be associated and each one of them may lead to different expectations and the same to different reclassifications of objects. In this paper we will focus on a new strategy of constructing action rules directly from single classification rules instead of pairs of classification rules. It presents a gain on the simplicity of the method of action rules construction, as well as on its time complexity. We present A*-type heuristic strategy for discovering only interesting action rules, which satisfy user-defined constraints such as: feasibility, maximal cost, and minimal confidence. We, therefore, propose a new method for fast discovery of interesting action rules.

## 1 Introduction

There are two aspects of interestingness of rules that have been studied in data mining literature, objective and subjective measures [1], [5]. Objective measures are data-driven and domain-independent. Generally, they evaluate the rules based on their quality and similarity between them. Subjective measures, including unexpectedness, novelty and actionability, are user-driven and domain-dependent.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 322–329, 2007.

Action rules, introduced in [6] and investigated further in [10], [11], [8], are constructed from certain pairs of association rules. Interventions, defined in [3], are conceptually very similar to action rules.

The notion of a cost of an action rule, which is a subjective measure, was introduced in [11]. It is associated with changes of values of classification attributes in a rule. The strategy for replacing the initially extracted action rule by a composition of new action rules, dynamically built and leading to the same reclassification goal, was proposed in [11]. This composition of rules uniquely defines a new action rule. Objects supporting the new action rule also support the initial action rule but the cost of reclassifying them is lower or even much lower for the new rule. In [8] authors propose a new simplified strategy for constructing action rules. This paper presents a heuristic strategy for discovering interesting action rules which satisfy user-defined constraints such as: feasibility, maximal cost, and minimal confidence. There is a similarity between the rules generated by Tree-Based Strategy [10] and rules constructed by this new method.

## 2 Action Rules

In the paper by [6], the notion of an action rule was introduced. The main idea was to generate, from a database, special type of rules which basically form a hint to users showing a way to re-classify objects with respect to some distinguished attribute (called a decision attribute). Values of some attributes, used to describe objects stored in a database, can be changed and this change can be influenced and controlled by user. However, some of these changes (for instance *profit* can not be done directly to a decision attribute. In such a case, definitions of this decision attribute in terms of other attributes (called classification attributes) have to be learned. These new definitions are used to construct action rules showing what changes in values of some attributes, for a given class of objects, are needed to re-classify these objects the way users want. But, users may still be either unable or unwilling to proceed with actions leading to such changes. In all such cases, we may search for definitions of a value of any classification attribute listed in an action rule. By replacing this value of attribute by its definition extracted either locally or at remote sites (if system is distributed), we construct new action rules which might be of more interest to users than the initial rule [11]. We start with a definition of an information system given in [4].

By an information system we mean a pair $S = (U, A)$, where:

1. $U$ is a nonempty, finite set of objects (object identifiers),
2. $A$ is a nonempty, finite set of attributes i.e. $a : U \rightarrow V_a$ for $a \in A$, where $V_a$ is called the domain of $a$.

Information systems can be seen as decision tables. In any decision table together with the set of attributes a partition of that set into conditions and decisions is given. Additionally, we assume that the set of conditions is partitioned into stable and flexible [6].

Attribute $a \in A$ is called stable for the set $U$ if its values assigned to objects from $U$ can not be changed in time. Otherwise, it is called flexible. *Place of birth*

is an example of a stable attribute. *Interest rate* on any customer account is an example of a flexible attribute. For simplicity reason, we consider decision tables with only one decision. We adopt the following definition of a decision table:

By a decision table we mean an information system $S = (U, A_{St} \cup A_{Fl} \cup \{d\})$, where $d \notin A_{St} \cup A_{Fl}$ is a distinguished attribute called the decision. The elements of $A_{St}$ are called stable conditions, whereas the elements of $A_{Fl} \cup \{d\}$ are called flexible. Our goal is to change values of attributes in $A_{Fl}$ for some objects in $U$ so the values of attribute $d$ for these objects may change as well. Certain relationships between attributes from $A_{St} \cup A_{Fl}$ and the attribute $d$ will have to be discovered first.

By $Dom(r)$ we mean all attributes listed in the *IF* part of a rule $r$ extracted from $S$. For example, if $r = [(a_1, 3) \wedge (a_2, 4) \rightarrow (d, 3)]$ is a rule, then $Dom(r) = \{a_1, a_2\}$. By $d(r)$ we denote the decision value of rule $r$. In our example $d(r) = 3$.

If $r_1, r_2$ are rules and $B \subseteq A_{Fl} \cup A_{St}$ is a set of attributes, then $r_1/B = r_2/B$ means that the conditional parts of rules $r_1, r_2$ restricted to attributes $B$ are the same. For example if $r1 = [(a1, 3) \rightarrow (d, 3)]$, then $r1/\{a1\} = r/\{a1\}$.

We assume that $(a, v \rightarrow w)$ denotes the fact that the value of attribute $a$ has been changed from $v$ to $w$. Similarly, the term $(a, v \rightarrow w)(x)$ means that the property $(a, v)$ of an object $x$ has been changed to property $(a, w)$.

Assume now that rules $r_1, r_2$ are extracted from $S$ and

$r_1/[Dom(r_1) \cap Dom(r_2) \cap A_{St}] = r_2/[Dom(r_1) \cap Dom(r_2) \cap A_{St}]$, $d(r_1) = k_1$, $d(r_2) = k_2$. Also, assume that $(b_1, b_2, \ldots, b_p)$ is a list of all attributes in $Dom(r_1) \cap Dom(r_2) \cap A_{Fl}$ on which $r_1, r_2$ differ and $r_1(b_1) = v_1$, $r_1(b_2) = v_2, \ldots$, $r_1(b_p) = v_p$, $r_2(b_1) = w_1$, $r_2(b_2) = w_2, \ldots$, $r_2(b_p) = w_p$.

By $(r_1, r_2)$-action rule we mean statement $r$:

$[r_2/A_{St} \wedge (b_1, v_1 \rightarrow w_1) \wedge (b_2, v_2 \rightarrow w_2) \wedge \ldots \wedge (b_p, v_p \rightarrow w_p)] \Rightarrow [(d, k_1 \rightarrow k_2)]$.

Object $x \in U$ supports action rule $r$, if $x$ supports the description $[r_2/A_{St} \wedge (b_1, v_1) \wedge (b_2, v_2) \wedge \ldots \wedge (b_p, v_p) \wedge (d, k_1)]$. The set of all objects in $U$ supporting $r$ is denoted by $U^{<r>}$. The term $r_2/A_{St}$ is called the header of action rule.

Extended action rules, introduced in [10], form a special subclass of action rules. We construct them by extending headers of action rules in a way that their confidence is getting increased. The support of extended action rules is usually lower than the support of the corresponding action rules.

## 3 Action Rule Discovery from Single Classification Rule

Let us assume that $S = (U, A_{St} \cup A_{Fl} \cup \{d\})$ is a decision system, where $d \notin A_{St} \cup A_{Fl}$ is a distinguished attribute called the decision. Assume also that $d_1 \in V_d$ and $x \in U$. We say that $x$ is a $d_1$-object if $d(x) = d_1$. Finally, we assume that $\{a_1, a_2, ..., a_p\} \subseteq A_{Fl}$, $\{b_1, b_2, ..., b_q\} \subseteq A_{St}$, $a_{[i,j]}$ denotes a value of attribute $a_i$, $b_{[i,j]}$ denotes a value of attribute $b_i$, for any $i, j$ and that

$r = [[a_{[1,1]} \wedge a_{[2,1]} \wedge ... \wedge a_{[p,1]}] \wedge [b_{[1,1]} \wedge b_{[2,1]} \wedge ..... \wedge b_{[q,1]}] \rightarrow d_1]$

is a classification rule extracted from $S$ supporting some $d_1$-objects in $S$. By $sup(r)$ and $conf(r)$ we mean *support* and *confidence* of $r$, respectively. Class $d_1$ is a preferable class and our goal is to reclassify $d_2$-objects into $d_1$ class, where $d_2 \in V_d$.

By an action rule $r_{[d2\rightarrow d1]}$ associated with $r$ and the reclassification task $(d, d_2 \rightarrow d_1)$ we mean the following expression [8]:

$r_{[d2\rightarrow d1]} = [[a_{[1,1]} \wedge a_{[2,1]} \wedge ... \wedge a_{[p,1]}] \wedge$

$[(b_1, \rightarrow b_{[1,1]}) \wedge (b_2, \rightarrow b_{[2,1]}) \wedge ..... \wedge (b_q, \rightarrow b_{[q,1]})] \Rightarrow (d, d_2 \rightarrow d_1)]$.

In a similar way, by an action rule $r = [\rightarrow d_1]$ associated with $r$ and the reclassification task $(d, \rightarrow d_1)$ we mean the following expression:

$r_{[\rightarrow d1]} = [[a_{[1,1]} \wedge a_{[2,1]} \wedge ... \wedge a_{[p,1]}] \wedge$

$[(b_1, \rightarrow b_{[1,1]}) \wedge (b_2, \rightarrow b_{[2,1]}) \wedge ..... \wedge (b_q, \rightarrow b_{[q,1]})] \Rightarrow (d, \rightarrow d_1)]$.

The term $[a_{[1,1]} \wedge a_{[2,1]} \wedge ... \wedge a_{[p,1]}]$, built from values of stable attributes, is called the header of action rule and its values can not be changed.

The support set of the action rule $r_{[d_2\rightarrow d_1]}$ is defined as:

$Sup(r_{[d_2\rightarrow d_1]}) = \{x \in U : (a_1(x) = a_{[1,1]}) \wedge (a_2(x) = a_{[2,1]}) \wedge ... \wedge (a_p(x) = a_{[p,1]}) \wedge (d(x) = d_2)\}$.

In the following paragraph we show how to calculate the confidence of action rules. Let $r_{[d_2\rightarrow d_1]}, r'_{[d_2\rightarrow d_3]}$ are two action rules extracted from $S$. We say that these rules are $p$-equivalent ($\approx$), if the condition given below holds for every $b_i \in A_{Fl} \cup A_{St}$:

if $r/b_i$, $r'/b_i$ are both defined, then $r/b_i = r'/b_i$.

Let us take $d_2$-object $x \in Sup(r_{[d_2\rightarrow d_1]})$. We say that $x$ positively supports $r_{[d_2\rightarrow d_1]}$ if there is no classification rule $r'$ extracted from $S$ and describing $d_3 \in V_d$, $d_3 \neq d_1$, which is $p$-equivalent to $r$, such that $x \in Sup(r'_{[d_2\rightarrow d_3]})$. The corresponding subset of $Sup(r_{[d_2\rightarrow d_1]})$ is denoted by $Sup^+(r_{[d_2\rightarrow d_1]})$. Otherwise, we say that $x$ negatively supports $r_{[d_2\rightarrow d_1]}$. The corresponding subset of $Sup(r_{[d_2\rightarrow d_1]})$ is denoted by $Sup^-(r_{[d_2\rightarrow d_1]})$. By the confidence of $r_{[d_2\rightarrow d_1]}$ in $S$ we mean:

$Conf(r_{[d2\rightarrow d1]}) = [card[Sup^+(r_{[d2\rightarrow d1]})]/card[Sup(r_{[d2\rightarrow d1]})]] \cdot conf(r)$.

## 4 Cost and Feasibility of Action Rules

Depending on the cost of actions associated with the classification part of action rules, business user may be unable or unwilling to proceed with them.

Assume that $S = (X, A, V)$ is an information system. Let $Y \subseteq X$, $b \in A$ is a flexible attribute in $S$ and $b_1, b_2 \in V_b$ are its two values. By $\wp_S(b_1, b_2)$ we mean a number from $(0, +\infty]$ which describes the average cost of changing the attribute value $b_1$ to $b_2$ for any of the qualifying objects in $Y$. These numbers are provided by experts. Object $x \in Y$ qualifies for the change from $b_1$ to $b_2$, if $b(x) = b_1$. If the above change is not feasible, then we write $\wp_S(b_1, b_2) = +\infty$. Also, if $\wp_S(b_1, b_2) < \wp_S(b_3, b_4)$, then we say that the change of values from $b_1$ to $b_2$ is more feasible than the change from $b_3$ to $b_4$.

Let us assume that

$r = [(b_1, v_1 \rightarrow w_1) \wedge (b_2, v_2 \rightarrow w_2) \wedge \ldots \wedge (b_p, v_p \rightarrow w_p)] \Rightarrow (d, k_1 \rightarrow k_2)$

is an action rule.

By the *cost* of $r$ in $S$, denoted by $cost(r)$, we mean the value $\sum\{\wp_S(v_k, w_k) : 1 \leq k \leq p\}$. We say that $r$ is *feasible*, if $cost(r) < \wp_S(k_1, k_2)$.

Now, let us assume that $R_S[(d, k_1 \rightarrow k_2)]$ denotes the set of action rules in $S$ having the term $(d, k_1 \rightarrow k_2)$ on their decision side. Sometimes, for simplicity reason, attribute $d$ will be omitted. An action rule in $R_S[(d, k_1 \rightarrow k_2)]$ which has the lowest cost value may still be too expensive to be of any help. Let us notice that the cost of an action rule $r = [(b_1, v_1 \rightarrow w_1) \wedge (b_2, v_2 \rightarrow w_2) \wedge \ldots \wedge (b_p, v_p \rightarrow w_p)] \Rightarrow (d, k_1 \rightarrow k_2)$ might be high because of the high cost value of one of its sub-terms in the conditional part of the rule. Let us assume that $(b_j, v_j \rightarrow w_j)$ is that term. In such a case, we may look for an action rule in $R_S$ $[(b_j, v_j \rightarrow w_j)]$, which has the smallest cost value. Assume that

$r_1 = [[(b_{j1}, v_{j1} \rightarrow w_{j1}) \wedge (b_{j2}, v_{j2} \rightarrow w_{j2}) \wedge \ldots \wedge (b_{jq}, v_{jq} \rightarrow w_{jq})] \Rightarrow$
$(b_j, v_j \rightarrow w_j)]$ is such a rule which is also feasible in $S$.

Now, we can compose $r$ with $r_1$ getting a new feasible action rule:

$[(b_1, v_1 \rightarrow w_1) \wedge \ldots \wedge [(b_{j1}, v_{j1} \rightarrow w_{j1}) \wedge (b_{j2}, v_{j2} \rightarrow w_{j2}) \wedge \ldots \wedge$
$(b_{jq}, v_{jq} \rightarrow w_{jq})] \wedge \ldots \wedge (b_p, v_p \rightarrow w_p)] \Rightarrow (d, k_1 \rightarrow k_2)$.

Clearly, the cost of this new rule is lower than the cost of $r$. However, if its support in $S$ gets too low, then such a rule has no value to the user. Otherwise, we may recursively follow this strategy trying to lower the cost of re-classifying objects from the group $k_1$ into the group $k_2$. Each successful step will produce a new action rule which cost is lower than the cost of the current rule. Obviously, this heuristic strategy always ends.

## 5 A*-Type Algorithm for Action Rules Construction

Let us assume that we wish to reclassify objects in $S$ from the class described by value $k_1$ of the attribute $d$ to the class $k_2$.

The term $k_1 \rightarrow k_2$ jointly with its cost $\wp_S(k_1, k_2)$ is stored in the initial node $n_0$ of the search graph G built from nodes generated recursively by feasible action rules taken initially from $R_S$ [(d, $k_1 \rightarrow k_2$)] .

For instance, the rule

$r = [[(b_1, v_1 \rightarrow w_1) \wedge (b_2, v_2 \rightarrow w_2) \wedge \ldots \wedge (b_p, v_p \rightarrow w_p)] \Rightarrow (d, k_1 \rightarrow k_2)]$
applied to the node $n_0 = \{[k_1 \rightarrow k_2, \wp_S(k_1, k_2)]\}$ generates the node
$n_1 = \{[v_1 \rightarrow w_1, \wp_S(v_1, w_1)], [v_2 \rightarrow w_2, \wp_S(v_2, w_2)], \ldots, [v_p \rightarrow w_p, \wp_S(v_p, w_p)]\}$
and from $n_1$ we can generate the node $n_2 = \{[v_1 \rightarrow w_1, \wp_S(v_1, w_1)],$
$[v_2 \rightarrow w_2, \wp_S(v_2, w_2)], \ldots, [v_{j1} \rightarrow w_{j1}, \wp_S(v_{j1}, w_{j1})], [v_{j2} \rightarrow w_{j2}, \wp_S(v_{j2}, w_{j2})],$
$\ldots, [v_{jq} \rightarrow w_{jq}, \wp_S(v_{jq}, w_{jq})], \ldots, [v_p \rightarrow w_p, \wp_S(v_p, w_p)]\}$
assuming that the action rule
$r_1 = [[(b_{j1}, v_{j1} \rightarrow w_{j1}) \wedge (b_{j2}, v_{j2} \rightarrow w_{j2}) \wedge \ldots \wedge (b_{jq}, v_{jq} \rightarrow w_{jq})] \Rightarrow$
$(b_j, v_j \rightarrow w_j)]$ from $R_S[(b_j, v_j \rightarrow w_j)]$ is applied to $n_1$.
This information can be written equivalently as:
$r(n_0) = n_1$, $r_1(n_1) = n_2$, $[r_1 \circ r](n_0) = r_1(r(n_0)) = n_2$.
By $Dom_S(r)$ we mean the set of objects in $S$ supporting $r$.

Search graph $G$ is dynamically built by applying action rules to its nodes. Its initial node $n_0$ contains information given by the user. Any other node $n$ in $G$ shows an alternative way to achieve the same reclassification with a cost that is lower than the cost assigned to all nodes which are preceding $n$ in $G$. Clearly, the

confidence of action rules labelling the path from the initial node to the node $n$ is as much important as the information about reclassification and its cost stored in node $n$.

The $A^{\star}$-type strategy for identifying a node in $G$, built for a desired reclassification of objects in $S$, with a cost possibly the lowest among all the nodes reachable from the node n, was given in [11]. This strategy was controlled by three threshold values: $\lambda_1$ - minimum confidence of action rules, $\lambda_2$ - maximum cost of action rules, and $\lambda_3$ - feasibility of action rules. The last threshold was introduced to control the minimal acceptable decrease in the cost of action rule to be constructed. If the search is stopped by threshold $\lambda_1$, then we do not continue the search along that path. If the search is stopped by threshold $\lambda_2$, then we can either stop or continue the search till it is stopped by threshold $\lambda_1$.

Assume that $N$ is the set of nodes in graph $G$ for $S$ and $n_0$ is its initial node.

For any node $n \in N$, by $F(n) = (Y_n, \{[v_{n,j} \rightarrow w_{n,j}, \wp_S(v_{n,j}, w_{n,j}, Y_n)]\}_{j \in In})$ we mean its domain (set of objects in $S$ supporting $r$, the reclassification steps for objects in $Y_n$ and their cost, all assigned by reclassification function $F$ to the node $n$, where $Y_n \subseteq X$.

The cost of node $n$ is defined as: $cost(n) = \Sigma\{\wp_S(v_{n,j}, w_{n,j}, Y_n) : j \in In\}$.

We say that action rule $r$ is applicable to a node $n$ if:

$[Y_n \cap Dom_S(r) \neq \emptyset]$ and $[(\exists k \in In)[r \in R_S[v_{n,kj} \rightarrow w_{n,k}]]]$.

If node $n_1$ is a successor of node $n$ in $G$ obtained by applying the action rule $r$ to $n$, then $Y_{n_1} = Y_n \cap Dom_S(r)$.

We assume here that the cost function $h(n_i) = \lceil [cost(n, Y_i) - \lambda_2]/\lambda_3 \rceil$ is associated with any node $n_i$ in $G$. It shows the maximal number of steps that might be needed to reach the goal from the node $n_i$.

By $conf(n)$, we mean the confidence of action rule associated with node $n$.

A search node in a graph $G$ associated with node $m$ is a pair

$p(m) = ([conf(m), f(m)], [m, n_1, n_2, n_o])$, where $f(m) = g(m) + h(m)$ and $g(m)$ is the cost function defined as the length of the path $[m, n_1, n_2, n_o]$ in $G$ (without loops) from the initial state $n_o$ to the state $m$.

The search node associated with the initial node $n_o$ of $G$ is defined as $([conf(n_o), f(n_o)], [n_o])$. It is easy to show that $f(m)$ is admissible and never overestimates the cost of a solution through the node $m$.

## 6 New A*-Type Algorithm for Action Rules Construction

In this section we propose a modified version of A*-type heuristic strategy discussed in Section 5 which is based on the method of constructing action rules directly from single classification rules instead of their pairs [8]. It presents a gain on the simplicity of the method of action rules construction, as well as on its time complexity.

First, we introduce the notion of a cost linked with the attribute value itself as $\wp_S(b_1)$, where $b_1 \in V_b$, which again is a number from *(0, +∞]* describing the average cost associated with changing any value of attribute $b$ to value $b_1$.

Next, assume that

$R = [[(a, a_1) \wedge (b, b_1) \wedge (c, c_1) \wedge (e, e_1) \wedge (m, m_1) \wedge (k, k_1) \wedge (n, n_1) \wedge (r, r_1)] \rightarrow (d, d_1)]$ is a classification rule extracted from $S$:

Assume that attributes in $St(R) = \{a, b, c, e\}$ are stable and in $Fl(R) = \{m, k, n, r\}$ flexible. Also, assume that class $d_1 \in V_d$ is of highest preference. The rule $R$ defines the concept $d_1$. Assume that $V_d = \{d_1, d_2, d_3, d_4\}$.

Clearly, there may be other classification rules that define concept $d_1$. We pick the rule which has the lowest total cost on the flexible part, i.e. the sum of cost of all flexible attributes $\sum\{\wp_S(Fl(R)_i) : i = m, k, \ldots, r\}$ is minimal.

Next, we are picking objects from $X$ which have property, let's say, $d_2$ i.e. objects of class $d_2$, which satisfy the header of stable attribute values in $R$:

$Y = \{x : a(x) = a_1, b(x) = b_1, c(x) = c_1, e(x) = e_1, d(x) = d_2\}$

In order to 'grab' these objects into $d_1$, we construct action rule:

$[(a_1 \wedge b_1 \wedge c_1 \wedge e_1] \wedge [(m, \rightarrow m_1) \wedge (k, \rightarrow k_1) \wedge (r, \rightarrow r_1)] \Rightarrow (d, d_2 \rightarrow d_1)$

In other words, if we make the specified changes to the attributes in $Fl(R)$, the expectation is that the objects in $Y$ will move to the desired class $d_1$. Looking at the changes needed, the user may notice that the change $(k, \rightarrow k_1)$ is the worst, i.e. it has the highest cost, and it contributes most to the cost of the sum (total cost) of all changes. Therefore, we may search for new classification rules, which define the concept $k_1$, and compose the feasible action rule $R_1 = [St(R_1)] \wedge [Fl(R_1)]$ which suggests the reclassification to $k_1$ at the lowest cost, where $St(R) \subseteq St(R_1)$. As defined earlier, such action rule will be feasible if the sum (total cost) of all changes on the left hand side of the rule is lower, than the right side. Therefore, the action rule $R_1$ will specify an alternative way to achieve the reclassification to $k_1$ at a cost lower than the currently known cost to the user. Next, we concatenate the two action rules $R$ and $R_1$ by replacing $(k, \rightarrow k_1)$ in $R$, with $[Fl(R_1)]$, and modifying the header to include $St(R) \cup St(R_1)$.

$[(a_1 \wedge b_1 \wedge c_1 \wedge e_1) \wedge St(R_1)] \wedge [(m, \rightarrow m_1) \wedge Fl(R_1) \wedge (r, \rightarrow r_1)] \Rightarrow (D, d_2 \rightarrow d_1)$.

Clearly, there may be many classification rules that we can choose from. We only consider the ones which stable part does not contradict with *St(R)*. Among them, we choose rules with a minimal number of new stable attributes, as each time we add a new stable attribute to the current rule we may decrease the total number of objects in $Y$ which can be moved to the desired class $d_1$. In relation to flexible attributes, they have to be the same on the overlapping part of a new classification rule and the rule $R$. This may further decrease the number of potential objects in $Y$ which can be moved to the desired class $d_1$.

Therefore, we need a heuristic strategy, similar to the one presented in the previous section for classical action rules, to look for classification rules to be concatenated with $R$ and which have the minimal number of new stable attributes in relation to $R$ and minimal number of new flexible attributes jointly with flexible attributes related to the overlapping part with $R$.

We propose a modified version of $A^\star$-algorithm we saw in the previous section. Again, we assume that user will provide the following thresholds related to action rules: $\lambda_1$ - minimum confidence, $\lambda_2$ - maximum cost, and $\lambda_3$ - feasibility.

Clearly, it is expensive to build the complete graph $G$ and next search for a node of the lowest cost satisfying both thresholds $\lambda_1$, $\lambda_2$. The heuristic value associated with a node $n$ in $G$ is defined as $h(n) = \lceil [cost(n) - \lambda_2]/\lambda_3 \rceil$. It shows the maximal number of steps that might be needed to reach the goal. The cost function $g(m)$ is defined as the length of the path in $G$ (without loops) from the initial state $n_o$ to the state $m$. It is easy to show that $f(m) = g(m) + h(m)$ is admissible and never overestimates the cost of a solution through the node $m$.

## 7 Conclusion and Acknowledgements

The new algorithm for constructing action rules of the lowest cost is a significant improvement of the algorithm presented in [11] because of its simplicity in constructing headers of action rules and because the concatenation of action rules is replaced by concatenation of classification rules.

This research was partially supported by the National Science Foundation under grant IIS-0414815.

## References

1. Adomavicius, G., Tuzhilin, A. (1997) Discovery of actionable patterns in databases: the action hierarchy approach, in **Proceedings of KDD′97 Conference**, Newport Beach, CA, AAAI Press
2. Hilderman, R.J., Hamilton, H.J. (2001) **Knowledge Discovery and Measures of Interest**, Kluwer
3. Greco, S., Matarazzo, B., Pappalardo, N., Slowiński, R. (2005) Measuring expected effects of interventions based on decision rules, in **Journal of Experimental and Theoretical Artificial Intelligence**, Taylor Francis, Vol. 17, No. 1-2
4. Pawlak, Z., (1991) Information systems - theoretical foundations, in **Information Systems Journal**, Vol. 6, 205-218
5. Silberschatz, A., Tuzhilin, A., (1995) On subjective measures of interestingness in knowledge discovery, in **Proceedings of KDD′95 Conference**, AAAI Press
6. Raś, Z., Wieczorkowska, A. (2000) Action Rules: how to increase profit of a company, in **Principles of Data Mining and Knowledge Discovery**, LNAI, No. 1910, Springer, 587-592
7. Raś, Z.W., Tzacheva, A., Tsay, L.-S. (2005) Action rules, in **Encyclopedia of Data Warehousing and Mining**, (Ed. J. Wang), Idea Group Inc., 1-5
8. Raś, Z.W., Dardzińska, A. (2006) Action rules discovery, a new simplified strategy, in **Foundations of Intelligent Systems**, F. Esposito et al. (Eds.), LNAI, No. 4203, Springer, 445-453
9. Tsay, L.-S., Raś, Z.W. (2005) Action rules discovery system DEAR, method and experiments, in **Journal of Experimental and Theoretical Artificial Intelligence**, Taylor & Francis, Vol. 17, No. 1-2, 119-128
10. Tsay, L.-S., Raś, Z.W. (2006) Action rules discovery system DEAR3, in **Foundations of Intelligent Systems**, LNAI, No. 4203, Springer, 483-492
11. Tzacheva, A., Raś, Z.W. (2005) Action rules mining, in **International Journal of Intelligent Systems**, Wiley, Vol. 20, No. 7, 719-736

# Data Confidentiality Versus Chase

Zbigniew W. Raś[1,2], Osman Gürdal[3], Seunghyun Im[4], and Angelina Tzacheva[5]

[1] Univ. of North Carolina, Dept. of Comp. Science, Charlotte, N.C. 28223
[2] Polish-Japanese Institute of Information Technology, 02-008 Warsaw, Poland
[3] Johnson C. Smith Univ., Dept. of Comp. Sci. and Eng., Charlotte, NC 28216
[4] Univ. of Pittsburgh at Johnstown, Dept. of Comp. Science, Johnstown, PA 15904
[5] Univ. of South Carolina Upstate, Dept. of Informatics, Spartanburg, SC 29303

**Abstract.** We present a generalization of a strategy, called $SCIKD$, proposed in [7] that allows to reduce a disclosure risk of confidential data in an information system $S$ [10] using methods based on knowledge discovery. The method proposed in [7] protects confidential data against Rule-based Chase, the null value imputation algorithm driven by certain rules [2], [4]. This method identifies a minimal subset of additional data in $S$ which needs to be hidden to guarantee that the confidential data are not revealed by Chase. In this paper we propose a bottom-up strategy which identifies, for each object $x$ in $S$, a maximal set of values of attributes which do not have to be hidden and still the information associated with secure attribute values of $x$ is protected. It is achieved without examining all possible combinations of values of attributes. Our method is driven by classification rules extracted from $S$ and takes into consideration their confidence and support.

## 1 Introduction

This article discusses an important issue in data mining: how to provide meaningful knowledge without compromising data confidentiality. In conventional database systems, data confidentiality is achieved by hiding sensitive data from unauthorized users. However, hiding is not sufficient in knowledge discovery systems ($KDS$) due to null imputation method like rule-based Chase ([2], [4]) which are designed to predict null or missing values. Suppose that attributes in a database contain medical information about patients; some portions are not confidential while others are confidential (they are hidden from users). In this case, part or all of the confidential data in the attribute may be revealed. In other words, self-generated rules extracted from non-confidential portions of data can be used to find secret data.

Security in $KDS$ is studied in many research areas, such as cryptography, statistics, and data mining. A well known security problem in cryptography area is how to acquire global knowledge without revealing the data stored in each local site in a distributed autonomous information system ($DAIS$). Proposed solutions are based primarily on secure multiparty protocol ([12], [5]) which ensures that each participant cannot learn more than its own input data and outcome of

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 330–337, 2007.

a public function. Various authors expanded the idea. Clifton and Kantarcioglou employed the protocol for association rule mining for vertically and horizontally partitioned data [8]. Authors Du and Zhan pursued a similar idea to build a decision tree system [6]. Protection of sensitive rules has been discussed by Oliveira and Zaiane [9]. Authors suggested a solution to protecting sensitive association rules in the form of "sanitization process" that hides selective patterns from frequent itemsets. The data security problem discussed in this article is different from other researches in the following ways. First, we focus on the accuracy of existing data or knowledge instead of statistical characteristics of data. Second, we aim to protect sensitive data in a database instead of sensitive rules.

Our paper takes the definition of an information system proposed by Pawlak [10] as a simplified model of a database. However, the notion of its incompleteness differs from the classical rough set approach by allowing a set of weighted attribute values as a value of an attribute. We also assume that the sum of these weights has to be equal 1. If weights assigned to attribute values have to be greater than a user specified threshold value $\lambda$, then we get information system of type $\lambda$ as introduced in [4].

Additionally we assume that one or more attributes in an information system $S$ of type $\lambda$ contain confidential data that have to be protected and $S$ is a part of a distributed autonomous information system ($DAIS$) which provides a set of rules applicable at $S$ as a $KB$ [11]. We have to be certain that values of any confidential attribute can not be revealed from the available data in $S$ and $KB$ by *Chase* [2] or any other null value imputation method while minimizing the changes in the original information system. Also, we assume that we can hide the precise information about objects from the user but we can not replace existing data by false data. For instance, if someone is 18 years old, we can say that she is young or her age is unknown but we can not say that she is 24 years old. In pursue of such requirements, we propose a protection method named as *SCIKD* for information systems of type $\lambda$. The method identifies weighted transitive closure of attribute values involved in confidential data reconstruction, and uses the result to identify the maximum number of attribute values that can remain unchanged.

## 2 Chase as Tool for Revealing Hidden Values

We briefly provide some background on a null value imputation algorithm *Chase* based on rule-discovery strategy called *ERID* [2]. Assume that $S = (X, A, V)$, where $V = \bigcup\{V_a : a \in A\}$ and each $a \in A$ is a partial function from $X$ into $2^{V_a} - \{\emptyset\}$. In the first step, Chase algorithm identifies all incomplete attributes in $S$. An attribute is incomplete if there is an object in $S$ with incomplete information on this attribute. The values of all incomplete attributes in $S$ are treated as concepts to be learned (in a form of rules) either directly from $S$ or from $S$ and its remote sites (if $S$ is a part of $DAIS$). The second step of Chase algorithm is to extract all these rules and store them in a knowledge base $D$ for $S$ [11]. The next step is to replace incomplete information in $S$ by values

provided by rules in $D$. This process is recursively repeated till no new hidden values in $S$ can be revealed.

**Definition**
We say that $S = (X, A, V)$ is a partially incomplete information system of type $\lambda$, if the following four conditions hold:

- $X$ is the set of objects, $A$ is the set of attributes, and $V = \bigcup\{V_a : a \in A\}$ is the set of values of attributes,
- $(\forall x \in X)(\forall a \in A)[a_S(x) \in V_a$ or $a_S(x) = \{(v_i, p_i) : 1 \leq i \leq m\}]$ ,
- $(\forall x \in X)(\forall a \in A)[(a_S(x) = \{(v_i, p_i) : 1 \leq i \leq m\}) \rightarrow \sum_{i=1}^{m} p_i = 1]$,
- $(\forall x \in X)(\forall a \in A)[(a_S(x) = \{(v_i, p_i) : 1 \leq i \leq m\}) \rightarrow (\forall i)(p_i \geq \lambda)]$.

An example of an information system of type $\lambda = \frac{1}{5}$ is given in Table 1.

**Table 1.** Information System $S$

| $X$ | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ | $g$ |
|---|---|---|---|---|---|---|---|
| $x_1$ | $(a_1, \frac{2}{3})(a_2, \frac{1}{3})$ | $b_1$ | $c_1$ | $d_1$ | $e_1$ | $f_1$ | $g_1$ |
| $x_2$ | $(a_2, \frac{2}{5})(a_3, \frac{3}{5})$ | $(b_1, \frac{1}{3})(b_2, \frac{2}{3})$ | | $d_2$ | $e_1$ | $f_2$ | |
| $x_3$ | $a_1$ | $b_2$ | $(c_1, \frac{1}{2})(c_3, \frac{1}{2})$ | $d_1$ | $e_3$ | $f_2$ | |
| $x_4$ | $a_3$ | | $c_2$ | $d_1$ | $(e_1, \frac{2}{3})(e2, \frac{1}{3})$ | $f_2$ | |
| $x_5$ | $(a_1, \frac{2}{3})(a_3, \frac{1}{3})$ | $(b_1, \frac{1}{2})(b_2, \frac{1}{2})$ | $c_2$ | $d_1$ | $e_1$ | $f_2$ | $g_1$ |
| $x_6$ | $a_2$ | $b_2$ | $c_3$ | $d_1$ | $(e_2, \frac{1}{3})(e_3, \frac{2}{3})$ | $f_3$ | |
| $x_7$ | $a_2$ | $b_1$ | $(c_1, \frac{1}{3})(c_2, \frac{2}{3})$ | | $e_2$ | $f_3$ | |
| | | | . | | | | |
| | | | . | | | | |
| $x_i$ | $(a_3, \frac{1}{2})(a_4, \frac{1}{2})$ | $b_1$ | $c_2$ | | $e_3$ | $f_2$ | |

Let us assume that another information system $S_2$ has the same values as $S$ except $a(x_1)$=$\{(a_1, \frac{3}{4}), (a_2, \frac{1}{4})\}$ and $b(x_5)$=$\{(b_1, \frac{3}{4}), (b_2, \frac{1}{4})\}$. In both cases, an attribute value assigned to an object in $S_2$ is less general than in $S_1$.

Now, let us assume that $S$, $S_2$ are partially incomplete information systems, both of type $\lambda$. They provide descriptions of the same set of objects $X$ using the same set of attributes $A$. The meaning and granularity of values of attributes in $A$ for both systems $S$, $S_2$ is also the same. Additionally, we assume that $a_S(x) = \{(a_i, p_i) : i \leq m\}$ and $a_{S_2}(x) = \{(a2_i, p2_i) : i \leq m_2\}$.

Now, we introduce the relation $\Psi$, called containment relation. We say that $(S, S_2) \in \Psi$, if the following two conditions hold:

- $(\forall x \in X)(\forall a \in A)[card(a_{S(x)}) \geq card(a_{S_2(x)})]$,
- $(\forall x \in X)(\forall a \in A)[[card(a_S(x)) = card(a_{S_2}(x))] \rightarrow [\sum_{i \neq j} |p2_i - p2_j| > \sum_{i \neq j} |p_i - p_j|]]$.

Instead of saying that containment relation holds between $S$ and $S_2$, we can equivalently say that $S$ was transformed into $S_2$ by containment mapping $\Psi$. Algorithm $Chase_2$, described by Dardzińska and Raś in [2], converts an information system $S$ of type $\lambda$ to a new more complete information system $Chase_2(S)$ of the same type. The algorithm differs from other known strategies for chasing incomplete data in relational tables because of the assumption concerning partial incompleteness of data (sets of weighted attribute values can be assigned by $Chase_2$ to an object as its new value). This assumption forced authors in [3] to develop a new discovery algorithm, called *ERID*, for extracting rules from incomplete information systems of type $\lambda$. The syntax of classification rules discovered by *ERID* is the same as syntax of similar rules discovered by classical methods, like *LERS* or *RSES*. However, the method of computing their confidence and support is different.

**Table 2.** Information System $S_d$

| $X$ | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ | $g$ |
|---|---|---|---|---|---|---|---|
| $x_1$ | $(a_1,\frac{2}{3})(a_2,\frac{1}{3})$ | $b_1$ | $c_1$ | | $e_1$ | $f_1$ | $g_1$ |
| $x_2$ | $(a_2,\frac{2}{5})(a_3,\frac{3}{5})$ | $(b_1,\frac{1}{3})(b_2,\frac{2}{3})$ | | | $e_1$ | $f_2$ | |
| $x_3$ | $a_1$ | $b_2$ | $(c_1,\frac{1}{2})(c_3,\frac{1}{2})$ | | $e_3$ | $f_2$ | |
| $x_4$ | $a_3$ | | $c_2$ | | $(e_1,\frac{2}{3})(e2,\frac{1}{3})$ | $f_2$ | |
| $x_5$ | $(a_1,\frac{2}{3})(a_3,\frac{1}{3})$ | $(b_1,\frac{1}{2})(b_2,\frac{1}{2})$ | $c_2$ | | $e_1$ | $f_2$ | $g_1$ |
| $x_6$ | $a_2$ | $b_2$ | $c_3$ | | $(e_2,\frac{1}{3})(e_3,\frac{2}{3})$ | $f_3$ | |
| $x_7$ | $a_2$ | $b_1$ | $(c_1,\frac{1}{3})(c_2,\frac{2}{3})$ | | $e_2$ | $f_3$ | |
| | | | . | | | | |
| | | | . | | | | |
| $x_i$ | $(a_3,\frac{1}{2})(a_4,\frac{1}{2})$ | $b_1$ | $c_2$ | | $e_3$ | $f_2$ | |

Algorithm $Chase_2$ based on *ERID* can be used as a null value imputation tool to reveal hidden symbolic data. The method proposed in [7] protects confidential data against $Chase_2$ assuming that it is driven by certain rules. It identifies a minimal subset of additional data in $S$ which needs to be hidden to guarantee that the confidential data can not be revealed by Chase. In this paper we generalize this strategy by proposing an algorithm which protects confidential data against $Chase_2$ driven by *ERID*. It is a bottom-up strategy which identifies, for each object $x$ in $S$, a maximal set of values of attributes which do not have to be entirely hidden and still the information associated with secure attribute values of $x$ is protected.

## 3 Algorithm Protecting Confidential Data Against Rule-Based Chase

In this section we present an algorithm which protects values of a hidden attribute over null value imputation $Chase_2$ based on *ERID*. Suppose we have an information system $S$ as shown in Table 1 of type $\lambda = \frac{1}{5}$. $S$ is transformed to

$S_d$ by hiding the confidential attribute $d$ as shown in Table 2. The rules in the knowledge base $KB$ are summarized in Table 3. For instance $r_1 = [b_1 \cdot c_1 \rightarrow a_1]$ is an example of a rule belonging to $KB$ and its confidence is 1.

**Table 3.** Rules contained in *KB*. Values in parenthesis are decision values

| *Rule* | *Conf* | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ | $g$ |
|---|---|---|---|---|---|---|---|---|
| $r_1$ | 1 | $(a_1)$ | $b_1$ | $c_1$ | | | | |
| $r_2$ | 1 | $(a_1)$ | | $c_1$ | | | $f_1$ | |
| $r_3$ | $\frac{2}{3}$ | | $(b_1)$ | $c_1$ | | | | |
| $r_4$ | 1 | | $(b_1)$ | | | $e_1$ | | |
| $r_5$ | 1 | $a_1$ | | $(c_1)$ | | | $f_1$ | |
| $r_6$ | 1 | $a_1$ | | $c_1$ | | $(e_1)$ | | |
| $r_7$ | $\frac{2}{3}$ | | | $(c_1)$ | | $e_1$ | | $g_1$ |
| $r_8$ | 1 | $a_1$ | | $c_1$ | $(d_1)$ | | | |
| $r_9$ | 1 | | $b_1$ | $c_1$ | $(d_1)$ | | | |
| $r_{10}$ | 1 | | | | $(d_1)$ | | $f_1$ | |

To describe the algorithm, first we define the following sets,

- $\alpha(x) = \{a \in A : a(x) \neq Null\}$, the set of attribute values in $S_d$ used to describe $x$
- $\alpha(t)$, the set of attribute values used in $t$, where $t$ is their conjunction
- $R(x) = \{(t \rightarrow c) \in KB : \alpha(t) \subseteq \alpha(x)\}$, the set of rules in $KB$ where the attribute values used in $t$ are contained in $\alpha(x)$
- $\beta(x) = \cup\{\alpha(t) \cup \{c\} : [t \rightarrow c] \in R(x)\}$.

In our example $R(x_1) = \{r_1,r_2,r_3,r_4,r_5,r_6,r_7,r_8,r_9,r_{10}\}$, and $\beta(x_1) = \{a_1, b_1, c_1, d_1, e_1, f_1, g_1\}$. By using $Chase_2$ based on *ERID*, $d_1$ replaces the hidden slot $d(x_1)$ by rules from $\{r_8, r_9, r_{10}\}$. Rules $r_9$, $r_{10}$ guarantee the confidence 1 assigned to $d_1$, whereas the rule $r_8$ only guarantees the confidence $\frac{2}{3}$ which is above the threshold value $\lambda = \frac{1}{5}$. In addition, other rules from $R(x_1)$ also predict attribute values listed in $\{t_8, t_9, t_{10}\}$. These interconnections often build up a complex chain of inferences. The task of blocking such inference chains and identifying the minimal set of concealing values is not straightforward [7], especially that the confidence assigned to rules in $KB$ and the confidence assigned to attribute values in $S_d$ have to be taken into consideration.

To reduce the complexity and minimize the size of the set of hidden values, a bottom up approach has been adapted. We check the values that remain unchanged starting from a singleton set containing attribute value $a$ by using weighted transitive closure [4] (if $a \rightarrow b$ and $b \rightarrow c$, then $a \rightarrow c$, which gives us the set $\{a, b, c\}$). What about computing the weights assigned to $a$, $b$, $c$? Let us assume that $a \rightarrow b$ has a confidence $\lambda_1$ and $b \rightarrow c$ has a confidence $\lambda_2$. Then, weight 1 is assigned to $a$, weight $\lambda_1$ is assigned to $b$, and weight $(\lambda_1 \cdot \lambda_2)$ is assigned to $c$. If $\lambda_3$ is a weight associated with $a$, then weight $(\lambda_3 \cdot \lambda_1)$ is assigned to $b$, and weight $(\lambda_3 \cdot \lambda_1 \cdot \lambda_2)$ is assigned to $c$. If the weight assigned to any of the

elements in $\{a, b, c\}$ is below the threshold value $\lambda$, then this element is removed from $\{a, b, c\}$. Our goal is to increase the initial set size as much as possible. Let us notice that any element of the resulting set can be generated by following two different paths. Each path assigns a different weight to that element. In all such cases, the highest weight is chosen by our algorithm. This approach automatically rules out any superset of must-be-hidden values, and minimizes the computational cost. The justification of this is quite simple. Weighted transitive closure has the property that the superset of a set $s$ also contains $s$. Clearly, if a set of attribute values predicts $d_1$, then the set must be hidden regardless of the presence/abscence of other attribute values.

To outline the procedure, we start with a set $\beta(x) = \{(a_1, \frac{2}{3}), b_1, c_1, e_1, f_1, g_1\}$ for the object $x_1$ which construction is supported by 10 rules from $KB$, and check the transitive closure of each singleton subset $\delta(x)$ of that set. If the transitive closure of $\delta(x)$ contains classified attribute value $d_1$ and the weight associated with $d_1$ is greater than $\lambda$, then $\delta(x)$ does not sustain, it is marked, and it is not considered in later steps. Otherwise, the set remains unmarked. In the second iteration of the algorithm, all two-element subsets of $\beta(x)$ built only from unmarked sets are considered. If the transitive closure of any of these sets does not contain $d_1$ with weight associated to it greater than $\lambda$, then such a set remains unmarked and it is used in the later steps of the algorithm. Otherwise, the set is getting marked. If either all sets in a currently executed iteration step are marked or we have reached the set $\beta(x)$, then the algorithm stops. Since only subsets of $\beta(x)$ are considered, the number of iterations will be usually not large.

So, in our example the following singleton sets are considered:
$\{(a_1, \frac{2}{3})\}^+ = \{(a_1, \frac{2}{3})\}$ is unmarked
$\{b_1\}^+ = \{b_1, \}$ is unmarked
$\{c_1\}^+ = \{(a_1, \frac{2}{3}), (b_1, \frac{2}{3}), c_1, (e_1, \frac{4}{9}), (d_1, \frac{4}{9})\}$ contains $d_1$ and $\frac{4}{9} \geq \lambda$ so it is marked
$\{e_1\}^+ = \{b_1, e_1\}$ is unmarked
$\{f_1\}^+ = \{d_1, f_1\}$ contains $d_1$ so it is marked
$\{g_1\}^+ = \{g_1\}$ is unmarked

Clearly, $c_1$ and $f_1$ have to be concealed. The next step is to build sets of length 2 and determine which of them can sustain. We take the union of two sets only if they are both unmarked and one of them is a singleton set.

$\{(a_1, \frac{2}{3}), b_1\}^+ = \{(a_1, \frac{2}{3}), b_1\}$ is unmarked
$\{(a_1, \frac{2}{3}), e_1\}^+ = \{(a_1, \frac{2}{3}), b_1, e_1\}$ is unmarked
$\{(a_1, \frac{2}{3}), g_1\}^+ = \{(a_1, \frac{2}{3}), g_1\}$ is unmarked
$\{b_1, e_1\}^+ = \{b_1, e_1\}$ is unmarked
$\{b_1, g_1\}^+ = \{b_1, g_1\}$ is unmarked
$\{e_1, g_1\}^+ = \{(a_1, \frac{2}{3}), (b_1, \frac{2}{3}), (c_1, \frac{2}{3}), (d_1, \frac{2}{3}), e_1, g_1\}$ contains $d_1$ and $\frac{2}{3} \geq \lambda$ so it is marked

Now we build 3-element sets from previous sets that have not been marked.

$\{(a_1, \frac{2}{3}), b_1, e_1\}^+ = \{(a_1, \frac{2}{3}), b_1, e_1\}$ is unmarked
$\{(a_1, \frac{2}{3}), b_1, g_1\}^+ = \{(a_1, \frac{2}{3}), b_1, g_1\}$ is unmarked
$\{b_1, e_1, g_1\}^+$ is not considered as a superset of $\{e_1, g_1\}$ which was marked.

We have $\{a_1, b_1, e_1\}$ and $\{a_1, b_1, g_1\}$ as unmarked sets that contain the maximum number of elements and do not have the transitive closure containing $d$ with associated weight greater than $\lambda$. In a similar way, we compute the maximal sets for any object $x_i$.

The corresponding algorithm, called *G-SCIKD*, is a generalization of *SCIKD* strategy presented in [7]. If an attribute values revealed by *G-SCIKD* has a confidence below $\lambda$, then this attribute value is removed from consideration. This constraint is semantically similar to the constraint $\lambda$ used in *ERID* [2].

## 4 Experiment

We implemented *G-SCIKD* on a PC running Windows XP and Oracle database version 10g. The code was written in PL/SQL language with PL/SQL Developer version 6.

The sampling data table containing 4,000 objects with 10 attributes was extracted randomly from a complete database describing personal income reported in the Census data [1]. The data table was randomly partitioned into 4 tables that each have 1,000 tuples. One of these tables is called *client* and the remaining 3 are called *servers*. Now, we hide all values of one attribute that includes income data at the client site. From the servers, 13 rules are extracted by *ERID* and stored in $KB$ of the client. Additionally, 75 rules describing incomplete or partially hidden attributes at the client site are extracted by *ERID*. All these rules are used to reveal values of incomplete attributes by *Chase* algorithm [2]. It appears that 739 attribute values (7.39% of the total number of attribute values in client table) have to be additionally hidden. The presented method can easily be used to protect two or more confidential attributes in an information system. In this case, a set of attribute values in $x_i$ should be hidden if the closure of this set contains any of the classified data.

**Acknowledgment.** This research was supported by NIH Grant No.5G11HD 38486-06.

## References

1. UCI Machine Learning Rep., http://www.ics.uci.edu/ mlearn/MLRepository.html
2. Dardzińska, A., Raś, Z.W. (2003) Rule-Based Chase Algorithm for Partially Incomplete Information Systems, in **Proceedings of the Second International Workshop on Active Mining (AM'2003)**, Maebashi City, Japan, October, 42-51

3. Dardzińska, A., Raś, Z.W. (2003) On Rules Discovery from Incomplete Information Systems, in **Proceedings of ICDM'03 Workshop on Foundations and New Directions of Data Mining**, (Eds: T.Y. Lin, X. Hu, S. Ohsuga, C. Liau), Melbourne, Florida, IEEE Computer Society, 31-35
4. Dardzińska, A., Raś, Z.W. (2003) Chasing Unknown Values in Incomplete Information Systems, in **Proceedings of ICDM'03 Workshop on Foundations and New Directions of Data Mining**, (Eds: T.Y. Lin, X. Hu, S. Ohsuga, C. Liau), Melbourne, Florida, IEEE Computer Society, 24-30
5. Du, W. and Atallah, M. J. (2001) Secure Multi-party Computation Problems and their Applications: a review and open problems, in **New Security Paradigms Workshop**
6. Du, W. and Zhan, Z. (2002), Building decision tree classifier on private data, in **Proceedings of the IEEE ICDM Workshop on Privacy, Security and Data Mining**
7. Im, S., Raś, Z.W., Dardzińska, A. (2005) SCIKD: Safeguarding Classified Information from Knowledge Discovery, in **Foundations of Semantic Oriented Data and Web Mining**, Proceedings of 2005 IEEE ICDM Workshop in Houston, Texas, Published by Math. Dept., Saint Mary's Univ., Nova Scotia, Canada, 34-39
8. Kantarcioglou, M. and Clifton, C. (2002), Privacy-preserving distributed mining of association rules on horizontally partitioned data, in **Proceedings of the ACM SIGMOD Workshop on Research Isuues in Data Mining and Knowledge Discovery**, 24-31
9. Oliveira, S. R. M. and Zaiane, O. R. (2002), Privacy preserving frequent itemset mining, in **Proceedings of the IEEE ICDM Workshop on Privacy, Security and Data Mining**, 43-54
10. Pawlak, Z. (1991) Information Systems - theoretical foundations, in **Information Systems Journal**, Vol. 6, 205-218
11. Raś, Z.W., Dardzińska, A. (2006) Solving Failing Queries through Cooperation and Collaboration, Special Issue on Web Resources Access, (Editor: M.-S. Hacid), in **World Wide Web Journal**, Springer, Vol. 9, No. 2, 173-186
12. Yao, A. C. (1996) How to generate and exchange secrets, in **Proceedings of the 27th IEEE Symposium on Foundations of Computer Science**, 162-167

# Relationship Between Loss Functions and Confirmation Measures

Krzysztof Dembczyński[1], Salvatore Greco[2], Wojciech Kotłowski[1], and Roman Słowiński[1,3]

[1] Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland
{kdembczynski,wkotlowski,rslowinski}@cs.put.poznan.pl
[2] Faculty of Economics, University of Catania, 95129 Catania, Italy
salgreco@unict.it
[3] Institute for Systems Research, Polish Academy of Sciences, 01-447 Warsaw, Poland

**Abstract.** In the paper, we present the relationship between loss functions and confirmation measures. We show that population minimizers for weighted loss functions correspond to confirmation measures. This result can be used in construction of machine learning methods, particularly, ensemble methods.

## 1 Introduction

Let us define the prediction problem in a similar way as in [4]. The aim is to predict the unknown value of an attribute $y$ (sometimes called *output*, *response variable* or *decision attribute*) of an object using the known joint values of other attributes (sometimes called *predictors*, *condition attributes* or *independent variables*) $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. We consider binary classification problem, in which we assume that $y \in \{-1, 1\}$. All objects for which $y = -1$ constitute decision class $Cl_{-1}$, and all objects for which $y = 1$ constitute decision class $Cl_1$. The goal of a learning task is to find a function $F(\mathbf{x})$ (in general, $F(\mathbf{x}) \in \Re$) using a set of training examples $\{y_i, \mathbf{x}_i\}_1^N$ that predicts accurately $y$ (in other words, classifies accurately objects to decision classes). The optimal classification procedure is given by:

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{y\mathbf{x}} L(y, F(\mathbf{x})), \tag{1}$$

where the expected value $E_{y\mathbf{x}}$ is over joint distribution of all variables $(y, \mathbf{x})$ for the data to be predicted. $L(y, F(\mathbf{x}))$ is a loss or cost for predicting $F(\mathbf{x})$ when the actual value is $y$. $E_{y\mathbf{x}} L(y, F(\mathbf{x}))$ is often called *prediction risk*. Nevertheless, the learning procedure can use only a set of training examples $\{y_i, \mathbf{x}_i\}_1^N$. Using this set, it tries to construct $F(\mathbf{x})$ to be the best possible approximation of $F^*(\mathbf{x})$. The typical loss function in binary classification tasks is, so called, 0-1 loss:

$$L_{0-1}(y, F(\mathbf{x})) = \begin{cases} 0 & \text{if } yF(\mathbf{x}) > 0, \\ 1 & \text{if } yF(\mathbf{x}) \leq 0. \end{cases} \tag{2}$$

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 338–345, 2007.

It is possible to use other loss functions than (2). Each of these functions has some interesting properties. One of them is a population minimizer of prediction risk. By conditioning (1) on $\mathbf{x}$ (i.e., factoring the joint distribution $P(y, \mathbf{x}) = P(\mathbf{x})P(y|\mathbf{x})$), we obtain:

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{\mathbf{x}} E_{y|\mathbf{x}} L(y, F(\mathbf{x})). \tag{3}$$

It is easy to see that it suffices to minimize (3) pointwise:

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{y|\mathbf{x}} L(y, F(\mathbf{x})). \tag{4}$$

The solution of the above is called *population minimizer*. In other words, this is an answer to a question: what does a minimization of expected loss estimate on a population level? Let us remind that the population minimizer for 0-1 loss function is:

$$F^*(\mathbf{x}) = \begin{cases} 1 & \text{if } P(y = 1|\mathbf{x}) \geq \frac{1}{2}, \\ -1 & \text{if } P(y = -1|\mathbf{x}) > \frac{1}{2}. \end{cases} \tag{5}$$

From the above, it is easy to see that minimizing 0-1 loss function one estimates a region in predictor space in which class $Cl_1$ is observed with the higher probability than class $Cl_{-1}$. Minimization of some other loss functions can be seen as an estimation of conditional probabilities $P(y = 1|\mathbf{x})$ (see Section 2).

From the other side, Bayesian confirmation measures (see, for example, [5,9]) have paid a special attention in knowledge discovery [7]. Confirmation measure $c(H, E)$ says in what degree a piece of evidence $E$ confirms (or disconfirms) a hypothesis $H$. It is required to satisfy:

$$c(H, E) = \begin{cases} > 0 & \text{if } P(H|E) > P(H), \\ = 0 & \text{if } P(H|E) = P(H), \\ < 0 & \text{if } P(H|E) < P(H), \end{cases} \tag{6}$$

where $P(H)$ is the probability of hypothesis $H$ and $P(H|E)$ is the conditional probability of hypothesis $H$ given evidence $E$. In Section 3, two confirmation measures of a particular interest are discussed.

In this paper, we present relationship between loss functions and confirmation measures. The motivation of this study is a question: what is the form of the loss function for estimating a region in predictor space in which class $Cl_1$ is observed with the positive confirmation, or alternatively, for estimating confirmation measure for a given $\mathbf{x}$ and $y$? In the following, we show that population minimizers for *weighted* loss functions correspond to confirmation measures. Weighted loss functions are often used in the case of imbalanced class distribution, i.e., when probabilities $P(y = 1)$ and $P(y = -1)$ are substantially different. This result is described in Section 4. The paper is concluded in the last section.

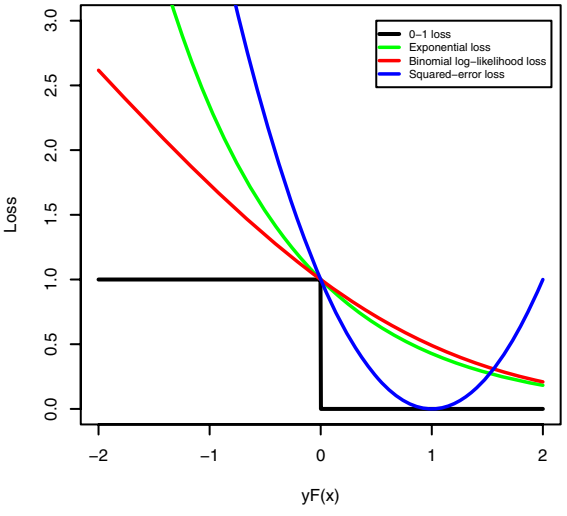



**Fig. 1.** The most popular loss functions (figure prepared in R [10]; similar figure may be found in [8], also prepared in R)

## 2 Loss Functions

There are different loss functions used in prediction problems (for a wide discussion see [8]). In this paper, we consider, besides 0-1 loss, the following three loss functions for binary classification:

– exponential loss:

$$L_{exp}(y, F(\mathbf{x})) = \exp(-yF(\mathbf{x})), \tag{7}$$

– binomial negative log-likelihood loss:

$$L_{log}(y, F(\mathbf{x})) = \log(1 + exp(-2yF(\mathbf{x}))), \tag{8}$$

– squared-error loss:

$$L_{sqr}(y, F(\mathbf{x})) = (y - F(\mathbf{x}))^2 = (1 - yF(\mathbf{x}))^2. \tag{9}$$

These loss functions are presented in Figure 1. Exponential loss is used in AdaBoost [6]. Binomial negative log-likelihood loss is common in statistical approaches. It is also used in Gradient Boosting Machines [3]. The reformulation of the squared-error loss (9) is possible, because $y \in \{-1, 1\}$. Squared-error loss is not a monotone decreasing function of increasing $yF(\mathbf{x})$. For values $yF(\mathbf{x}) > 1$ it increases quadratically. For this reason, one has to use this loss function in classification task very carefully.

The population minimizers for these loss functions are as follows:

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{y|\mathbf{x}} L_{exp}(y, F(\mathbf{x})) = \frac{1}{2} \log \frac{P(y = 1|\mathbf{x})}{P(y = -1|\mathbf{x})},$$

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{y|\mathbf{x}} L_{log}(y, F(\mathbf{x})) = \frac{1}{2}\log\frac{P(y=1|\mathbf{x})}{P(y=-1|\mathbf{x})},$$
$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{y|\mathbf{x}} L_{sqr}(y, F(\mathbf{x})) = P(y=1|\mathbf{x}) - P(y=-1|\mathbf{x}).$$

From these formulas, it is easy to get values of $P(y=1|\mathbf{x})$.

## 3 Confirmation Measures

There are two confirmation measures of a particular interest:

$$l(H,E) = \log\frac{P(E|H)}{P(E|\neg H)}, \tag{10}$$
$$f(H,E) = \frac{P(E|H) - P(E|\neg H)}{P(E|H) + P(E|\neg H)}, \tag{11}$$

where $H$ is hypothesis, and $E$ is evidence. Measures $l$ and $f$ satisfy two desired properties that are:

- hypothesis symmetry: $c(H,E) = -c(\neg H, E)$ (for details, see for example [5]),
- and monotonicity property $M$ defined in terms of rough set confirmation measures (for details, see [7]).

Let us remark that in the binary classification problem, one tries for a given $\mathbf{x}$ to predict value $y \in \{-1, 1\}$. In this case, evidence is $\mathbf{x}$, and hypotheses are then $y = -1$ and $y = 1$. Confirmation measures (10) and (11) take the following form:

$$l(y=1|\mathbf{x}) = \log\frac{P(\mathbf{x}|y=1)}{P(\mathbf{x}|y=-1)}, \tag{12}$$
$$f(y=1|\mathbf{x}) = \frac{P(\mathbf{x}|y=1) - P(\mathbf{x}|y=-1)}{P(\mathbf{x}|y=1) + P(\mathbf{x}|y=-1)}. \tag{13}$$

## 4 Population Minimizers for Weighted Loss Functions

In this section, we present our main results that show relationship between loss functions and confirmation measures. We prove that population minimizers for weighted loss functions correspond to confirmation measures. Weighted loss functions are often used in the case of imbalanced class distribution, i.e., when probabilities $P(y=1)$ and $P(y=-1)$ are substantially different. Weighted loss function can be defined as follows:

$$L^w(y, F(\mathbf{x})) = w \cdot L(y, F(\mathbf{x})),$$

where $L(y, F(\mathbf{x}))$ is one of the loss functions presented above. Assuming that $P(y)$ is known, one can take $w = 1/P(y)$, and then:

$$L^w(y, F(\mathbf{x})) = \frac{1}{P(y)} \cdot L(y, F(\mathbf{x})). \tag{14}$$

In the proofs presented below, we use the following well-known facts: Bayes theorem: $P(y=1|\mathbf{x}) = P(y=1 \cap \mathbf{x})/P(\mathbf{x}) = P(\mathbf{x}|y=1)P(y=1)/P(\mathbf{x})$; and $P(y=1) = 1 - P(y=-1)$ and $P(y=1|\mathbf{x}) = 1 - P(y=-1|\mathbf{x})$.

Let us consider the following weighted 0-1 loss function:

$$L^w_{0-1}(y, F(\mathbf{x})) = \frac{1}{P(y)} \cdot \begin{cases} 0 & \text{if } yF(\mathbf{x}) > 0, \\ 1 & \text{if } yF(\mathbf{x}) \leq 0. \end{cases} \tag{15}$$

**Theorem 1.** *Population minimizer of* $E_{y|\mathbf{x}} L^w_{0-1}(y, F(\mathbf{x}))$ *is:*

$$\begin{aligned} F^*(\mathbf{x}) &= \begin{cases} 1 & \text{if } P(y=1|\mathbf{x}) \geq P(y=1), \\ -1 & \text{if } P(y=-1|\mathbf{x}) > P(y=-1) \end{cases} \\ &= \begin{cases} 1 & \text{if } c(y=1,\mathbf{x}) \geq 0, \\ -1 & \text{if } c(y=-1,\mathbf{x}) > 0. \end{cases} \end{aligned} \tag{16}$$

*where c is any confirmation measure.*

*Proof.* We have that

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{y|\mathbf{x}} L^w_{0-1}(y, F(\mathbf{x})).$$

Prediction risk is then:

$$E_{y|\mathbf{x}} L^w_{0-1}(y, F(\mathbf{x})) = P(y=1|\mathbf{x}) L^w_{0-1}(1, F(\mathbf{x})) + P(y=-1|\mathbf{x}) L^w_{0-1}(-1, F(\mathbf{x})),$$

$$E_{y|\mathbf{x}} L^w_{0-1}(y, F(\mathbf{x})) = \frac{P(y=1|\mathbf{x})}{P(y=1)} L_{0-1}(1, F(\mathbf{x})) + \frac{P(y=-1|\mathbf{x})}{P(y=-1)} L_{0-1}(-1, F(\mathbf{x})).$$

This is minimized, if either $P(y=1|\mathbf{x})/P(y=1) \geq P(y=-1|\mathbf{x})/P(y=-1)$ for any $F(\mathbf{x}) > 0$, or $P(y=1|\mathbf{x})/P(y=1) < P(y=-1|\mathbf{x})/P(y=-1)$ for any $F(\mathbf{x}) < 0$ (in other words, only the sign of $F(\mathbf{x})$ is important). From $P(y=1|\mathbf{x})/P(y=1) \geq P(y=-1|\mathbf{x})/P(y=-1)$, we have that:

$$\frac{P(y=1|\mathbf{x})}{P(y=1)} \geq \frac{1 - P(y=1|\mathbf{x})}{1 - P(y=1)},$$

which finally gives $P(y=1|\mathbf{x}) \geq P(y=1)$ or $c(y=1,\mathbf{x}) \geq 0$. Analogously, from $P(y=1|\mathbf{x})/P(y=1) < P(y=-1|\mathbf{x})/P(y=-1)$, we obtain that $P(y=-1|\mathbf{x}) > P(y=-1)$ or $c(y=-1,\mathbf{x}) > 0$. From the above we get the thesis. □

From the above theorem, it is easy to see that minimization of $L^w_{0-1}(y, F(\mathbf{x}))$ results in estimation of a region in predictor space in which class $Cl_1$ is observed with a positive confirmation. In the following theorems, we show that minimization of a weighted version of an exponential, a binomial negative log-likelihood, and a squared-loss error loss function gives an estimate of a particular confirmation measure, $l$ or $f$.

Let us consider the following weighted exponential loss function:

$$L^w_{exp}(y, F(\mathbf{x})) = \frac{1}{P(y)} \exp(-y \cdot F(\mathbf{x})). \tag{17}$$

**Theorem 2.** *Population minimizer of* $E_{y|\mathbf{x}}L^w_{exp}(y, F(\mathbf{x}))$ *is:*

$$F^*(\mathbf{x}) = \frac{1}{2}\log\frac{P(\mathbf{x}|y=1)}{P(\mathbf{x}|y=-1)} = \frac{1}{2}l(y=1,\mathbf{x}). \tag{18}$$

*Proof.* We have that

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{y|\mathbf{x}}L^w_{exp}(y, F(\mathbf{x})).$$

Prediction risk is then:

$$E_{y|\mathbf{x}}L^w_{exp}(y, F(\mathbf{x})) = P(y=1|\mathbf{x})L^w_{exp}(1, F(\mathbf{x})) + P(y=-1|\mathbf{x})L^w_{exp}(-1, F(\mathbf{x})),$$
$$E_{y|\mathbf{x}}L^w_{exp}(y, F(\mathbf{x})) = \frac{P(y=1|\mathbf{x})}{P(y=1)}\exp(-F(\mathbf{x})) + \frac{P(y=-1|\mathbf{x})}{P(y=-1)}\exp(F(\mathbf{x})).$$

Let us compute a derivative of the above expression:

$$\frac{\partial E_{y|\mathbf{x}}L^w_{exp}(y, F(\mathbf{x}))}{\partial F(\mathbf{x})} = -\frac{P(y=1|\mathbf{x})}{P(y=1)}\exp(-F(\mathbf{x})) + \frac{P(y=-1|\mathbf{x})}{P(y=-1)}\exp(F(\mathbf{x})).$$

Setting the derivative to zero, we get:

$$\exp(2F(\mathbf{x})) = \frac{P(y=1|\mathbf{x})P(y=-1)}{P(y=1|\mathbf{x})P(y=1)},$$
$$F(\mathbf{x}) = \frac{1}{2}\log\frac{P(y=1|\mathbf{x})P(y=-1)}{P(y=1|\mathbf{x})P(y=1)} = \frac{1}{2}l(y=1,\mathbf{x}). \qquad \square$$

Let us consider the following weighted binomial negative log-likelihood loss function:

$$L^w_{log}(y, F(\mathbf{x})) = \frac{1}{P(y)}\log(1 + exp(-2y \cdot F(\mathbf{x}))). \tag{19}$$

**Theorem 3.** *Population minimizer of* $E_{y|\mathbf{x}}L^w_{log}(y, F(\mathbf{x}))$ *is:*

$$F^*(\mathbf{x}) = \frac{1}{2}\log\frac{P(\mathbf{x}|y=1)}{P(\mathbf{x}|y=-1)} = \frac{1}{2}l(y=1,\mathbf{x}). \tag{20}$$

*Proof.* We have that

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{y|\mathbf{x}}L^w_{log}(y, F(\mathbf{x})).$$

Prediction risk is then:

$$E_{y|\mathbf{x}}L^w_{log}(y, F(\mathbf{x})) = P(y=1|\mathbf{x})L^w_{log}(1, F(\mathbf{x})) + P(y=-1|\mathbf{x})L^w_{log}(-1, F(\mathbf{x}))$$
$$= \frac{P(y=1|\mathbf{x})}{P(y=1)}\log(1+\exp(-2F(\mathbf{x}))) + \frac{P(y=-1|\mathbf{x})}{P(y=-1)}\log(1+\exp(2F(\mathbf{x}))).$$

Let us compute a derivative of the above expression:

$$\frac{\partial E_{y|\mathbf{x}} L^w_{log}(y, F(\mathbf{x}))}{\partial F(\mathbf{x})} = -2\frac{P(y=1|\mathbf{x})}{P(y=1)}\frac{\exp(-2F(\mathbf{x}))}{1+\exp(-2F(\mathbf{x}))} + \\ + 2\frac{P(y=-1|\mathbf{x})}{P(y=-1)}\frac{\exp(2F(\mathbf{x}))}{1+\exp(2F(\mathbf{x}))}.$$

Setting the derivative to zero, we get:

$$\exp(2F(\mathbf{x})) = \frac{P(y=1|\mathbf{x})P(y=-1)}{P(y=-1|\mathbf{x})P(y=1)}$$
$$F(\mathbf{x}) = \frac{1}{2}\log\frac{P(y=1|\mathbf{x})P(y=-1)}{P(y=-1|\mathbf{x})P(y=1)} = \frac{1}{2}l(y=1,\mathbf{x}). \qquad \square$$

Let us consider the following weighted squared-error loss function:

$$L^w_{sqr}(y, F(\mathbf{x})) = \frac{1}{P(y)}(y - F(\mathbf{x}))^2. \tag{21}$$

**Theorem 4.** *Population minimizer of* $E_{y|\mathbf{x}} L^w_{sqr}(y, F(\mathbf{x}))$ *is:*

$$F^*(\mathbf{x}) = \frac{P(\mathbf{x}|y=1) - P(\mathbf{x}|y=-1)}{P(\mathbf{x}|y=1) + P(\mathbf{x}|y=-1)} = f(y=1,\mathbf{x}). \tag{22}$$

*Proof.* We have that

$$F^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} E_{y|\mathbf{x}} L^w_{sqr}(y, F(\mathbf{x})).$$

Prediction risk is then:

$$E_{y|\mathbf{x}} L^w_{sqr}(y, F(\mathbf{x})) = P(y=1|\mathbf{x})L^w_{sqr}(1, F(\mathbf{x})) + P(y=-1|\mathbf{x})L^w_{sqr}(-1, F(\mathbf{x})),$$
$$E_{y|\mathbf{x}} L^w_{sqr}(y, F(\mathbf{x})) = \frac{P(y=1|\mathbf{x})}{P(y=1)}(1 - F(\mathbf{x}))^2 + \frac{P(y=-1|\mathbf{x})}{P(y=-1)}(1 + F(\mathbf{x}))^2.$$

Let us compute a derivative of the above expression:

$$\frac{\partial E_{y|\mathbf{x}} L^w_{log}(y, F(\mathbf{x}))}{\partial F(\mathbf{x})} = -2\frac{P(y=1|\mathbf{x})}{P(y=1)}(1 - F(\mathbf{x})) + 2\frac{P(y=-1|\mathbf{x})}{P(y=-1)}(1 + F(\mathbf{x})).$$

Setting the derivative to zero, we get:

$$F(\mathbf{x}) = \frac{P(y=1|\mathbf{x})/P(y=1) - P(y=-1|\mathbf{x})/P(y=-1)}{P(y=1|\mathbf{x})/P(y=1) + P(y=-1|\mathbf{x})/P(y=-1)},$$
$$F(\mathbf{x}) = \frac{P(\mathbf{x}|y=1) - P(\mathbf{x}|y=-1)}{P(\mathbf{x}|y=1) + P(\mathbf{x}|y=-1)} = f(y=1,\mathbf{x}). \qquad \square$$

## 5 Conclusions

We have proven that population minimizers for weighted loss functions correspond directly to confirmation measures. This result can be applied in construction of machine learning methods, for example, ensemble classifiers producing a linear combination of base classifiers. In particular, considering ensemble of decision rules [1,2], a sum of outputs of rules that cover $\mathbf{x}$ can be interpreted as an estimate of a confirmation measure for $\mathbf{x}$ and a predicted class.

Our future research will concern investigation of general conditions that loss function has to satisfy to be used in estimation of confirmation measures.

## References


1. Błaszczyński, J., Dembczyński, K., Kotłowski, W., Słowiński, R., Szeląg, M.: Ensemble of Decision Rules. Research Report RA-011/06, Poznań University of Technology (2006)
2. Błaszczyński, J., Dembczyński, K., Kotłowski, W., Słowiński, R., Szeląg, M.: Ensembles of Decision Rules for Solving Binary Classification Problems with Presence of Missing Values. In: Greco et al. (eds.): Rough Sets and Current Trends in Computing, Lecture Notes in Artificial Intelligence, Springer-Verlag **4259** (2006) 318–327
3. Friedman, J. H.: Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 5 **29** (2001) 1189–1232
4. Friedman, J. H.: Recent Advances in Predictive (Machine) Learning. Dept. of Statistics, Stanford University, `http://www-stat.stanford.edu/~jhf` (2003)
5. Fitelson, B.: Studies in Bayesian Confirmation Theory. Ph.D. Thesis, University of Wisconsin, Madison (2001)
6. Freund, Y., Schapire, R. E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. System Sci. **55** 1997 119–139
7. Greco, S., Pawlak, Z., Słowiński, R.: Can Bayesian confirmation measures be useful for rough set decision rules? Engineering Applications of Artificial Intelligence **17** (2004) 345–361
8. Hastie, T., Tibshirani, R., Friedman, J. H.: Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer (2003)
9. Kyburg, H.: Recent work in inductive logic. In: Lucey, K.G., Machan, T.R. (eds.): Recent Work in Philosophy. Rowman and Allanheld, Totowa, NJ, (1983) 89–150
10. R Development Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, `http://www.R-project.org`, Vienna, (2005)

# High Frequent Value Reduct in Very Large Databases

Tsau Young Lin[1] and Jianchao Han[2]

[1] Department of Computer Science, San Jose State University,
San Jose, CA 95192, USA
tylin@cs.sjsu.edu
[2] Department of Computer Science, California State University Dominguez Hills,
Carson, CA 90747, USA
jhan@csudh.edu

**Abstract.** One of the main contributions of rough set theory to data mining is data reduction. There are three reductions: attribute (column) reduction, row reduction, and value reduction. Row reduction is merging the duplicate rows. Attribute reduction is to find important attributes. Value reduction is to reduce the decision rules to a logically equivalent minimal length. Most recent attentions have been on finding attribute reducts. Traditionally, the value reduct has been searched through the attribute reduct. This paper observes that this method may miss the best value reducts. It also revisits an old rudiment idea [11], namely, a rough set theory on high frequency data: The notion of high frequency value reduct is extracted in a bottom-up fashion without finding attribute reducts. Our method can discover concise and important decision rules in large databases, and is described and illustrated by an example.

**Keywords:** Rough set theory, high frequency, decision rule, relational database.

## 1 Introduction

Rough set theory (RST) is an elegant and powerful methodology in extracting and minimizing rules from decision tables and has been extensively studied in the field and applied in real-life applications since the pioneer work by Pawlak in 1982 [12]. The essence of RST is to reduce a given decision table small enough so that decision rules can be directly extracted [13]. However, due to the complexity, the idea works only for small and clean data sets. RST has a fundamental assumption, namely, every piece of data is precise and important. For example, every tuple in a decision table is a rule [13]. This paper revisits an old rudiment idea in 1996-98 [3], [4], [10], namely, adopting RST to high frequency data in very large databases. The underlying assumption is high frequent data is important and clean [10], where we had regarded that a high support tuple is an important rule from the point of view of case based reasoning.

The reduction in rough set theory can be summarized in three aspects: attribute (column) reduction, row (tuple) reduction, and value reduction. Row reduction is only merging duplicate rows, attribute reduction is to find important attributes, while value reduct simplifies decision rules. Attribute reduction has been paid much attention in

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 346–354, 2007.

finding attribute reducts. The central notions in this research are core, reduct and knowledge dependency [12], [13]. An attribute *reduct* of a decision table is a subset of condition attributes that suffice to define the decision attributes. More than one reduct for each decision table may exist. The intersection of all the possible reducts is called the *core*, which represents the most important information of the decision table. Finding all attribute reducts in a decision table is NP-hard [14] unfortunately, so the full power of rough set methodology may only be effective on clean and small sets of data. Though approximation algorithms have been proposed to build reducts from a decision table either top-down or bottom-up [10], with gigabytes of data in modern database applications, direct applications of rough set methodology are prohibitively expensive. Additionally, using any attribute reduct may still miss some important decision rules. On the other hand, very little effort has been made to find value reducts directly without finding minimum attribute reducts.

We present an approach to extracting a series of interconnected information table that represent certain patterns of data. Our method uses the itemset concept exploited in mining association rules [1], [2], [6] and applies rough set methodology to such a series of information tables. In essence, we integrate relational database techniques and rough set methodology into an effective procedure of mining decision rules in very large databases. Although some methods that integrate RDBMS capabilities into rough set theory have been studied [3], [4], [5], [7], [8], [9], our contribution in this paper can be summarized as follows: 1) Unlike traditional rough set theory where decision rules are extracted from attribute reducts, we propose an approach to inducing decision rules by finding high frequent value reduct directly without finding any attribute reducts. A bottom-up algorithm is proposed to generate itemsets which are actually sub-decision tables of the original one. 2) The algorithm proposed in this paper can be easily implemented in the relational database environment by taking advantage of efficient SQL statements, and thus can be used to mine decision rules from very large databases.

The rest of this paper is organized as follows. In Section 2, the rough set theory is reviewed with an example to distinguish various reductions. A bottom-up approach to finding value reduct without attribute reducts is presented and its implementation in RDBMS environments is discussed in Section 3, The approach proposed is illustrated in Section 4 with an example, and the conclusion is in Section 5.

## 2 Rough Set Methodology

In this section, we will review rough set methodology as explained by Pawlak [12], [13]. We especially demonstrate by an example the attribute reduction, row reduction, and value reduction to induce a small set of decision rules. To simplify the problem, we assume that the decision table is consistent. Let us consider the decision table shown in Table 1: The first column ID# is transaction id. RESULT is the decision attribute. TEST, LOW, HIGH, CASE and NEW are conditional attributes.

(1) Row reduction by merging duplicate rows

Step 1: An equivalence relation can be defined by RESULT:

ID-i $\cong$ ID-j iff ID-i.RESULT = ID-j.RESULT

It partitions the transaction into three decision classes:

DECISION1={ID-1, ID-2, ID-3, ID-4, ID-5, ID-6, ID-7, ID-8, ID-9}={1}
DECISION2={ID-10, ID-11, ID-12, ID-13, ID-14}={2}
DECISION3={ID-15, ID-16, ID-17, ID-18}={3}

Step 2: For the conditional attributes {TEST, LOW, HIGH, CASE, NEW}, we have the following condition classes: CONDITION1 = {ID-1, ID-2};
CONDITION2 = {ID-3}; CONDITION3 = {ID-4, …, ID-9};
CONDITION4 = {ID-10}; CONDITION5 = {ID-11, …, ID-14};
CONDITION6 = {ID-15}; CONDITION7 = {ID-16, ID-17, ID-18}.

**Table 1.** An decision table

| ID# | TEST | LOW | HIGH | CASE | NEW | RESULT |
|---|---|---|---|---|---|---|
| ID-1. | 1 | 0 | 0 | 2 | 1 | 1 |
| ID-2. | 1 | 0 | 0 | 2 | 1 | 1 |
| ID-3. | 1 | 1 | 1 | 2 | 1 | 1 |
| ID-4. | 0 | 1 | 1 | 3 | 2 | 1 |
| ID-5. | 0 | 1 | 1 | 3 | 2 | 1 |
| ID-6. | 0 | 1 | 1 | 3 | 2 | 1 |
| ID-7. | 0 | 1 | 1 | 3 | 2 | 1 |
| ID-8. | 0 | 1 | 1 | 3 | 2 | 1 |
| ID-9. | 0 | 1 | 1 | 3 | 2 | 1 |
| ID-10. | 0 | 1 | 1 | 2 | 1 | 2 |
| ID-11 | 1 | 1 | 0 | 2 | 1 | 2 |
| ID-12. | 1 | 1 | 0 | 2 | 1 | 2 |
| ID-13. | 1 | 1 | 0 | 2 | 1 | 2 |
| ID-14. | 1 | 1 | 0 | 2 | 1 | 2 |
| ID-15. | 0 | 1 | 0 | 2 | 1 | 3 |
| ID-16. | 1 | 0 | 1 | 2 | 1 | 3 |
| ID-17. | 1 | 0 | 1 | 2 | 1 | 3 |
| ID-18. | 1 | 0 | 1 | 2 | 1 | 3 |

Step 3: Compare condition and decision classes obtained above. We have seven inclusions that give us seven decision rules and can be represented in Table 2, where the column *# of items* indicates the number of rows that match the corresponding rule.

R1: CONDITION1 → DECISION1; R2: CONDITION2→ DECISION1;
R3: CONDITION3 → DECISION1; R4: CONDITION4 → DECISION2;
R5: CONDITION5 → DECISION2; R6: CONDITION6 → DECISION3;
R7: CONDITION7 → DECISION3.

(2) Attribute reduction by finding the attribute reduct. One can observe that CASE and NEW are RESULT-dispensable, but not both together. We can drop either CASE or NEW without affecting decision rules. Other attributes are indispensable. Hence we have two minimal attribute reducts: {TEST, LOW, HIGH, CASE}, and {TEST, LOW, HIGH, NEW}. Without loss of generality, we consider the first attribute reduct.

**Table 2.** Decision rules

| Rule# | TEST | LOW | HIGH | CASE | NEW | RESULT | # of items |
|---|---|---|---|---|---|---|---|
| R1 | 1 | 0 | 0 | 2 | 1 | 1 | 2 |
| R2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 |
| R3 | 0 | 1 | 1 | 3 | 2 | 1 | 6 |
| R4 | 0 | 1 | 1 | 2 | 1 | 2 | 1 |
| R5 | 1 | 1 | 0 | 2 | 1 | 2 | 4 |
| R6 | 1 | 1 | 0 | 2 | 1 | 3 | 1 |
| R7 | 1 | 0 | 0 | 2 | 1 | 3 | 3 |

(3) Value reduction by finding the value reduct for each rule. To illustrate the idea, we will compute the value reduct for first rule. Let $[R1]_{TEST}$ denotes the equivalence class of rules which are induced from the attribute TEST in Table 2, namely

$[R1]_{TEST} = \{R1, R2, R5, R7\}$; $[R1]_{LOW} = \{R1, R7\}$;

$[R1]_{HIGH} = \{R1, R5, R6\}$; $[R1]_{CASE} = \{R1, R2, R4, R5, R6, R7\}$.

$F = \{[R1]_{TEST}, [R1]_{LOW}, [R1]_{HIGH}, [R1]_{CASE}\}$.

$\cap F = [R1]_{TEST} \cap [R1]_{LOW} \cap [R1]_{HIGH} \cap [R1]_{CASE} = \{R1\}$.

By dropping each component, we find the minimal subfamilies of F, called value reduct, such that the following inclusion holds: $[R1]_{LOW} \cap [R1]_{HIGH} \subseteq \cap F$. So for rule R1, we have a set of minimal conditions. We summarize all the value reducts, shown in Table 3, which represents the minimal conditions for all rules R1 through R7.

**Table 3.** Value reducts of decision rules

| Rule# | TEST | LOW | HIGH | CASE | NEW | RESULT | # of items |
|---|---|---|---|---|---|---|---|
| R1 | | 0 | 0 | | | 1 | 2 |
| R2 | 1 | 1 | 1 | | | 1 | 1 |
| R3 | | | | 3 | | 1 | 6 |
| R4 | 0 | | 1 | 2 | | 2 | 1 |
| R5 | 1 | 1 | 0 | | | 2 | 4 |
| R6 | 0 | | 0 | | | 3 | 1 |
| R7 | | 0 | 1 | | | 3 | 3 |
| R3’ | | | | | 2 | 1 | 6 |
| R4’ | 0 | | 1 | | 1 | 2 | 1 |

If the other attribute reduct {TEST, LOW, HIGH, NEW} is chosen, one can find two more decision rules, shown as rules R3’ and R4’ in Table 3.

One can see that no matter which attribute reduct is taken, two decision rules will be missing. Though they are equivalent in the table, there is no general way to tell from which is more important. Consider two rules R3 and R3’. For a future instance that misses the CASE value, R3’ can be applied, while if the instance misses the NEW value, R3 can be applied. If we are not interested in rules with low supporting

cases, we can drop all the rules with *# of items* less than a predefined threshold. Table 3 shows that rules R2, R4, R6, and R4' have only support of 1, and thus can be ignored.

## 3 Finding High Frequent Value Reduct

The method introduced in Section 2 is an elegant approach when data is clean and small. One of the main contributions of rough set theory to data mining is data reduction. There are three reductions: attribute (column) reduction, row reduction, and value reduction. Value reduction is to reduce the decision rules to a logically equivalent minimal subset of minimal length. Row reduction is merging only the duplicate rows. Attribute reduction is to find important attributes. Traditionally, the value reduct has been searched through the attribute reduct. This method may miss important decision rules with any attribute reducts chosen, not to mention, finding all or minimum reducts is a NP-hard problem [14].

On the other hand, if Table 1 is a very large database, then Table 3 is also large. One should note that the transaction table is usually sparse; most entries are null. For such databases, it is extremely difficult to apply rough set methodology directly. Fortunately, in a sparse table, each item can be represented and stored as attribute-value pair, i.e., (attribute, integer), where the attribute is an encoding of an item name and the integer is the number of items purchased. We will refer to its encoding as encoded pair, encoded item or simply item. Each customer transaction is a variable length record. Each record is a sequence of encoded items. As usual, these data are indexed by B+-trees. The total size of non-leaf nodes are small, we can assume the tree stays in main memory at all time.

Adopting the frequent itemset idea from association rules [1] and case based reasoning [10], we redevelop/refine a bottom-up approach of rough set methodology [11] in this section to find value reduct in very large databases without finding attribute reducts. Our method can discover concise and important decision rules.

Assume a given decision table or information table is stored as a relational table as above. As in [1], an itemset consists of encoded items of uniform length, say k, k=2, 3, ,,,. We distinguish condition items that are formed by condition columns from decision items that are generated from decision attribute (column). Each k-item is constituted of k-1 condition attributes and 1 decision attribute, thus k-itemsets are information tables or sub-decision tables. All itemsets can be constructed iteratively until either all items are exhausted or no more interesting itemsets can be generated.

With the required minimum support of transactions, we can form a sequence of information tables. In database mining, we do not know our targets, so information tables are more suitable than decision tables. However, we will base our discussions on decision tables. Our approach is described in the following algorithm.

**Algorithm:** Finding all decision rules in a decision table by value reduction
**Input:** A decision table *T* in a relational table with condition attribute set *C*, and a decision attribute *d;* a minimum support threshold *s*
**Output:** *RB,* a set of decision rules

**Procedure:**

1. RB ← empty
2. For k=1 to |C| Do
3. RBk ← empty
4. For each subset of C, A of size k, Do
5. TA ← create a subset from T with all columns in A
6. Remove all inconsistent and insufficient support tuples from TA
7. For each remaining tuple r in TA Do
8. If r is not covered by R Then RBk ← RBk ∪ {r}
9. If RBk = empty Then Return RB
10. Else R ← R ∪ RBk
11. Return R

**End**

The main operations in the algorithm are creating a subset of T from a given subset A of all condition columns C and removing all inconsistent and insufficient support tuples in Lines 5 and 6. In relational databases, these can be easily implemented in SQL [4] by creating a view from the relational table, and thus saving storage spaces. Assume A = {A1, A2, …, Ap}, then the following SQL statement works:

```
CREATE VIEW TA
SELECT A1, A2, …, Ap, d, sum(support)
FROM (SELECT A1, A2, … Ap, d, count(*) support
       FROM T
       GROUP BY A1, A2, …, Ap, d)
GROUP BY A1, A2, …, Ap
HAVING count(*)  = 1 and sum(support) >= s
```

The inner SELECT statement groups (merges) all duplicate tuples and count their support, while the outer SELECT statement removes all inconsistent tuples which have the same conditions but different decisions.

In Lines 7 and 8, tuples that are covered by current RB is discarded. A tuple is covered by RB if it is a super-tuple of a tuple in RB.

## 4 An Example

In this section, let's reconsider the example in Section 2, shown in Table 1, and demonstrate the execution of the algorithm proposed in Section 3. For readers' sake, we represent items by tables, though they should be represented in sequences of encoded items. To save the space, each column in the following table represents a k-item relational table, where sequences in parenthesis are the value sequences of attributes listed in the column title, and the numbers after parentheses are supports of the preceding items. Assume the support threshold s=1.

**Loop 1:** Finding 2-itemset. Table 4 shows all 2-items with one condition item and one decision item before removing inconsistent tuples. One can see there are only two consistent 2-items in shading, which should be added to RB2 and RB.

**Table 4.** 2-items with one condition item and one decision item

| TEST, RESULT | LOW, RESULT | HIGH, RESULT | CASE, RESULT | NEW, RESULT |
|---|---|---|---|---|
| (1, 1), 3 | (0, 1), 2 | (0, 1), 2 | (2, 1), 3 | (1, 1), 3 |
| (1, 1), 6 | (1, 1), 7 | (1, 1,), 7 | (3, 1), 6 | (2, 1), 6 |
| (0, 2), 1 | (1, 2), 5 | (1, 2), 1 | (2, 2), 5 | (1, 2), 5 |
| (1, 2), 4 | (1, 3), 1 | (0, 2), 4 | (2, 3), 4 | (1, 3), 4 |
| (0, 3), 1 | (0, 3), 3 | (0, 3), 1 | | |
| (1, 3), 3 | | (1, 3), 3 | | |

**Loop 2:** Finding 3-itemset. Table 5 only shows the remaining 3-items after removing all inconsistent tuples as well as those tuples that are covered by RB. For example, in the third column, (((TEST, 0), (CASE, 3), (RESULT, 1)), 6) is covered by (((CASE, 3), (RESULT, 1)), 6) in RB and thus is removed. After this loop, RB3 contains three 3-items and is added to RB.

**Table 5.** 3-items with two condition items and one decision item

| TEST, LOW, RESULT | TEST, HIGH, RESULT | TEST, CASE, RESULT | TEST, NEW, RESULT | LOW, HIGH, RESULT |
|---|---|---|---|---|
| | (0, 0, 3), 1 | | | (0, 0, 1), 2 |
| | | | | (0, 1, 3), 3 |
| LOW, CASE, RESULT | LOW, NEW, RESULT | HIGH, CASE, RESULT | HIGH, NEW, RESULT | NEW, CASE, RESULT |

**Loop 3:** Finding 4-items. Table 6 illustrates all consistent 4-items with support generated from the original table. RB4 contains four 4-items and is added to RB.

**Table 6.** 4-items with three condition items and one decision item

| TEST, LOW, HIGH, RESULT | TEST, LOW, CASE, RESULT | TEST, LOW, NEW, RESULT | TEST, HIGH, CASE, RESULT | TEST, HIGH, NEW, RESULT |
|---|---|---|---|---|
| (1, 1, 1, 1), 1 | | | (0, 1, 2, 2), 1 | (0,1,1,2), 1 |
| (1, 1, 0, 2), 4 | | | | |
| TEST, NEW, CASE, RESULT | LOW, HIGH, CASE, RESULT | LOW, HIGH, NEW, RESULT | LOW, NEW, CASE, RESULT | HIGH, NEW, CASE, RESULT |

**Loop 4:** Finding 5-itemset. One can verify that there are no any 5-items which are consistent and not covered by RB. Thus the procedure stops.

In summarization, the above procedure outputs 9 decision rules:

R1'': CASE = 3 ➔ RESULT = 1 with support = 6
R2'': NEW = 2 ➔ RESULT = 1 with support = 6
R3'': TEST = 0, HIGH = 0 ➔ RESULT = 3 with support = 1
R4'': LOW = 0, HIGH = 0 ➔ RESULT = 1 with support = 2
R5'': LOW = 0, HIGH = 1 ➔ RESULT = 3 with support = 3
R6'': TEST = 1, LOW = 1, HIGH = 1 ➔ RESULT = 1 with support 1
R7'': TEST = 1, LOW = 1, HIGH = 0 ➔ RESULT = 2 with support 4
R8'': TEST = 0, HIGH = 1, CASE = 2 ➔ RESULT = 2 with support 1
R9'': TEST = 0, HIGH = 1, NEW = 1 ➔ RESULT = 2 with support 1

Comparing above rules with Table 3, one can find that they are exactly the same. If the minimum support threshold is set to 1, then only five items remain, indicating five decision rules, also shown in Table 3.

## 5 Conclusion

Traditional rough set theory induces decision rules in a decision table by finding attribute reducts first and value reducts second. Unfortunately, generating decision rules from any attribute reducts may miss some important rules, and finding all attribute reducts is NP-hard. In this paper, we presented an approach to mining important decision rules by finding value reducts directly without finding attribute reducts. Our approach integrates the itemset idea of mining association rules and can be implemented with efficient RDBMS operations, and thus can be applied in very large databases. The algorithm was described and illustrated with an example. We also discussed the algorithm implementation in SQL statements.

The approach proposed in this paper mines only fully confident decision rules. Our future work is to relax this constraint so that soften decision rules can be discovered.

## References

1. Agrawal, R., Imielinski, T., and Swami, A.: Mining Association Rules between Sets of Items in Large Databases, Proc. of the ACM SIGMOD Conference (1993) 207-216.
2. Agrawal, R. and Srikant, R.: Fast algorithms for mining association rules, Proc. of the 20th International Conference on Very Large Data Bases, Morgan Kaufmann (1994) 487-499.
3. R. Chen and T. Y. Lin, "Supporting Rough Set Theory in Very Large Database Using ORACLE RDBMS," *Soft Computing in Intelligent Systems and Information Processing*, Proceedings of 1996 Asian Fuzzy Systems Symposium, Kenting, Taiwan, December 11-14, (1996) 332-337
4. R. Chen and T. Y. Lin, "Finding Reducts in Very Large Databases," *Proceedings of Joint Conference of Information Science*, Research Triangle Park, North Carolina, March 1-5, (1997) 350-352.
5. Fernandez-Baizan, M., Ruiz, E., and Wasilewska, A.: A Model of RSDM Implementation, Lecture Notes in Computer Science 1424, Springer (1998) 186-193.
6. Garcia-Molina, H., Ullman, J., and Widom, J.: Database Systems: The Complete Book, Prentice Hall (2001).

7. Han, J., Hu, X., and Lin, T.: A new computation model for rough set theory based on database systems, Lecture Notes in Computer Science 2737, Springer (2003) 381-390.
8. Houtsma, M., and Swami, A.: Set-Oriented Mining for Association Rules in Relational Databases, Proc. of Internal Conf. on Data Engineering (1995) 25-33,.
9. Hu, X., Lin, T., Han, J.: A new rough sets model based on database systems, J. of Fundamenta Informaticae 59(2-3) (2004) 135-152
10. Lin, T. Y.: "Neighborhood Systems and Approximation in Database and Knowledge Base Systems", *Proceedings of the Fourth International Symposium on Methodologies of Intelligent Systems*, Poster Session, October 12-15 (1989) 75-86.
11. T. Y. Lin, Rough Set Theory in Very Large Database Mining**,** in: *Symposium on Modeling, Analysis and Simulation*, CESA'96 IMACS Multi Conference (Computational Engineering in Systems Applications), Lille, France, July 9-12, Vol. 2 (1996) 936-94.
12. Pawlak, Z., Rough Sets, International Journal of Information and Computer Science, 11(5) (1982) 341-356.
13. Pawlak, Z.: Rough sets. Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers (1991).
14. Skowron, A. and Rauszer, C.: The discernibility matrices and functions in information systems, Decision Support by Experience - Application of the Rough Sets Theory, R. Slowinski (ed.), Kluwer Academic Publishers (1992) 331-362.

# A Weighted Rough Set Approach for Cost-Sensitive Learning

Jinfu Liu and Daren Yu

Harbin Institute of Technology, 150001 Harbin, China
{liujinfu,yudaren}@hcms.hit.edu.cn

**Abstract.** In many real-world applications, the costs of different errors are often unequal. Therefore, the inclusion of costs into learning, also named cost-sensitive learning, has been regarded as one of the most relevant topics of future machine learning research. Rough set theory is a powerful mathematic tool dealing with inconsistent information for attribute dependence analysis, knowledge reduction and decision rule extraction. However, it is insensitive to the costs of misclassification due to the absence of a mechanism of considering the subjective knowledge. This paper discusses problems connected with introducing the subjective knowledge into rough set learning and proposes a weighted rough set approach for cost-sensitive learning. In this method, weights are employed to represent the subjective knowledge of costs and a weighted information system is defined firstly. With the introduction of weights, weighted attribute dependence analysis is carried out and an index of weighted approximate quality is given. Furthermore, weighted attribute reduction algorithm and weighted rule extraction algorithm are designed to find the reducts and rules with the consideration of weights. Based on the proposed weighted rough set, a series of comparing experimentations with several familiar general techniques on cost-sensitive learning are constructed. The results show that the approach of weighted rough set produces averagely the minimum misclassification costs and the lowest high cost errors.

**Keywords:** Weighted rough set, knowledge reduction, rule extraction, cost-sensitive learning.

## 1 Introduction

In many real-world applications, the costs of different errors are often unequal. For example, in medical diagnosis, the cost of erroneously diagnosing a patient to be healthy may be much bigger than that of mistakenly diagnosing a healthy person as being sick, because the former kind of error may result in the loss of a life. Recently, these kinds of learning problems have been recognized as a crucial problem in machine learning and data mining, and many cost-sensitive learning methods have been developed [1]. However, most of the research efforts make decision trees, neural networks and SVM cost-sensitive [2,3,4].

Rough set theory, proposed by Pawlak [5], has been a powerful tool to deal with inconsistent information. However, it is insensitive to the costs of misclassification

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 355–362, 2007.

due to the absence of a mechanism of considering the subjective knowledge. Through assigning each attribute an appropriate weight in the reduction process, Xu C.-Z. introduced some subjective knowledge of experts into attribute reduction [6]. But the subjective knowledge of objects related with misclassification costs can't be considered in this method yet. In [7] and [8], probability rough set was introduced and each object was associated with a probability $p(x)$, which may include some subjective knowledge of objects. However, how to determine the probability in applications was not given. What's more, specific knowledge acquiring algorithms were not presented and systemic experimental analyses were not carried out.

Note that there are some general learning techniques addressing the cost-sensitive learning problem outside the realm of rough set. Oversampling and undersampling is one kind of these popular methods, which resamples each class until the appearance of examples of each class is proportional to its costs. However, many studies have shown that oversampling usually increases the training time and may lead to overfitting since it involves the exact copies of some examples, and undersampling discards potentially useful training examples and the performance of the resulting classifier may be degraded[9]. Another method for cost-sensitive learning is to employ the minimum expected cost criterion in selecting a predicted class during classification [10]. This criterion affects only the classification process, not the training process. Since the inconsistent use of costs from training to classification, it can't usually achieve the minimum misclassification costs [2].

In order to make rough set cost-sensitive, we propose a weighted rough set approach for cost-sensitive learning in this paper. The rests is organized as follows. Weighted rough set learning is proposed in section 2. Experiments of cost-sensitive learning based on weighted rough set are carried out in section 3. Section 4 concludes.

## 2 Weighted Rough Set Learning

A weighted information system is formally denoted by $WIS =< U, W, A, V, f >$, where $U$ is a finite set of objects, $W$ is a weight distribution on $U$, $A$ is a finite set of attributes, $V$ is the value domain of $A$, and $f$ is an information function $f : U \times A \to V$. If $A = C \bigcup D$, where $C$ is the condition attribute set and $D$ is the decision attribute set, $WIS$ is called a weighted decision table.

In WIS, weights provide some necessary and additional information about applications, which can not be given by data. Since the equivalence class is the elementary information granule that expresses knowledge, it is necessary to obtain the weight of a set of objects. Let $w(X)$ be the weight of $X \subseteq U$, $w(Y)$ be the weight of $Y \subseteq U$ and $w(X \bigcup Y)$ be the weight of $X \bigcup Y$, if $X \bigcap Y = \varnothing$, then

$$w(X \bigcup Y) = \frac{w(X)p(X) + w(Y)p(Y)}{p(X \bigcup Y)}, \tag{1}$$

where $p(X)$, $p(Y)$ and $p(X \bigcup Y)$ represent respectively the probability of the set of objects $X$, $Y$ and $X \bigcup Y$.

After the introduction of weights into an information system, since the family of equivalence class associated with the set of attributes A is the same as that in classical information system, lower and upper approximation of the decision class don't vary. However, the quality of classification under the subjective knowledge represented by weights will be quite different. For $WIS =< U, W, A = C \bigcap D, V, f >$, if $B \subseteq C$ is the condition attribute set, $D$ is the decision attribute set and $Pos_B(D)$ is the $B$-positive region of classification induced by $D$, the weighted quality of classification induced by D by set of attributes $B$, denoted by $\gamma_B^W(D)$, is defined as

$$\gamma_B^W(D) = \frac{w(pos_B(D))\left|pos_B(D)\right|}{w(U)\left|U\right|}. \tag{2}$$

If the weight of each object of $U$ is equal, the weighted quality of classification degenerates into the classical one.

Attribute reduction is a core problem in rough set. Based on the weighted quality of classification, we design a heuristic attribute reduction algorithm under the subjective knowledge as Algorithm 1. In our algorithm, in order to restrain the noise, a threshold $\varepsilon$ is introduced. This algorithm selects the attribute with the greatest weighted quality of classification in sequence until $\gamma_C^W(D) - \gamma_B^W(D) \le \varepsilon$, where $B \subseteq C$.

**Algorithm 1.** Weighted attribute reduction under the subjective knowledge

```
Input: WIS =< U, W, A = C ∪ D, V, f > and a threshold ε.
Output: a D-reduct B of C.
begin
   compute the maximal weighted quality of
   classification γ_C^W(D);
   B ← ∅;
   while B ⊂ C do
   begin
      for each a ∈ C − B do
         compute γ_{B∪{a}}^W(D);
      select a_max such that γ_{B∪{a}}^W(D) is maximum;
      B ← B ∪ {a_max};
      if γ_C^W(D) − γ_B^W(D) ≤ ε then exit the loop;
   end
   for each a ∈ B
      if γ_C^W(D) − γ_{B−{a}}^W(D) ≤ ε then B ← B − {a};
   return B;
end
```

One of the most important problems which can be solved using rough set is rule extraction. For $IS =< U, A, V, f >$, if $B \subseteq A$ and $E \in U / IND(B)$, $Des(E, B) = \wedge(a = f_a(E))$, where $a \in B$, is called the description of class $E$ with

respect to $B$. For a decision table $IS =< U, A = C \bigcap D, V, f >$, if $B \subseteq C$, $X \in U / IND(B)$ and $Y \in U / IND(D)$, a decision rule $r$, is an assertion of the form

$$Des(X, B) \rightarrow Des(Y, D), \tag{4}$$

where $Des(X, B)$ is the condition part of $r$ and $Des(Y, D)$ is the decision part of $r$.

Nowadays, there are many known rule extraction algorithms inspired by the rough set theory. Among these algorithms, LEM2 algorithm, proposed by Grzymala in [11], is one of the most used rough set based rule induction algorithm in real-life applications. In LEM2, a generalized decision is defined firstly, which is a decision class, or is the union of more than one decision classes. According to the generalized decisions, the set of objects is partitioned as a family of several disjoint subsets of objects associated with the generalized decisions, denoted by $\tilde{Y}$. Each of $\tilde{Y}$ is the lower approximation of a decision classification $Y \in U / IND(D)$, or is one of the disjoint subsets of the boundary of a decision classification. For instance, let us assume that three decision classifications, $Y_1, Y_2, Y_3$, are roughly defined in the

**Algorithm 2.** Weighted rule extraction under the subjective knowledge

```
Input: a set of objects K ∈ Ỹ .
Output: rule set R of K .
begin
    G ← K , R ← ∅ ;
    while G ≠ ∅ do
    begin
        Φ ← ∅ , Φ_G ← {c : [c] ⋂ G ≠ ∅} ;
        while ( Φ = ∅ ) or (not( [Φ] ⊆ K )) do
        begin
            for each c ∈ Φ_G , select c_max such that
            w([c] ⋂ G) / |[c] ⋂ G| is maximum;
            Φ ← Φ ⋃ {c_max} , G ← [c_max] ⋂ G ;
            Φ_G ← {c : [c] ⋂ G ≠ ∅} , Φ_G ← Φ_G − Φ ;
        end
        for each c ∈ Φ do
            if [Φ − c] ⊆ K then Φ ← Φ − {c} ;
        create rule r basing on the conjunction Φ ;
        R ← R ⋃ {r} , G ← K − ⋃_{r∈R} [r] ;
    end
    for each r ∈ R do
        if ⋃_{s∈R−r} [S] = K then R ← R − r ;
end
```

decision table. The boundary of the class $Y_1$ consists of three disjoint subsets, i.e. $BND(Y_1) = (\overline{BY}_1 \bigcap \overline{BY}_2 - \overline{BY}_3) \bigcup (\overline{BY}_1 \bigcap \overline{BY}_3 - \overline{BY}_2) \bigcup (\overline{BY}_1 \bigcap \overline{BY}_2 \bigcap \overline{BY}_3)$. Obviously, $\tilde{Y}$

is consistent for every generalized decision. For each $K \in \tilde{Y}$, LEM2 uses a heuristic strategy to generate a minimal rule set of *K*.

Based on LEM2, we design a rule extraction algorithm under the subjective knowledge as Algorithm 2. In Algorithm 2, $c$ is an elementary condition of the description of class with respect to the condition attribute set and $\Phi$ is a conjunction of such elementary conditions being a candidate for condition part of a decision rule. Additionally, $\Phi_G$ denotes the set of elementary conditions currently considered to be added to the conjunction $\Phi$ and $[\Phi]$ denotes the cover of $\Phi$.

In order to evaluate discovered rules, the weighted support coefficient and confidence coefficient can be defined as follows.

For a weighted decision table $WIS =< U, W, A = C \bigcap D, V, f >$, if $B \subseteq C$, $X \in U / IND(B)$ and $Y \in U / IND(D)$, the weighted support coefficient and weighted confidence coefficient of a decision rule $r : Des(X, B) \to Des(Y, D)$, respectively denoted by $\mu_s^W(r)$ and $\mu_c^W(r)$, are defined as

$$\mu_s^W(r) = \frac{w(X \bigcap Y)|X \bigcap Y|}{w(U)|U|}, \quad \mu_c^W(r) = \frac{w(X \bigcap Y)|X \bigcap Y|}{w(X)|X|}. \tag{5}$$

## 3 Experiments on Cost-Sensitive Learning

In order to carry out cost-sensitive learning, the weight of each class associated with misclassification cost must be given firstly. Suppose that $Cost(i, j)$ denotes the cost of misclassifying an object of the *i*th class to the *j*th class and $w(i)$ denotes the weight of the *i*th class associated with misclassification cost. $w(i)$ can be usually derived from $Cost(i, j)$ and a popular rule of the derivation, where $Cost(i, j)$ is partitioned into three types, is defined as follows [2]:

(a) $1.0 < Cost(i, j) \le 10.0$ only for a single value of $j = J$ and $Cost(i, j \ne J) = 1.0$ for all $j \ne i$. Define $w(i) = Cost(i, J)$ for $j \ne J$ and $w(J) = 1.0$.

(b) $1.0 \le Cost(i, j) = H_i \le 10.0$ for each $j \ne i$ and at least one $H_i = 1.0$. Define $w(i) = H_i$.

(c) $1.0 \le Cost(i, j) \le 10.0$ for all $j \ne i$ and at least one $Cost(i, j) = 1.0$. Define $w(i) = \sum_j Cost(i, j)$.

Based on the proposed weighted rough set (WRS), 19 UCI data sets[12], which consist of 10 two-class data sets(echocardiogram, Hepatitis, heart_s, breast, horse, votes, credit, breast_w, tictoc, german) and 9 multi-class data sets(zoo, lymphography, wine, machine, glass, audiology, heart, solar, soybean), are used in the empirical study. In these data sets, missing values on continuous attributes are set to

the average value while those nominal attributes are set to the majority value, and all the continuous attributes in each data set are discretized via entropy (MDLP) [13].

Moreover, along with WRS, several familiar general techniques on cost-sensitive learning, including oversampling(OS), undersampling(US) and the minimum expected cost criterion(MC), are also selected to perform cost-sensitive learning in rough set. For every method, some specific configurations are summarized as follows:

1) WRS: assigning every objects with the weight of misclassification costs, then using the proposed weighted rough set to learning and using the majority voting of weighted support coefficient to classification.

2) RS: assigning every objects with the equal weight, then using the proposed weighted rough set to learning and using the majority voting of weighted support coefficient to classification.

3) OS: random oversampling the *k*th class, which have $N_k$ training objects, until the appearance of training objects of each class is proportional to its weight of misclassification costs, then using RS to learning and classification. If the $\lambda$ -class has the smallest number of training objects to be duplicated, then ( $N_k^* - N_k$ ) number of training objects of the *k*th class will be resampled random, where $N_k^* = \lfloor N_\lambda * w(k) / w(\lambda) \rfloor$ [9].

4) US: random undersampling the *k*th class according to the contrary process of oversampling, then using RS to learning and classification.

5) MC: assigning every objects with the equal weight, then using RS to learning, and using the minimum expected cost criterion to classification. The expected cost for predicting class $i$ with respect to object $x$ is given by $EC_i(x) = \sum_j Vote_j(x) \cos t(j,i)$, where $Vote_j(x)$ denotes the number of votes for predicting class *j*.

In cost-sensitive learning, three measures are usually used to evaluate the performance. They are the total misclassification costs, the number of high cost errors and the total number of misclassification on unseen data. The first and the second are the most important measures. While the aim of cost-sensitive is to minimize the total misclassification costs, it is important to measure the number of high cost errors since the aim is achieved through high cost errors minimization.

Via 10-fold cross-validations with randomly generated cost matrices belonging to the same cost type, we carried out experiments on cost-sensitive learning using the above five methods. The detailed results on two-class data sets are shown in Table 1 and the average results under each type of cost matrix on multi-class data sets are given in Table 2. The results suggest that WRS achieves averagely the minimum misclassification costs and the lowest high cost errors under each type of cost matrix on both two-class and multi-class data sets. As compared with WRS, though US achieves the minimum misclassification costs on the two-class data sets, it obtains the worst results on the multi-class data sets. MC and OS acquire the middle results. Additionally, RS, WRS and MC cost the similar learning time, and US needs the minimum learning time, and OS costs the maximum learning time.

**Table 1.** Detail results of cost-sensitive learning on two-class data sets

| Dataset | Misclassification costs | | | | | No. high cost errors | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | RS | WRS | MC | OS | US | RS | WRS | MC | OS | US |
| echo | 13.80 | 0.543 | 0.558 | 0.615 | 0.605 | 1.8 | 0.222 | 0.277 | 0.388 | 0.333 |
| hepa | 12.25 | 0.644 | 0.844 | 0.836 | 0.387 | 1.3 | 0.769 | 0.846 | 1.000 | 0.307 |
| heart_s | 16.50 | 0.848 | 0.890 | 0.954 | 0.806 | 3.0 | 0.800 | 0.833 | 0.866 | 0.533 |
| breast | 32.50 | 0.881 | 0.920 | 0.870 | 0.561 | 4.2 | 0.881 | 0.904 | 0.857 | 0.500 |
| horse | 4.65 | 0.817 | 0.849 | 0.763 | 0.903 | 0.7 | 0.714 | 0.857 | 0.714 | 0.428 |
| votes | 6.25 | 0.920 | 0.952 | 0.880 | 0.584 | 0.8 | 0.750 | 0.875 | 0.625 | 0.250 |
| credit | 29.35 | 0.875 | 0.836 | 0.870 | 0.724 | 4.3 | 0.860 | 0.767 | 0.860 | 0.441 |
| breast_w | 13.25 | 0.637 | 0.894 | 0.818 | 0.773 | 2.0 | 0.500 | 0.800 | 0.750 | 0.500 |
| tictoc | 31.00 | 0.827 | 0.787 | 0.830 | 0.811 | 5.1 | 0.823 | 0.803 | 0.827 | 0.490 |
| german | 102.3 | 0.516 | 0.517 | 0.521 | 0.554 | 17.8 | 0.033 | 0.033 | 0.043 | 0.078 |
| Mean | 26.18 | 0.751 | 0.805 | 0.796 | 0.671 | 4.1 | 0.635 | 0.699 | 0.693 | 0.386 |
| Dataset | No. errors | | | | | Time | | | | |
| | RS | WRS | MC | OS | US | RS | WRS | MC | OS | US |
| echo | 4.0 | 1.375 | 1.400 | 1.550 | 1.650 | 0.353 | 1.048 | 1.008 | 6.789 | 0.648 |
| hepa | 2.7 | 0.925 | 0.963 | 1.037 | 1.037 | 2.559 | 0.986 | 1.166 | 7.324 | 0.366 |
| heart_s | 6.7 | 0.970 | 1.044 | 1.014 | 1.358 | 8.294 | 1.028 | 1.128 | 7.103 | 0.328 |
| breast | 11.4 | 0.964 | 0.991 | 0.964 | 1.017 | 11.72 | 1.163 | 1.136 | 6.608 | 0.363 |
| horse | 1.6 | 0.937 | 0.937 | 1.125 | 2.312 | 3.826 | 0.993 | 1.003 | 42.94 | 0.333 |
| votes | 2.1 | 1.238 | 1.190 | 1.190 | 1.428 | 11.79 | 1.001 | 1.012 | 10.75 | 0.321 |
| credit | 14.5 | 1.013 | 1.006 | 1.013 | 1.262 | 187.8 | 1.161 | 1.261 | 6.470 | 0.361 |
| breast_w | 4.8 | 0.958 | 1.062 | 0.854 | 1.520 | 14.55 | 0.987 | 1.178 | 143.3 | 0.278 |
| tictoc | 13.4 | 1.074 | 1.007 | 1.091 | 1.529 | 162.2 | 1.013 | 1.113 | 35.23 | 0.463 |
| german | 29.5 | 1.671 | 1.674 | 1.574 | 1.783 | 23.80 | 1.002 | 1.042 | 8.928 | 0.342 |
| Mean | 9.07 | 1.113 | 1.127 | 1.141 | 1.490 | 42.70 | 1.038 | 1.105 | 27.55 | 0.380 |

The table entries present the real results of RS or the ratio of other method against RS

**Table 2.** Average results under each type of cost matrix on multi-class data sets

| | Data set | No. errors | | | | |
|---|---|---|---|---|---|---|
| | | RS | WRS | MC | OS | US |
| (a) | Misclassification costs | 10.7165 | 0.8702 | 0.9104 | 0.9311 | 1.9290 |
| | No. high cost errors | 1.8222 | 0.7118 | 0.8354 | 0.9213 | 1.8931 |
| | No. errors | 6.4333 | 1.1947 | 1.2797 | 1.1989 | 2.2487 |
| | Time | 44.162 | 1.1921 | 1.2307 | 49.6926 | 0.4163 |
| (b) | Misclassification costs | 30.796 | 0.8969 | 0.9862 | 0.9884 | 1.9782 |
| | No. high cost errors | 4.8556 | 0.8838 | 0.9774 | 0.9874 | 2.0149 |
| | No. errors | 6.23 | 1.0751 | 1.1189 | 1.3274 | 2.6128 |
| | Time | 43.0187 | 1.1373 | 1.2108 | 93.7216 | 0.1752 |
| (c) | Misclassification costs | 32.5504 | 0.9616 | 0.9992 | 1.1207 | 2.0169 |
| | No. high cost errors | 5.6222 | 0.9332 | 0.9850 | 1.0939 | 2.1971 |
| | No. errors | 6.4333 | 1.1377 | 1.2218 | 1.4966 | 3.0306 |
| | Time | 45.1938 | 1.0925 | 1.1937 | 98.4257 | 0.1527 |

The table entries present the real results of RS or the ratio of other method against RS

## 4 Conclusions

In this paper, we proposed a weighted rough set learning method for cost-sensitive learning. With the introduction of weights of misclassification costs, some basic definitions of classical rough set are extended, and weighted attribute reduction algorithm and weighted rule extraction algorithm are design to find the reducts and rules with the consideration of weights.

Based on the proposed weighted rough set, a series of comparing experimentations with several familiar general techniques on cost-sensitive learning, including oversampling, undersampling and the minimum expected cost criterion, are constructed. The results show that the approach of weighted rough set produces averagely the minimum misclassification costs and the lowest high cost errors on all data sets.

## References

1. Domingos P.: MetaCost: A General Method for Making Classifiers Cost-Sensitive. Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (1999) 155–164
2. Ting K.M.: An Instance-Weighting Method to Induce Cost-Sensitive Trees. IEEE Trans. Knowledge and Data Eng. 14 (3) (2002) 659–665
3. Zhou Z.-H., Liu X.-Y.: Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. IEEE Trans. Knowledge and Data Eng. 18 (1) (2006) 63–77
4. Brefeld U., Geibel P., Wysotzki F.: Support Vector Machines with Example Dependent Costs. Proc. 14th European Conf. Machine Learning (2003) 23–34
5. Pawlak Z.: Rough Sets. International Journal of Computer and Information Sciences 11 (1982) 341–356
6. Xu C.-Z., Min F.: Weighted Reduction for Decision Tables. Fuzzy Systems and Knowledge Discovery, Proceedings Lecture Notes in Computer Science (2006) 246–255
7. Ma T.-H., Tang M.-L.: Weighted Rough Set Model. Sixth International Conference on Intelligent Systems Design and Applications (2006) 481–485
8. Hu Q.-H., Yu D.-R., Xie Z.-X., Liu J.-F.: Fuzzy Probabilistic Approximation Spaces and Their Information Measures, IEEE Transactions on Fuzzy Systems 14 (2) (2006) 191–201
9. Batista G., Prati R. C., Monard M. C.: A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. SIGKDD Explorations 6 (1) (2004) 20–29
10. Michie D., Spiegelhalter D.J., Taylor C.C.: Machine Learning, Neural and Statistical Classification. Ellis Horwood Limited (1994)
11. Grzymala-Busse J. W.: LERS - A System for Learning From Examples Based on Rough Sets. In R. Slowinski, (ed.) Intelligent Decision Support, Kluwer Academic Publishers (1992) 3–18
12. Blake C., Keogh E., Merz C.J.: UCI Repository of Machine Learning Databases, Dept. of Information and Computer Science, Univ. of California, Irvine, http://www.ics.uci.edu/~mlearn/MLRepository.html (1998)
13. Fayyad U., Irani K.: Discretizing Continuous Attributes While Learning Bayesian Networks. In Proc. Thirteenth International Conference on Machine Learning, Morgan Kaufmann, (1996) 157–165

# Jumping Emerging Pattern Induction by Means of Graph Coloring and Local Reducts in Transaction Databases*

Pawel Terlecki and Krzysztof Walczak

Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
{P.Terlecki,K.Walczak}@ii.pw.edu.pl

**Abstract.** This paper demonstrates how to employ rough set framework in order to induce JEPs in transactional data. The algorithm employs local reducts in order to generate desired JEPs and additional EPs. The number of the latter is decreased by preceding reduct computation with item aggregation. The preprocessing is reduced to graph coloring and solved with efficient classical heuristics. Our approach is contrasted with JEP-Producer, the recommended method for JEP induction. Moreover, a formal apparatus for classified transactional data has been proposed.

**Keywords:** jumping emerging pattern, pattern with negation, transaction database, local reduct, condensed decision table, rough set, graph coloring.

## 1 Introduction

Transactional patterns are present in various areas of knowledge discovery. Association rules based on frequent itemsets ([1]) are a common example. However, also for either clustering or classification tasks many successful approaches have been already proposed ([2,3]). Due to the size of a search space, efficient induction and concise representation remain the most important challenges.

On the other hand, classical problems defined for transaction databases have their analogues in relational approach, where the rough set theory ([4]) provides a robust and convenient framework. In particular, one can seek associations by means of association reducts ([5]) or generate classification rules by reducts. Since transformations between both data representations are trivial, natural questions emerge about interdependencies among existing methods. As far as original data is relational, it has been demonstrated that rough set approach tends to be more efficient in inducing classification rules ([6]) than respective KDD tools. Following this fact, we check if it can be also a successful alternative for transactional data.

Our paper focuses on classification in transaction databases and extends the formality introduced for emerging patterns ([2]). A key notion is a jumping

* The research has been partially supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 363–370, 2007.

emerging pattern (JEP), defined as a set of items (over a given item space) that exists in some of transactions of one class and is absent in transactions of others. Due to their generality, we are particularly interested in minimal JEPs.

In [6] it has been demonstrated that jumping emerging patterns found for a transformed relational data refer to local reducts in original database narrowed to a positive region. Following this fact one could expect a similar association in our case. However, this intuition is misleading. While values 1 in a respective binary decision table refer to items that belong to transactions, values 0 refer to these items which do not exist. In data mining it is equivalent to say that transactions contain relevant negated items, e.g. the pattern $a\overline{ce}$ is supported by transactions containing $a$ and not containing $c$ or $e$. Thus, minimal patterns obtained by local reducts can contain negated items that are indispensable for holding discernibility. If negated items are ignored, the residual pattern will be a regular emerging pattern (EP), i.e. pattern supported in both classes.

Two common features of transactional data are the large size of an itemspace and sparsity. Although we can obtain the superset of the set of minimal JEPs by means of local reduct computation, the number of additionally generated EPs is often too high. Therefore, we propose to perform a certain item aggregation that do not lead to any information loss on JEPs even for suboptimal aggregations. This step can be very efficient due to sparsity and can significantly decrease the number of additionally generated EPs. It is demonstrated that preprocessing can be reduced to graph coloring and solved with well-known heuristics.

In Section 2 we extend the formal apparatus for emerging patterns and negative knowledge and give preliminaries on the rough set theory. Section 3 explains how item aggregation can be employed to rule induction. It also covers reduction to the graph coloring problem and details of the algorithm. The testing procedure and the results are given in Sect. 4. The paper is concluded in Sect. 5.

## 2 Formal Background

***Emerging Patterns.*** Let a transaction system be a pair $(\mathcal{D}, \mathcal{I})$, where $\mathcal{D}$ is a finite sequence of transactions $(T_1, .., T_n)$ (database) such as $T_i \subseteq \mathcal{I}$ for $i = 1, .., n$ and $\mathcal{I}$ is a non-empty set of items (itemspace). A support of an itemset $X \subseteq \mathcal{I}$ in a sequence $D = (T_i)_{i \in K \subseteq \{1,..,n\}} \subseteq \mathcal{D}$ is defined as $supp_D(X) = \frac{|\{i \in K : X \subset T_i\}|}{|K|}$.

Let a decision transaction system be a tuple $(\mathcal{D}, \mathcal{I}, \mathcal{I}_d)$, where $(\mathcal{D}, \mathcal{I} \cup \mathcal{I}_d)$ is a transaction system and $\forall_{T \in \mathcal{D}} |T \cap \mathcal{I}_d| = 1$. Elements of $\mathcal{I}$ and $\mathcal{I}_d$ are called condition and decision items, respectively. A support for a decision transaction system $(\mathcal{D}, \mathcal{I}, \mathcal{I}_d)$ is understood as a support in the transaction system $(\mathcal{D}, \mathcal{I} \cup \mathcal{I}_d)$.

For each decision item $c \in \mathcal{I}_d$, we define a decision class sequence $C_c = (T_i)_{i \in K}$, where $K = \{k \in \{1, .., n\} : c \in T_k\}$. Notice that each of the transactions from $\mathcal{D}$ belongs to exactly one class sequence. In addition, for a database $D = (T_i)_{i \in K \subseteq \{1,..,n\}} \subseteq \mathcal{D}$, we define a complement database $D' = (T_i)_{i \in \{1,..,n\} - K}$.

Given two databases $D_1, D_2 \subseteq \mathcal{D}$ we define a growth rate $gr_{D_1 \to D_2}(X) = 0$, if $supp_{D_1}(X) = supp_{D_2}(X) = 0$; $\infty$, if $supp_{D_1}(X) = 0 \wedge supp_{D_2}(X) > 0$; $\frac{supp_{D_2}(X)}{supp_{D_1}(X)}$, otherwise. An itemset $X \subseteq \mathcal{I}$ is a $\varrho$-emerging pattern (EP) from $D_1$ to $D_2$, if

$gr_{D_1 \to D_2}(X) > \varrho$ and a jumping emerging pattern (JEP), if $gr_{D_1 \to D_2} = \infty$. A set of all JEPs from $D_1$ to $D_2$ is called a JEP space and denoted by $JEP(D_1, D_2)$. For the data in Tab. 1, $eh$ is a minimal JEP with $supp_{C_1}(eh) = 1/3$ and $ce$ is an $EP$ with $gr_{C_1 \to C_2}(ce) = 1$.

One of the most useful features of jumping emerging patterns is the possibility of storing and maintaining a JEP space in a concise manner [2].

Consider a set $S$. A border is an ordered pair $< \mathcal{L}, \mathcal{R} >$ such that $\mathcal{L}, \mathcal{R} \subseteq 2^S$ are antichains and $\forall_{X \in \mathcal{L}} \exists_{Z \in \mathcal{R}} X \subseteq Z$. $\mathcal{L}$ and $\mathcal{R}$ are called a left and a right bound, respectively. A border $< \mathcal{L}, \mathcal{R} >$ represents a set interval $[\mathcal{L}, \mathcal{R}] = \{Y \in 2^S : \exists_{X \in \mathcal{L}} \exists_{Z \in \mathcal{R}} X \subseteq Y \subseteq Z\}$.

Consider a decision transaction database $(\mathcal{D}, \mathcal{I}, \mathcal{I}_d)$ and two databases $D_1, D_2 \subseteq \mathcal{D}$. According to [2], a collection $JEP(D_1, D_2)$ can be uniquely represented by a border. For $d \in I_d$, we use a border $< \mathcal{L}_d, \mathcal{R}_d >$ to represent the JEP space $JEP(C'_d, C_d)$. Members of left bounds are minimal JEPs.

**Lemma 1 ([2]).** $\forall_{J \subseteq \mathcal{I}} J$ *is minimal (maximal) in* $JEP(C'_d, C_d) \Longleftrightarrow J \in \mathcal{L}_d(\mathcal{R}_d)$.

***Emerging Patterns with Negation.*** This section introduces the notion of negation to our study upon JEPs in decision transaction databases.

Consider a decision transaction system $(\mathcal{D}, \mathcal{I}, \mathcal{I}_d)$. Let $\overline{\mathcal{I}} = \{\overline{I}\}_{I \in \mathcal{I}}$ be a negative itemspace. A set of items and negated items is called an itemset with negation. The set of all such itemsets is defined as $\mathcal{P} = \{X \subseteq \mathcal{I} \cup \overline{\mathcal{I}} : \forall_{I \in \mathcal{I}} I \in X \Longrightarrow \overline{I} \notin X\}$. For an itemset $X \in \mathcal{P}$, we define a positive part $X_p = X \cap \mathcal{I}$, a negative part $X_n = X \cap \overline{\mathcal{I}}$ and a negated pattern $\overline{X} = \{\overline{i}\}_{i \in X}$, assuming $\overline{\overline{I}} = I$.

An extended support of an itemset $X \in \mathcal{P}$ in a database $D \subseteq \mathcal{D}$ is defined as $exsupp_D(X) = \frac{|\{i \in K : X_p \subseteq T_i \wedge X_n \subseteq \overline{\mathcal{I} - T_i}\}|}{|K|}$. An extended growth rate, an emerging pattern and a jumping emerging pattern with negation (JEPN) from $D_1$ to $D_2$ are defined accordingly by means of an extended support. A set of all JEPNs from $D_1$ to $D_2$ is called a JEPN space and denoted by $JEPN(D_1, D_2)$. For the data in Tab. 1, $e\overline{g}$ is a minimal JEPN and $exsupp_{C_1}(e\overline{g}) = 2/3$.

***Elements of the Rough Set Theory.*** Let a decision table be a triple $(\mathcal{U}, \mathcal{C}, d)$, where $\mathcal{U}$ (universum) is a non-empty, finite set of objects, $\mathcal{C}$ is a non-empty finite set of condition attributes and $d$ is a decision attribute. A set of all attributes is denoted by $\mathcal{A} = \mathcal{C} \cup \{d\}$. The domain of an attribute $a \in \mathcal{A}$ is denoted by $V_a$ and its value for an object $u \in \mathcal{U}$ is denoted by a(u). In particular, $V_d = \{c_1, .., c_{|V_d|}\}$ and the decision attribute induces a partition of $\mathcal{U}$ into decision classes $\{U_d\}_{c \in V_d}$. Hereinafter, we use the term *attribute* to denote a condition attribute.

Consider $B \subseteq \mathcal{A}$. An indiscernibility relation $IND(B)$ is defined as $IND(B) = \{(u, v) \in \mathcal{U} \times \mathcal{U} : \forall_{a \in B}\ a(u) = a(v)\}$. Since $IND(B)$ is an equivalence relation it induces a partition of $\mathcal{U}$ denoted by $\mathcal{U}/IND(B)$. Let $B(u)$ be a block of the partition containing $u \in \mathcal{U}$. A $B$-lower approximation of a set $X \subseteq \mathcal{U}$ is defined as follows: $B_*(X) = \{u \in \mathcal{U} \mid B(u) \subseteq X\}$ and a $B$-positive region with respect to a decision attribute $d$ is defined as $POS(B, d) = \bigcup_{X \in \mathcal{U}/IND(\{d\})} B_*(X)$.

A local reduct for an object $u \in \mathcal{U}$ is a minimal attribute set $B \subseteq \mathcal{C}$ such that $\forall_{c \in V_d}(\mathcal{C}(u) \cap U_c = \emptyset \Longrightarrow B(u) \cap U_c = \emptyset)$. It means that the object $u$ can be

differentiated by means of $B$ from all the objects from other classes as well as using $\mathcal{C}$. The set of all local reducts for an object $u$ is denoted by $REDLOC(u,d)$.

**Lemma 2 ([7]).** *$B \in REDLOC(u,d)$ for $u \in POS(\mathcal{C},d) \iff B$ is a minimal set such that $B(u) \subseteq U_{d(u)}$.*

## 3 Item Aggregation in JEP Induction

***Transactional to Relational Transformation.*** Hereinafter, we assume that our data is given by a decision transaction system $\mathcal{DTS} = (\mathcal{D}, \mathcal{I}, \mathcal{I}_d)$, where $\mathcal{D} = (T_1, .., T_n)$, $\mathcal{I} = \{I_1, .., I_m\}$, $\mathcal{I}_d = \{c_1, .., c_p\}$.

A binary decision table for a decision transaction system $\mathcal{DTS}$ is a decision table $BDT_{\mathcal{DTS}} = (\mathcal{U}, \mathcal{C}, d)$ such that $\mathcal{U} = \{u_1, .., u_n\}$, $\mathcal{C} = \{a_1, .., a_m\}$, $V_d = \{c_1, .., c_p\}$; $a_j(u_i) = \begin{cases} 0, I_j \notin T_i \\ 1, I_j \in T_i \end{cases}, \forall_{i \in 1..n, j \in 1..m}$; $d(u_i) = T_i \cap \mathcal{I}_d, \forall_{i \in 1..n}$.

This representation provides distinct values for items which belong to a respective transaction and which do not. Thus, patterns obtained by rough set methods can contain negated items. Since negative knowledge introduce a significant overhead to computation and JEP induction involves only positive patterns, we propose to condense the database by aggregating its items.

We say that a partition $\{p_1, .., p_r\}$ of $\mathcal{I}$ is proper iff $\forall_{T \in \mathcal{D}} \forall_{j \in \{1,..,r\}} |T \cup p_j| <= 1$. A condensed decision table for $DTS$, a decision transaction system, $P = \{p_1, .., p_r\}$, a proper partition of $\mathcal{I}$, $F = \{f_1, .., f_r\}$, where $f_j : 2^{p_j} \mapsto \mathbb{N}$ and $f_j$ is a bijection for each $j \in \{1, .., r\}$ is a decision table $CDT_{\mathcal{DTS},P,F} = (\mathcal{U}, \mathcal{C}, d)$ such that $\mathcal{U} = \{u_1, .., u_n\}$, $\mathcal{C} = \{a_1, .., a_r\}$, $V_d = \{d_1, .., d_p\}$; $a_j(u_i) = f_j(T_i \cup p_j), \forall_{i \in 1..n, j \in 1..r}$; $d(u_i) = T_i \cap \mathcal{I}_d, \forall_{i \in 1..n}$.

For the sake of convenience, we introduce the notation: $condPatt(u,B) = \bigcup_{k \in K} f_k^{-1}(a_k(u))$, where $u \in \mathcal{U}$, $B = \{a_k\}_{k \in K \subseteq \{1,..,r\}}$.

**Table 1.** A sample decision transaction system $DTS = \{\{T_1, .., T_6\}, \{a, b, c, d, e, f, g, h\}, \{c_0, c_1\}\}$, a respective binary decision table and a condensed decision table for a proper partition $\{\{a, b, c\}, \{d, e, f\}, \{g, h\}\}$

| | | |
|---|---|---|
| $T_1$ | $adh$ | $c_0$ |
| $T_2$ | $afg$ | $c_0$ |
| $T_3$ | $ceg$ | $c_0$ |
| $T_4$ | $ce$ | $c_1$ |
| $T_5$ | $beh$ | $c_1$ |
| $T_6$ | $bfg$ | $c_1$ |

$\Longrightarrow$

| | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ | $g$ | $h$ | $d$ |
|---|---|---|---|---|---|---|---|---|---|
| $u_1$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| $u_2$ | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| $u_3$ | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| $u_4$ | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| $u_5$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| $u_6$ | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

$\Longrightarrow$

| | $a_1$ | $a_2$ | $a_3$ | $d$ |
|---|---|---|---|---|
| $u_1$ | 0 | 0 | 0 | 0 |
| $u_2$ | 0 | 1 | 1 | 0 |
| $u_3$ | 1 | 2 | 1 | 0 |
| $u_4$ | 1 | 2 | 2 | 1 |
| $u_5$ | 2 | 2 | 0 | 1 |
| $u_6$ | 2 | 1 | 1 | 1 |

*Example 1.* In Tab. 1 we have a transformation from a sample transactional dataset, through a binary table, to a condensed table generated for a proper partition $\{\{a, b, c\}, \{d, e, f\}, \{g, h\}\}$. Each attribute of a condensed table refers

to a block of a partition and each attribute value to an at most one item. It holds: $condPatt(u_4, a_3) = \emptyset$, $condPatt(u_5, a_3) = h$, $condPatt(u_6, a_3) = g$. Note that the partition $\{\{a, b, c\}, \{d, e, f, g, h\}\}$ is not proper, since $|T_1 \cap \{d, e, f, g, h\}| = 2 > 1$.

Let us consider a condensed decision table $CDT_{\mathcal{DTS},P,F} = (\mathcal{U}, \mathcal{C}, d)$. The following theorem demonstrates that an object from a positive region of a condensed decision table can be used to generate a JEP when one applies an attribute set with each attribute mapping to some not empty set of items.

**Theorem 1.** $\{condPatt(u, R) : u \in POS(R, d) \cap U_{d(u)} \wedge R = \{a_k\}_{k \in K \subseteq \{1,..,r\}} \wedge \forall_{k \in K} a_k(u) \neq f_k(\emptyset)\} = JEP(C'_{d(u)}, C_{d(u)})$

*Proof.* Let us start with $\{condPatt(u, R) : u \in POS(R, d) \cap U_{d(u)} \wedge R = \{a_k\}_{k \in K \subseteq \{1,..,r\}} \wedge \forall_{k \in K} a_k(u) \neq f_k(\emptyset)\} \subseteq JEP(C'_{d(u)}, C_{d(u)})$.

Let $u_g \in POS(R, d) \cap U_c$, $R = \{a_k\}_{k \in K}$, $K \subseteq \{1, .., r\}$, $c = d(u_g)$, $g \in \{1, .., n\}$, $H = \{h \in \{1, .., n\} : u_h \in U - U_c\}$.

Note that we have $\forall_{k \in K} \emptyset \neq f_k^{-1}(a_k(u_g)) = (T_g \cap p_k) \in condPatt(u, R)$ (1).

We have $u_g \in R_*(U_c) \iff R(u_g) \subseteq U_c \iff \{v \in \mathcal{U} : \forall_{k \in K} a_k(u_g) = a_k(v)\} \subseteq U_c \iff \forall_{h \in H} \exists_{k \in K} a_k(u_g) \neq a_k(u_h) \iff \forall_{h \in H} \exists_{k \in K} T_g \cap p_k \neq T_h \cap p_k \iff \forall_{h \in H} \exists_{k \in K} (T_g \cap p_k) \not\subseteq T_h$. According to (1), we have $\forall_{h \in H} \exists_{k \in K} (T_g \cap p_k) \not\subseteq T_h \iff \forall_{T \in C'_c} condPatt(u_g, R) \not\subseteq T \iff supp_{C'_c}(condPatt(u_g, R)) = 0$. On the other hand, $condPatt(u_g, R) \subseteq T_g \in C_c \implies supp_{C_c}(condPatt(u_g, R)) \neq 0$, thus, we have $condPatt(u_g, R) \in JEP(C'_c, C_c)$.

Now, consider the opposite inclusion. Let $J = \{I_j\}_{j \in M \subseteq \{1,..,m\}} \in JEP(C'_c, C_c)$, $H = \{h \in \{1, .., n\} : T_h \in D'_c\}$. We have $supp_{C_c}(J) > 0 \iff \exists_{g \in \{1,..,n\}} J \subseteq T_g$.

Since $P$ is a proper partition of $\mathcal{I}$, we have $\forall_{j \in M} \exists_{k_j \in \{1,..,r\}} I_j \in p_{k_j} \wedge |p_{k_j} \cap T_g| = 1$ (2); let $K = \{k_j\}_{j \in M}$. Now, we have $J = \{I_j\}_{j \in M} = \bigcup_{j \in M} \{I_j\}$. According to (2), we have $\bigcup_{j \in M} \{I_j\} = \bigcup_{j \in M} T_g \cap p_{k_j} = \bigcup_{k \in K} T_g \cap p_k = \bigcup_{k \in K} f_k^{-1}(a_k(u_g)) = condPatt(u_g, \{a_k\}_{k \in K})$.

In addition, we have $supp_{C'_c}(J) = 0 \iff \forall_{h \in H} J \not\subseteq T_h \iff \forall_{h \in H} \exists_{j \in M} I_j \notin T_h \iff \forall_{h \in H} \exists_{j \in M} T_g \cap p_{k_j} \not\subseteq T_h \implies \forall_{h \in H} \exists_{k \in K} T_g \cap p_k \not\subseteq T_h \cap p_k \iff \forall_{h \in H} \exists_{k \in K} f_k^{-1}(a_k(u_g)) \neq f_k^{-1}(a_k(u_h)) \iff \forall_{h \in H} \exists_{k \in K} a_k(u_g) \neq a_k(u_h) \iff \{v \in \mathcal{U} : \forall_{k \in K} a_k(u_g) = a_k(v)\} \subseteq U_c \iff R(u_g) \subseteq U_c \iff u_g \in R_*(U_c) \iff u_g \in POS(R, d) \cap U_c$.

As a continuation, the following theorem states that if this attribute set is a local reduct, it generates a minimal JEP. We omit the proof due to space limitations. It uses Theorem 1 and remains analogical to Theorem 1 from [6].

**Theorem 2.** $\{condPatt(u, R) : u \in POS(\mathcal{C}, d) \cap U_{d(u)} \wedge R = \{a_k\}_{k \in K \subseteq \{1,..,r\}} \in REDLOC(u, d) \wedge \forall_{k \in K} a_k(u) \neq f_k(\emptyset)\} = \mathcal{L}_{d(u)}$

*Example 2.* The number of additionally generated EPs depends on a chosen partition. For the $CDT$ from Tab. 1, only one additional EP $condPatt(u_4, a_3) = \emptyset$ has been generated. The worst partition is a trivial one-to-one mapping leading to a binary table. Let us consider a condensed table $CDT'$ looking like $BTD$ in order to show analogy to finding JEPNs in $BTD$. For example, the pattern for

the object $u_3$ and the attribute set $\{f, g\}$ in the $BDT$ can be interpreted as $\overline{f}g \in JEPN(C_1, C_0)$. When we ignore negations, we have $condPatt(u_3, \{f, g\}) = g$ that is not a JEP in $CDT'$ and has too be pruned. As we can see the spaces (Tab. 2) contain additional JEPNs for $BTD$, which become EPs with a not infinite growth rate in $CDT'$ after ignoring negated items by a $condPatt$ notation.

**Table 2.** JEP an JEPN spaces for $DTS$

| Space | Border |
|---|---|
| $JEP(D_1, D_0)$ | $< \{eg, d, cg, a\}, \{adh, afg, ceg\} >$ |
| $JEP(D_0, D_1)$ | $< \{eh, b\}, \{ce, beh, bfg\} >$ |
| $JEPN(D_1, D_0)$ | $< \{\overline{f}g, \overline{e}h, \overline{eg}, \overline{ef}, eg, d, cg, \overline{b}h, \overline{b}g, \overline{b}f, \overline{be}, \overline{bc}, a\},$ $\{a\overline{b}\overline{c}d\overline{e}\overline{f}\overline{g}h, a\overline{b}\overline{c}d\overline{e}fg\overline{h}, a\overline{b}\overline{c}de\overline{g}\overline{h}\} >$ |
| $JEPN(D_0, D_1)$ | $< \{\overline{g}\overline{h}, eh, e\overline{g}, \overline{d}\overline{h}, \overline{dg}, \overline{c}e, \overline{cdf}, c\overline{g}, b, \overline{a}h, \overline{ag}, \overline{af}, \overline{ae}, \overline{ac}\},$ $\{\overline{a}\overline{b}\overline{c}d\overline{e}\overline{f}\overline{g}\overline{h}, \overline{a}b\overline{c}\overline{d}e\overline{f}\overline{g}h, \overline{a}b\overline{c}\overline{d}\overline{e}fg\overline{h}\} >$ |

***Problem Reduction to Graph Coloring.*** It is not obvious which partition is optimal for a given dataset and a reduct finding algorithm. Since dimensionality is usually the most significant issue, we have chosen a criterium stating that higher aggregations lead to better performance. This optimization problem can be easily reduced to graph coloring. Let us consider an undirected graph $(V, E)$ such that $V = \{v_1, .., v_m\}$, $\forall_{x,y \in \{1,..,m\}}(v_x, v_y) \in E \iff \forall_{T \in \mathcal{D}} i_x \notin T \vee i_y \notin T$. Every coloring $\{w_1, .., w_r\}$ of this graph defines a proper partition of items $\{p_1, .., p_r\}$ such that $\forall_{j \in \{1,..,m\}} \forall_{k \in \{1,..,r\}} v_j \in w_k \iff i_j \in p_k$. A respective graph for the data in Tab. 1 is presented in Fig. 1.

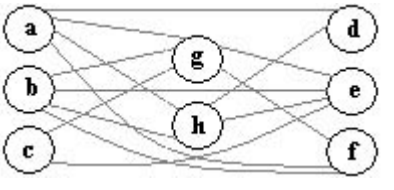



**Fig. 1.** The graph for the decision transaction system $DTS$ from Tab. 1. Vertices connected by an edge refer to attributes that cannot be aggregated into one block. The partitions $\{\{a, b, c\}, \{d, e, f\}, \{g, h\}\}$ and $\{\{a, e\}, \{c, d, f\}, \{g, h\}\}$ are proper.

***Algorithm Details.*** Our task is to find JEP spaces for all the classes of a decision transaction system $DTS$. The first stage of our method is to find an aggregation of items with a possibly small number of blocks. For this purpose, we construct an aforementioned graph and solve a graph coloring problem. A suboptimal solution is obtained by widely known heuristics, like LF, SLR, RLF, SR ([8]) and used to build a condensed decision table $CDT$ for $DTS$.

The most time-consuming step is discovery of minimal patterns for $CDT = \{\mathcal{U}, \mathcal{C}, d\}$. It involves finding the set $REDLOC(u,d)$ for each object $u \in POS(\mathcal{U},d)$.

**Table 3.** Experimental results

| Dataset | Obj | Items | Attr | JEP | Other EPs | Part. Time | RS1 Time | RS2 Time | JEP-Producer Time |
|---|---|---|---|---|---|---|---|---|---|
| balance | 625 | 20 | 4 | 303 | 0 | 0 | 125 | 112 | 691 |
| car | 1728 | 21 | 6 | 246 | 0 | 0 | 906 | 922 | 4628 |
| cmc | 1473 | 29 | 9 | 1943 | 0 | 3 | 1668 | 1737 | 1737 |
| dna | 500 | 80 | 20 | 444716 | 0 | 9 | 3997162 | 532440 | 464803 |
| geo | 402 | 78 | 10 | 7361 | 808 | 9 | 893 | 787 | 2103 |
| house | 435 | 48 | 16 | 6986 | 0 | 0 | 6722 | 11925 | 3359 |
| krkopt | 28056 | 43 | 6 | 21370 | 0 | 371 | 325525 | 328472 | 5474234 |
| lung | 32 | 220 | 56 | 203060 | 13860 | 25 | 8353406 | 2320684 | 1987296 |
| lymn | 148 | 59 | 18 | 6794 | 0 | 6 | 3940 | 3356 | 1375 |
| mushroom | 8124 | 117 | 23 | 3635 | 194 | 868 | 175196 | 112881 | 1271456 |
| nursery | 12960 | 27 | 8 | 638 | 0 | 15 | 102153 | 102865 | 523959 |
| tic-tac-toe | 958 | 27 | 9 | 2858 | 0 | 0 | 2853 | 3178 | 2659 |
| vehicle | 846 | 72 | 19 | 20180 | 6162 | 12 | 29515 | 88149 | 16281 |

Every local reduct $B \subseteq \mathcal{C}$ refers to the pattern $condPatt(u, B)$. We ignore EPs based on reducts with at least one attribute mapping to an empty itemset. The rest of patterns for the objects from a particular class constitutes the left bound of a respective JEP space. Reduct finding algorithm can be chosen arbitrarily. A detailed discussion of a similar phase for relational data is presented in [6].

## 4 Experimental Results

An experiment has been performed in order to check the usefulness of the aggregation step and the overall efficiency of the presented method towards JEP-Producer. Two different methods for reduct computation are taken into account: finding prime implicants of a monotonous boolean function ([4], RS1) and space traversing with an attribute set dependence pruning ([9], RS2). Because of space limitations, we present results (Tab. 3) only for one coloring heuristics, LF. Datasets originate from the UCI repository ([10], transformed to transactional form). The results are averaged over several repetitions. To avoid bias of a particular attribute order, a permutation has been picked up randomly each time.

In most of the cases, the items have been aggregated optimally, greatly reducing the search space. Only for 4 datasets, other EPs have been generated. At the same time, the coloring stage have not influenced the total computation time significantly. Rough set approach outperformed JEP-Producer for 3 larger sets: krkopt, mushroom, nursery, remained comparable for dna, lung, and more efficient for the majority of smaller sets.

## 5 Conclusions

In this paper, we have proposed a novel rough set approach for finding jumping emerging patterns (JEPs) in a decision transaction database. A database can be

represented by a respective decision table and minimal patterns can be induced by means of local reduct methods. Unfortunately, this approach generates a significant overhead of additional EPs and remains inefficient. Therefore, we have introduced a preprocessing phase, in which a database is transformed to a relevant condensed decision table by aggregating its items into attributes.

It has been also demonstrated that finding of an optimal aggregation can be reduced to graph coloring. Since any suboptimal solution still leads to a complete set of minimal JEPs, we employ heuristics for graph coloring, like LF, SRL etc.

Tests performed on originally relational and transactional datasets show that the proposed rough set method is an efficient alternative for JEP-Producer, which operates on borders and is a widely recommended algorithm. Aggregation phase introduces an almost unnoticeable cost and allow us to reduce the number of additionally generated EPs and, thus, the total computation time.

We hope that our approach will be a valuable extension for other rough set algorithms applied to transactional data.

## References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB '94. (1994) 487–499
2. Dong, G., Li, J.: Mining border descriptions of emerging patterns from dataset pairs. Knowl. Inf. Syst. **8** (2005) 178–202
3. Li, W., Han, J., Pei, J.: Cmar: Accurate and efficient classification based on multiple class-association rules. In: ICDM '01. (2001) 369–376
4. Bazan, J., Nguyen, H.S., Nguyen, S.H., Synak, P., Wroblewski, J.: Rough set algorithms in classification problem. Rough set methods and applications: new developments in knowl. disc. in inf. syst. (2000) 49–88
5. Slezak, D.: Association reducts: A framework for mining multi-attribute dependencies. (Volume 3488 of LNCS.) 354–363
6. Terlecki, P., Walczak, K.: Local reducts and jumping emerging patterns in relational databases. (Volume 4259 of LNCS.) 268–276
7. Wroblewski, J.: Covering with reducts - a fast algorithm for rule generation. In: Rough Sets and Current Trends in Computing. (1998) 402–407
8. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. 2nd edn. The MIT Press, Cambridge, MA, USA (2001)
9. Terlecki, P., Walczak, K.: Attribute set dependence in apriori-like reduct computation. In: LNCS. (Volume 4062.) 268–276
10. D.J. Newman, S. Hettich, C.B., Merz, C.: UCI repository of machine learning databases (1998)

# Visualization of Rough Set Decision Rules for Medical Diagnosis Systems

Grzegorz Ilczuk[1] and Alicja Wakulicz-Deja[2]

[1] Siemens AG Medical Solutions,
Allee am Roethelheimpark 2, 91052 Erlangen, Germany,
Grzegorz.Ilczuk@ilczuk.com
[2] Institute of Informatics, University of Silesia,
Bedzinska 39, 41-200 Sosnowiec, Poland,
wakulicz@us.edu.pl

**Abstract.** An ability of Pawlak's Rough Sets Theory to handle imprecision and uncertainty without any need of preliminary or additional information about analyzed data makes this theory very interesting for analyzing medical data. Using Rough Sets Theory knowledge extracted from raw data may be stored in form of decision rules. But increasing number and complexity of decision rules make their analysis and validation by domain experts difficult. In this paper we focus on this problem and propose an approach to visualize decision rules in form of decision trees. Afterwards domain experts validate transformed decision trees and compare the results with general guidelines proposed by the American College of Cardiology Foundation and the American Heart Association.

## 1 Introduction

Visual techniques have a special place in data exploration process because of the phenomenal abilities of the human visual system to detect structures in images. This product of aeons of evolution makes learning from visually presented information faster and is used by visual methods to present abstract data graphically. This approach is quite opposite to formal methods of model building and testing but it is ideal for searching through data to find unexpected relationships. Therefore our research is focused on a vertical solution for analyzing medical data, which joins advantages of several data mining techniques in one system, which consist of following parts:

- Import subsystem-responsible for importing data from medical information systems into storage database
- Data recognition-this subsystem transforms raw data to a form suited for further data processing. Additionally noise and redundant data are removed based on a statistical analysis. Partly results were described in [1]
- Feature subset selection - responsible for selecting an optimal set of attributes for a generation of decision rules.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 371–378, 2007.

- Rule induction subsystem - uses based on Rough Sets [2,3] MLEM2 algorithm proposed by Grzymala-Busse in [4] for generating decision rules. Early research on this area was described in [5,6].
- Visualization of the collected knowledge in a form easily understandable by humans. Partly results based on decision trees were published in [7]

In this paper we extend our initial research on the data visualization and present an implementation of AQDT-2 method proposed by Michalski in [8] for transforming decision rules into decision trees models. Visualization of tree models is realized by a renderer class, which is also discussed in this paper. Generated decision trees are also validated by domain experts and compared with general guidelines proposed by the American College of Cardiology Foundation and the American Heart Association [9].

## 2 Decision Trees

### 2.1 Transformation of Decision Rules:AQDT-2

Decision trees are an effective tool for describing a decision process but they also show some limitations if their structure must be adapted for new requirements. This limitation is attributable to the fact, that decision structure (tree) stores information in form of procedural representation, which imposes an evaluation order of tests. In contrary declarative representation of knowledge such as decision rules can be evaluated in any order, so that it is possible to generate a large number of logically equivalent decision trees which differ in test ordering. This way decision rules may be easily modified and adapted for specified requirements and at the end this declarative knowledge representation may be transformed into procedural one (decision trees). In this paper we describe our transformation results achieved with an extended version of AQDT-2 method based on the idea presented by Michalski in [8]. The core of this method is a selection of the most suitable attribute from decision rules for further processing based on four criteria:

- **Disjointness** - specifies an effectiveness of an attribute in discriminating among decision rules for different decision classes.
- **Importance** - this measure gives more "points" to an attribute which appears in strong decision rules.
- **Value distribution** - an attribute which has a smaller number of legal values scores better.
- **Dominance** - this criteria prefers attributes which appears in a large number of rules.

These criteria build the attribute ranking system (LEF) [10] which firstly selects the best attribute based on value of Disjointness. If there are more then one attribute which have the same maximal value of Disjointness then the next criteria is checked in the following order: Importance, Value distribution and Dominance. The algorithm of transforming decision rules into decision tree consist of following steps:

1. From the given decision rules extract attributes and select the best attribute using LEF.
2. Create a node of the tree (first node is the root) and assign to its branches all legal values of the attribute.
3. Associate each branch with rules from the parent node which satisfy condition for the branch.
4. Remove satisfied condition from rules belonging to the branch.
5. If all rules in the branch belong to the same class create a leaf node and assign to it that class.
6. If there are no more unprocessed attributes in rules assigned to the branch create a leaf and assign to it all left decision classes.
7. If all branches were processed stop otherwise repeat steps 1 to 6.

In our research we have extended the initial algorithm to support: a rule filtering, an interactive attribute selection and an advanced tree generalization and pruning.

### 2.2 Visualization of Decision Tree Model

For rendering a generated decision tree we implemented an algorithm which mostly follows the esthetics defined in [11,12] for keeping a displayed tree as tight as possible without compromising its readability. This implementation in contrary to Bloesch uses a non-recursive algorithm which avoids stack overflows during a processing of large tree models. Main steps of the render algorithm are following:

1. Let maximum_level be the deepest tree level (starting from 0 for a root node).
2. Create a one dimension table Yfree[maximum_level] for storing a next possible Y position of nodes at each level and initialize it with start values.
3. Starting from the root node find the deepest leaf of its first branch.
4. If the node is a leaf calculate its positions based on its level. Otherwise calculate its positions based on its level and positions of its child's nodes.
5. Actualize Yfree table for the node's level.
6. Go to node's parent and continue with the next unprocessed branch. If all branches are processed then repeat the Step 6.
7. Repeat steps 4 to 6. If all root branches are expanded the calculate coordinates for the root node and stop.

An example of tree rendering is presented at figure 1, where numbers displayed in tree nodes mean node's draw order starting from 1 and nodes having a dashed outline symbolize not yet processed nodes. Rendering of a tree model starts with some initialization (steps 1 and 2) and then the algorithm finds the deepest node which belongs to the first branch of the root node. This start node is marked at figure 1 with a number 1 and it belongs to level 1 (marked as 'L:1' column at figure 1). During the next step X and Y positions of the node are calculated. Node's X position depends on its level and is the same for all nodes belonging to the same level. Node's Y position depends on the entry in the Yfree table. For the first node it will be the initial value saved in the Yfree table during initialization. Each time X and Y positions for the processed node have been

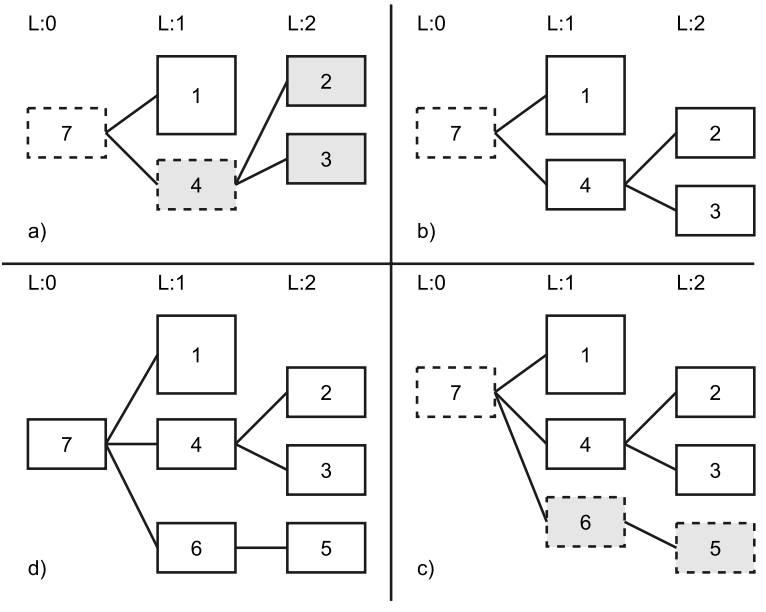



**Fig. 1.** An example of the tree rendering process: a) Subtree (nodes 2 and 3) will be moved b) Subtree (nodes 2 and 3) has been moved c) Adjusting node 6 and its child (node 5) d) End result of a tree rendering

assigned the Yfree table is actualized. This step ends a processing for the node and the algorithm repeats itself for the next unprocessed node. The selection of a next unprocessed node is done in following order: if the parent node of the currently processed node has other unprocessed branches then again the deepest node of the first unprocessed branch is selected for further processing. If all parent nodes branches have been processed then the algorithm tries to find a next unprocessed branch of a node one level higher. The calculation stops if all root's branches and the root node are processed. If during the processing at some level a calculated node's Y position is smaller then a possible Y position stored in the Yfree table for that level then its position and positions of its children will be adjusted as shown in cases a), b) and c) of figure 1. Case d) of the figure 1 shows the completely rendered tree.

The presented method was optimized to utilize a horizontal layout of a tree for saving the space needed for tree rendering. Additionally, as it can be seen at figure 2, it supports a variable height of tree nodes.

## 3 Material and Method

### 3.1 Dataset Preparation

Data used in our research was obtained from the Cardiology Department of Silesian Medical Academy in Katowice-the leading Electrocardiology Department in

Poland specializing in hospitalization of severe heart diseases (over 1200 pacemaker implantations yearly). Data of 4318 patients hospitalized in this Department between 2003 and 2005 were imported and transformed into 14 grouped attributes such as: Atrioventricular block, Cardiomyopathy, Chronic ischaemic heart disease, Sick Sinus Syndrome and Paroxysmal tachycardia. Additionally a decision attribute (**PM_DDD**, value range:[0,1]) was specified which represents a decision (0=no, 1=yes) about an implementation of a dual chamber DDD pace-maker. Decision rules were generated from a testing set ($\frac{2}{3}$ of the initial data) using our implementation of Rough Sets MLEM2 algorithm. The rules were then used as an input for AQDT-2 method.

## 4 Results

Figure 2 shows a generalized decision tree generated from decision rules achieved for the training set containing all 14 attributes. Following steps can be used to simplify a tree structure and thus allow its analysis:

- Limitation of tree depth: Only first four levels of the decision tree were calculated.
- Automatic selection of strong attributes: We used *filter* method [13] coped with $\chi^2$ to select the strongest attributes in tree generation phase. It would be also possible to use a set of user-selected attributes for tree generation.
- Generalization of tree structure: Each tree node was only expanded if the weakest class in the node reached a defined ratio. This ratio (set in the presented case at 20%) is calculated as a ratio between the class strength and the strength of the strongest class in a processed node. Strength of a class is summarized from decision rules belonging to the class and assigned to the currently processed node.

Presented at figure 2 decision tree shows, that the decision about implantation of a DDD pace-maker (Class=1.0) was taken if at least two indicators where diagnosed: **Sick Sinus Syndrome (SSS)** were present (values between 1 and 3 depending on disease advance) and **Atrial fibrillation and flutter (AFF)** had values: either 'Not diagnosed=value 0.0' or 'Paroxysmic=value 1.0'. It can be also seen, that alone the diagnosis about **Paroxysmal tachycardia (PTACH)** was not sufficient for decision making. These results validated by domain experts got a very positive opinion which is deeply rooted in both a common praxis in the Cardiology Department and a match of ACC/AHA/NSAPE guidelines [9].

From table 1 it can be seen, that a non-reduced data set with 14 attributes performed very well on the training dataset (above 82%) but on the test dataset it was able to classify correctly only 71% of new cases. Comparing the true positive rate (TP) for decision class PM_DDD=1 shows, that this rate with 68% is the lowest in comparison to the other sets of decision rules. Additionally a distribution of coverage ratio for each class and the lowest total number of recognized cases from the test dataset led us to the conclusion, that these decision rules demonstrate an overfitting effect. The rules show a good prediction accuracy

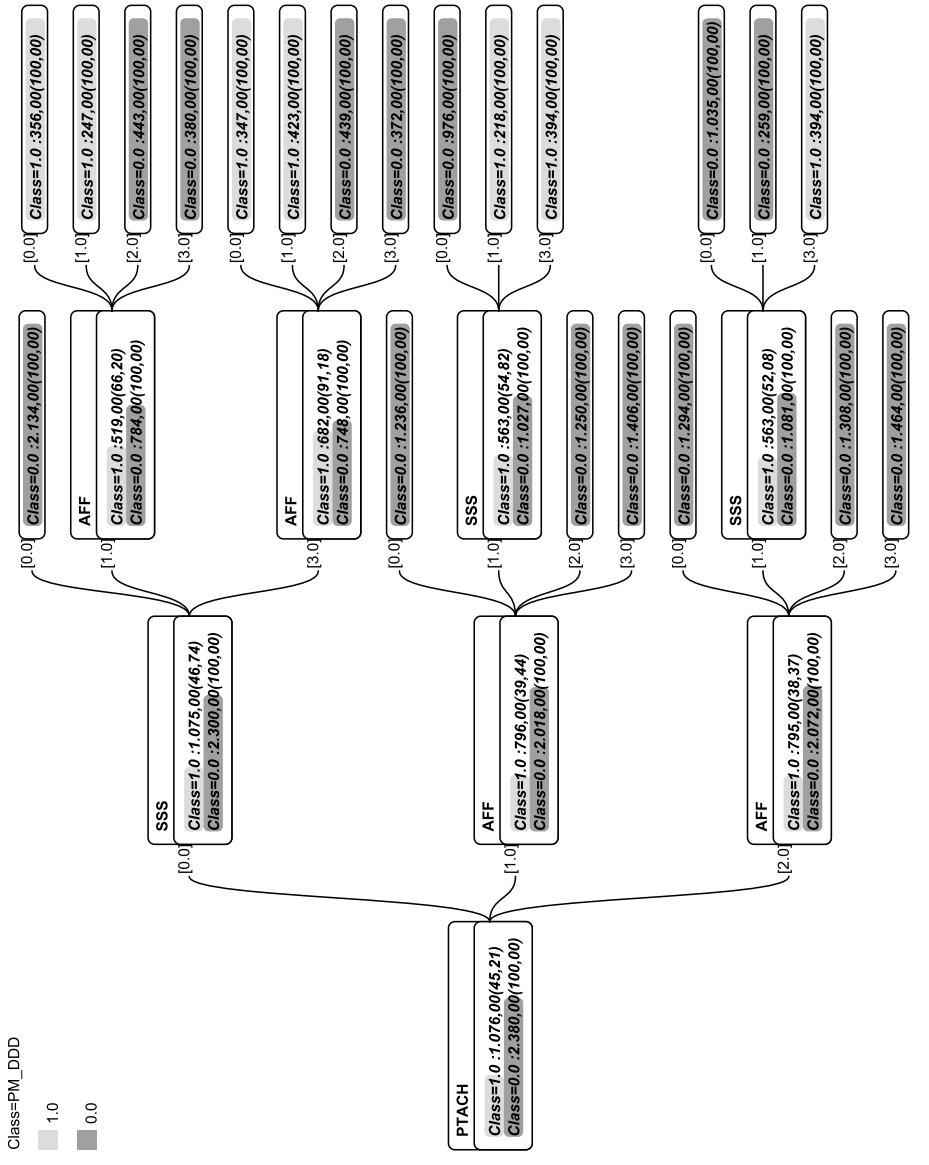



**Fig. 2.** Decision tree generated for decision rules with all 14 attributes in full display mode. Format of a node description: ⟨decision attribute⟩=⟨value of the decision attribute⟩ : ⟨strength of decision rules in node⟩ (⟨percentage of the class strength in compare to strength of all classes in a node⟩).

on the training dataset but are less able to recognize new cases. Classification results achieved for a reduced set of attributes showed a better classification accuracy on the test dataset (75%) but the accuracy on the training set was with 76% lower as for the non-reduced dataset. It is worth of mentioning that,

**Table 1.** Classification results for PM_DDD decision attribute

| Decision Rules | Correct Classif. | C:1 TP | C:0 TP | C:1 Coverage | C:0 Coverage | Correctly classified | Rules |
|---|---|---|---|---|---|---|---|
| **Train set 66%, Test set 33% - C:1 619 objects, C:0 839 objects** | | | | | | | |
| 14 attributes | 71.33 | 68.44 | 73.06 | 88.05 | 108.82 | 1040 | 238 |
| 4 attributes | 75.38 | 73.13 | 76.79 | 90.79 | 106.79 | 1099 | 41 |
| 14 attributes (from tree) | 75.79 | 73.33 | 77.36 | 92.08 | 105.84 | 1105 | 22 |
| **Train set 66% used also as Test set - C:1 1089 objects, C:0 1740 objects** | | | | | | | |
| 14 attributes | 82.15 | 75.70 | 86.47 | 104.32 | 97.30 | 2324 | 238 |
| 4 attributes | 76.53 | 68.02 | 82.61 | 108.26 | 94.83 | 2165 | 41 |
| 14 attributes (from tree) | 76.14 | 67.69 | 82.10 | 107.44 | 95.34 | 2154 | 22 |

these results were achieved with a noticeable reduced number of decision rules. The best results in terms of classification accuracy and the smallest number of rules where achieved for decision rules transformed back from the decision tree shown at figure 2. Prediction accuracy of these rules is comparable with results achieved for a reduced set of attributes what is attributable to the same subset of strong attributes used in both cases. A very small number of 22 decision rules is an effect of a performed indirectly generalization during a transformation decision rules→decision tree. This step simplifies an input concept, as mentioned by Michalski in [10].

## 5 Conclusions

In this paper we presented the AQDT-2 method for transformation of decision rules into decision trees and mentioned our extensions of this method. We also presented a solution to render the generated tree and thus allow its verification. In our experiments we generated an tree model from decision rules and then we compared prediction accuracy of different sets of decision rules. Results of these experiments shown, that AQDT-2 method has several advantages in medical domain over methods natively generating decision trees:

1. Knowledge stored as decision rules can be easily transformed to a graphical format, which is easily understandably and verifiable by domain experts.
2. An order of attributes in a decision tree (tree structure) can be easily changed according to user preferences. This allows partly decision making in situations where getting a value of an attribute is impossible, dangerous or costly.
3. Transformation from calculated decision tree back to decision rules simplify an original concept without a significant loss of a recognition accuracy. Similar results were also achieved by Bohanec and Bratko in [14].
4. A possibility to interactively change a structure of a generated tree allows 'what-if' analysis and can be easily used to reveal new patterns from processed data.

Presented methods join data mining techniques implemented in our system into a complete solution where both analytical methods and human senses are used for knowledge exploration.

## References


1. Ilczuk, G., Wakulicz-Deja, A.: Attribute selection and rule generation techniques for medical diagnosis systems. RSFDGrC 2005,LNCS **3642** (2005) 352–361
2. Pawlak, Z.: Knowledge and uncertainty: A rough set approach. In: Workshop on Incompleteness and Uncertainty in Information Systems. (1993) 34–42
3. Pawlak, Z., Grzymala-Busse, J.W., Slowinski, R., Ziarko, W.: Rough sets. Commun. ACM **38** (1995) 88–95
4. Grzymala-Busse, J.W.: Mlem2-discretization during rule induction. Proc. of the Int'l IIS (2003) 499–508
5. Ilczuk, G., Wakulicz-Deja, A.: Rough sets approach to medical diagnosis system. AWIC 2005,LNCS **3528** (2005) 204–210
6. Ilczuk, G., Mlynarski, R., Wakulicz-Deja, A., Drzewiecka, A., Kargul, W.: Rough sets techniques for medical diagnosis systems. Computers in Cardiology 2005 **32** (2005) 837–840
7. Mlynarski, R., Ilczuk, G., Wakulicz-Deja, A., Kargul, W.: Automated decision support and guideline verification in clinical practice. Computers in Cardiology 2005 **32** (2005) 375–378
8. Michalski, R.S., Imam, I.F.: Learning problem-oriented decision structures from decision rule: The AQDT-2 system. In: Int'l Symposium on Methodologies for Intelligent Systems. (1994) 416–426
9. Gregoratos, e.A.: Acc/aha/naspe 2002 guideline update for implantation of cardiac pacemakers and antiarrhythmia devices. Technical report, American College of Cardiology Foundation and the American Heart Association, Inc. (2002)
10. Michalski, R.S.: Aqval/1–computer implementation of a variable-valued logic system vl1 and examples of its application to pattern recognition. In: Proc. First Int'l Joint Conf. Pattern Recognition. (1973) 3–17
11. Bloesch, A.: Aestetic layout of generalized trees. Software-Practice and Experience **23** (1993) 817–827
12. Vaucher, J.G.: Pretty-printing of trees. Software-Practice and Experience **10** (1980) 553–561
13. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: ICML. (1994) 121–129
14. Bohanec, M., Bratko, I.: Trading accuracy for simplicity in decision trees. Machine Learning Journal **15** (1994) 223–250

# Attribute Generalization and Fuzziness in Data Mining Contexts

Shusaku Tsumoto

Department of Medical Informatics,
Shimane University, School of Medicine
89-1 Enya-cho, Izumo 693-8501 Japan
tsumoto@computer.org

**Abstract.** This paper shows problems with combination of rule induction and attribute-oriented generalization, where if the given hierarchy includes inconsistencies, then application of hierarchical knowledge generates inconsistent rules. Then, we introduce two approaches to solve this problem, one process of which suggests that combination of rule induction and attribute-oriented generalization can be used to validate concept hiearchy. Interestingly, fuzzy linguistic variables play an important role in solving these problems.

## 1 Introduction

Conventional studies on rule discovery based on rough set methods[1,2,3] mainly focus on acquisition of rules, the targets of which have mutually exclusive supporting sets. Supporting sets of target concepts form a partition of the universe, and each method search for sets which covers this partition. Especially, Pawlak's rough set theory shows the family of sets can form an approximation of the partition of the universe. These ideas can easily extend into probabilistic contexts, such as shown in Ziarko's variable precision rough set model[4]. However, mutual exclusiveness of the target does not always hold in real-world databases, where conventional probabilistic approaches cannot be applied.

In this paper, first, we show that these phenomena are easily found in data mining contexts: when we apply attribute-oriented generalization to attributes in databases, generalized attributes will have fuzziness for classification, which causes rule induction methods to generate inconsistent rules. Then, we introduce two solutions. The first one is to introduce aggregation operators to recover mathematical consistency. The other one is to introduce Zadeh's linguistic variables, which describes one way to represent an interaction between lower-level components in an upper level components and which gives a simple solution to deal with the inconsistencies. Finally, we briefly discuss the mathematical generalization of this solution in which context-free fuzzy sets is a key idea. In this inconsistent problem, we have to take care about the conflicts between each attributes, which can be viewed as a problem with multiple membership functions.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 379–386, 2007.

## 2 Attribute-Oriented Generalization and Fuzziness

### 2.1 Probabilistic Rules

**Accuracy and Coverage.** In the subsequent sections, we adopt the following notations, which is introduced in [5].

Let $U$ denote a nonempty, finite set called the universe and A denote a nonempty, finite set of attributes, i.e., $a : U \rightarrow V_a$ for $a \in A$, where $V_a$ is called the domain of $a$, respectively.Then, a decision table is defined as an information system, $A = (U, A \cup \{d\})$.

The atomic formulas over $B \subseteq A \cup \{d\}$ and $V$ are expressions of the form $[a = v]$, called descriptors over B, where $a \in B$ and $v \in V_a$. The set $F(B, V)$ of formulas over B is the least set containing all atomic formulas over $B$ and closed with respect to disjunction, conjunction and negation.

For each $f \in F(B, V)$, $f_A$ denote the meaning of $f$ in $A$, i.e., the set of all objects in U with property $f$, defined inductively as follows.

1. If $f$ is of the form $[a = v]$ then, $f_A = \{s \in U | a(s) = v\}$
2. $(f \wedge g)_A = f_A \cap g_A$; $(f \vee g)_A = f_A \vee g_A$; $(\neg f)_A = U - f_a$

By the use of this framework, classification accuracy and coverage, or true positive rate is defined as follows.

**Definition 1**
*Let $R$ and $D$ denote a formula in $F(B, V)$ and a set of objects which belong to a decision $d$. Classification accuracy and coverage(true positive rate) for $R \rightarrow d$ is defined as:*

$$\alpha_R(D) = \frac{|R_A \cap D|}{|R_A|} (= P(D|R)), \text{ and } \kappa_R(D) = \frac{|R_A \cap D|}{|D|} (= P(R|D)),$$

*where $|A|$ denotes the cardinality of a set $A$, $\alpha_R(D)$ denotes a classification accuracy of $R$ as to classification of $D$, and $\kappa_R(D)$ denotes a coverage, or a true positive rate of $R$ to $D$, respectively.*

**Definition of Rules**
By the use of accuracy and coverage, a probabilistic rule is defined as:

$$R \stackrel{\alpha,\kappa}{\rightarrow} d \quad s.t. \quad R = \wedge_j \vee_k [a_j = v_k], \ \alpha_R(D) \geq \delta_\alpha, \ \kappa_R(D) \geq \delta_\kappa.$$

This rule is a kind of probabilistic proposition with two statistical measures, which is an extension of Ziarko's variable precision model(VPRS) [4].

It is also notable that both a positive rule and a negative rule are defined as special cases of this rule, as shown in the next subsections.

### 2.2 Attribute-Oriented Generalization

Rule induction methods regard a database as a decision table[1] and induce rules, which can be viewed as reduced decision tables. However, those rules extracted from tables do not include information about attributes and they are too simple. In practical situation, domain knowledge of attributes is very important to gain the comprehensiblity of induced knowledge, which is one of the reasons why databases are implemented as relational-databases[6]. Thus, reinterpretation of induced rules by using information about attributes is needed to acquire comprehensive rules. For example, terolism, cornea, antimongoloid slanting of palpebral fissures, iris defects and long eyelashes are symptoms around eyes. Thus, those symptoms can be gathered into a category "eye symptoms" when the location of symptoms should be focused on. symptoms should be focused on. The relations among those attributes are hierarchical as shown in Figure 1. This process, grouping of attributes, is called attribute-oriented generalization[6].

Attribute-oriented generalization can be viewed as transformation of variables in the context of rule induction. For example, an attribute "iris defects" should be transformed into an attribute "eye symptoms=yes".It is notable that the transformation of attributes in rules correspond to that of a database because a set of rules is equivalent to a reduced decision table. In this case, the case when eyes are normal is defined as "eye symptoms=no". Thus, the transformation rule for iris defects is defined as:

$$[\textit{iris-defects} = yes] \rightarrow [\textit{eye-symptoms} = yes] \tag{1}$$

In general, when $[A_k = V_l]$ is a upper-level concept of $[a_i = v_j]$, a transforming rule is defined as:

$$[a_i = v_j] \rightarrow [A_k = V_l],$$

and the supporting set of $[A_k = V_l]$ is:

$$[A_i = V_l]_A = \bigcup_{i,j} [a_i = v_j]_a,$$

where $A$ and $a$ is a set of attributes for upper-level and lower level concepts, respectively.

### 2.3 Examples

Let us illustrate how fuzzy contexts is observed when attribute-oriented generalization is applied by using a small table (Table 1). Then, it is easy to see that a rule of "Aarskog",

$$[\textit{iris-defects} = yes] \rightarrow Aarskog \quad \alpha = 1.0, \kappa = 1.0$$

is obtained from Table 1.

When we apply transforming rules shown in Figure 1 to the dataset of Table 1, the table is transformed into Table 2. Then, by using transformation rule 1, the above rule is transformed into:

$$[\textit{eye-symptoms} = yes] \rightarrow Aarskog.$$

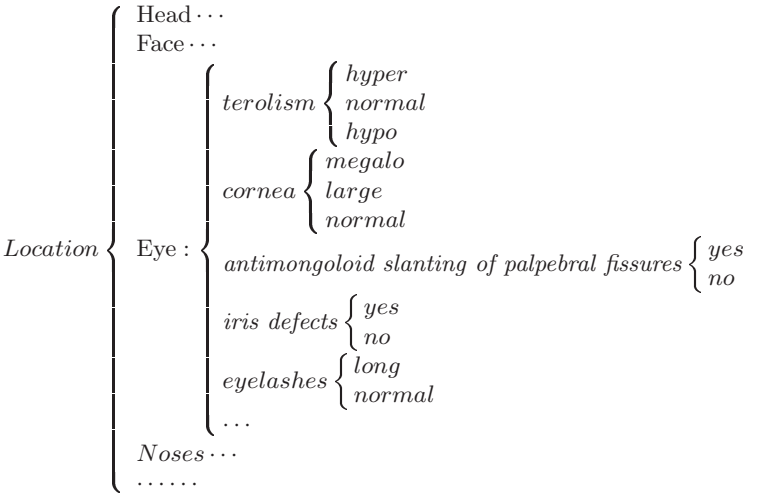



**Fig. 1.** An Example of Attribute Hierarchy

It is notable that mutual exclusiveness of attributes has been lost by transformation. Since five attributes (telorism, cornea, slanting, iris-defects and eyelashes) are generalized into *eye-symptoms*, the candidates for accuracy and coverage will be (2/4, 2/3), (2/4, 3/3), (3/4, 3/3), (3/4, 3/3), (3/3, 3/3) and (3/4, 3/3), respectively. Then, we have to select which value is suitable for the context of this analysis.

In [7], Tsumoto selected the minimum value in medical context: accuracy is equal to 2/4 and coverage is equal to 2/3.

Thus, the rewritten rule becomes the following probabilistic rule:

$$[\textit{eye-symptoms} = \textit{yes}] \rightarrow \textit{Aarskog},$$
$$\alpha = 3/4 = 0.75, \kappa = 2/3 = 0.67.$$

**Table 1.** A Small Database on Congenital Disorders

| U | round | telorism | cornea | slanting | iris-defects | eyelashes | class |
|---|---|---|---|---|---|---|---|
| 1 | no | normal | megalo | yes | yes | long | Aarskog |
| 2 | yes | hyper | megalo | yes | yes | long | Aarskog |
| 3 | yes | hypo | normal | no | no | normal | Down |
| 4 | yes | hyper | normal | no | no | normal | Down |
| 5 | yes | hyper | large | yes | yes | long | Aarskog |
| 6 | no | hyper | megalo | yes | no | long | Cat-cry |

DEFINITIONS: round: round face, slanting: antimongoloid slanting of palpebral fissures, Aarskog: Aarskog Syndrome, Down: Down Syndrome, Cat-cry: Cat Cry Syndrome.

**Table 2.** A Small Database on Congenital Disorders (Transformed)

| U | eye | eye | eye | eye | eye | eye | class |
|---|---|---|---|---|---|---|---|
| 1 | no | no | yes | yes | yes | yes | Aarskog |
| 2 | yes | yes | yes | yes | yes | yes | Aarskog |
| 3 | yes | no | no | no | no | no | Down |
| 4 | yes | yes | no | no | no | no | Down |
| 5 | yes | yes | yes | yes | yes | yes | Aarskog |
| 6 | no | yes | yes | yes | no | yes | Cat-cry |

DEFINITIONS: eye: eye-symptoms

This examples show that the loss of mutual exclusiveness is directly connected to the emergence of fuziness in a dataset. It it notable that the rule used for transformation is a deterministic one. When this kind of transformation is applied, whether applied rule is deterministic or not, fuzziness will be observed. However, no researchers has pointed out this problem with combination of rule induction and transformation.

It is also notable that the conflicts between attributes with respect to accuracy and coverage corresponds to the vector representation of membership functions shown in Lin's context-free fuzzy sets[8].

### 2.4 What Is a Problem ?

The illustrative example in the last subsection shows that simple combination of rule induction and attribute-oriented generalization easily generates many inconsistent rules. One of the most important features of this problem is that simple application of transformation violates mathematical conditions.

Attribute-value pairs can be viewed as a mapping in a mathematical context, as shown in Section 2. For example, in the case of an attribute "*round*", a set of values in "*round*", $\{yes, no\}$ is equivalent to a domain of "round". Then, since the value of *round* for the first example in a dataset, denoted by "1" is equal to 1, $round(1)$ is equal to *no*. Thus, an attribute is a mapping from examples to values. In a reverse way, a set of examples is related to attribute-value pairs:

$$round^{-1}(no) = \{1, 6\}.$$

In the same way, the following relation is obtained:

$$eyeslashes^{-1}(normal) = \{3, 4\}.$$

However, simple transformation will violate this condition on mapping because transformation rules will change different attributes into the same name of generalized attributes. For example, if the following four transformation rules are applied:

$$round \rightarrow \text{eye-symptoms},$$
$$eyeslashes \rightarrow \text{eye-symptoms},$$
$$normal \rightarrow no, \qquad long \rightarrow yes,$$

then the following relations are obtained:

$$\text{eye-symptoms}^{-1}(no) = \{1, 6\},$$
$$\text{eye-symptoms}^{-1}(no) = \{3, 4\},$$

which leads to contradiction. Thus, transformed attribute-value pairs are not mapping because of one to many correspondence.

In this way, violation is observed as generation of logically inconsistent rules, which is equivalent to mathematical inconsistencies.

## 3 Solutions

### 3.1 Join Operators

In Subsection 2.3, since five attributes (telorism, cornea, slanting, iris-defects and eyelashes) are generalized into *eye-symptoms*, the candidates for accuracy and coverage will be (2/4, 2/3), (2/4, 3/3), (3/4, 3/3), (3/4,3/3), (3/3, 3/3), and (3/4, 3/3), respectively. Then, we show one approach reported in [7]. the minimum value is selected: accuracy is equal to 2/4 and coverage is equal to 2/3. This selection of minimum value is a kind of *aggregation*, or *join* operator. In join operators, conflict values will be integrated into one values, which means that one to many correspondence is again transformed into one to one correspondence, which will recover consistencies.

Another example of join operators is "average". In the above example, the average of accuracy is 0.71, so if the average operator is selected for aggregation, then the accuracy of the rule is equal to 0.71. This solution can be generalized into context-free fuzzy sets introduced by Lin[8], which is shown in Section 4.

### 3.2 Zadeh's Linguistic Variables

**Concept Hierarchy and Information.** Another solution is to observe this problem from the viewpoint of information. After the application of transformation, it is clear that some information is lost. In other words, transformation rules from concept hierarchy are kinds of projection and usually projection loses substantial amounts of information. Intuitively, over-projection is observed as fuzziness.

For example, let me consider the following three transformation rules:

$$[Round = yes] \rightarrow [\text{Eye-symptoms} = yes],$$
$$[\text{Iris-Defects} = yes] \rightarrow [\text{Eye-symptoms} = yes],$$
$$[Telorism = hyper] \rightarrow [\text{Eye-symptoms} = yes]$$

One of the most important questions is whether eyes only contribute to these symptoms.

Thus, one way to solve this problem is to recover information on the hierarchical structure for each symptoms. For example, let us summarize the components of each symptom and corresponding accuracy into Table 3.

**Table 3.** Components of Symptoms

| Symptoms | Components | Accuracy |
|---|---|---|
| $[Round = yes]$ | : [Eye, Nose, Frontal] | $\alpha = 1/2$ |
| $[Iris - Defects = yes]$ | : [Substructure of Eye] | $\alpha = 3/3$ |
| $[Telorism = hyper]$ | : [Eye, Nose, Frontal] | $\alpha = 2/3$ |

It is notable that even if components of symptoms are the same, the values of accuracy are not equal to each other. These phenomena suggest that the degrees of contribution of components are different in those symptoms. In the above examples, the degrees of contribution of Eye in $[Round = yes]$, $[Iris - Defects]$ and $[Telorism]$ are estimated as 1/2 (0.5), 3/3 (1.0) and 2/3(0.67), respectively.

**Linguistic Variables and Knowledge Representation.** Zadeh proposes linguistic variables to approximate human linguistic reasoning[9]. One of the main points in his discussion is that when human being reasons hierarchical structure, he/she implicitly estimates the degree of contribution of each components to the subject in an upper level.

In the case of a symptom $[Round = yes]$, this symptom should be described as the combination of Eye, Nose and Frontal part of face. From the value of accuracy in Aarskog syndromes, since the contribution of Eye in [Round=yes] is equal to 0.5, the linguistic variable of $[Round = yes]$ is represented as:

$$[Round = yes] = 0.5 * [Eye] + \theta * [Nose] + (0.5 - \theta) * [Frontal],$$

where 0.5 and $\theta$ are degrees of contribution of eyes and nose to this symptom, respectively. It is interesting to see that the real hierarchical structure is recovered by Zadehfs linguistic variable structure, which also suggests that linguistic variables captures one aspect of human reasoning about hierarchical structure. Especially, one important issue is that Zadeh's linguistic variables, although partially, represent the degree of interactions between sub-components in the same hierarchical level, which cannot be achieved by simple application of object-oriented approach.

Another important issue is that the degree of contribution, which can be viewed as a subset of a membership function, can be estimated from data. Estimation of membership function is one of the key issues in application of fuzzy reasoning, but it is a very difficult to extract such membership functions from data and usually they are given by domain experts[10].

## 4 Conclusions

This paper shows that combination of attribute-oriented generalization and rule induction methods generate inconsistent rules and proposes one solution to this problem. It is surprising that tranformation of attributes will easily generate this situation in data mining from relation databases: when we apply attribute-oriented generalization to attributes in databases, generalized attributes will

have fuzziness for classification. In this case, we have to take care about the conflicts between each attributes, which can be viewed as a problem with linguistic uncertainty or multiple membership functions. Finally, the author pointed out that these contexts should be analyzed by using two kinds of fuzzy techniques: one is introduction of aggregation operators, which can viewed as those on multiple membership functions. The other one is linguistic variables, which captures the degree of contribution.

## References

1. Pawlak, Z.: Rough Sets. Kluwer Academic Publishers, Dordrecht (1991)
2. Tsumoto, S.: Automated induction of medical expert system rules from clinical databases based on rough set theory. Information Sciences **112** (1998) 67–84
3. Tsumoto, S.: Knowledge discovery in clinical databases and evaluation of discovered knowledge in outpatient clinic. Information Sciences (2000) 125–137
4. Ziarko, W.: Variable precision rough set model. J. Comput. Syst. Sci. **46** (1993) 39–59
5. Skowron, A., Grzymala-Busse, J.: From rough set theory to evidence theory. In Yager, R., Fedrizzi, M., Kacprzyk, J., eds.: Advances in the Dempster-Shafer Theory of Evidence. John Wiley & Sons, New York (1994) 193–236
6. Cai, Y.G., Cercone, N., Han, J.: Attribute-oriented induction in relational databases. In Shapiro, G.P., Frawley, W.J., eds.: Knowledge Discovery in Databases. AAAI press, Palo Alto, CA (1991) 213–228
7. Tsumoto, S.: Knowledge discovery in medical databases based on rough sets and attribute-oriented generalization. In: Proceedings of IEEE-FUZZ98, Anchorage, IEEE Press (1998)
8. Lin, T.Y.: Context free fuzzy sets and information tables. In: Proceedings of the Sixth European Congress on Intelligent Techniques and Soft Computing (EUFIT'98), Aachen, Verlag Mainz (1998) 76–80
9. Zadeh, L.: The concept of linguistic variable and its application to approximate reasoning (part i). Information Sciences **8** (1975) 199–249
10. Pedrycz, W., Gomide, F.: An Introduction to Fuzzy Sets – Analysis and Design. MIT Press, Cambridge, MA (1996)

# A Hybrid Method for Forecasting Stock Market Trend Using Soft-Thresholding De-noise Model and SVM

Xueshen Sui, Qinghua Hu, Daren Yu, Zongxia Xie, and Zhongying Qi

Harbin Institute of Technology, Harbin 150001, China
Suixueshen@Gmail.com

**Abstract.** Stock market time series are inherently noisy. Although support vector machine has the noise-tolerant property, the noised data still affect the accuracy of classification. Compared with other studies only classify the movements of stock market into up-trend and down-trend which does not concern the noised data, this study uses wavelet soft-threshold de-noising model to classify the noised data into stochastic trend. In the experiment, we remove the stochastic trend data from the SSE Composite Index and get de-noised training data for SVM. Then we use the de-noised data to train SVM and to forecast the testing data. The hit ratio is 60.12%. Comparing with 54.25% hit ratio that is forecasted by noisy training data SVM, we enhance the forecasting performance.

**Keywords:** Soft-thresholding, De-noise, SVM, Stock market, Financial time series.

## 1 Introduction

Stock market trend forecasting gives information on the corresponding risk of the investments and it also will influence the trading behavior. Stock market time series are inherent noisy, non-stationary, and deterministically chaotic [1]. It has been shown that data extrapolated from stock markets are almost corrupted by noise and it appears that no useful information can be extracted from such data. Modeling such noisy and non-stationary time series is expected to be a challenging task [2].

In recent years, numerous studies have demonstrated that neural networks are a more effective method in describing the dynamics of non-stationary time series due to their unique non-parametric, non-assumable, noise-tolerant and adaptive properties [3]. However, neural networks still have several limitations.

SVM originates from Vapnik's statistical learning theory. Unlike most of the traditional methods which implement the empirical risk minimization principal, SVM implements the structural risk minimization principal which seeks to minimize an upper bound of the generalization error rather than minimize the training error [4]. Many applications of the SVM to forecast financial time series have been reported. Cao and Tay used the theory of SVM in regression to forecast the S&P 500 Daily Index in the Chicago Mercantile. They measured the degree of accuracy and the acceptability of certain forecasts by the estimates' deviations from the observed values [3]. Kim forecasted the direction of the change in daily Korea composite stock

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 387–394, 2007.

price index (KOSPI) with the theory of SVM in classification. The best prediction performance for the holdout data is 57.83% [5]. Tony Van Gestel designed the LS-SVM time series model in the evidence framework to predict the daily closing price return of the German DAX30 index (Deutscher Aktien Index) [6].Many of the previous studies have compared the performance of SVM with BP neural network, case-based reasoning (CBR) and so on. All of the results prove that the general performance for SVM is better than the traditional methods.

Many studies had selected optimum parameters of SVM when they would enhance the forecasting performance. This study proposes dealing with the noise of the stock market in order to enhance the forecasting performance of SVM. According to the wavelet de-noising model of soft-thresholding, we classify the stock market short-term trend into up-trend, stochastic trend and down-trend. We remove the stochastic trend data from the original Index data and take the rest data which belong to the up-trend and down-trend as the training data. Then we use the trained SVM to forecast the stock market trends.

## 2 Theoretical Backgrounds

### 2.1 Soft-Thresholding De-noise Model

Supposing $f(i)$ is the original signal, the polluted image signal is $s(i)$, and noise signal is $e(i)$. Then, the model of the noised imaged is

$$s(i) = f(i) + \sigma e(i) \tag{1}$$

where $\sigma$ denotes a noise level and $e(i)$ is a Gauss white noise

Figure 1 is the block diagram of signal de-noising with wavelet transformation. The three blocks in figure 1 represent the three basic steps of de-nosing respectively.

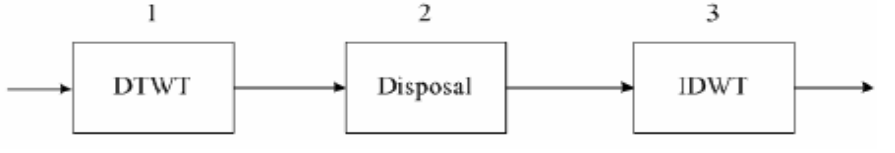



**Fig. 1.** The block diagram of wavelet de-noising

Wavelet decomposition is the first step: selecting wavelet and decomposition Level, and calculating the coefficients of the transformation from $s(i)$ to the layer $J$. The second step which is the threshold manipulation step: selecting the threshold and dealing with the coefficients according to the equation as follows:
The soft-threshold de-noising function

$$d'_{j,k} = \begin{cases} d_{j,k} & ,|d_{j,k}| \geq t \\ 0 & ,|d_{j,k}| < t \end{cases} \tag{2}$$

where $d_{j,k}$ denotes the coefficient of the transformation, $d_{j,k}$ denotes the coefficient of the threshold manipulation, $t = \sigma \cdot \sqrt{2\ \log(N)}$ is the threshold, and $N$ is the total number of the image pixel. The final step is the reconstruction step: reconstructing the image with the coefficients $d_{j,k}$ by inverse wavelet transformation [7, 8, 9].

### 2.2 Support Vector Machine in Classification

In this section, we only briefly introduce the final classification function. For the detailed theory of SVM in classification, please refer to [10,11,12]. The final classification function is

$$f(x) = Sign\left( \sum_{i=1}^{N} \alpha_i y_i \varphi(x_i)^T \varphi(x_i) + \frac{1}{N_s} \sum_{0<\alpha_j<C} \left( y_j - \sum_{i=1}^{N} \alpha_i y_i \varphi(x_i)^T \varphi(x_j) \right) \right) \tag{3}$$

If there is a Kernel function such as $K\left(x_i^T, x_j\right) = \varphi(x_i)^T \varphi(x_j)$, it is usually unnecessary to explicitly know what $\varphi(x)$ is, and we only need to work with a kernel function in the training algorithm. The non-linear classification function is

$$f(x) = Sign\left( \sum_{i=1}^{N} \alpha_i y_i K\left(x_i, x\right) + \frac{1}{N_s} \sum_{0<\alpha_j<C} \left( y_j - \sum_{i=1}^{N} \alpha_i y_i K\left(x_i, x_j\right) \right) \right) \tag{4}$$

There are some different kernels for generating the inner products to construct machines with different types of nonlinear decision surfaces in the input space. Choosing among different kernels the model that minimizes the estimate, one chooses the best model. Common examples of the kernel function are the polynomial kernel $K(x, y) = (xy + 1)^d$ and the Gaussian radial basis function $K(x, y) = \exp\left(-1/\sigma^2 (x - y)^2\right)$ where $d$ is the degree of the polynomial kernel function and $\sigma$ is the bandwidth of the Gaussian radial basis function kernel. It has proved that the upper bound $C$ and the kernel parameter $\sigma^2$ play an important role in the performance of SVM.

## 3 Experiment Design

### 3.1 Data Collection and Preprocessing by De-noice Model

In our empirical analysis, we set out to examine the five-day moving trend of the Shanghai Stock Exchange (SSE) Composite Index. The original data points cover the time period from 28/04/1997 up to 12/09/2006 which is 2261 data. We select 1920

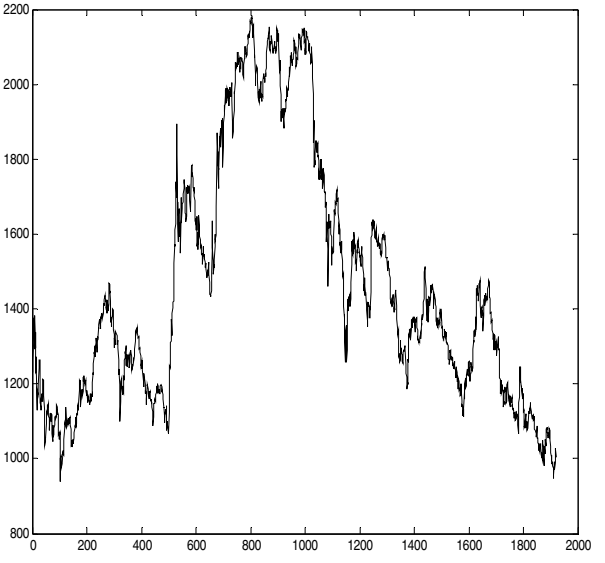



**Fig. 2.** SSE composite index

data from the 2261 data as training data and take the rest 341 data as testing data. As shown in Fig. 2, the 1920 data are illustrated.

Figure 3 illustrate the 1920 smooth SSE Composite Index data which had been de-noised by soft-thresholding.

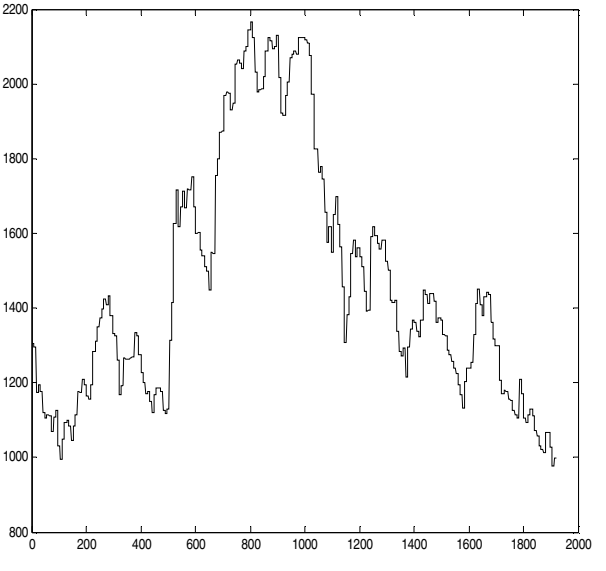



**Fig. 3.** Smoothed SSE composite index

Based on the soft-threshold which determined by the wavelet de-noise Model, we classify the stock market into up-trend, stochastic trend and down-trend which have 357, 1171,392 data respectively. Figure 4 illustrates the details residual of SSE Composite Index and soft-threshold [13].

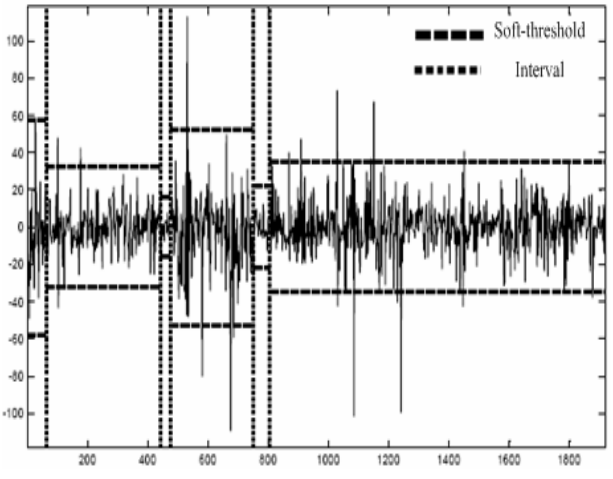



**Fig. 4.** Residual of SSE composite index and soft-thresholds

We define the noisy data whose five-day SSE Composite Index change value between the up and lower soft-threshold as the stochastic trend. In consequence, the value of data above the upper soft-threshold is defined as the up-trend and the value of data below the lower soft-threshold are defined as the down-trend. Then, we remove the 1171 stochastic trend data from the original SSE Composite Index and take the rest 749 data which belong to the up-trend and down-trend as the training data.

### 3.2 The Input Data of SVM

The input data used in this study is technical indicators and the direction of change in the five-day SSE Composite Index. The selected 12 technical indicators are the initial attributes which are presented in Table 1.

1) $C_t, h_t, l_t$ is the closing , highest and lowest price at time $t$ , $V_t$ is trading volume at time $t$ ; 2) $AU(AD)$ 14 days $C_t$ up (down) average rang; 3) $EMA$ is the exponential moving average; 4) $HH_t$ , $LL_t$ mean highest high and lowest low.

The forecasting performance $P$ is evaluated using the following equation:

$$P = \frac{1}{m}\sum_{i=1}^{m} D_i \qquad (i = 1, 2, \cdots, m) \tag{5}$$

where $D_i$ is the forecasting result for the $i$ th up-trend and down-trend trading day which is not including the stochastic trading day . It is defined by

**Table 1.** Selected technical indicators and their formulas

| Technical indicators | Formula |
|---|---|
| ALF(Alexander's Filter) | $(C_{t-5}/C_t\text{-}1)\times 100$ 1) |
| RS(Relative Strong) | $AU / AD$ 2) |
| RSI(Relative Strong Index) | $100 - 100 \times AD/(AD + AU)$ |
| MFI(Money Flow Index) | $(+MF)/((+MF) + (-MF)) \times 100$ |
| | $+(-)MF = EMA(U(D), n)$ |
| %BB(Bollinger's Band) | $(C - LB)/(UB - LB) \times 100$ |
| | $UB(LB) = MA + (-)V \times 2$ |
| V(Volatility) | $SD(Y)\times 100\%, Y=\ln(C_t/C_{t-1})$ |
| VB(Volatility Band) | $UB-LB$ |
| CHO (Chaikin Oscillator) | $MA(A,5) - MA(A,10)$ |
| | $A = (C_t/((h_t + l_t)/2) - 1) \times V_t$ |
| MACD(Moving Average Convergence/Divergence) | $EMA(C_t,12)-EMA(C_t,26)$ 3) |
| %K | $(C_t\text{-}LL_{t-n})/(HH_{t-n}\text{-}LL_{t-n})\times 100$ 4) |
| A/D Osc (Accumulation and distribution oscillator) | $(H_t\text{-}C_{t-1})/(H_t\text{-}L_t)$ |
| Williams %R | $(H_n\text{-}C_t)/(H_n\text{-}C_n)\times 100$ |

$$D_i = \begin{cases} 1 & if\ PO_i = AO_i, \\ 0 & otherwise \end{cases} \tag{6}$$

where $PO_i$ is the forecasting output from the model for the $i$ th trading day, $AO_i$ is the actual output for the $i$ th trading day, $m$ is the number of the test examples [5].

## 4 Experiment Result

In this study, we select the data points covering the time period from 28/04/1997 to 12/09/2006. There are 2261 data points of Shanghai Stock Exchange (SSE) Composite Index. We use the first 1920 data of the 2261 original data as training set and take the rest 341 data as testing data.

We use wavelet soft-threshold de-noise Model to de-noise the 1920 training data. As illustrated in Fig.2- Fig.4, we can see the detail process of de-noise. As a result of the soft-threshold de-noising, we get 1171 noised data which are classified as the stochastic trend data. We remove the 1171stochastic trend data from the original SSE Composite Index and take the rest 749 data which belong to the up-trend and down-trend as the training set for SVM.

The Gaussian radial basis function is used as the kernel function of the SVM. We conduct the experiment with respect to various kernel parameters and the upper bound $C$ . The range for kernel parameter is between 1 and 100 and the range for $C$

is between 1 and 100. We use the 749 data mentioned above to train the SVM and apply the SVM to classify the 341 test data. For comparison, we also use the 1920 data mentioned above to train SVM and employ SVM to classify the same 341 test data. The forecasting results of two methods are shown in table 2.

**Table 2.** Best forecasting results of two methods

| SVM | Testing/training data | C | $\sigma$ | Hit ratio |
|---|---|---|---|---|
| De-noise | Testing data | 90 | 20 | 60.12% |
| | Training data | 30 | 10 | 99.87% |
| Noisy | Testing data | 10 | 100 | 54.25% |
| | Training data | 50 | 10 | 99.95% |

The results in table 2 show that best hit ratio of the de-noise SVM is 60.12% which are better than the best hit ratio 54.25% of noisy SVM.

## 5 Conclusion

Many applications of SVM to forecast financial time series have been reported. Most of the researches only paid attention to select optimum parameters of SVM when they want to enhance forecasting performance. However, as SVM has the noise-tolerant property, little study discusses about preprocessing the noisy input data to enhance the forecasting performance. In this study, on the condition of selecting optimum parameters of SVM, we use soft-thresholding to de-noise the training data and get a better optimal hyperplane than the optimal hyperplane learned with noisy training data. Consequently, compared with the 54.25% hit ratio of the noisy SVM, the forecasting performance of the de-noised SVM is 60.12% hit ratio. The hit ratio is also better than Kim's 57.83%, which is the best prediction performance in forecasting the trend of Korea composite stock price index (KOSPI) with SVM [6]. This study proves that de-noising the training data can effectively enhance the forecasting performance of SVM.

## References

1. Deboeck, G. J.: Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets. Wiley, New York (1994)
2. Cao, L. J., Tay, F. E. H.: Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting. IEEE Transactions on Neural Networks, 14 (2003) 33-52
3. Cao, L. J., Tay, F. E. H.: Financial Forecasting Using Support Vector Machines. Neural Comput, (2001)184–192
4. Vapnik, V.: The Nature of Statistical Learning Theory. Spring-Verlag, New York (1995)
5. Kim, Kyoung-jae: Financial time series forecasting using support vector machines. Neurocomputing, (2003) 307–319

6. Gestel, T.V., Suykens, J. A. K., Baestaens, D.E., Lambrechts, A., Lanckriet, G., Vandaele, B., Moor, B.D., Vandewalle, J. : Financial Time Series Prediction Using Least Squares Support Vector Machines within the Evidence Framework. IEEE Transactions on Neural Networks, 12 (2001) 809-821
7. Donoho, D. L.: Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data. Proceedings of Symposia in Applied Mathematics, 47 (1993) 173–205
8. Donoho, D. L.: De-noising by soft-thresholding. IEEE Trans. on Information Theory, 41 (1995) 613–6274
9. Kecman, V.: Learning and Soft Computing, Support Vector machines, Neural Network and Fuzzy Logic Models.The MIT Press, Cambridge, MA (2001)
10. Vapnik, V.: Statistical learning theory. Wiley, New York (1998)
11. Wang, L.P., Fu.X.J. : Data Mining with Computation Intelligence. Springer, Berlin (2005)
12. Han, M., Xi, J., Xu, S., Yin, F.L.: Prediction of chaotic time series based on the recurrent predictor neural network IEEE Trans. Signal Processing, 52(2004)3409-3416
13. Teo, K.K., Wang, L.P., Lin, Z.: Wavelet packet multi-layer perception for chaotic time series prediction: effects of weight initialization. Lecture Notes in computer Science, 2074 (2001)210-317

# Attribute Granules in Formal Contexts

Wei-Zhi Wu

School of Mathematics, Physics and Information Science,
Zhejiang Ocean University, Zhoushan, Zhejiang, 316004, P.R. China
wuwz@zjou.edu.cn

**Abstract.** Granular computing is a basic issue in knowledge representation and data mining. In this paper, the concept of attribute granules in formal contexts is introduced. The mathematical structure of attribute granules is investigated.

**Keywords:** Concept lattices; Formal concept analysis; Formal contexts; Granular computing; Granules.

## 1 Introduction

Ever since the introduction of the concept of "Granular computing" (GrC) [11,20], we have witnessed a rapid development of and a fast growing interest in the topic [2,4,9,12,13,14,18,19]. Many models and methods of GrC concentrating on concrete models in particular contexts have been proposed and studied. A primitive notion in GrC is called a granule which may be interpreted as one of the numerous small particles forming a larger unit. The set of granules provide a representation of the unit with respect to a particular level of granularity. Thus one of main directions in the study of GrC is to deal with the construction, interpretation, and representation of granules.

The theory of formal concept analysis, proposed by Wille [16], provides a framework for the discovery and design of concept hierarchies from relational information systems. It starts with the notion of a formal context specifying which objects have what attributes. It is based on the perspective that a concept is constituted by two parts: its extension and its intension. An important notion in formal concept analysis is thus a formal concept which is a pair consisting of a set of objects (the extension) and a set of attributes (the intension) such that the intension consists of exactly those attributes that the objects in the extension have in common, and the extension contains exactly those objects that share all attributes in the intension. All concepts associated with the context form a complete lattice called the concept lattice. The concept lattice reflects the relationship of generalization and specialization among concepts. It is thus an intuitive and effective way to represent, discover and design knowledge structure. Formal concept analysis is now emerging as a powerful methodology for information retrieval, machine learning and knowledge discovery [5,8,10,15,17].

Since the basic structure of a concept lattice is the set of attribute-concepts which we call them attribute granules and every formal concept in the concept lattice can be represented as a meet of some attribute-concepts, we investigate in this paper the structure of attribute granules in formal contexts.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 395–402, 2007.

## 2 Formal Contexts and Concept Lattices

**Definition 1.** *A formal context is a triplet* $(U, A, I)$*, where* $U = \{x_1, x_2, \ldots, x_n\}$ *is a non-empty, finite set of objects,* $A = \{a_1, a_2, \ldots, a_m\}$ *is a non-empty, finite set of attributes, and* $I \subseteq U \times A$ *is a binary relation between* $U$ *and* $A$*,* $(x, a) \in I$ *means that object* $x$ *has attribute* $a$*.*

In this paper, we assume that the binary relation $I$ is regular, that is, it satisfies the following conditions: for any $(x, a) \in U \times A$,

(1) there exist $a_1, a_2 \in A$ such that $(x, a_1) \in I$ and $(x, a_2) \notin I$,
(2) there exist $x_1, x_2 \in U$ such that $(x_1, a) \in I$ and $(x_2, a) \notin I$.

For $X \subseteq U$ and $B \subseteq A$, we define

$$X^* = \{a \in A : (x, a) \in I, \forall x \in X\}, \quad B^* = \{x \in U : (x, a) \in I, \forall a \in B\}.$$

$X^*$ is the maximal set of attributes shared by all objects in $X$. Similarly, $B^*$ is the maximal set of objects that have all attributes in $B$. For $x \in U$ and $a \in A$, we denote $x^* = \{x\}^*$ and $a^* = \{a\}^*$.

The pair of $^*$ functions induces a Galois connection [1,3] between $\mathcal{P}(U)$ (where $\mathcal{P}(X)$ denotes the power set of $X$) and $\mathcal{P}(A)$. The two functions $^* : \mathcal{P}(U) \to \mathcal{P}(A)$ and $^* : \mathcal{P}(A) \to \mathcal{P}(U)$ are called derivation operators and satisfy the following properties [17]:

*Property 1.* Let $(U, A, I)$ be a formal context. If $X, X_1, X_2 \subseteq U$ and $B, B_1, B_2 \subseteq A$, then

(1) $X_1 \subseteq X_2 \Longrightarrow X_1^* \supseteq X_2^*, \quad B_1 \subseteq B_2 \Longrightarrow B_1^* \supseteq B_2^*$,
(2) $X \subseteq X^{**}, \quad B \subseteq B^{**}$,
(3) $X^* = X^{***}, \quad B^* = B^{***}$,
(4) $(X_1 \cup X_2)^* = X_1^* \cap X_2^*, \quad (B_1 \cup B_2)^* = B_1^* \cap B_2^*$,
(5) $X \subseteq B^* \Longleftrightarrow B \subseteq X^* \Longleftrightarrow X \times B \subseteq I$.

**Definition 2.** *Let* $(U, A, I)$ *be a formal context. A pair* $(X, B), X \subseteq U, B \subseteq A$*, is called a formal concept of the context* $(U, A, I)$ *if* $X^* = B$ *and* $B^* = X$*. The set of objects* $X$ *and the set of attributes* $B$ *are respectively called the extension and the intension of the formal concept* $(X, B)$*.*

Thus in a formal concept $(X, B)$, objects in $X$ share all attributes $B$, and only attributes $B$ are possessed by all objects in $X$. By Property 1(3), for any object set $X \subseteq U$, $(X^{**}, X^*)$ is a formal concept, and similarly, and for any attribute set $B \subseteq A$, $(B^*, B^{**})$ is also a formal concept. In particular, $(x^{**}, x^*)$ and $(a^*, a^{**})$ are formal concepts for all $x \in U$ and $a \in A$, $(x^{**}, x^*)$ and $(a^*, a^{**})$ are called an object concept and an attribute concept respectively [7].

The set of all formal concepts forms a complete lattice called a concept lattice [7] and is denoted by $L(U, A, I)$. The meet and join of the lattice are given by:

$$\begin{aligned} (X_1, B_1) \wedge (X_2, B_2) &= (X_1 \cap X_2, (B_1 \cup B_2)^{**}), \\ (X_1, B_1) \vee (X_2, B_2) &= ((X_1 \cup X_2)^{**}, B_1 \cap B_2). \end{aligned} \tag{1}$$

The corresponding partial order relation $\leq$ in the concept lattice $L(U, A, I)$ is given as follows: for $(X_1, B_1), (X_2, B_2) \in L(U, A, I)$,

$$(X_1, B_1) \leq (X_2, B_2) \Longleftrightarrow X_1 \subseteq X_2 \Longleftrightarrow B_1 \supseteq B_2.$$

*Example 1.* Table 1 depicts an example of formal context $F = (U, A, I)$, where $U = \{1, 2, 3, 4, 5\}$, $A = \{a, b, c, d, e, f\}$, and for each $(x_i, a_j) \in U \times A$, $(x_i, a_j) \in I$ if and only if (iff) object $x_i$ has value 1 in attribute $a_j$, i.e., $a_j(x_i) = 1$. Figure 1 is the Hasse diagram of the concept lattice derived from Table 1.

**Table 1.** A formal context

| $U$ | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 2 | 1 | 0 | 1 | 0 | 0 | 0 |
| 3 | 1 | 1 | 0 | 0 | 0 | 0 |
| 4 | 0 | 1 | 1 | 1 | 1 | 0 |
| 5 | 1 | 0 | 0 | 0 | 1 | 1 |

Following the standard notions in formal concept analysis, set notions are separator-free in the sequel to follow, e.g., 235 stands for $\{2, 3, 5\}$.

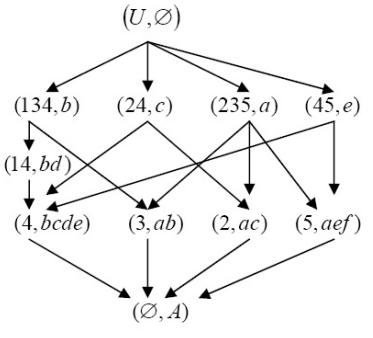



**Fig. 1.** $L(U, A, I)$

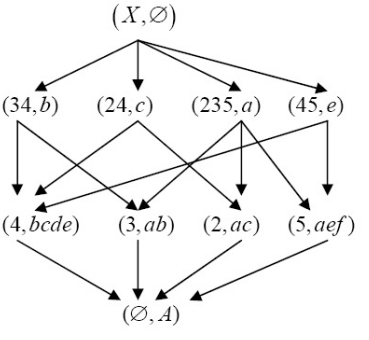



**Fig. 2.** $L(X, A, I_X)$

## 3 Characterization of Attribute Granules

**Definition 3.** *Let $F = (U, A, I)$ be a formal context. For any $X \subseteq U$, we can obtain a formal context $F_X = (X, A, I_X)$ which is called a sub-context of $F$, where $I_X = I \cap (X \times A)$. We define the functions ${}^{*_X} : \mathcal{P}(A) \to \mathcal{P}(X)$ and ${}^{*_X} : \mathcal{P}(X) \to \mathcal{P}(A)$ in the sub-context $F_X = (X, A, I_X)$ as follows:*

$$B^{*_X} = \{x \in X : (x, a) \in I, \forall a \in B\}, \quad B \subseteq A.$$
$$Y^{*_X} = \{a \in A : (x, a) \in I, \forall x \in Y\}, \quad Y \subseteq X.$$

It can easily be observed that $B^{*_X} = B^* \cap X$, and, of course, $B^{*_U} = B^*$. By Property 1 we can obtain the following Property 2.

*Property 2.* Let $(U, A, I)$ be a formal context and $Y \subseteq X \subseteq U$. Then
(1) $Y^{*X} = Y^{*U}$,
(2) $B^{*Y} \subseteq B^{*X}$ for all $B \subseteq A$,
(3) $a^{*Y} \subseteq a^{*X}$ for all $a \in A$,
(4) $B^{*X*X} \subseteq B^{*Y*Y}$ for all $B \subseteq A$,
(5) $a^{*X*X} \subseteq a^{*Y*Y}$ for all $a \in A$.

Combining Properties 1 and 2 we can conclude Property 3.

*Property 3.* Let $(U, A, I)$ be a formal context, $X \subset U$ and $Y = U - X$. Then
(1) $B^{*U} = B^{*X} \cup B^{*Y}, \quad B \subseteq A$,
(2) $a^{*U} = a^{*X} \cup a^{*Y}, \quad a \in A$,
(3) $B^{*U*U} = B^{*X*X} \cap B^{*Y*Y}, \quad B \subseteq A$,
(4) $a^{*U*U} = a^{*X*X} \cap a^{*Y*Y}, \quad a \in A$.

It is well-known that a formal concept $(X, B)$ in the concept lattice $L(U, A, I)$ can be represented as a meet of the attribute concepts of its intension [7], that is, $(X, B) = \bigwedge_{a \in B} (a^*, a^{**})$. Thus we can see that the set of all attribute concepts $\{(a^*, a^{**}) : a \in A\}$ forms a basis of the concept lattice $L(U, I, A)$, i.e., the set of all attribute concepts in a concept lattice reflects the information granules of the concept lattice structure. We call the set of attribute concepts $\{(a^*, a^{**}) : a \in A\}$ the attribute granules of the concept lattice $L(U, A, I)$.

For any $X \subseteq U$, we define a binary relation $R_X$ on $A$ as follows:

$$R_X = \{(a, b) \in A \times A : a^{*X} \subseteq b^{*X}\}.$$

Clearly, $R_X$ is reflexive and transitive but may not be symmetric. For any $a \in A$, denote $R_X(a) = \{b \in A : (a, b) \in R_X\}$.

*Property 4.* Let $(U, A, I)$ be a formal context, $X \subseteq A$, and $a \in A$. Then

$$R_X(a)^{*X} = a^{*X}. \tag{2}$$

*Proof.* Since
$$\begin{aligned} R_X(a)^{*X} &= \{y \in X : (y, b) \in I, \forall b \in R_X(a)\} \\ &= \bigcap_{b \in R_X(a)} \{y \in X : (y, b) \in I\} = \bigcap_{b \in R_X(a)} b^{*X}, \end{aligned}$$
note that $a^{*X} \subseteq b^{*X}$ for all $b \in R_X(a)$, we have $\bigcap_{b \in R_X(a)} b^{*X} = a^{*X}$. Therefore Eq.(2) holds.

**Theorem 1.** *Let $F = (U, A, I)$ be a formal context. Then $(R_U(a), R_U(a)^*)$ is a formal concept of $F$ and $R_U(a) = a^{**}$.*

*Proof.* Since from Property 4 we have $a^{**} = R_U(a)^{**}$, by Property 1(2) we obtain

$$R_U(a) \subseteq R_U(a)^{**} = a^{**}. \tag{3}$$

On the other hand, for any $b \in R_U(a)^{**}$, that is, $\{b\} \subseteq a^{**}$, by Property 1 we have $a^* = a^{***} \subseteq \{b\}^* = b^*$. Hence, $a^* \subseteq b^*$, that is, $b \in R_U(a)$, from which we conclude that

$$R_U(a)^{**} = a^{**} \subseteq R_U(a). \tag{4}$$

It follows from Eqs.(3) and (4) that $R_U(a)^{**} = a^{**} = R_U(a)$. Thus $(R_U(a)^*, R_U(a))$ is a formal concept of $F$.

*Property 5.* Let $(U, A, I)$ be a formal context, $X \subset U$, $B \subseteq A$, and $B_i \subseteq A$, $i = 1, 2, \ldots, k$. If $B^{*U} = \bigcap_{i=1}^{k} B_i^{*U}$, then $B^{*X} = \bigcap_{i=1}^{k} B_i^{*X}$.

*Proof.* Let $Y = U - X$. On one hand, we have

$$B^{*X} = B^{*U} \cap X, \quad B_i^{*X} = B_i^{*U} \cap X, \ i = 1, 2, \ldots, k.$$

On the other hand, by Property 3(1) we see that

$$B^{*U} = B^{*X} \cup B^{*Y}, \quad B_i^{*U} = B_i^{*X} \cup B_i^{*Y}, \ i = 1, 2, \ldots, k.$$

Then by Property 1(4) we conclude

$$\begin{aligned} B^{*X} &= B^{*U} \cap X = (\bigcap_{i=1}^{k} B_i^{*U}) \cap X = (\bigcap_{i=1}^{k} (B_i^{*X} \cup B_i^{*Y})) \cap X \\ &= \bigcap_{i=1}^{k} ((B_i^{*X} \cup B_i^{*Y}) \cap X) = \bigcap_{i=1}^{k} B_i^{*X}. \end{aligned}$$

*Property 6.* Let $(U, A, I)$ be a formal context, $X \subset U$, $a \in A$, and $a_i \in A$, $i = 1, 2, \ldots, k$. If $a^{*U} = \bigcap_{i=1}^{k} a_i^{*U}$, then $a^{*X} = \bigcap_{i=1}^{k} a_i^{*X}$.

**Theorem 2.** *Let $(U, A, I)$ be a formal context, $X \subset U$, $B \subseteq A$, and $B_i \subseteq A$, $i = 1, 2, \ldots, k$. If*

$$(B^{*U}, B^{*U*U}) = \bigwedge_{i=1}^{k} (B_i^{*U}, B_i^{*U*U}), \tag{5}$$

*then*

$$(B^{*X}, B^{*X*X}) = \bigwedge_{i=1}^{k} (B_i^{*X}, B_i^{*X*X}). \tag{6}$$

*Proof.* From Eqs.(1) and (5) we have $B^{*U} = \bigcap_{i=1}^{k} B_i^{*U}$. Then by Property 5 we obtain

$$B^{*X} = \bigcap_{i=1}^{k} B_i^{*X}. \tag{7}$$

Hence

$$(\bigcup_{i=1}^{k} B_i^{*X*X})^{*X*X} = (\bigcap_{i=1}^{k} B_i^{*X*X*X})^{*X} = (\bigcap_{i=1}^{k} B_i^{*X})^{*X} = (B^{*X})^{*X} = B^{*X*X}. \tag{8}$$

Combining Eq.(7) and Eq.(8), we conclude Eq.(6).

*Remark* 1. Theorem 2 tells us that a concept lattice induced from a sub-context of a formal context $F$ inherits the hierarchical structure of the concept lattice derived from $F$. For example, Figure 2 is the Hasse diagram of the concept lattice derived from sub-context $(X, A, I_X)$ of $(U, A, I)$ in Example 1, where $X = \{2, 3, 4, 5\}$.

**Theorem 3.** *Let $(U, A, I)$ be a formal context, $X \subset U$, $a \in A$, and $a_i \in A$, $i = 1, 2, \ldots, k$. If $(a^{*U}, a^{*U*U}) = \bigwedge_{i=1}^{k} (a_i^{*U}, a_i^{*U*U})$, then $(a^{*X}, a^{*X*X}) = \bigwedge_{i=1}^{k} (a_i^{*X}, a_i^{*X*X})$.*

## 4 Meet-Irreducible Elements in Concept Lattices

**Definition 4.** [6] *Let $L$ be a lattice. An element $a \in L$ is said to be meet-irreducible if $a = \bigwedge_{x \in X} x$ implies $a \in X$.*

*Property 7.* [6] Let $L$ be a finite lattice. Every element in $L$ is a meet of some meet-irreducible elements.

**Theorem 4.** *Let $(U, A, I)$ be a formal context, $X \subset U$, and $a \in A$. If $(a^{*X}, a^{*X*X})$ is a meet-irreducible element in $L(X, A, I_X)$, then $(a^{*U}, a^{*U*U})$ is a meet-irreducible element in $L(U, A, I)$.*

*Proof.* If $(a^{*U}, a^{*U*U})$ is not a meet-irreducible element in $L(U, A, I)$, then the attribute concept $(a^{*U}, a^{*U*U})$ is a meet of some meet-irreducible elements in $L(U, A, I)$, i.e., there exist $a_i \in A, i = 1, 2, \ldots, k$, $k \geq 2$, such that $(a_i^{*U}, a_i^{*U*U})$ is meet-irreducible in $L(U, A, I)$ for each $i = 1, 2, \ldots, k$ and $(a^{*U}, a^{*U*U}) = \bigwedge_{i=1}^{k} (a_i^{*U}, a_i^{*U*U})$. By Theorem 3 we then have $(a^{*X}, a^{*X*X}) = \bigwedge_{i=1}^{k} (a_i^{*X}, a_i^{*X*X})$. Since, for each $i = 1, 2, \ldots, k$, $(a_i^{*X}, a_i^{*X*X})$ is a meet-irreducible element or can be represented as a meet of some meet-irreducible elements in $L(X, A, I_X)$, we conclude that $(a^{*X}, a^{*X*X})$ can be represented as a meet of some meet-irreducible elements in $L(X, A, I_X)$, which contradicts the assumption that $(a^{*X}, a^{*X*X})$ is a meet-irreducible element in $L(X, A, I_X)$. Thus we have proved that $(a^{*U}, a^{*U*U})$ is a meet-irreducible element in $L(U, A, I)$.

According to Proposition 13 in [7], we can obtain the following theorem.

**Theorem 5.** *Let $(U, A, I)$ be a formal context and $a \in A$. Then $(a^{*U}, a^{*U*U})$ is a meet-irreducible element in $L(U, A, I)$ iff there is an object $x \in U$ such that $(x, a) \notin I$ and $(x, b) \in I$ for all $b \in \{c \in A : c^{*U} \supset a^{*U}\}$.*

The following theorem can help us to determine whether or not an attribute concept is meet-irreducible.

**Theorem 6.** *Let $(U, A, I)$ be a formal context and $a \in A$. Then the attribute concept $(a^{*U}, a^{*U*U})$ is a meet-irreducible element in $L(U, A, I)$ iff one of the following two conditions holds:*

*(1) there does not exist* $b \in A$ *such that* $b^{*U} \supset a^{*U}$*, i.e.,* $\{b \in A : b^{*U} \supset a^{*U}\} = \emptyset$*,*

*(2)* $\{b \in A : b^{*U} \supset a^{*U}\} \neq \emptyset$ *and* $\cap\{b^{*U} - a^{*A} : b^{*U} \supset a^{*U}\} \neq \emptyset$.

*Proof.* If there does not exist $b \in A$ such that $b^{*U} \supset a^{*U}$, then by definition it is obvious to see that $(a^{*U}, a^{*U*U})$ is a meet-irreducible element in $L(U, A, I)$.

Now we assume that $\{b \in A : b^{*U} \supset a^{*U}\} \neq \emptyset$.

Case I. $\cap\{b^{*U} - a^{*U} : b^{*U} \supset a^{*U}\} \neq \emptyset$.

In such a case, we can find an object $x \in \cap\{b^{*U} - a^{*U} : b^{*U} \supset a^{*U}\}$. Since $x \notin a^{*U}$, we have $(x, a) \notin I$. On the other hand, for any $b \in A$ satisfying $b^{*U} \supset a^{*U}$, we have $x \in b^{*U}$, that is, $(x, b) \in I$ for all $b \in \{c \in A : c^{*U} \supset a^{*U}\}$. Consequently, by Theorem 5 we conclude that $(a^{*U}, a^{*U*U})$ is a meet-irreducible element in $L(U, A, I)$.

Case II. $\cap\{b^{*U} - a^{*U} : b^{*U} \supset a^{*U}\} = \emptyset$.

In such a case, we cannot find any object $x \in U$ such that $x \notin a^{*U}$ and $x \in b^{*U}$ for all $b \in A$ satisfying $b^{*U} \supset a^{*U}$. Alternatively, we cannot find any object $x \in U$ such that $(x, a) \notin I$ and $(x, b) \in I$ for all $b \in A$ satisfying $b^{*U} \supset a^{*U}$. Thus by Theorem 5 we conclude that $(a^{*U}, a^{*U*U})$ is not a meet-irreducible element in $L(U, A, I)$.

*Remark* 2. From Theorem 6 we can see that an attribute concept $(a^{*U}, a^{*U*U})$ is not a meet-irreducible element in $L(U, A, I)$ iff $\text{card}(\{b \in A : b^{*U} \supset a^{*U}\}) \geq 2$ and $\cap\{b^{*U} - a^{*U} : b^{*U} \supset a^{*U}\} = \emptyset$.

*Example 2.* In Example 1, it can easily be verified that $(w^*, w^{**})$ is a meet-irreducible element in $L(U, A, I)$ for $w \in \{a, b, c, d, e\}$. Since $\{w \in A : w^* \supset f^*\} = \{a, e\} \neq \emptyset$ and $\cap\{w^* - f^* : w^* \supset f^*\} = \{2, 3\} \cap \{4\} = \emptyset$, then by Theorem 6 we conclude that $(f^*, f^{**})$ is not a meet-irreducible element in $L(U, A, I)$. In fact, we can observe from Figure 1 that $(f^*, f^{**}) = (a^*, a^{**}) \wedge (e^*, e^{**})$.

## 5 Conclusion

Granular computing is a basic issue in knowledge representation and data mining. The class of all attribute concepts are the basic granules of the concept lattice derived from a formal context and each formal concept in a concept lattice can be represented a meet of some attribute concepts. We have investigated in this paper the mathematical structure of attribute granules in a concept lattice. For further study, we will investigate the mathematical structure of granules in more complex systems such as fuzzy concept lattices derived from fuzzy formal contexts.

## Acknowledgement

This work was supported by grants from the National Natural Science Foundation of China (No. 60373078 and No. 60673096).

## References


1. Barbut, M., Monjardet, B.: Order et Classification: Algeèbre et Combinatoire. Hachette, 1970
2. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Boston, 2002
3. Carpineto, C., Romano, G.: Galois: an order-theoretic approach to conceptual clustering. In: Utgoff, P. (ed.): Proceedings of ICML 293, Elsevier, Amherst, 1993, pp.33–40
4. Chen, Y.H., Yao, Y.Y.: Multiview intelligent data analysis based on granular computing. In: Proceedings of 2006 IEEE International Conference on Granular Computing, 2006, pp.281–286
5. Cole, R., Eklund, P., Stumme, G.: Document retrieval for e-mail search and discovery using formal concept analysis. Applied Aartificial Intelligence **17**(2003) 257–280
6. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge University Press, Cambridge, 2002
7. Ganter, B., Wille, R.: Formal Concept Analysis, Mathematical Foundations. Springer, Berlin, 1999
8. Hereth, J., Stumme, G., Wille, R., et al.: Conceptual knowledge discovery and data analysis. Lecture Notes in Artificial Intelligence 1867, Springer-Verlag, Berlin, 2000, pp.421–437
9. Inuiguchi, M., Hirano, S., Tsumoto, S. (eds.): Rough Set Theory and Granular Computing. Springer, Berlin, 2003
10. Kuznetsov, S.O.: Machine learning and formal concept analysis. In: Eklund, P. (ed.): ICFCA 2004. Lecture Notes in Artificial Intelligence 2961, Springer-Verlag, Berlin, 2004, pp.287–312
11. Lin, T.Y.: Granular computing, announcement of the BISC Special Interest Group on Granular Computing, 1997
12. Lin, T.Y., Yao, Y.Y., Zadeh, L.A. (eds.): Data Mining, Rough Sets and Granular Computing. Physica-Verlag, Heidelberg, 2002
13. Pedrycz, W. (ed.): Granular Computing: An Emerging Paradigm. Physica-Verlag, Heidelberg, 2001
14. Skowron, A., Stepaniuk, J.: Information granules: towards foundations of granular computing. International Journal of Intelligent Systems **16**(2001) 57–85
15. Valtchev, P., Missaoui, R., Godin, R.: Formal concept analysis for knowledge discovery and data mining: the new challenges. In: Eklund, P. (ed.): ICFCA 2004. Lecture Notes in Artificial Intelligence 2961, Springer-Verlag, Berlin, 2004, pp. 352–371
16. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.): Ordered Sets. Reidel, Dordrecht-Boston, 1982, pp.445–470
17. Wille, R.: Formal concept analysis as mathematical theory of concepts and concept hierarchies. In: Ganter, B., Stumme, G., Wille, R. (eds.): Formal Concept Analysis. Lecture Notes in Artificial Intelligence 3626, Springer-Verlag, Berlin, 2005, pp.1–33
18. Yao, Y.Y.: Perspectives of granular computing. Proceedings of 2005 IEEE International Conference on Granular Computing, Vol. 1, 2005, pp.85–90
19. Yao, Y.Y.: Modeling data mining with granular computing. Proceedings of the 25th Annual International Computer Software and Applications Conference (COMPSAC 2001), Chicago, USA, October 8-12, 2001, IEEE Computer Society, Los Alamitos, California, pp. 638–643
20. Zadeh, L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems **19**(1997) 111–127

# An Incremental Updating Algorithm for Core Computing in Dominance-Based Rough Set Model

Xiuyi Jia[1], Lin Shang[1], Yangsheng Ji[1], and Weiwei Li[2]

[1] National Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210093, China
jiaxiuyi@gmail.com, shanglin@nju.edu.cn, johnson_cs@163.com
[2] College of Automation, Chongqing University, Chongqing, 400030, China
amy107weiwei@tom.com

**Abstract.** This paper analyzes incremental updating for core computing in a dominance-based rough set model, which extends previous reduct studies in capability of dynamic updating and dominance relation. Then we redefine the dominance discernibility matrix and present an incremental updating algorithm. In this algorithm, when new samples arrive, the proposed solution only involves a few modifications to relevant rows and columns in the dominance discernibility matrix instead of recalculation.Both of theoretical analysis and experimental results show that the algorithm is effective and efficient in dynamic computation.

**Keywords:** Rough sets, dominance discernibility matrix, incremental updating, core.

## 1 Introduction

As a kind of mathematical tool, Rough sets [1] can be used to deal with imprecise, incomplete and inconsistent data. Rough set theory has been applied in several fields as machine learning, data mining and pattern recognition et.al. In practice, more often than not the attributes domains and classes are preference-ordered. The attributes with preference-ordered domains are called *criteria*. As pointed out in [2,3] the Classical Rough Set Approach(CRSA) cannot be applied to *multiple-criteria decision problems*, because it does not consider criteria but only regular attributes.

For this reason, Greco, Matarazzo and Slowinski [2,4] have proposed an extension of the rough set theory, called Dominance-based Rough Set Approach (DRSA). Many contributions on core and reduction had been reported [5,6,7]. Discernibility matrix concept is one of important fundamental concepts, and the discernibility matrix-based algorithms are an important member in the family of reduction algorithms. Besides its use in finding the reduct, it can also facilitate the computation of core. But little is contributed on incremental updating core. Yang [8] has proposed an incremental updating algorithm of the computation of a core based on discernibility matrix under CRSA, which inspires our algorithm.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 403–410, 2007.

In the paper, an approach to incremental updating for core computation under DRSA is proposed to avoid complete regeneration. The approach involving only a few modifications to relevant rows and columns in the matrix avoids complete recalculation. Not only does the theoretical analysis prove its correctness but also experimental results on several UCI data sets show its efficiency and effectiveness.

The rest of this paper is organized as follows. Section 2 briefly introduces some preliminary knowledge. Section 3 discusses the elementary relevant notions of DRSA and core. Section 4 presents the incremental updating algorithm of the computation of a core. Section 5 reports on the experiments. Finally, Section 5 concludes.

## 2 Preliminaries

### 2.1 Dominance-Based Rough Set Approach(DRSA)

Let us assume that learning examples are represented in *decision table* $DT = (U, C \cup D)$, where $U$ is a set of examples(objects), $C$ is a set of *condition attribute* describing examples, Let $f(x, q)$ denote the value of attribute $q \in C$ taken by object $x \in U$, $V_q$ is a domain of $q$ [9].

Assuming that all condition attributes $q \in C$ are criteria, let $S_q$ be an *outranking relation* on $U$ with respect to criterion $q$ such that $xS_qy$ means "$x$ is at least as good as $y$ with respect to criterion $q$". Furthermore, assuming that the set of decision attributes $D$(possibly a singleton $\{d\}$) makes a partition of $U$ into a finite number of classes, let $\mathbf{Cl} = \{Cl_t, t \in T\}$, $T = \{1, \ldots, n\}$, be a set of these classes such that each $x \in U$ belongs to one and only one $Cl_t \in \mathbf{Cl}$. We suppose that the classes are ordered, i.e. for all $r, s \in T$, $r > s$, such that the objects from $Cl_r$ are preferred to the objects from $Cl_s$.

The sets to be approximated are *upward union* and *downward union* of classes, respectively: $Cl_t^{\geq} = \bigcup_{s \geq t} Cl_s, Cl_t^{\leq} = \bigcup_{s \leq t} Cl_s, t = 1, \ldots, n$. Usually we do not take $Cl_n^{\leq}$ and $Cl_1^{\geq}$ into consideration because both of their value are $U$, there is no real meaning here.

The indiscernibility relation is substituted by a *dominance relation*. We say that $x$ *dominates* $y$ with respect to $P \subseteq C$, denoted by $xD_Py$, if $xS_qy$ for all $q \in P$. The dominance relation is reflexive and transitive. Given $P \subseteq C$ and $x \in U$, the "granules of knowledge" used for approximation in DRSA are:

-a set of objects dominating $x$, called *P-dominating set*, $D_P^+(x) = \{y \in U : yD_Px\}$,

-a set of objects dominated by $x$, called *P-dominated set*, $D_P^-(x) = \{y \in U : xD_Py\}$.

**Definition 1.** *Using $D_P^+(x)$ sets, P-lower and P-upper approximation of $Cl_t^{\geq}$ and $Cl_t^{\leq}$ are defined as:*

$$\underline{P}(Cl_t^{\geq}) = \{x \in U : D_P^+(x) \subseteq Cl_t^{\geq}\}, \overline{P}(Cl_t^{\geq}) = \bigcup_{x \in Cl_t^{\geq}} D_P^+(x), for\ t = 1, \ldots, n.$$

$$\underline{P}(Cl_t^{\leq}) = \{x \in U : D_P^-(x) \subseteq Cl_t^{\leq}\}, \overline{P}(Cl_t^{\leq}) = \bigcup_{x \in Cl_t^{\leq}} D_P^-(x), for\ t = 1, \ldots, n.$$

**Definition 2.** *The* P-boundaries *of* $Cl_t^{\geq}$ *and* $Cl_t^{\leq}$ *are defined as:*

$$Bn_P(Cl_t^{\geq}) = \overline{P}(Cl_t^{\geq}) - \underline{P}(Cl_t^{\geq}), Bn_P(Cl_t^{\leq}) = \overline{P}(Cl_t^{\leq}) - \underline{P}(Cl_t^{\leq}), for\ t = 1, \ldots, n.$$

### 2.2 Reduct and Core

In the following we will present some definitions of reduct and core.

**Definition 3.** *An information system is an ordered quadruple* $S = (U, C \cup D, V, f)$*, where* $U$ *is a non-empty finite set of objects,* $C \cup D$ *is a non-empty finite set of attributes,* $C$ *denotes the set of condition attributes and* $D$ *denotes the set of decision attributes.* $V$ *is the union of attribute domains,* $f : U \times (C \cup D) \rightarrow V$ *is an information function which associates an unique value of each attribute with every object belonging to* $U$*.* $f(x_i, q)$ *denotes the value of object* $x_i$ *in attribute* $q$*.*

**Definition 4.** *(See [10]) The quality of approximation of partition* $\boldsymbol{Cl}$ *by the set of attributes and criteria* $P$*:* $\gamma_P(\boldsymbol{Cl}) = \frac{|U - \bigcup_{n=1}^{N} Bn_P(d_n^{\geq})|}{|U|} = \frac{|U - \bigcup_{n=1}^{N} Bn_P(d_n^{\leq})|}{|U|}$

The *quality of approximation* expresses the ratio of all $P$-correctly sorted actions to all actions in the table.

**Definition 5.** *(See [10]) Let* $P \subseteq C$*, we call* $P$ *is one reduct of* $C$ *if it satisfies* $\gamma_P = \gamma_C$ *and* $\gamma_R \neq \gamma_C$ *for every* $R \subset P$*. The intersection of all reducts is called the core and denoted by* $Core(C)$*.*

**Definition 6.** *(See [10]) The attribute* $a \in C$ *is defined as indispensable if it satisfies* $\gamma_{C-\{a\}} < \gamma_C$*, otherwise, it is redundant.*

**Proposition 1.** *For every attribute* $a \in C$*,* $a \in Core(C) \Leftrightarrow a$ *is indispensable.*

## 3 Dominance Discernibility Matrix and Core Computing

The use of discernibility matrix is common to compute core. In paper [11], similar discernibility matrix based on dominance relation is defined. It has been proved wrong to compute the core using the matrix in some cases by Wu [12]. The error reason is there exists inconsistent data in decision table. To address this issue, Wu [12] proposed a new definition of discernibility matrix to compute the core, called dominance discernibility matrix, but the cost of constructing the matrix is too high. In what follows we will redefine the matrix to compute the core with less computing cost.

For further study we will give the following two propositions:

**Proposition 2.** $\underline{C}(Cl_i^{\leq}) \subseteq \underline{C}(Cl_j^{\leq}), \underline{C}(Cl_i^{\geq}) \supseteq \underline{C}(Cl_j^{\geq}), \quad 1 \leq i \leq j \leq n$

*Proof.* For every $x \in \underline{C}(Cl_i^{\leq})$, so $D_C^-(x) \subseteq Cl_i^{\leq}$. Since $i \leq j$ implies $Cl_i^{\leq} \subseteq Cl_j^{\leq}$, we have $D_C^-(x) \subseteq Cl_j^{\leq}$ and $x \in \underline{C}(Cl_j^{\leq})$, therefore, $\underline{C}(Cl_i^{\leq}) \subseteq \underline{C}(Cl_j^{\leq})$. Similarly, $\underline{C}(Cl_i^{\geq}) \supseteq \underline{C}(Cl_j^{\geq})$ can be proved correct. End of proof.

**Proposition 3.** $D_C^-(x) \subseteq D_{C-\{a\}}^-(x), D_C^+(x) \subseteq D_{C-\{a\}}^+(x)$

*Proof.* Immediate from the definition of *P-dominating set* and *P-dominated set.* End of proof.

Our redefined notion of dominance discernibility matrix is presented as the following:

**Definition 7.** *For given information system, the dominance discernibility matrix* $\boldsymbol{M} = \{a^{\#}(x_i, x_j)\}$,

$$a^{\#}(x_i, x_j) = \begin{cases} a_1^* & , \quad x_i \in U1 \\ a_2^* & , \quad x_j \in U1 \\ C & , \quad otherwise \end{cases}$$

$a_1^* = \{a \in C | f(x_i, a) > f(x_j, a), f(x_i, D) > f(x_j, D)\}$
$a_2^* = \{a \in C | f(x_i, a) < f(x_j, a), f(x_i, D) < f(x_j, D)\}$
$U1 = \underline{C}(Cl_{n-1}^{\leq}) \cup \underline{C}(Cl_2^{\geq})$

where $U1 = \underline{C}(Cl_{n-1}^{\leq}) \cup \underline{C}(Cl_2^{\geq}) = \bigcup_{i=1}^{n-1} \underline{C}(Cl_i^{\leq}) \cup \bigcup_{i=2}^{n} \underline{C}(Cl_i^{\geq})$ follows from Proposition 2. We do not take $\underline{C}(Cl_n^{\leq})$ and $\underline{C}(Cl_1^{\geq})$ into consideration because both of their values are $U$ as mentioned above.

**Theorem 1.** *For given information system IS, let* $IDM(C, \boldsymbol{M}) = \{m_{ij} | m_{ij} \in \boldsymbol{M} \wedge |m_{ij}| = 1\}$, *then* $IDM(C, \boldsymbol{M}) = Core(C)$.

*Proof.* In this theorem, $|m_{ij}| = 1$ means that $m_{ij}$ includes only a single attribute.

(1) First we prove that $IDM(C, \mathbf{M}) \subseteq Core(C)$.

For every $a \in IDM(C, \mathbf{M})$, we know $\exists m_{ij}$ made $a_1^* = \{a\}$ or $a_2^* = \{a\}$.

Suppose $x_i \in U1$, then $x_i \in \underline{C}(Cl_{n-1}^{\leq})$ or $x_i \in \underline{C}(Cl_2^{\geq})$.

(a) Suppose $x_i \in \underline{C}(Cl_{n-1}^{\leq})$.

From the definition of *P-boundaries* and Proposition 2 we can easily get $\bigcup_{n=1}^{N} Bn_C(Cl_n^{\leq}) = \overline{C}(Cl_{n-1}^{\leq}) - \underline{C}(Cl_{n-1}^{\leq})$. Since $\forall x \in \bigcup_{n=1}^{N} Bn_C(Cl_n^{\leq})$ implies $x \in \overline{C}(Cl_{n-1}^{\leq})$ and $x \notin \underline{C}(Cl_{n-1}^{\leq})$, we have $D_C^-(x) \not\subseteq Cl_{n-1}^{\leq}$, then $D_{C-\{a\}}^-(x) \not\subseteq Cl_{n-1}^{\leq}$ and $x \notin \underline{C - \{a\}}(Cl_{n-1}^{\leq})$ follow from proposition 3.

Since $x \in \overline{C}(Cl_{n-1}^{\leq})$ implies $D_C^+(x) \cap Cl_{n-1}^{\leq} \neq \emptyset$, then $D_{C-\{a\}}^+(x) \cap Cl_{n-1}^{\leq} \neq \emptyset$ again follows from proposition 3 and $x \in \overline{C - \{a\}}(Cl_{n-1}^{\leq})$,

$x \in \bigcup_{n-1}^{N} Bn_{C-\{a\}}(Cl_{n-1}^{\leq})$. $\bigcup_{n-1}^{N} Bn_C(Cl_{n-1}^{\leq}) \subseteq \bigcup_{n-1}^{N} Bn_{C-\{a\}}(Cl_{n-1}^{\leq})$ has been gotten now.

We will prove $|Bn_{C-\{a\}}| > |Bn_C|$ next, $|A|$ means the cardinality of $A$. For $x_i$, while if $f(x_i, D) > f(x_j, D)$, then $x_i \in D^-_{C-\{a\}}$, so $x_i \in \overline{C-\{a\}}(Cl^{\leq}_{f(x_j,D)})$. Since $f(x_i, D) > f(x_j, D)$ implies $x_i \notin \underline{C-\{a\}}(Cl^{\leq}_{f(x_j,D)})$, we have $x_i \in Bn_{C-\{a\}}(Cl^{\leq}_{n-1})$. We can get $x_i \notin Bn_C(Cl^{\leq}_{n-1})$ follows from the definition of $x_i$ and $|Bn_{C-\{a\}}| > |Bn_C|$. According to the Definition 4 and above result, $\gamma_{C-\{a\}} < \gamma_C$ can be gotten easily, so $a \in Core(C)$.

(b) Suppose $x_i \in \underline{C}(Cl^{\geq}_2)$.

Analogously with (a), we can prove $a \in Core(C)$.

Analogously for $x_j \in U1$ with above process, $a \in Core(C)$ can be proved correct easily. So $IDM(C, \mathbf{M}) \subseteq Core(C)$.

(2) Now we will prove that $IDM(C, \mathbf{M}) \supseteq Core(C)$.

Disprove: assume each $a \in Core(C)$, neither does there exist $a^*_1 = \{a\}$ nor $a^*_2 = \{a\}$.

(a) First we suppose that there doesn't exist $a^*_1 = \{a\}$.

For every $x \notin \bigcup_{n=1}^{N} Bn_C(Cl^{\leq}_{n-1})$ implies $x \notin Bn_C(Cl^{\leq}_{f(x,D)})$ and $x \in \overline{C}(Cl^{\leq}_{f(x,D)})$, so $x \in \underline{C}(Cl^{\leq}_{f(x,D)})$ and $D^-_C(x) \subseteq Cl^{\leq}_{f(x,D)}$. For every $y \in D^-_{C-\{a\}}(x)$, suppose $y \notin Cl^{\leq}_{f(x,D)}$, then $f(y, D) < f(x, D)$. Since $D^-_C(x) \subseteq Cl^{\leq}_{f(x,D)}$ implies $y \notin D^-_C(x)$, we have $f(y, q) \leq f(x, q)$ and $f(y, a) > f(x, a)$ for every $q \in C - \{a\}$. Therefore, according to the definition we get $a^*_1 = \{a\}$ that is inconsistent with the assumption. So $y \in Cl^{\leq}_{f(x,D)}$.

Owing to the arbitrariness of selection, we can get $D^-_{C-\{a\}} \subseteq Cl^{\leq}_{f(x,D)}$, so $x \notin Bn_{C-\{a\}}(Cl^{\leq}_{n-1})$. Because $x$ is also selected arbitrarily, $Bn_{C-\{a\}} \subseteq Bn_C$. Since $a$ is belong to the core implies $Bn_C \subseteq Bn_{C-\{a\}}$, $Bn_{C-a} = Bn_C$. This result is paradoxical with the definition of core.

(b) Analogously with above process, we can prove that its paradox between not exists $a^*_2 = \{a\}$ with the definition of core.

With (a) and (b) we can get $IDM(C, \mathbf{M}) \supseteq Core(C)$.

So it is also easy to see that $IDM(C, \mathbf{M}) = Core(C)$ from (1) and (2). End of proof.

## 4 Incremental Updating Algorithm of the Computation of a Core

### 4.1 Incremental Updating Algorithm

We will present an incremental updating algorithm of the computation of a core based on our Definition 7 and Theorem 1.

For simplification, we assume the value domain of decision attribute(s) $D$ are $1, 2, \ldots, n$ when decision table increases dynamically. For object $x$ and object $y$, we say they are inconsistent when they have same condition value but different decision value. Otherwise, we say they are consistent. Let $U1 = \underline{C}(Cl^{\leq}_{n-1}) \cup \underline{C}(Cl^{\geq}_2)$, for an added new object $x$, if $\forall y \in U1$, $x$ and $y$ are consistent, then we

say $x$ is consistent with $U1$. While if $\exists y \in U1$, $x$ and $y$ are inconsistent, then $x$ is inconsistent with $U1$.

For $U1 = \underline{C}(Cl_{n-1}^{\leq}) \cup \underline{C}(Cl_2^{\geq})$, we can get the dominance discernibility matrix **M2** for Definition 7. For the new added object $x$, we just get the dominance discernibility matrix(**M2(x)**) of $(U1 \cup U \cup \{x\})$ and compute the core based on Theorem 1. So the incremental updating of core is the updating problem of dominance discernibility matrix essentially. $U2$ is the set of inconsistent objects discovered in computing. The updating of **M2** is given as follows:

1) If $x$ is consistent with $U1$, $x \notin U2$ and $\forall y \in U1$ there exists $x \neq y$, then add a row and a column corresponding to object $x$, $U1 = U1 \cup \{x\}$.

2) If $x$ is consistent with $U1$ and either $x \in U2$ or $\exists y \in U1$ there exists $x = y$, then **M2** remains unchanged.

3) If $x$ is inconsistent with $U1$, then $\exists y \in U1$, $x$ is inconsistent with $y$. So delete the row and remend the column corresponding to object $y$, $U1 = U1 - \{y\}, U2 = U2 + \{x, y\}$.

Then we give description of the incremental updating algorithm as following:

**Algorithm.** Incremental Updating Algorithm of the Computation of a Core (IUA)

**Input**: (1)$U1 = \underline{C}(Cl_{n-1}^{\leq}) \cup \underline{C}(Cl_2^{\geq})$, $U2$=set of inconsistent objects computed, dominance discernibility matrix **M2**.
(2)new object $x$.

**Output**: **M2**(x) and Core(C).

**BEGIN**

```
IF x is consistent with U1 THEN
    IF ∀y ∈ U1 there exists x ≠ y and x ∉ U2 THEN
        insert a new row and a column into M2, and complete the
        matrix according to Definition 7;
        U1 = U1 ∪ {x};
    ELSE
        remains unchanged;
    End IF
ELSE
    find the inconsistent object y with x in U1;
    delete the row corresponding to object y;
    remend the column corresponding to object y;
    U1 = U1 − {y};U2 = U2 + {x, y};
END IF
compute the core according to Theorem 1;
END BEGIN
```

Through proposed algorithm *IUA*, we can get the right core. Suppose $U1 = \underline{C}(Cl_{n-1}^{\leq}) \cup \underline{C}(Cl_2^{\geq})$, **M2** is the dominance discernibility matrix followed from Definition 7, the added new object is $x$. For $(U1 \cup U \cup \{x\})$, the constructed matrix **M2'** according to Definition 7, then we can know that $m \in$ **M2'** if and only if $m \in \mathbf{M2}(x)$ from algorithm *IUA*. We can get right core according to **M2'** and Theorem 1, so algorithm *IUA* is correct.

### 4.2 Computation Complexity Analysis

For $U1 = \underline{C}(Cl_{n-1}^{\leq}) \cup \underline{C}(Cl_2^{\geq})$ and new object $x$, we first discuss the computation complexity of algorithm $IUA$.

(1)Space Complexity

The dominance discernibility matrix has $(|U1| \times |U|)$ elements, so the space complexity of algorithm $IUA$ is $O(|U1| \times |U|)$.

(2)Time Complexity

When adding a new object $x$, it requires at most $(|U1| + |U|)$ operations to assert the consistency between $x$ with $|U1|$. We need at most $(|U1| + |U|)$ operations when updating the matrix, and at most $(|U1| + 1) \times (|U| + 1)$ operations when scanning the matrix to compute the core. So the time complexity of algorithm $IUA$ is $O(|U1| \times |U|)$.

## 5 Experiment

The following experiments were conducted on an Pentium(R) D-2.8GHz CPU with 512MB main memory running Windows XP. All algorithms were implemented in C# and executed on the Visual Studio.NET 2003.

The datasets named *iris* and *mushroom* were selected from the *UCI Machine Learning Repository*. The dataset *iris* has 150 objects and 5 attributes with 3 classes. First we compute the dominance discernibility matrix with selected 100 objects from the dataset, then select 20, 50 objects respectively as incremental added objects. The dataset *mushroom* has 8124 objects and 22 attributes with 2 classes. We select 800 objects to compute the dominance discernibility matrix and choose respectively 200, 400 objects as incremental add objects. The results of computation times are presented in Table 1.

**Table 1.** The computational time(in seconds) of two algorithms worked in two data sets.

| Data Set | iris | | | mushroom | | |
|---|---|---|---|---|---|---|
| | number of objects | | | number of objects | | |
| Algorithm | 100 | 120 | 150 | 800 | 1000 | 1200 |
| Algorithm *Wu* | 0.187 | 1.093 | 2.765 | 276 | 606 | 1076 |
| *IUA* | 0 | 0.015 | 0.031 | 0 | 4.03 | 9.89 |

The algorithm followed from Definition 7 was named Algorithm *Wu* while our incremental updating algorithm was named *IUA*.

From Table 1 we can see that our algorithm could save more time than Wu's algorithm when adding new objects dynamically.

## 6 Conclusion

In the paper, We analyze and prove a new method to compute the core based on the redefinition of dominance discernibility matrix, and then propose an

incremental updating algorithm based on the definition. In this algorithm, when new samples arrive, the solution only involves a few modifications to relevant rows and columns in the dominance discernibility matrix instead of recalculation. So the running time is less than non-incremental algorithm, both shown by the theoretical complexity analysis and experimental results. In the future, we will optimize the algorithm and emphasize the experiments for the applicability.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 60503022.

## References

1. Pawlak, Z.: Rough set. *International Journal of Information and Computer Science*, **11** (1982) 341-356.
2. Greco, S., Matarazzo, B., Slowinski, R.: The use of rough sets and fuzzy sets in MCDM, in: *Adcances in Multiple Criteria Decision Making*, Gal,T., Stewart,T. Hanne,T.: Kluwer (1999) 14.1–14.59.
3. Slowinski, R., Stefanowski, J. Greco, S., Matarazzo,B.: Rough sets processing of inconsistent information. *Control and Cyberbetics*, **29**(2000) 379-404.
4. Greco, S., Inuiguchi, M., Slowinski, B.: Dominance-based rough set approach using possibility and necessity measures. In: *RSCTC2002*, J.J.Alpigini et al.(Eds.): LNAI2475 (2002) 85-92.
5. Hu, X.H., Cercone, N.: Learning in relational databases: A rough set approach. *Computational Intelligence*, **11(2)** (1995) 323-338.
6. Jelonek, J., Krawiec, K., Slowinski, R.: Rough set reduction of attributes and their domains for neural networks. *Computational Intelligence*, **11(2)** (1995) 339-347.
7. Wang, J, Wang, J.: Reduction algorithm based on discernibility matrix the ordered attributes method. *Journal of Computer Science and Technology*, **16(6)** (2001) 489-504.
8. Yang, M.: An incremental updating algorithm of the computation of a core based on the improved discernibility matrix. *Chinese Journal of Computers(in chinese)*, **29(3)** (2006) 407-413.
9. Greco, S., Matarazzo, B., Slowinski, R., Stefanowinski, J.: An algorithm for induction of decision rules consistent with the dominance principle. In: *RSCTC2000*, W.Ziarko and Y.Yao(Eds.): LNAI2005 (2001) 304-313.
10. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets methodology for sorting problems in presence of multiple attributes and criteria. *European Journal of Operational Research*, **138** (2002) 247-259.
11. Li, K.W., Wu, M.D.: Rough set based on order relations. In: *Proceedings of 2003 National Conference on Artificial Intelligence(in chinese)*, (2003) 1359-1363.
12. Wu, Y.M., Ye, D.Y.: Computation of a core in a rough set model based on dominance relation. *Journal of Computer Science and Technology(in chinese)*, **31(10A)** (2004) 138-139.

# A Ranking Approach with Inclusion Measure in Multiple-Attribute Interval-Valued Decision Making

Hong-Ying Zhang and Ya-Juan Su

Institute for Information and System Sciences, Faculty of Science,
Xi'an Jiaotong University, Xi'an, Shaan'xi 710049, P.R. China
zhyemily@mail.xjtu.edu.cn, yiyi1806@126.com

**Abstract.** This paper first presents a brief survey of the existing works on comparing and ranking any two interval numbers and then, on the basis of this, gives the inclusion measure approach to compare any two interval numbers. The monotonic inclusion measure is defined over the strict partial order set proposed by Moore and illustrate that the possibility degrees in the literature are monotonic inclusion measures defined in this paper; Then a series of monotonic inclusion measures are constructed based on t-norms. Finally, we give illustrations by using the monotonic inclusion measures and gain good results.

**Keywords:** Inclusion measure; Ranking of interval number; Multiple-attribute decision making.

## 1 Introduction

In reality, interval coefficients are frequently used to describe and treat imprecise and uncertain elements present in a decision problem. An interval number can be thought as an extension of the concept of a real number and also as a subset of the real line R [6]. An interval signifies the extent of tolerance that the parameter can possibly take. In the formulation of realistic problems, Set of intervals may appear as coefficients in the inequality (or equality) constraints of an optimization problem or in the selection of best alternative in a decision making problem [8]. Consequently, the comparison and ranking of any two interval numbers is one key question.

Moore [6] studied the arithmetic of interval numbers first and gave two transitive order relations defined over intervals one as an extension of $' <'$ on the real line and another as an extension of $' \subseteq'$, the concept of set inclusion. But these order relations cannot explain ranking between two partially or fully overlapping intervals. Ishibuchi and Tanaka [5] suggested two order relations $' \leq'_{LR}$ and $' \leq'_{MW}$. However, there exist a set of pair of intervals for which both the order relations do not hold. The literature ([1-4], [7], [8]) discussed degree to which one given interval is higher than another. Xu et.al [3] pointed out that the possibility degree was same to those proposed in [1] and [2] and gave the basic

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 411–418, 2007.

properties. Qiu et.al [4] introduced the inclusion measure approach to ranking of interval number over a partial order set defined in [5]. The inclusion measure defined in [4] should satisfy $I(I_1, I_2) = 1$ when $I_1 = [a_1, b_1] \leq I_2 = [a_2, b_2]$, where $I_1 = [a_1, b_1] \leq I_2 = [a_2, b_2] \Leftrightarrow a_1 \leq a_2, b_1 \leq b_2$. We think there exists inconsistency in the application of these inclusion measures. Such as $I_1 = [0.1, 0.6]$ and $I_2 = [0.2, 0.7]$ $I_3 = [0.59, 0.9]$ which satisfy $I(I_1, I_2) = 1, I(I_1, I_2) = 1$, but the difference between $I_2$ and $I_3$ is ignored evidently.

Inclusion measure is an important concept in the area of fuzzy sets. It is a generalization of the existed approximate reasoning, such as probability reasoning, fuzzy inference, evidential reasoning and so on [9]. It surfaces in knowledge discovery, tuning rules and determining the coincidence measure of rules in fuzzy logic. In this paper, a series of more rational inclusion measures are introduced to rank any two interval numbers.

In section 2, The monotonic inclusion measure is defined over the strict partial order set proposed by Moore [6]; The possibility degrees in paper [1], [2] and [3] prove to be monotonic inclusion measures; In section 3, a series of monotonic inclusion measures are constructed based on t-norms. We give illustrations by using the monotonic inclusion measures and gain good results in section 4. section 5, the conclusion.

## 2 Preliminaries

By an implicator we mean a function $\mathscr{I} : I^2 \to I$ satisfying $\mathscr{I}(1, 0) = 0$ and $\mathscr{I}(1, 1) = \mathscr{I}(0, 1) = \mathscr{I}(0, 0) = 1$.

*Remark 1.* It is easy to see that $\mathscr{I}(\alpha, 1) = 1$ for all $\alpha \in I$ when $I$ is a left monotonic implicator, and if $I$ is right monotonic then $\mathscr{I}(0, \alpha) = 1$ for all $\alpha \in I$.

Some axioms have been postulated by Smets and Magrez [12] as axiomatically appropriate for an implicator. Such as Hybrid Monotonicity: $\forall(x, y) \in [0, 1]^2$, $\mathscr{I}(., y)$ is decreasing, yet $\mathscr{I}(x, .)$ is increasing; Confinement Principle(*CP* principle): $\forall(x, y) \in [0, 1]^2, x \leq y \iff \mathscr{I}(x, y) = 1$ and Border Principle: $\forall x \in [0, 1], \mathscr{I}(1, x) = x$.

An R-implicator [10](residual implicator) based on a left-continuous t-norm $\mathscr{T}$ if for every x, y $\in [0, 1]$, $\mathscr{I}(x, y) = \sup\{\gamma \in [0, 1], \mathscr{T}(x, \gamma) \leq y\}$. The well-known $R$-implicators are the Łukasiewicz implicator $\mathscr{I}_L(x, y) = \min(1, 1-x+y)$, based on $\mathscr{T}_L$, the Gödel implicator $\mathscr{I}_G(x, y) = 1$ for $x \leq y$ and $\mathscr{I}_G(x, y) = y$ elsewhere, based on $\mathscr{T}_M$, and the Gaines implicator $\mathscr{I}_P(x, y) = 1$ for $x \leq y$ and $\mathscr{I}_P(x, y) = \frac{y}{x}$ elsewhere, based on $\mathscr{T}_P$.

**Proposition 1.** [10] *Every R-implicator is a Hybrid Monotonic, Border and* CP *implicator.*

**Definition 1.** [9] *Let* $' \leq'$ *be a binary relation on a nonempty set X,* $(X, \leq)$ *is a partial order set if it satisfies the following conditions:*

*(1) Reflexivity:* $x \leq x, \forall x \in X$;
*(2) Antisymmetry:* $x \leq y, y \leq x, then x = y, \forall x, y \in X$;
*(3) Transitivity:* $x \leq y, y \leq z, then x = z, \forall x, y, z \in X$.

**Definition 2.** *Let $'<'$ be a binary relation on a nonempty set X, if $'<'$ is transitive, namely, $x < y, y < z \Rightarrow x < z$, then $(X, <)$ is called as strict partial order set.*

## 3 Monotonic Inclusion Measure

Zadeh[13] gave the definition of fuzzy set subhood first: $A$ is contained in $B$ if and only if $A(x) \leq B(x)$ for all $x \in X$. Kosko[14] argued that if the inequality hold for all just a few x, $A$ can be considered to be a subset of $B$ by some degree. He generalized Zadeh's definition by using a popular concept of fuzzy set inclusion which is a fuzzy analog of conditional probability: $|A \wedge B|/|A|$. A variation of this measure and its application appear in [14-17]. [18,19] gave the knowledge processing method for intelligent systems based on inclusion degree and rough set. At the same time, many authors argued the axioms (properties) which a reasonable fuzzy inclusion measure should satisfy. Sinha and Dougherty [20], the prime work of the axiomatic approach, offered axioms for fuzzy inclusion (subsethood) measure.

**Definition 3.** *(*Sinha-Dougherty[20]*) Let* I *be a mapping* I*: $\mathscr{F}(X) \times \mathscr{F}(X) \to [0,1]$, and* A, B *and* C *fuzzy sets in a universe* X*. The Sinha-Dougherty axioms imposed on I are as follows:*

***Axiom 1.*** $\mathrm{I}(A, B) = 1 \Leftrightarrow A \subseteq B$*, that is $A(x) \leq B(x)$, for all $x \in X$;*

***Axiom 2.*** $\mathrm{I}(A, B) = 0 \Leftrightarrow \exists x \in X$ *such that $A(x) = 1$ and $B(x) = 0$.*

***Axiom 3.*** $A \subseteq B \Rightarrow \mathrm{I}(B, C) \leq \mathrm{I}(A, C)$*, i.e. the operator has decreasing first partial mappings.*

***Axiom 4.*** $A \subseteq B \Rightarrow \mathrm{I}(C, A) \leq \mathrm{I}(C, B)$*, i.e. the operator has increasing second partial mappings.*

***Axiom 5.*** $\mathrm{I}(A,B)=\mathrm{I}(S(A),S(B))$ *where S is a $\mathscr{F}(X)\times\mathscr{F}(X)$ mapping defined by, for every $x \in X, S(A)(x) = A(s(x))$, s denoting an $X \to X$ mapping.*

***Axiom 6.*** $\mathrm{I}(A, B) = \mathrm{I}(coA, coB)$

***Axiom 7.*** $\mathrm{I}(A \cup B, C) = \min(\mathrm{I}(A, C), \mathrm{I}(B, C))$

***Axiom 8.*** $\mathrm{I}(A, B \cap C) = \min(\mathrm{I}(A, B), \mathrm{I}(A, C))$

The ninth axiom, $I(A, B \cup C) \geq \max(I(A, B), I(A, C))$, was indicated by Frago [21] that it is redundant because it is equivalent to axiom 4. Young [22] indicated that one loses much of the relative structure of fuzzy sets $A$ and $B$ by letting one point determine when $I$ is 0. They gave Axiom $2'$ as a substitute for Axiom 2.

**Axiom** $2'$**.** if $[1/2] \subseteq A$, then $I(A, A^c) = 0$ if and only if $A = X$;

**Definition 4.** [9] *Let $(X, \leq)$ is a partial order set for all $x, y \in X$, a inclusion measure is denoted by $\mathcal{I}(x, y)$ if it satisfies the following conditions:*

*(1) $0 \leq \mathcal{I}(x, y) \leq 1, \forall x, y \in X$;*

*(2) $x \leq y \Leftrightarrow \mathcal{I}(x, y) = 1$;*

*(3) if $x \leq y \leq z \in X$, then $\mathcal{I}(y, z) \leq \mathcal{I}(x, z)$ and $\mathcal{I}(z, x) \leq \mathcal{I}(z, y)$ hold.*

So inclusion degree is a measure on the partial order set, namely when $x \leq y$, then $S(x, y) = 1$, otherwise the degree of $x \leq y$ is given. So it is important than the partial relation.

Let $P([0,1])$ is the set of all the closed subset of [0,1], the strict partial order $\prec$ is defined as $I_1 = [a_1, b_1] \prec I_2 = [a_2, b_2] \Leftrightarrow b_1 \leq a_2$ by Moore [6]. It obvious that $(P([0,1]), \prec)$ is a strict partial order set.

Now we introduce the monotonic inclusion measure on $(P([0,1]), \prec)$ which possesses some of the axioms postulated by Sina and Dougherty [20].

**Definition 5.** *For all $I_1 = [a_1, b_1], I_2 = [a_2, b_2] \in (P([0,1]), \prec)$, $\mathcal{I}(I_1, I_2)$ is called as a monotonic inclusion measure on $(P([0,1]), \prec)$ if it satisfies the following conditions:*

*(1) $0 \leq \mathcal{I}(I_1, I_2) \leq 1, \forall I_1, I_2 \in (P([0,1]), \prec)$;*

*(2) $I_1 \prec I_2 \Leftrightarrow \mathcal{I}(I_1, I_2) = 1$;*

*(3) if $I_1 \prec I_2$, for all $I_3 = [a_3, b_3] \in (P([0,1]), \prec)$, $S(I_2, I_3) \leq S(I_1, I_3)$ and $S(I_3, I_1) \leq S(I_3, I_2)$ hold.*

The literature [3] mentioned that the possibility degrees in [1],[2] and [3] were same, so we just show one of them is monotonic inclusion measure.

**Lemma 1.** *If $\alpha, \beta, a, b$ are all positive number, then*

$$\frac{\alpha}{\alpha + a} \leq \frac{\beta}{\beta + b} \Leftrightarrow \alpha b \leq \beta a.$$

**Theorem 1.** *If $I_1 = [a_1, b_1], I_2 = [a_2, b_2] \in (P([0,1]), \prec)$ then*

$$p(I_1, I_2) = \frac{\max\{0, b_1 - a_1 + b_2 - a_2 - \max\{b_1 - a_2, 0\}\}}{b_1 - a_1 + b_2 - a_2} [2]$$

*is a monotonic inclusion measure over $(P([0,1]), \prec)$.*

*Proof.* See [3] for details about the proof of conditions (1) and (2), we argue the third condition holds as follows:

If $I_1 = [a_1, b_1] \prec I_2 = [a_2, b_2]$, for all $I_3 = [a_3, b_3]$

$$p(I_3, I_1) = \frac{\max\{0, b_1 - a_1 + b_3 - a_3 - \max\{b_3 - a_1, 0\}\}}{b_1 - a_1 + b_3 - a_3},$$

$$p(I_3, I_2) = \frac{\max\{0, b_2 - a_2 + b_3 - a_3 - \max\{b_3 - a_2, 0\}\}}{b_2 - a_2 + b_3 - a_3},$$

Since $I_1 \prec I_2 \Leftrightarrow b_1 \leq a_2$, namely $a_1 \leq b_1 \leq a_2 \leq b_2$. Then we have $b_3 - a_2 \leq b_3 - a_1$.

It follows that

1) if $b_3 - a_2 \leq 0$, then $p(I_3, I_2) = 1$, so $p(I_3, I_1) \leq p(I_3, I_2)$;

2) if $0 < b_3 - a_2 \leq b_3 - a_1$, then we have

$$p(I_3, I_1) = \frac{\max\{0, b_1 - a_3\}}{b_1 - a_1 + b_3 - a_3}, p(I_3, I_2) = \frac{\max\{0, b_2 - a_3\}}{b_2 - a_2 + b_3 - a_3}$$

If $b_1 - a_3 \leq 0$, then $p(I_3, I_1) = 0$, we have $p(I_3, I_1) \leq p(I_3, I_2)$. If $0 < b_1 - a_3 \leq b_2 - a_3$, then

$$p(I_3, I_1) = \frac{b_1 - a_3}{b_1 - a_3 + b_3 - a_1}, p(I_3, I_2) = \frac{b_2 - a_3}{b_2 - a_3 + b_3 - a_2}$$

Since $0 < b_3 - a_2 \leq b_3 - a_1, 0 < b_1 - a_3 \leq b_2 - a_3$, then $(b_1 - a_3)(b_3 - a_2) \leq (b_2 - a_3)(b_3 - a_1)$. By lemma 3.1, we have

$$p(I_3, I_1) = \frac{b_1 - a_3}{b_1 - a_3 + b_3 - a_1} \leq p(I_3, I_2) = \frac{b_2 - a_3}{b_2 - a_3 + b_3 - a_2}.$$

$p(I_2, I_3) \leq p(I_1, I_3)$ can be proved by the same way.

So $p(I_1, I_2)$ is a partial inclusion measure on $(P([0,1]), \prec)$.

**Lemma 2.** *If $a_1, b_1, a_2, b_2$ are positive real number, then*

$$\frac{a_1 - b_1}{a_1 + b_1} \geq \frac{a_2 - b_2}{a_2 + b_2} \Leftrightarrow a_1 b_2 \geq a_2 b_1.$$

**Theorem 2.** *If $I_1 = [a_1, b_1], I_2 = [a_2, b_2] \in (P([0,1]), \prec)$ then*

$$\mathcal{I}(I_1, I_2) = 0.5\left(\frac{b_2 - a_1 + a_2 - b_1}{|b_2 - a_1| + |a_2 - b_1|} + 1\right),$$

*is a monotonic inclusion measure on $(P([0,1]), \prec)$.*

*Proof.* (1)It's obvious that $0 \leq \mathcal{I}(I_1, I_2) \leq 1$;

(2) Since

$$\mathcal{I}(I_1, I_2) = 1 \Leftrightarrow 0.5\left(\frac{b_2 - a_1 + a_2 - b_1}{|b_2 - a_1| + |a_2 - b_1|} + 1\right) = 1,$$

we have $b_2 - a_1 \geq 0, a_2 - b_1 \geq 0$ which equals to $I_1 \leq I_2$;

(3) If $I_1 \prec I_2$, namely $a_1 \leq b_1 \leq a_2 \leq b_2$, then

$$\mathcal{I}(I_1, I_3) = 0.5\left(\frac{b_3 - a_1 + a_3 - b_1}{|b_3 - a_1| + |a_3 - b_1|} + 1\right), S(I_2, I_3) = 0.5\left(\frac{b_3 - a_2 + a_3 - b_2}{|b_3 - a_2| + |a_3 - b_2|} + 1\right);$$

It follows that

1) if $a_3 \geq b_1$, then $\mathcal{I}(I_1, I_3) = 1$, so we have $\mathcal{I}(I_1, I_3) \geq \mathcal{I}(I_2, I_3)$;

2) if $a_3 < b_1, b_3 \leq a_2$, then $\mathcal{I}(I_2, I_3) = 0 \leq \mathcal{I}(I_1, I_3)$;

3) if $a_3 < b_1, b_3 > a_2$, then

$$\mathcal{I}(I_1, I_3) = 0.5\left(\frac{b_3 - a_1 - (b_1 - a_3)}{b_3 - a_1 + b_1 - a_3} + 1\right), \mathcal{I}(I_2, I_3) = 0.5\left(\frac{b_3 - a_2 - (b_2 - a_3)}{b_3 - a_2 + b_2 - a_3} + 1\right);$$

Since $a_1 \leq b_1 \leq a_2 \leq b_2, a_3 < b_1, b_3 > a_2$, we have $0 < b_3 - a_2 \leq b_3 - a_1, 0 < b_1 - a_3 \leq b_2 - a_3$, then $(b_3 - a_2)(b_1 - a_3) \leq (b_3 - a_1)(b_2 - a_3)$. By Lemma 3.2, we have

$$\frac{b_3 - a_2 - (b_2 - a_3)}{b_3 - a_2 + b_2 - a_3} \leq \frac{b_3 - a_1 - (b_1 - a_3)}{b_3 - a_1 + b_1 - a_3}$$

Namely $\mathcal{I}(I_2, I_3) \leq \mathcal{I}(I_1, I_3)$; By the same way, we can prove $\mathcal{I}(I_3, I_1) \leq \mathcal{I}(I_3, I_2)$ is true. So $\mathcal{I}(I_1, I_2)$ is a monotonic inclusion measure.

## 4 Inclusion Measure Based on t-Norm

**Theorem 3.** *$\mathscr{I}(a,b)$ is a monotonic inclusion measure over the partial order set $([0,1],\leq)$ where $\mathscr{I}$ is a R-implicator.*

It is obvious by the properties of R-implicator.

**Theorem 4.** *Let $I_1=[a_1,b_1], I_2=[a_2,b_2]$, $\mathcal{I}_1(a,b)$ and $\mathcal{I}_2(a,b)$ are monotonic inclusion measure on $([0,1],\leq)$, $T$ is a t-norm on [0,1], then $\mathcal{I}(I_1,I_2) = T(\mathcal{I}_1(a_1,b_2),\mathcal{I}_2(b_1,a_2))$ is a monotonic inclusion measure on $(P([0,1]),\prec)$.*

*Proof.* (1)By the definition of t-norm, it is obvious that $0\leq I(I_1,I_2)\leq 1$;

(2) $\mathcal{I}(I_1,I_2)=1 \Leftrightarrow T(\mathcal{I}_1(a_1,b_2),\mathcal{I}_2(b_1,a_2))=1 \Leftrightarrow \mathcal{I}_1(a_1,b_2)=1, \mathcal{I}_2(b_1,a_2)=1 \Leftrightarrow a_1\leq b_2, b_1\leq a_2 \Leftrightarrow I_1\prec I_2$;

(3)If $I_1\prec I_2$, then $a_1\leq b_1\leq a_2\leq b_2$. For all $I_3=[a_3,b_3]$, then

$$\mathcal{I}(I_1,I_3)=T(\mathcal{I}_1(a_1,b_3),\mathcal{I}_2(b_1,a_3)), \mathcal{I}(I_2,I_3)=T(\mathcal{I}_1(a_2,b_3),\mathcal{I}_2(b_2,a_3))$$

Combined $a_1\leq a_2, b_1\leq b_2$ with the monotonicity of $\mathcal{I}_1$ and $\mathcal{I}_2$, we get $\mathcal{I}_1(a_2,b_3)\leq\mathcal{I}_1(a_1,b_3), \mathcal{I}_2(b_2,a_3)\leq\mathcal{I}_2(b_1,a_3)$. By the increasing monotonicity of t-norm, we have $\mathcal{I}(I_2,I_3)\leq\mathcal{I}(I_1,I_3)$.

By the same way, $\mathcal{I}(I_3,I_1)\leq\mathcal{I}(I_3,I_2)$. So $\mathcal{I}(I_1,I_2)=T(\mathcal{I}_1(a_1,b_2),\mathcal{I}_2(b_1,a_2))$ is a monotonic inclusion measure on $(P([0,1]),\prec)$.

According to Theorem 4, We can construct a series of inclusion measure of interval numbers taking different inclusion measures of real number and t-norms.

## 5 Illustration

We select a example about evaluation of five colleges in [11]. These five colleges are represented by $A_1, A_2, A_3, A_4$ and $A_5$ respectively. By integrated weight, we get the integrative interval evaluation value $d_1=[0.1890, 0.1976]$, $d_2=[0.2022, 0.2154]$, $d_3=[0.2021, 0.2112]$, $d_4=[0.1865, 0.1964]$, $d_5=[0.1888, 0.1983]$

Then we compute the comparing matrix [4] by $D=(\mathscr{I}(d_i,d_j))$ and get the decision value by $D_i=\Sigma_{j=1}^{5}\mathscr{I}(d_i,d_j)$.

Now we make the decision taking the monotonic inclusion measure defined by t-norm.

(1) Let $S_1=S_2=\mathcal{I}_L, T=T_M$, we get the ranking order: $A_4\prec_{0.9926}A_1\prec_{0.9912}A_5\prec_1 A_3\prec_{0.991}A_2$;

(2) Let $S_1=S_2=\mathcal{G}_L, T=T_M$, we get $A_4\prec_{0.1888}A_5\prec_{0.1890}A_1\prec_1 A_3\prec_{0.2022}A_2$;

(3) Let $S_1=S_2=\mathcal{I}_P, T=T_P$, we get $A_4\prec_{0.9613}A_5\prec_{0.9531}A_1\prec_1 A_3\prec_{0.9573}A_2$.

We get a decision that $A_2$ is the best, while $A_4$ is the worst by taking different inclusion measure. This decision is approximately consistent with the order

$A_4 \prec_{0.6} A_1 \prec_{0.5138} A_5 \prec_1 A_3 \prec_{0.5865} A_2$ computed by the inclusion measure in [3]. At the same time, we get that $A_1$ is approximately equal to $A_5$, because $S_{15} = 0.9912 = S_{51} = 0.9912$, $S_{15} = 0.1888 \approx S_{51} = 0.1890$ and $S_{15} = 0.9554 \approx S_{51} = 0.9531$ respectively. We get $S_{15} = 0.5138 \approx S_{51} = 0.4862$ by taking the inclusion measure in [3]. So we get approving decision by the monotonic inclusion measures of interval numbers defined in this paper.

## 6 Conclusion

A ranking approach with monotonic inclusion measures for interval-valued decisions has been proposed in this paper. A series of inclusion measures based on t-norm are proposed, which is convenient for the decision-maker's selection in different problems. In the future, we will deal with interval-valued information systems and interval-valued grey systems and so on by the monotonic inclusion measure.

## Acknowledgement

This work is supported by the National 973 Program of China(No.2002CB312200) and the National Natural Science Foundation of China (F030101-60574021).

## References

1. Facchinetti, G., Ricci, R. G., Muzzioli: Note on ranking fuzzy triangular numbers[J]. International Journal of Intelligent Systems 1998 (13) 613-622
2. Da, Q.-L., Liu, X.-W.: Interval Number Linear Programming and Its Satisfactory Solution,Systems Engineering-theory and Practice 19 (1999) 3-7
3. Xu, Z.-S., Da, Q.-L.: Possibility degree method for ranking interval numbers andits application, Systems Engineering-theory and Practice 18 (2003) 67-70
4. Quo-Fang Qiu,Huai-Zu Li: Measurement and Construct with Inclusion Degree for Priority of Interval Numbers, Operations Research and Management Science 12 (2003) 13-17
5. H.Ishibuchi, H.Tanaka: Multiobjective programming in optimization of the onterval objective function, European Journal of Operational Resarch, 48 (1990) 219-225
6. R.E. Moore: Method and Application of Interval Analysis, SIAM, Philadelphia,1979
7. Qiu, G.-F., Li,H.-Z., Xu, L.-D. and Zhang, W.-X.: A knowledge processing for intelligent systems based on inclusion degree[J]. Expert Systems, 4 (2000) 187-195
8. Sengupta, A., Pal, T. K.: On comparing interval numbers[J]. European Journal of Operational Research, 127 (2000) 28-43
9. Zhang, W.-X., LEUNG, Y.: The Uncertainty Reasoning Principles, Xi'an: Xi'an Jiaotong University Press,(1996a)
10. G.J. Klir, B.Yuan, Fuzzy Logic: Theory and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1995
11. Zhang, Q. et.al : A Ranking Approach for Interval Numbers in Uncertain Multiple Attribute Decision Making Problems, Systems Engineering-theory and Practice, 19 (1999) 129-132

12. Smets, P.,Magrez, P.: Implication in fuzzy logic, Int.J.Approximate reasoning, 1 (1987) 327-347
13. L.A. Zadeh: Fuzzy Sets, Informat. Control 8 (1965) 338-353
14. B.Kosko: Fuzzy entropy and conditioning, Inform.Sci.40 (1986) 165-174
15. S. Bodjanova: Approximation of fuzzy concepts in decision making, Fuzzy Sets and Systems 85 (1997) 23-29
16. V.G. Kaburlasos, V. Petridis: Fuzzy lattice neurocomputing (FLN): a novel connectionist scheme for versatile learning and decision daking by clustering,Internat.J.Comput.Appl. 4 (1997) 31-43
17. V. Petridis, V.G. Kaburlasos: Fuzzy lattice neural network (FLNN): a hybrid model for learning,IEEE Trans. Neural Networks 9 (1998) 877-890
18. Zhang, W.-X., Qiu, G.-F.: Uncertain Decision Making Based On Rough Sets, Tsinghua University Press, Beijing, 2005
19. Zhang, W.-X., Leung, Y., Wu, W.-Z.: Information Systems and Knowledge Discovery, Science Press,Beijing, 2003
20. D. Sinha, E.R. Dougherty: Fuzzication of set inclusion: theory and applications,Fuzzy Sets and Systems 55 (1993) 15-42
21. N.Frago: Morfologia matematica borrosa basada en operadores generalizados de Łukasiewicz: procesiamento de imagines, Ph.D. Thesis, Universidal publica de Navarra, 1996
22. V.R. Young: Fuzzy subsethood, Fuzzy Sets and Systems 77 (1996) 371-384

# Granulations Based on Semantics of Rough Logical Formulas and Its Reasoning

Qing Liu[1,2], Hui Sun[1], and Ying Wang[1]

[1] Department of Computer Science & Technology
Nanchang Institute of Technology, Nanchang 330099, China
qliu_ncu@yahoo.com.cn
[2] Department of Computer Science & Technology
Nanchang University, Nanchang 330031, China

**Abstract.** In this article, the granulation based on the meaning of rough logical formula in a given information system $IS = (U, A)$ is proposed. Which is considered the granular formulas of form $m(F)$, where $F$ is a rough logical formula on $IS$. Relative properties of the granulations are discussed. Deductive reasoning of the granulations and $\lambda$-granular resolution strategies are also studied in this article. The practicability of the granulations will offer the new idea for studying meaning of classical logic and the meaning of other nonstandard logic. It could also be a theoretical development for granular computing.

**Keywords:** Semantics of rough logical formula, granulation, $\lambda$-inclusion, $\lambda$-closeness,granular reasoning.

## 1 Introduction

Successful applications of Rough Set theory show that the rough sets proposed by Pawlak are successful, significant and contributive. However rough sets based on indiscernibility relation are extended necessarily [1 − 4, 17, 18]. The indiscernibility relation we defined is difficult in some specialization area, but a binary relation or general relation is easily defined. So the granulations based on the binary relation or general relation are also easily generated [9, 10, 16] . Therefore, the research of granular computing is proposed [4, 17, 18]. We may also see that granulations based on the meaning of rough logical formulas on $IS$ could hopefully be the theoretical development of granular computing studying

Proposed granulation based on the meaning of rough logical formulas is thought of as granular formulas of form $m(F)$, where $F$ is a rough logical formula on $IS$. We also discuss Skolem granular clause form, granular deductive reasoning and $\lambda$-granular resolution strategies in this article. We have discussed related properties of the granulations derived from rough logical formulas.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 419–426, 2007.

## 2 Granulations Based on the Meaning of Rough Logical Formulas

Rough logic (RL) based on Rough Sets is thought of as a nonstandard logic defined on $IS$. Defined domain of formulas in the logic is considered as the universe $U$ in $IS$. Predicates in the logic are considered as attributes in the attribute set $A$ [16, 19 – 22].

Let $I_R$ be an interpretation of rough logical formula $F$, and $u_R$ be an assignment symbol to individual variable occurring in $F$. $T_{I_R u_R}$ be an united assignment symbol to each constant, variable, function and predicate occurring in $F$, that is,

1. If the term $\tau$ occurring in $F$ is a constant, then $T_{I_R u_R}(F) = I_R(\tau) = e$.
2. If the term $\tau$ occurring in $F$ is a variable, then $T_{I_R u_R}(F) = u_R(\tau) = e$.
3. If the term $\tau$ occurring in $F$ is a function symbol of form $\pi(\tau_1, \cdots, \tau_n)$, then $T_{I_R u_R}(F) = I_R(\tau) = g(x_1, \cdots, x_n)$, where $g$ is a given function symbol defined on $IS$.
4. If the term $\tau$ occurring in $F$ is a predicate symbol of form $\theta(\tau_1, \cdots, \tau_n)$, then $T_{I_R u_R}(F) = I_R(\tau) = P(x_1, \cdots, x_n)$, where $P$ is a relation symbol defined on $IS$.

The truth value of formula in the logic is denoted by $T_{I_R u_R}(F)$, which is defined as follows:

1. $T_{I_R u_R}(F) =\mid m(F) \mid / \mid U \mid$.
2. $T_{I_R u_R}(\neg F) = 1 - T_{I_R u_R}(F)$.
3. $T_{I_R u_R}(F_1 \vee F_2) \geq\mid (m(F_1) \cup m(F_2) \mid / \mid U \mid$.
4. $T_{I_R u_R}(F_1 \wedge F_2) \leq\mid (m(F_1) \cap m(F_2) \mid / \mid U \mid$.

In particular, the granulation is of the form $m(F)$, called granular formula. Value of the granular formula is a set defined on $2^U$. The value is derived from rough logical formula $F$, hence, it depend on the properties of the formulas. The operation symbols of granular formulas of the form $m(F)$ are the inclusion symbol $\propto_\lambda$ to a degree $\lambda$ and the closeness symbol $\infty_\lambda$ to a degree $\lambda$ besides operation symbols in classical set theory [23 – 26].

## 3 Inclusion and Closeness of Granular Formulas

**Definition 1.** *(Inclusion) Let $\varphi$ and $\psi$ be rough logical formula on $IS$. The granular formula $m(\varphi)$ is included in granular formula $m(\psi)$ to degree at least $\lambda$. Formally:*

$$\propto_\lambda (m(\varphi), m(\psi)) = \begin{cases} Card(m(\varphi) \cap m(\psi))/Card(m(\varphi)) & m(\varphi) \neq \emptyset \\ 1 & m(\varphi) = \emptyset \end{cases} \quad (1)$$

**Definition 2.** *(Closeness) Let $\varphi$ and $\psi$ be rough logical formula. The granulation $m(\varphi)$ is close to granulation $m(\psi)$ to degree at least $\lambda$. Formally, it is defined as follows:*

$$| T_{I_{IS}u_{IS}}(\varphi) - T_{I_{IS}u_{IS}}(\psi) | < 1 - \lambda \wedge m(\varphi) \propto_\lambda m(\psi) \wedge m(\psi) \propto_\lambda m(\varphi) \quad (2)$$

*for short denoted by $\propto_\lambda(m(\varphi), m(\psi))$, where:*

1. *$\propto_\lambda$ is called* $\lambda$-closeness *relation, to have $m(\varphi) \quad \propto_\lambda \quad m(\psi)$,*
2. *$T_{I_{IS}u_{IS}}$ is the united assignment symbol defined by*

$$T_{I_{IS}u_{IS}}(\varphi) = \alpha \quad (3)$$

where $\alpha \in [0, 1]$,a real number. $I_{IS}$ is an interpretation symbol of granular formula $m(\varphi)$ on $IS$, and $u_{IS}$ is an evaluation symbol to individual variable occurring in granular formula on $IS$ (to see [24, 30, 31]).

**Definition 3.** *(Operations) Let $m(\varphi)$ and $m(\psi)$ be two granular formulas, the operations of them with respect to usual logical connectives $\neg$, $\vee$, $\wedge$, $\rightarrow$ and $\leftrightarrow$ in the rough logical formula are defined as follows* [2, 4 − 8, 20, 32]*:*

1. *$m(\neg\varphi) = U - m(\varphi)$;*
2. *$m(\varphi \vee \psi) = m(\varphi) \cup m(\psi)$;*
3. *$m(\varphi \wedge \psi) = m(\varphi) \cap m(\psi)$;*
4. *$m(\varphi \rightarrow \psi) = m(\neg\varphi) \cup m(\psi)$;*
5. *$m(\varphi \leftrightarrow \psi) = (m(\neg\varphi) \cap m(\neg\psi)) \cup (m(\psi) \cap m(\varphi))$.*

For $\forall\varphi \in RL_{IS}$, value of $m(\varphi)$ is a subset in $U$. Which is defined as follows:

$$m(\varphi) = \{x \in U : x \mid\approx_{IS} \varphi\} \quad (4)$$

Where $\mid\approx_{IS}$ is a degree satisfiability symbol in $IS$ [12 − 15, 19, 25, 28, 32].

## 4 Properties of the Granular Formulas

The granulations based on the meaning of rough logical formulas on $IS$ have following properties [24, 25, 27, 32]:

1. **Identity**
   For $\forall F \in RL_{IS}$, $m(F) \quad \propto_\lambda \quad m(F)$;
2. **Symmetry**
   For $\forall F_1, F_2 \in RL_{IS}$, $(m(F_1) \quad \propto_\lambda \quad m(F_2)) \rightarrow (m(F_2) \quad \propto_\lambda \quad m(F_1))$;
3. **Transitivety**
   For $\forall F_1, F_2, F_3 \in RL_{IS}$,
   $((m(F_1) \quad \propto_\lambda \quad m(F_2)) \wedge (m(F_2) \quad \propto_\lambda \quad m(F_3))) \rightarrow (m(F_1) \quad \propto_{\lambda'} \quad m(F_3))$,
   where $\lambda^{'} = 2\lambda - 1, \lambda^{'}, \lambda \in [0, 1]$;
4. **Modus Ponens**
   For $\forall F_1, F_2 \in RL_{IS}$,
   $((m(F_1 \rightarrow F_2) \quad \propto_\lambda \quad U) \wedge (m(F_1) \quad \propto_\lambda \quad U)) \rightarrow ((m(F_1) \cap m(F_1 \rightarrow F_2))$
   $\propto_\lambda \quad m(F_2) \quad \propto_\lambda \quad m(F_1 \rightarrow F_2))$;

5. **Absorbance laws**
   $m(F_1) \cap (m(F_1) \cup m(F_2)) \quad \infty_\lambda \quad m(F_1)$;
   $m(F_1) \cup (m(F_1) \cap m(F_2)) \quad \infty_\lambda \quad m(F_1)$;
6. **Forever true**
   $m(F) \cup \neg m(F) \quad \infty_\lambda \quad U$, where $U$ is the universe of objects;
7. **Forever false**
   $m(F) \cap \neg m(F) \quad \infty_\lambda \quad \emptyset$, where $\emptyset$ is an empty;
8. **Substitution**
   For $\forall \alpha, \beta \in RL_{IS}$,
   $(m(\alpha) \quad \infty_\lambda \quad m(\beta)) \rightarrow (m(P(\alpha)) \quad \infty_\lambda \quad m(P(\beta)))$,
   where $\alpha$,$\beta$ may be a constant, variable, function item or well-formed formula, $\lambda \in [0, 1]$.
9. **Special properties**
   Special properties of atomic granules based on the meaning of rough logical formulas defined on $IS$.
   (1). $m(a_v) \cap m(a_u) = \emptyset$, where $a \in A$, $v, u \in V_a$, and $v \neq u$.
   (2). $\bigcup_{v \in V_a} m(a_v) = U$, for each $a \in A$.
   (3). $\neg m(a_u) = \bigcup_{v \in V_a} m(a_v)$, for each $a \in A$, $v \neq u$.

## 5 Deductive Proof of Granular Formulas

We discuss the reasoning technique called granular deduction. It is similar to the deductive technique in usually logic [26, 27].

***Example 1.*** We will show that

$$m(((P \vee Q) \wedge \neg(\neg P \wedge \neg(Q \wedge R))) \vee \neg(P \vee Q) \vee \neg(P \vee R)) \quad \infty_\lambda \quad U \qquad (5)$$

Where $P, Q, R \in RL_{IS}$, $\lambda \in [0, 1]$, $U$ is the universe of objects.

**Proof**

1. $m(((P \vee Q) \wedge \neg(\neg P \wedge \neg(Q \wedge R))) \vee \neg(P \vee Q) \vee \neg(P \vee R)) \quad \infty_\lambda \quad m(((P \vee Q) \wedge (P \vee (Q \wedge R))) \vee \neg(P \vee Q) \vee \neg(P \vee R))$, by De Morgan's Laws.
2. $m(((P \vee Q) \wedge (P \vee (Q \wedge R))) \vee \neg(P \vee Q) \vee \neg(P \vee R)) \quad \infty_\lambda \quad m(((P \vee Q) \wedge ((P \vee Q) \wedge (P \vee R))) \vee \neg((P \vee Q) \wedge (P \vee R)))$, by distributive laws,De Morgan's Laws.
3. $m(((P \vee Q) \wedge ((P \vee Q) \wedge (P \vee R))) \vee \neg((P \vee Q) \wedge (P \vee R))) \quad \infty_\lambda \quad m((((P \vee Q) \wedge (P \vee R))) \vee \neg((P \vee Q) \wedge (P \vee R)))$, by $A \wedge (A \wedge B) = A \wedge B$.
4. $m((((P \vee Q) \wedge (P \vee R))) \vee \neg((P \vee Q) \wedge (P \vee R))) \quad \infty_\lambda \quad U$, by forever true.

**Definition 4.** Let $F$ be a rough logical formula on $IS$, it could be equivalently transformed into a Skolem standard form $F = C_1 \wedge \cdots \wedge C_m$, where each $C_i$ is a disjunction or set of atoms or negation of them [19, 27, 29, 31, 32]. By the granular operation definition (3) in the above, having $m(F) = m(C_1) \cap \cdots \cap m(C_m)$, each $m(C_i) = m(L_{i_1}) \cup \cdots \cup m(L_{i_k})$, $i = 1, \cdots, m$, we call it Skolem granular clause form [29, 31].

**Definition 5.** Consider ground granular clauses $m(C_1)$ and $m(C_2)$ specified by $m(C_1) : m(C_1') \cup m(a_v)$ and $m(C_2) : m(C_2') \cup m(b_u)$ respectively. The resolvent of $m(C_1)$ and $m(C_2)$, $GR(m(C_1), m(C_2))$, is defined as follows: If the ground granular atoms $m(a_v)$ in $m(C_1)$ and $m(b_u)$ in $m(C_2)$ are a complement granular literal pair [25, 29, 31, 32] in the granulations, then resolvent of $m(C_1)$ and $m(C_2)$ is defined as follows:

$$\frac{C_1 : m(C_1') \cup m(a_v) \quad C_2 : m(C_2') \cup m(b_u)}{C : m(C_1') \cup m(C_2')} \tag{6}$$

Namely, we have $GR(m(C_1), m(C_2)) = m(C_1') \cup m(C_2')$.

## 6 λ-Resolution Strategies in the Granulations

**Definition 6.** Let $m(L_1)$ and $m(L_2)$ be granular literal, where $m(L_1)$ is close to $U$ to degree at least $\lambda$, $m(L_2)$ is close to $U$ to degree at most $1-\lambda$, if $\lambda \geq 0.5$, $T_{IS u_{IS}}(m(L_1)) > \lambda$ and $T_{IS u_{IS}}(m(L_2)) \leq 1-\lambda$; Or $m(L_1)$ is close to $U$ to degree at most $\lambda$, $m(L_2)$ is close to $U$ to degree at least $1-\lambda$, if $\lambda < 0.5$, $T_{IS u_{IS}}(m(L_1)) < \lambda$ and $T_{IS u_{IS}}(m(L_2)) \geq 1-\lambda$, where $L_1$ and $L_2$ are any description on $IS$ or atom in $RL$, such as $L_1 = a_v$, $L_2 = b_u$, then $m(L_1)$ and $m(L_2)$ is called as $\lambda$-complement granular literal pair [24, 31].

**Definition 7.** Let $m(C_1)$ and $m(C_2)$ be without common variable granular clause form, and $m(L_1)$ in $m(C_1)$ and $m(L_2)$ in $m(C_2)$ be $\lambda$-complement granular literal, then $\lambda$-resolvent of $m(C_1)$ and $m(C_2)$ is defined as follows:

$$GR_\lambda(m(C_1), m(C_2)) = (m(C_1) - m(L_1)) \cup (m(C_2) - m(L_2)) = m(C_1') \cup m(C_2') \tag{7}$$

**Where** $m(C_1') = m(C_1) - m(L_1)$, $m(C_2') = m(C_2) - m(L_2)$

**Table 1.** Information Table

| U | a | b | c | d | e |
|---|---|---|---|---|---|
| 1 | 5 | 4 | 0 | 1 | 1 |
| 2 | 3 | 4 | 0 | 2 | 1 |
| 3 | 3 | 4 | 0 | 2 | 2 |
| 4 | 0 | 2 | 0 | 1 | 2 |
| 5 | 3 | 2 | 1 | 2 | 2 |
| 6 | 5 | 2 | 1 | 1 | 0 |

***Example 2.*** Let $IS = (U, A, V, f)$ be an information system, as show on information table 1 in the above. We may construct a granular formula based on the meaning of rough logical formula on $IS$ [25 − 29, 31, 32]. We extract the formula $\varphi \in RL_{IS}$ as follows:

$$\varphi(a_5, b_2, b_4, c_0, \neg e_0) = (a_5 \vee b_4) \wedge b_2 \wedge (c_0 \vee \sim e_0) \tag{8}$$

Formula (8) may be written as the following granular formula:

$$m(\varphi(a_5, b_2, b_4, c_0, \neg e_0)) = (m(a_5) \cup m(b_4)) \cap m(b_2) \cap (m(c_0) \cup m(\neg e_0)) \quad (9)$$

This is a granular clause form, where each intersection item is a granular clause. By Definition 5, the ground granular clause form of the granular formula is defined as follows:

$$m(\varphi(a_5, b_2, b_4, c_0, \neg e_0)) = (a_5^{\{1,6\}} \cup b_4^{\{1,2,3\}}) \cap b_2^{\{4,5,6\}} \cap (c_0^{\{1,2,3,4\}} \cup \neg e_0^{\{2,3,4,5\}}) \quad (10)$$

where each item is a ground granular clause. When $\lambda$ is defined as 0.6, obviously, by definition 7, $a_5^{\{1,6\}}$ and $c_0^{\{1,2,3,4\}}$ is a $\lambda$-complement ground granular literal pair. So, the resolvent $GR_\lambda(m(C_1), m(C_2))$ of $a_5^{\{1,6\}} \cup b_4^{\{1,2,3\}}$ in $m(C_1)$ and $c_0^{\{1,2,3,4\}} \cup \neg e_0^{\{2,3,4,5\}}$ in $m(C_2)$ is defined as follows:

$$GR_\lambda(m(C_1), m(C_2)) = (a_5^{\{1,6\}} \cup b_4^{\{1,2,3\}} - a_5^{\{1,6\}}) \cup (c_0^{\{1,2,3,4\}} \cup \neg e_0^{\{2,3,4,5\}} - c_0^{\{1,2,3,4\}}) \quad (11)$$

Hence, the formula (10) could be rewritten as

$$(b_4^{\{1,2,3\}} \cup \neg e_0^{\{1,2,3,4,5\}}) \cap b_2^{\{4,5,6\}} \quad (12)$$

In face, when $\lambda = 0.6$, $a_5^{\{1,6\}}$ and $\neg e_0^{\{1,2,3,4,5\}}$ is also a $\lambda$-complement ground granular literal pair, hence the resolvent $GR_\lambda(m(C_1), m(C_2))$ by definition 7 could be obtained as follows:

$$(b_4^{\{1,2,3\}} \cup c_0^{\{1,2,3,4\}}) \cap b_2^{\{4,5,6\}} \quad (13)$$

## 7 Conclusion

We define the granulation based on the meaning of rough logical formula on $IS$ [32]. Based on reference [32], we further proposed a $\lambda$-resolution strategies of granular formula in this article. So the content of this article is an extension with respect to the references [32].

The granular studying based on the meaning of rough logical formulas will offer a new idea for studying classical logic and nonstandard logic. Studying of the granulations of meaning based on rough logical formulas is also an extension of Rough Logic proposed by Pawlak [20]. The derived granulations from rough logical formulas are axiomatized to get the deductive systems of granulations. We could prove some relationships between granulations in the axiomatic systems. So the theorems and properties of the granulations derived from rough logical formulas could be proved in the systems and used in theoretical study of granular computing [25, 32].

Further work will be to study the derived granulations of the meaning based on other nonstandard logical formulas and classical logical formulas,to study the related properties and reasoning of the granulations.

## Acknowledgement

We would like to thank the support of Natural Science Fund of China (NSFC-60173054) and the Natural Science Fund of Jiangxi province (JXPNSF- 0311101) in China.

## References

1. Pawlak, Z., Rough Sets, Int. J. Inform. Comp. Sci., 11 (1982). 341-356.
2. Pawlak, Z., Rough Sets present State and Further Prospects, The Proceedings of Third International Workshop on Rough Sets and Soft Computing, Nov. 10-12,1994, 72-76.
3. Lin, T. Y., A Roadmap from Rough Set Theory to Granular Computing,LNAI 4062, The Proceedings of RSKT2006, by Springer, China, July 2006, 39-41.
4. Lin, T. Y., Granular Computing on Partitions, Coverings and Neighborhood Systems,The proceedings of International Forum on Theory of GrC from Rough Set Perspective (IFTGrCRSP2006), Journal of Nanchang Institute of Technology, Vol.25, No.2, Nanchang, China, July 2006, 1-7.
5. Pawlak, Z., Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht , 1991.
6. Skowron A., Rough Sets in KDD, Special invited Speaking, WCC 2000, in Beijing, Aug. 2000.
7. Skowron, A., Stepaniuk, J., Peters, J.F., Extracting patterns using information granules. In: Proceedings of International Workshop on Rough Set Theory and Granular Computing. Japan 2001, 135-142.
8. Skowron, A., Swiniarski, R., Information granulation and pattern recognition. In: Pal, S.K., Polkowski, L., Skowron, A. (eds): Rough Neurocomputing: Techniques for Computing with Words, Cognitive Technologies. Springer-Verlag, Berlin, 2003.
9. Lin, T.Y., Granular computing on binary relations I: Data mining and neighborhood systems. In: Skowron, A., Polkowski, L. (eds): Rough Sets in Knowledge Discovery. Physica-Verlag, Berlin 1998, 107-121.
10. Lin, T.Y., Granular computing on binary relations II: Rough set representations and belief functions. In: Skowron, A., Polkowski, L. (eds): Rough Sets in Knowledge Discovery. Physica-Verlag, Berlin, 1998.2 (2000) 113-124.
11. Skowron A., Synthesis of Adaptive Decision Systems from Experimental Data, In: A. Aamodi, J. Komorowski (eds.): Proc. Of the Fifth Scandinavian Conference on Artificial Intelligence (SCAI) 95, Tvondheim, NOrwaw, IOS Press., Amsterdam, 220-238
12. Liu, Q., Rough Sets and Rough Reasoning (Third) Press. Of Science, Beijing, 2005, (In Chinese).
13. Yao, Y.Y., Information granulation and rough set approximation. International Journal of Intelligence Systems 16, 2001, 87-104.
14. Yao, J.T., Yao, Y.Y., Induction of classification rules by granular computing. In: Proceedings of the International Conference on Rough Sets and Current Trends in Computing. Springer, Berlin. Philadelphia, 2002, 331-338.
15. Liu,Q., Jiang, F. and Deng, D.Y., Design and Implement for the Diagnosis Software of Blood Viscosity Syndrome Based on Hemorheology on GrC., Lecture Notes in Artificial Intelligence 2639, Springer-Verlag, 2003,413-420.

16. Lin, T.Y., From rough sets and neighborhood systems to information granulation and computing in Words. In: Proceedings of European Congress on Intelligent Techniques and Soft Computing (1997) 1602-1606.
17. Skowron A., Rough-Granular Computing,The proceedings of International Forum on Theory of GrC from Rough Set Perspective (IFTGrCRSP2006), Journal of Nanchang Institute of Technology, Vol.25, No.2, Nanchang, China, July 2006, 8-14.
18. Yao, Y.Y., Three Prospectives of Granular Computing,The proceedings of International Forum on Theory of GrC from Rough Set Perspective (IFTGrCRSP2006), Journal of Nanchang Institute of Technology, Vol.25, No.2, Nanchang, China, July 2006, 16-21.
19. Liu, Q., Liu, S.H. and Zheng, F., Rough Logic and Applications in Data Reduction, Journal of Software, Vol.12, No.3, March 2001, (In Chinese).
20. Pawlak, Z., Rough Logic, Bulletin of the Polish Academy of Sciences, Technical Sciences, Vol.35, No.5-6, 1987, 253-259.
21. Orlowska, E., A Logic of Indiscernibility Relation. In: E. Orlowska (ed.), Computation Theory, Lecture Notes in Computer Science 208, 1985, 177-186.
22. Orlowska, E., Logic of Nondeterministic Information, In: Lectuer Notes in Computer Science, 208, 1985, 469-473.
23. Skowron, A., Toward Intelligent Systems: Calculi of Information Granules, Bulletin of International Rough Set Society Vol.5, No.1/2, Japan, 2001,9-30.
24. Liu, Q. and Wang, Q.Y., Granular Logic with Closeness Relation and Its Reasoning, LNAI 3641, by Springer, Berlin, September 2005, 709-711.
25. Liu, Q. and Huang, Z.H., G-Logic and Its Resolution Reasoning, Chinese Journal of Computer, Vol.27,No.7, 2004, 865-873. (In Chinese).
26. Liu, Q., Granular Language and Its Reasoning, Data Mining and Knowledge Discovery: Theory, Tools, and Technology V, Proceeding of SPIE-The International Society for Optical Engineering, Orlando, Florida, USA, 21-22 April 2003, 279-287.
27. Lin, T.Y., Liu, Q., First order rough logic I: Approximate reasoning via rough sets, Fundamenta Informaticae 2-3, 1996, 137-153.
28. Yao, Y.Y. and Liu, Q., A Generalized Decision Logic in Interval-Set-Valued Information table, Lecture Notes in AI 1711, Springer-Verlag, Berlin, 1999, 285-294.
29. Chang, C.L., Lee, R.C.T., Symbolic Logic and Machine Theorem Proving [M]. New York, Academic Press (1993).
30. Liu, Q.: The OI-resolution of operator rough logic. In: Proceedings of the International Conference on Rough Sets and Current Trends in Computing. Springer, Berlin (1998) 432-435.
31. Liu, X., Fuzzy Logic and Fuzzy Reasoning [M], Press. of Jilin University, Jilin, 1989. (In Chinese) .
32. Liu, Q., Sun H., Theoretical Study of Granular Computing, LNAI 4062, The Proceedings of RSKT2006, by Springer, China, July 2006, 93-102.

# A Categorial Basis for Granular Computing

Mohua Banerjee[1,⋆] and Yiyu Yao[2]

[1] Department of Mathematics and Statistics,
Indian Institute of Technology, Kanpur 208 016, India
mohua@iitk.ac.in
[2] Department of Computer Science, University of Regina,
Regina, Saskatchewan, Canada S4S 0A2
yyao@cs.uregina.ca

**Abstract.** The theory of granular computing is accorded a formal mathematical basis, by presenting its main features using a category-theoretic language. A category $\mathcal{C}_\mathcal{G}$ of granulations is proposed. It is shown how two main operations between granulations, viz. coarsening and refinement, can be expressed in terms of $\mathcal{C}_\mathcal{G}$-morphisms. Examples of some special subcategories of $\mathcal{C}_\mathcal{G}$ and their relationships are given.

**Keywords.** Category theory, granular computing, granulation coarsening and refinement.

## 1 Introduction

Granular computing is an emerging field of study on human-centered, knowledge-intensive problem solving using multiple levels of granularity [4,11,13,18,22]. One of its key notions is hierarchical structures that define levels of differing granularity [19,20]. The study of such structures will play a crucial role in the development of a theory of granular computing.

Category theory is a general mathematical theory of structures and systems of structures. With its powerful language, we can easily see the universal components of a family of structures of a given kind, and the interrelations of structures of different kinds [16]. It can be immediately applied to the study of the structures considered in granular computing. The main objective of this paper is to present such a category-theoretic basis of granular computing.

In building a category $\mathcal{C}_\mathcal{G}$ of granulations (cf. Section 2), we impose a minimum requirement. Specifically, a granulation is interpreted as a family of granules. Each granulation is an *object* and "similar" granulations constitute a category. *Morphisms* reflect interactions between granulations. They may be used to represent switchings between different levels of granularity. In Section 3, we consider two typical operations between granulations, *coarsening* and *refinement*, and express them through $\mathcal{C}_\mathcal{G}$-morphisms. Section 4 highlights a few existing categories of granulations to show that these are just subcategories of $\mathcal{C}_\mathcal{G}$.

⋆ The author acknowledges the support of the Department of Computer Science, University of Regina, Canada, and NSERC Canada (through a Discovery Grant to Yiyu Yao), during a visit to which the work was done.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 427–434, 2007.

Lin [9] studied a slightly different granular structure, consisting of an object space, a data space, a neighborhood system, and a concept space, and pointed out that a category of such "granular structures" can be constructed. The extra requirements are imposed to reflect the physical interpretations of granular structures in some real world applications. Such a category would fall under the scheme presented in this paper.

The results from the category-theoretic study provide an abstract level characterization of granular structures and their connections. Moreover, specific models of granular computing may be interpreted and understood in terms of the category introduced in this paper. This has the potential to unify a wide range of diverse results in the existing studies of granular computing.

## 2 The Category $\mathcal{C}_\mathcal{G}$ of Granulations

Basic definitions of category theory are briefly introduced as follows [6]. A category consists of a collection of entities called *objects*, and for each pair $(X, Y)$ of objects, a collection of *morphisms* with *domain* $X$ and *codomain* $Y$. If $f$ is such a morphism, we write $f : X \to Y$. Further, if $X, Y, Z$ are objects and $f : X \to Y$, $g : Y \to Z$ are morphisms, then there is a *composite* morphism (i.e., composition) $g \circ f : X \to Z$. The associativity of compositions holds: $h \circ (g \circ f) = (h \circ g) \circ f$, whenever both sides are defined. Finally, with each object $X$, is associated an *identity* morphism $i_X : X \to X$, such that the identity laws hold: for any morphism $f : X \to Y$, $i_Y \circ f = f = f \circ i_X$.

We now present the category $\mathcal{C}_\mathcal{G}$ of granulations.

**Definition 1.** *An* object *of $\mathcal{C}_\mathcal{G}$ is a non-empty set $G$ of granules.*

We put no more conditions on an object. However, one may note that $G$ could be a collection with some structure (e.g., an order). $G$ may also consist of granules on a particular domain $X$ (or a part of it), so that members of $G$ are subsets of $X$. In this case, we write the object as the pair $(X, G)$ (or $(X, G_A, A)$, $A \subseteq X$). One can further stipulate properties of $G$. For instance, the granules of $G$ could be assumed to *cover* $X$. In particular, they could partition $X$. In this context, the finest possible granularity on a domain $X$ would be given by $G := \{\{x\} : x \in X\}$. We denote this special object as $(X, Id_X)$, where $Id_X$ stands for "identity" on $X$.

**Definition 2.** *If $G$ and $G'$ are objects of $\mathcal{C}_\mathcal{G}$, a* morphism *$f : G \to G'$ is a map $f : G \to \mathcal{P}(G')$, where $\mathcal{P}(G')$ denotes the powerset of $G'$. For practical purposes, we shall assume that $f(g)$, $g \in G$, is finite. The* composition *$h \circ f : G \to G''$ of two morphisms $f : G \to G'$ and $h : G' \to G''$ is a map $h \circ f : G \to \mathcal{P}(G'')$ defined for each $g \in G$ as:*

$$h \circ f(g) := \bigcup h(f(g)).$$

*The* identity *morphism $id : G \to G$ on the object $G$ is the map $i : G \to \mathcal{P}(G)$ such that $id(g) := \{g\}$, for each $g \in G$.*

Morphisms reflect the process of *articulation* between granulations [8]. Every granule in an object $G$ is associated through a morphism with a (possibly empty) collection of granules in $G'$. It may be noted that a morphism forms a *binary neighborhood system*, in the terminology of [9].

*Example 1.* Let us consider two granulations $G_1, G_2$ on the set $\mathcal{R}$ of real numbers, representing, say, temperature [8]. For any $t \in \mathcal{R}$, we define a granule of $G_1$ as $g_t := \{t' \in \mathcal{R} : |t - t'| < 2\}$, i.e., $g_t$ is the open interval $(t-2, t+2)$. It is clear that $G_1$ has overlapping granules, and that it forms a cover of $\mathcal{R}$. On the other hand, $G_2$ is the collection of semi-open intervals $\{[10t, 10(t+1)) : t \in \mathcal{Z}\}$, where $\mathcal{Z}$ is the set of integers. $G_2$ thus forms a partition of $\mathcal{R}$, and has granules of temperatures in 10's, 20's etc. One may now define a morphism $f : (\mathcal{R}, G_1) \to (\mathcal{R}, G_2)$ in a natural way as follows: for $t \in \mathcal{R},\ g_t \in G_1$,

$$f(g_t) := \{g' \in G_2 : g_t \cap g' \neq \emptyset\}.$$

For any $t \in \mathcal{R}$, we observe that the granule $g_t = (t-2, t+2)$ of $G_1$ either intersects two "adjacent" granules $[10t', 10(t'+1)), [10(t'+1), 10(t'+2))$, or is completely contained in a granule $[10t', 10(t'+1))$ of $G_2$. So in the former case, by definition of $f$ above,

$$f(g_t) = \{[10t', 10(t'+1)), [10(t'+1), 10(t'+2))\}.$$

In the latter,

$$f(g_t) = \{[10t', 10(t'+1))\}.$$

A morphism can also be defined from $(\mathcal{R}, G_2)$ to $(\mathcal{R}, G_1)$: for $g' \in G_2$,

$$f'(g') := \{g \in G_1 : g' \cap g \neq \emptyset\},$$

which is a family of granules in $G_1$.

Morphisms could also represent an order between granulated views (i.e., objects of $\mathcal{C}_\mathcal{G}$) [19,20], as we shall see in the next section. Further, different conditions may be imposed on morphisms. For instance, if the granulations have some structure (such as an order among component granules), then the morphisms may be required to preserve the order [23]. In Example 1, it is not difficult to define order relations among granules in $G_1$ and among collections of granules in $G_2$ and to find that $f$ in fact preserves the order relations.

*Example 2.* Let us illustrate through a simple example, the definition of composition of morphisms. Given three granulations: $G_0 := \{g_0\},\ G_1 := \{g_1, g_2\},\ G_2 := \{g'_1, g'_2, g''_1, g''_2\}$, consider morphisms $f_1 : G_0 \to G_1$ and $f_2 : G_1 \to G_2$ defined as:

$$f_1(g_0) := G_1; \qquad f_2(g_1) := \{g'_1, g'_2\}, \quad f_2(g_2) := \{g''_1\}.$$

By Definition 2, the composition $f_2 \circ f_1 : G_0 \to G_2$ is the map from $G_0$ to $\mathcal{P}(G_2)$ such that

$$f_2 \circ f_1(g_0) = \bigcup f_2(f_1(g_0)) = \bigcup f_2(\{g_1, g_2\}) = f_2(g_1) \cup f_2(g_2) = \{g'_1, g'_2, g''_1\}.$$

Based on the definitions of objects and morphisms, we have a category of granulations.

**Proposition 1.** *$\mathcal{C}_{\mathcal{G}}$ forms a category.*

Various subcategories of $\mathcal{C}_{\mathcal{G}}$ would be formed by considering (as mentioned earlier) granulations with some structure, or only objects of the kind $(X, G)$, or objects $(X, G)$ where $G$ satisfies some properties.

## 3 Coarsening and Refining Granulations

In this section, we consider two typical operations involving granulations, viz. *coarsening* and *refinement*, and show that these are instances of $\mathcal{C}_{\mathcal{G}}$-morphisms.

An object $G_0$ may be progressively coarsened to give rise to new objects, say through a sequence:

$$G_0 \longrightarrow G_1 \longrightarrow ... \longrightarrow G_i \longrightarrow ... \tag{1}$$

where $G_{i+1}$ is a granulation *coarser* than $G_i$. For example, one may like to distinguish between a "monitor" and a "printer" at some point, but at another, wish to erase the distinction – clubbing both together and referring to a "computer". So, at each stage, some granules "collapse" into a single granule, while others remain "unaffected".

**Definition 3.** *An object $G'$ is* coarser *than $G$, provided there is a morphism $f : G \to G'$ such that,*

(i). *For at least one pair of distinct granules $g_1, g_2 \in G$, $f(g_1) = f(g_2) = \{g_{12}\}$, for some $g_{12} \in G'$;*
(ii). *If a granule $g$ of $G$ is not part of any such pair, $f(g) = \{g'\}$, for some $g' \in G'$ that is distinct from members of $f$-images obtained as in (i).*

*$G'$ and the granule $g_{12}$ are referred to as* coarsened transforms *of $G$ and the granules $g_1$ and $g_2$, respectively. $f$ is called a* coarsening *of $G$ into $G'$. A granule $g$ as in (ii) is said to be* unaffected *by $f$.*

The images under coarsening are always singletons. In the sequence (1), there would be a coarsening $f^i : G_i \to G_{i+1}$ at each stage $i$, ensuring that at least two granules in $G_i$ collapse into a single granule of $G_{i+1}$. Every other granule $g$ of $G_i$ is "fixed" – it is assigned a single distinct granule of $G_{i+1}$. This also ascertains that only coarsening takes place, and no granule is "split" in the process.

**Proposition 2.** *All objects $G_i$ constituting a sequence (1) are coarsened transforms of the object $G_0$.*

*Proof.* Take any three consecutive objects $G_i, G_{i+1}, G_{i+2}$ in the sequence. We claim that the composite $f^{i+1} \circ f^i : G_i \to G_{i+2}$ is a coarsening of $G_i$ into $G_{i+2}$. For this we observe that there are only two ways in which distinct granules in $G_i$, say $g_1^i, g_2^i$, may be transformed by $f^{i+1} \circ f^i$ into a $G_{i+2}$-granule $g^{i+2}$:

(a). $g_1^i, g_2^i$ are transformed by $f^i$ into a $G_{i+1}$-granule $g^{i+1}$, and $f^{i+1}(g^{i+1}) := \{g^{i+2}\}$.
(b). Both $g_1^i$, and $g_2^i$ are unaffected by $f^i$, but their (distinct) $f^i$-images are transformed by $f^{i+1}$ into $g^{i+2}$.

Condition (i) of Definition 3 is clearly satisfied. Condition (ii) also holds: due to (a) and (b) above, if a granule $g$ is unaffected by $f^{i+1} \circ f^i$, it must be unaffected by $f^i$, *and* the granule in $f^i(g)$ (a singleton) must be unaffected by $f^{i+1}$. It is then easy to see that $f^{i+1} \circ f^i(g) = \bigcup f^{i+1}(f^i(g)) = f^{i+1}(f^i(g))$ is distinct from all images of $f^{i+1} \circ f^i$-transformed pairs. □

Using Proposition 2, one obtains a subcategory $\mathcal{C}^c(G_0)$ of $\mathcal{C}_\mathcal{G}$, that has as objects, all the coarsened transforms of $G_0$. Morphisms are coarsenings between objects (if any), apart from the identity morphisms. All objects in sequences such as (1) become part of $\mathcal{C}^c(G_0)$. It is not surprising then, that we get

**Proposition 3.** *A granulation $G$ consisting of a single granule, is a terminal object of $\mathcal{C}^c(G_0)$, i.e., there can be only one $\mathcal{C}^c(G_0)$-morphism with any $\mathcal{C}^c(G_0)$-object as domain and $G$ as codomain.*

One can have special cases of Proposition 3, such as in the subcategory of partitions: if $G_0$ is a partition on some domain $X$, it may be coarsened until it collapses into the terminal object $X \times X$ on $X$.

*Refinement*, expectedly, is a process reverse to coarsening. Some objects are "split" into collections of granules, leading to finer granulations. Others remain unaffected. Moreover, one makes sure that no coarsening takes place during a refinement.

**Definition 4.** *An object $G'$ is* finer *than $G$, provided there is a morphism $f : G \to G'$ satisfying the following:*

(i). *There is some granule $g \in G$ such that $f(g)$ contains at least two distinct granules of $G'$;*
(ii). *If $g'$ in $G$ is not such a granule, $f(g') = \{g_0\}$, for some granule $g_0$ of $G'$ that is distinct from members of the $f$-images obtained as in (i). Further, $f$-images of all granules such as $g'$, are mutually distinct.*

*$f$ is called a* refinement *of $G$ into $G'$. $G'$ is also termed as a refinement of $G$.*

Successive refinements reflect an order among granulations, or a transition from a coarse-grained view to progressively finer views [8,19,20], leading to a hierarchy of granulations. Any such hierarchy could be represented by a sequence:

$$G_0 \longrightarrow G_1 \longrightarrow ... \longrightarrow G_i \longrightarrow ... \tag{2}$$

where each $G_{i+1}$ is a refinement of $G_i$.

A result similar to Proposition 2 can be obtained.

**Proposition 4.** *All objects $G_i$ in the sequence (2) are refinements of the object $G_0$.*

As in case of coarsening, we obtain a subcategory $\mathcal{C}^f(G_0)$ of $\mathcal{C}_\mathcal{G}$, that has as objects, all refinements of the object $G_0$. Thus, a hierarchy of granulations, whether based on refinement or coarsening, is just a subcategory of $\mathcal{C}_\mathcal{G}$.

We now consider a special subcategory $\mathcal{C}^f(X, G_0)$ of $\mathcal{C}^f(G_0)$, where the granulations are on some fixed domain $X$. Further, in an object $(X, G)$, (i) if a $G$-granule $g$ is split by a refinement $f$, $\bigcup f(g) = g$; (ii) if $g'$ in $G$ is not split, $f(g') = \{g'\}$. We arrive at the following proposition.

**Proposition 5.** *The finest granulation on $X$, $(X, Id_X)$, is a terminal object of $\mathcal{C}^f(X, G_0)$.*

Thus the category $\mathcal{C}^f(X, G_0)$ is, in effect, a network of hierarchies of granulated views, rooted at the object $(X, G_0)$ and terminating at $(X, Id_X)$.

## 4 Some Subcategories of $\mathcal{C}_\mathcal{G}$ and Their Interrelationships

A *functor* between two categories provides a correspondence between the respective collections of objects and morphisms. In the present context, functors would lead to a passage from one category of granulations into another. In [3], a "tree" of categories is formed with the help of functors. Each category constituting the tree has as its identity morphism, an "approximate" identity – ranging from the crisp identity $Id$, to various fuzzy indistinguishability relations [5,7,12,14,17,21]. Objects of these categories may be regarded as granulations, the component granules of which are formed on the basis of these indistinguishability relations. We demonstrate this formally for three of the constituent categories: through a reformulation, these are shown to be subcategories of $\mathcal{C}_\mathcal{G}$.

***SET***: The classical category $SET$ has sets as objects and functions between sets as morphisms [6]. Any set $X$ can be identified with the $\mathcal{C}_\mathcal{G}$-object $(X, Id_X)$. A function $f : X \to Y$ can be identified with the $\mathcal{C}_\mathcal{G}$-morphism $f' : (X, Id_X) \to (Y, Id_Y)$, where $f' : X \to \mathcal{P}(Y)$ is a map defined as: $f'(x) := \{f(x)\}$ for any $x \in X$. If we restrict the collection of $\mathcal{C}_\mathcal{G}$-morphisms with domain $(X, Id_X)$ and codomain $(Y, Id_Y)$ to contain functions $f : X \to \mathcal{P}(Y)$ such that $range(f) := \{\{y\}_{y \in Y}\}$, a converse identification can directly be made.

*Note*:The collection $\{\{x\}_{x \in X}\}$ of granules is isomorphic to the set $X$. Any $\mathcal{C}_\mathcal{G}$-morphism $f' : (X, Id_X) \to (Y, Id_Y)$ may be looked upon as a map $f' : X \to \mathcal{P}(Y)$.

***Set(E)***: The category of crisp equivalences $Set(E)$ [1], has objects of the form $(X, G_A, A)$, where $G_A$ is a partition on $A \subseteq X$. A morphism with domain $(X, G_A, A)$ and codomain $(Y, G_B, B)$ is a function $f$ from $G_A$ to $G_B$. We observe that the subcategory of $\mathcal{C}_\mathcal{G}$ consisting of objects $(X, G)$ is also a subcategory of $Set(E)$. Now $(X, G_A, A)$ is a $\mathcal{C}_\mathcal{G}$-object. In a line similar to the exercise done for $SET$, we restrict the collection of $\mathcal{C}_\mathcal{G}$-morphisms with domain $(X, G_A, A)$ and codomain $(Y, G_B, B)$ to contain functions $f : G_A \to \mathcal{P}(G_B)$ such that $range(f) := \{\{g\}_{g \in G_B}\}$. This enables an identification of $Set(E)$ as a subcategory of $\mathcal{C}_\mathcal{G}$.

***ROUGH***: The objects of *ROUGH* [2], a category of rough sets, are of the form $(X, G, A)$, where $G$ is a partition of $X$, and $A \subseteq X$. If $\overline{\mathcal{A}}$ and $\underline{\mathcal{A}}$ denote the collections of classes of $G$ contained in the upper and lower approximations of $A$, a morphism in *ROUGH* with domain $(X, G, A)$ and codomain $(Y, G', B)$ is a map $f : \overline{\mathcal{A}} \rightarrow \overline{\mathcal{B}}$ such that $f(\underline{\mathcal{A}}) \subseteq \underline{\mathcal{B}}$. Now this can be translated into the framework of $\mathcal{C}_{\mathcal{G}}$ as follows. Let us make the identification mentioned in the Note above, viz. for any set $S$, the collection $\{\{x\}_{x \in S}\}$ is isomorphic to $S$. We consider $\mathcal{C}_{\mathcal{G}}$-objects of the form $(X, G, L, U)$, where $G$ is a partition on $X$, and $L \subseteq U \subseteq G$. A morphism $f$ in this subcategory, with domain $(X, G, L, U)$ and codomain $(Y, G', L', U')$, is a map $f : G \rightarrow \mathcal{P}(G')$ such that (i) $range(f) := G'$, (ii) $f(U) \subseteq U'$, (iii) $f(L) \subseteq L'$, and (iv) $f(G \setminus U) := \{g'\}$, for some fixed granule $g' \in G' \setminus U'$. This subcategory of $\mathcal{C}_{\mathcal{G}}$ may be identified with *ROUGH*.

***SET*** $\subset$ ***Set(E)*** $\subset$ ***ROUGH***: For two categories $\mathcal{C}$ and $\mathcal{D}$, $\mathcal{C} \subset \mathcal{D}$ denotes that there is a functor which is an *embedding* of $\mathcal{C}$ into $\mathcal{D}$ i.e., an isomorphism between $\mathcal{C}$ and a subcategory of $\mathcal{D}$. To get the above-mentioned relationships, we define the required functors as follows. It is easy to see that any object $(X, Id_X)$ of *SET* can be assigned the $Set(E)$-object $(X, Id_X, X)$. On the other hand, any $Set(E)$-object $(X, G_A, A)$ can be assigned the object $(A, G_A)$ in *ROUGH*. Note that one takes $L = U = G_A$ for such an object. Images of objects being thus defined, the morphism images are decided in a natural way. It is clear that these correspondences suffice for the result that $SET \subset Set(E) \subset ROUGH$.

## 5 Conclusions

A category $\mathcal{C}_{\mathcal{G}}$ of granulations is proposed that is able to bring under its fold, a wide variety of granular structures as well as interactions between them. It is demonstrated how two typical operations involving granulations, viz. coarsening and refinement, may be reflected through morphisms of $\mathcal{C}_{\mathcal{G}}$. Some special categories of granulations are shown to be subcategories of $\mathcal{C}_{\mathcal{G}}$, and embeddings between them are pointed out.

The objects of $\mathcal{C}_{\mathcal{G}}$ may be generalized, by taking not only a single granulation as done here, but a collection of granulations. Morphisms may then be appropriately defined as well. A complex theory would form such a generalized category, taking together granulations on various domains such as those of agents, objects, times, locations etc. The granular structure of [9] could then be accounted for too.

The study of Section 4 may be extended to categories of granulations in which component granules are formed on the basis of other indistinguishability/similarity/nearness relations. The remaining categories in the tree of [3] may be considered for this purpose, or for instance, those defined in [10].

As we have seen in Section 3, hierarchies of granulations are subcategories of $\mathcal{C}_{\mathcal{G}}$. Just as morphisms represent an articulation between granulations, at the next level, functors would represent articulation between hierarchies of granulations. It may be interesting to explore for examples of such articulation, in specific models of granular computing.

## References


1. Banerjee, M. *A Categorial Approach to the Algebra and Logic of the Indiscernible*, Ph.D. Thesis, University of Calcutta, India, 1995.
2. Banerjee, M. and Chakraborty, M.K. A category for rough sets, *Foundations of Computing and Decision Sciences*, **18**, 167-180, 1993.
3. Banerjee, M. and Chakraborty, M.K. Foundations of vagueness: a category-theoretic approach, *Electronic Notes in Theoretical Computer Science*, **82**, 2003.
4. Bargiela, A. and Pedrycz W. *Granular Computing: an Introduction*, Kluwer Academic Publishers, Boston, 2002.
5. Eytan, M. Fuzzy sets: a topos-logical view, *Fuzzy Sets and Systems*, **5**, 47-67, 1981.
6. Goldblatt, R. *Topoi, the Categorial Analysis of Logic*, North Holland, 1984.
7. Higgs, D. A categorial approach to Boolean-valued set theory, preprint, 1973.
8. Hobbs, J. R. Granularity, *Proceedings of IJCAI 1985*, 432-435, 1995.
9. Lin, T.Y. Granular computing on binary relations I: data mining and neighbourhood systems, In: *Rough Sets in Knowledge Discovery*, A. Skowron and L. Polkowski (eds.), Physica-Verlag, 107-121, 1998.
10. Lin, T.Y. Qualitative fuzzy sets: homotopy and perception, In: *Proceedings of 2004 IEEE International Conference on Fuzzy Systems*, 665-668, 2004.
11. Lin, T.Y., Yao, Y.Y. and Zadeh, L.A. (Eds.) *Data Mining, Rough Sets and Granular Computing*, Physica-Verlag, Heidelberg, 2002.
12. Menger, K. Probabilistic theory of relations, Proceedings of Nat. Acad. Sci. U.S.A., **37**, 178-180, 1951.
13. Pedrycz, W. (Ed.) *Granular Computing: An Emerging Paradigm*, Physica-Verlag, Heidelberg, 2001.
14. Ruspini, E. Recent developments in fuzzy clustering, In: *Fuzzy Set and Probability Theory: Recent Developments*, R.R. Yager (ed.), Pergamon, 1982, 133-147.
15. Skala, H.J., Termini, S. and Trillas E. (Eds.) *Aspects of Vagueness*, D. Reidel, 1984.
16. Stanford Encyclopedia of Philosophy, Category theory, http://plato.stanford.edu/entries/category-theory/ (accessed November 30, 2006).
17. Trillas, E. and Valverde L. An inquiry into indistinguishability operators, In [15], 231-256.
18. Yao, Y.Y. Granular computing, *Computer Science (Ji Suan Ji Ke Xue)*, **31**, 1-5, 2004.
19. Yao, Y.Y. Perspectives of granular computing, *Proceedings of 2005 IEEE International Conference on Granular Computing*, Vol. 1, 85-90, 2005.
20. Yao, Y.Y. Three perspectives of granular computing, *Journal of Nanchang Institute of Technology*, **25**, 16-21, 2006.
21. Zadeh, L.A. Similarity relations and fuzzy orderings, *Information Sciences*, **3**, 177-200, 1971.
22. Zadeh, L.A. Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems, *Soft Computing*, **2**, 23-25, 1998.
23. Zhang, B. and Zhang, L. *Theory and Applications of Problem Solving*, North-Holland, Amsterdam, 1992.

# Granular Sets – Foundations and Case Study of Tolerance Spaces

Dominik Ślęzak[1] and Piotr Wasilewski[2]

[1] Infobright Inc.
218 Adelaide St W., Toronto, ON, M5H 1W7 Canada
[2] Faculty of Psychology, Warsaw University
Stawki 5/7, 00-183 Warsaw, Poland
slezak@infobright.com, piotr@psych.uw.edu.pl

**Abstract.** A novel approach to extend the notions of definability and rough set approximations in information systems with non-equivalence relations is proposed. The upper approximation is defined as set-theoretic complement of negative region of a given concept; therefore, it does not need to be definable. Fundamental properties of new approximation operators are compared with the previous ones reported in literature. The proposed idea is illustrated within tolerance approximation spaces. In particular, granulation based on maximal preclasses is considered.

**Keywords:** Rough Sets, Tolerance Relations, Granular Computing.

## 1 Introduction

Although rough sets were introduced and originally mainly studied for information systems with equivalence relations, corresponding to data tables with crisp, symbolic values [10,11,12], a lot of research has been devoted to other relations and value types as well [3,5,6,9,14,15,16,18,21]. As in many other theories, what is simple and straightforward for equivalence relations, requires much deeper study for more general cases, beginning with most natural extension onto tolerance relations [5,16], ending with non-deterministic, fuzzy relations between vaguely defined values [1,3,4]. Generalizations are needed both at the level of fundamental notions, like those of definability or approximations, and at the level of derivation of relations from actual data, using discretization, neighborhoods, application-specific analysis of missing values etc.

We approach the above challenges from both abstract-driven and data-driven perspectives. Firstly, we reconsider the well-known rough set approximation operators at the level of abstract families of granules – subsets of universe. Comparing to the previous approaches, we redefine the upper approximation as a set-theoretic complement of negative region. We show that such a change improves both interpretation and mathematical properties of the obtained rough set model. Secondly, we focus on the case study of tolerance relations, as an example that we may expect while dealing with real-life data. In particular, granulation based on maximal tolerance preclasses is analyzed. Propositions 2 and 3 illustrate our contribution from the two above-described perspectives.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 435–442, 2007.

## 2 Preliminaries: Information Systems and Rough Sets

Rough sets were introduced as a tool for analyzing information systems – formal counterparts of information tables, where rows are labeled by names of objects and columns – by names of attributes. An information system is a triple $S = \langle Ob, At, Val_a \rangle$ where $Ob$ is a set of objects, $At$ is a set of attributes, and each $Val_a$ is a value domain of an attribute $a \in At$, where $a : Ob \longrightarrow \mathcal{P}(Val_a)$ ($\mathcal{P}(Val_a)$ is a power set of $Val_a$). If $a(x) \neq \emptyset$ for all $x \in Ob$ and $a \in At$, then $S$ is total. If $card(a(x)) = 1$ for every $x \in Ob$ and $a \in At$, then $S$ is deterministic. Otherwise $S$ is indeterministic.

**Table 1.** Example of indeterministic information system

| | Height (cm) | Degrees | | Height (cm) | Degrees |
|---|---|---|---|---|---|
| $x_1$ | 196 | {BSc, MSc, PhD} | $x_6$ | 183 | {BA, MA} |
| $x_2$ | 174 | {BSc, BA, MA, PhD} | $x_7$ | 190 | {BSc} |
| $x_3$ | 173 | {BSc, MSc} | $x_8$ | 192 | {BSc, MA, PhD} |
| $x_4$ | 179 | {BA, MA, PhD} | $x_9$ | 187 | {BA, MA} |
| $x_5$ | 178 | {BSc} | $x_{10}$ | 184 | {BSc, MSc, PhD} |

According to Pawlak, knowledge is based on ability to discern objects [11,12,13,14]. In information systems, this ability is presented by *indiscernibility relation*. Let $S = \langle U, At, Val_a \rangle$ be an information system, $B \subseteq At$ and $x, y \in U$. Indiscernibility relation $ind(B)$ is a relation such that $(x, y) \in ind(B) \Leftrightarrow a(x) = a(y)$ for all $a \in B$. $ind(B)$ can be further analyzed from abstract perspective of *approximation spaces* [11,12] – ordered pairs $(U, R)$, where $R$ is an equivalence relation on an arbitrary set $U$ called the universe of discourse. Information system $S = \langle Ob, At, Val_a \rangle$ determines approximation spaces $(Ob, ind(B))$ where $B \subseteq At$. Originally, Pawlak called the equivalence classes of $ind(B)$ as *atoms* [12]. Subsets of $U$ which are unions of atoms are called *definable* (or *composed*). Otherwise they are called *rough* [11,12,14]. For $(U, R)$ and $X \subseteq U$, *lower* and *upper approximations* $X$ in $(U, R)$ are defined as follows

$$R_*(X) = \bigcup \{Y \in U_{/R} : Y \subseteq X\} \qquad R^*(X) = \bigcup \{Y \in U_{/R} : Y \cap X \neq \emptyset\}.$$

Each concept $X \subseteq U$ determines two special definable subsets of the universe: *positive region* $POS(X)$ consisting of equivalence classes contained in $X$ and *negative region* $NEG(X)$ consisting of equivalence classes disconnected with $X$ [11,12]. We can present these regions by means of approximation operators:

$$POS(X) := R_*(X) \qquad NEG(X) := R_*(X'),$$

where $X' = U \setminus X$ . One can also define a *boundary* of $X$:

$$BN(X) := U \setminus (R_*(X) \cup R_*(X')) = R^*(X) \setminus R_*(X).$$

**Table 2.** Well-known properties of $R_*$ and $R^*$ [3,11,12,15]

| | |
|---|---|
| 1a. $R_*(X) \subseteq X$ | 1b. $X \subseteq R^*(X)$ |
| 2a. $X \subseteq Y \Rightarrow R_*(X) \subseteq R_*(Y)$ | 2b. $X \subseteq Y \Rightarrow R^*(X) \subseteq R^*(Y)$ |
| 3a. $R_*(\emptyset) = \emptyset$ 4a. $R_*(U) = U$ | 3b. $R^*(\emptyset) = \emptyset$ 4b. $R^*(U) = U$ |
| 5a. $R_*(R_*(X)) = R_*(X)$ | 5b. $R^*(R^*(X)) = R^*(X)$ |
| 6a. $R_*(X \cap Y) = R_*(X) \cap R_*(Y)$ | 6b. $R^*(X \cap Y) \subseteq R^*(X) \cap R^*(Y)$ |
| 7a. $R_*(X) \cup R_*(Y) \subseteq R_*(X \cup Y)$ | 7b. $R^*(X) \cup R^*(Y) = R^*(X \cup Y)$ |
| 8a. $R_*(X) = R^*(R_*(X))$ | 8b. $R_*(R^*(X)) = R^*(X)$ |
| 9a. $R_*(X)' = R^*(X')$ | 9b. $R^*(X)' = R_*(X')$ |
| 10. $X$ is definable $\Leftrightarrow R_*(X) = X \Leftrightarrow R^*(X) = X \Leftrightarrow R_*(X) = R^*(X)$ | |
| 11. If $X$ or $Y$ are definable, then $R_*(X) \cup R_*(Y) = R_*(X \cup Y)$ and $R^*(X \cap Y) = R^*(X) \cap R^*(Y)$ | |

A set $X \subseteq U$ is rough if and only if $BN(X) \neq \emptyset$.[1] $NEG(X)$ can be interpreted as the set of elements certainly not belonging to $X$, while $R^*(X)$ – the set of elements possibly belonging to $X$. One can show that

$$R^*(X) \cap NEG(X) = \emptyset.$$

Otherwise we would have a problem (*paradox*) of overlapped regions: there would be objects which both possibly belong to $X$ and certainly do not belong to $X$.

## 3 General Approach: Granules and Granular Sets

In [14] Pawlak admitted that approximation spaces can be defined over arbitrary binary relations. Earlier, approximation operators based on tolerance relations were considered in [16], while [6,21] provide results for arbitrary reflexive relations. In the mentioned approaches, information atoms are defined generally as images $R(x) := \{y \in U : (x, y) \in R\}$ of elements $x \in U$ within approximation space $(U, R)$. Nowadays, information atoms, among others, are called *information granules*. Various forms of information granules were extensively discussed in literature (see also e.g. [1,7,13,17,18,22]).

When granules are analyzed at more abstract level, we can overlook their information ancestry – the way, in which they were derived from information systems, and consider them as purely abstract objects – the subsets of the universe. Such granules can be viewed as *knowledge granules* [13,22]. At this level, the family of granules of $U$ will be simply denoted by $Gr(U) \subseteq \mathcal{P}(U)$. A natural additional assumption will be that $U = \bigcup Gr(U)$, i.e. $Gr(U)$ covers $U$. Original Pawlak's definitions can be now generalized as follows:

**Definition 1.** *Let $U$ and $Gr(U) \subseteq \mathcal{P}(U)$, $U = \bigcup Gr(U)$, be given. For any $X \subseteq U$ we define* lower *and* upper approximation *operators as follows:*

$$Gr_*(X) := \bigcup\{Y \in Gr(U) : Y \subseteq X\} \quad Gr^*(X) := \bigcup\{Y \in Gr(U) : Y \cap X \neq \emptyset\}.$$

[1] But this is not a definition. – Further we keep formulating the definable sets as unions of atoms/granules, which is not necessarily equivalent to their empty boundaries.

*Subset $X \subseteq U$ is* granularly definable*, iff there is $\mathcal{B} \subseteq Gr(U)$ such that $X = \bigcup \mathcal{B}$. The family of all granularly definable sets is denoted by $Def_{Gr}(U)$.*

The new operators preserve the idea of definability of sets by means of atoms-granules. Regions $POS_{Gr}(X) := Gr_*(X)$ and $NEG_{Gr}(X) := Gr_*(X')$ are granularly definable. Since granules may overlap now, we can observe that:

$$Gr^*(X) \cap NEG_{Gr}(X) \neq \emptyset,$$

which was mentioned in the previous section as not present in the classical case. One can see that this is not actually a problem of defining positive and negative regions, but rather the one related to the upper approximation and boundary. We solve it by introducing a new upper approximation operator. Informally, we may call it as a *bited upper approximation* because we use granules in $NEG_{Gr}(X)$ to *bite* the overlapping parts of granules in $Gr^*(X)$.

**Definition 2.** *For $U$, $Gr(U) \subseteq \mathcal{P}(U)$, $U = \bigcup Gr(U)$, and $X \subseteq U$, we put:*

$$Gr_b^*(X) := Gr^*(X) \setminus NEG_{Gr}(X).$$

With no change to $POS_{Gr}(X)$ and $NEG_{Gr}(X)$, we obviously have $Gr_b^*(X) \cap NEG_{Gr}(X) = \emptyset$. Actually we get $Gr_b^*(X) = U \setminus NEG_{Gr}(X)$. We can also put:

$$BN_{Gr}(X) := Gr_b^*(X) \setminus Gr_*(X) = U \setminus (POS_{Gr}(X) \cup NEG_{Gr}(X)).$$

Our approach follows an idea that the most fundamental notions are the regions gathering positive and negative examples of the concepts. These regions are *generic* and should be kept as definable in any generalization of the rough set model. On the other hand, the boundary as *the rest*, as well as the upper approximation as a sum of the positive region and the boundary are *derivable* from those generic notions, and not necessarily definable any more. This is a difference with respect to the previous approaches, where there was a focus rather on definability of lower and upper approximations. Consequently, while $Gr^*(X)$ remains comparable to other operators in literature, $Gr_b^*(X)$ is entirely new.

In general case of $Gr(U)$, granular definability of $X$ does not imply that its boundary is empty. This is again because the elements of $Gr(U)$ may overlap within $U$. Therefore, we suggest strengthening definability as follows:

**Definition 3.** *Let $U$ and $Gr(U) \subseteq \mathcal{P}(U)$, $U = \bigcup Gr(U)$, be given. For any $X \subseteq U$, we say that $X$ is* granularly crisp*, iff both $X$ and $X'$ are granularly definable. The family of all granularly crisp sets is denoted by $Cri_{Gr}(U)$.*

We may say that a given concept is granularly crisp, if the sets of its positive and negative examples are definable. Obviously, for the classical case of equivalence relations those two above definitions are the same:

**Proposition 1.** *For arbitrary $U$ and $Gr(U) \subseteq \mathcal{P}(U)$ which forms partition of $U$, there is $Def_{Gr}(U) = Cri_{Gr}(U)$. For each $X \subseteq U$, there is $Gr_b^*(X) = Gr^*(X)$.*

For general case of $Gr(U)$, properties of the proposed model look as follows:

**Proposition 2.** *For arbitrary $U$ and $Gr(U) \subseteq \mathcal{P}(U)$ such that $U = \bigcup Gr(U)$, the properties in Table 3 hold. Besides the cases described by 10,10',11,11', one can construct counterexamples to equalities in 6a,6b,7a,7b,8a,8b in Table 3.*

**Table 3.** Properties of $Gr_*$ and $Gr_b^*$ for arbitrary $Gr(U) \subseteq \mathcal{P}(U)$, $U = \bigcup Gr(U)$

| | |
|---|---|
| 1a. $Gr_*(X) \subseteq X$ | 1b. $X \subseteq Gr_b^*(X)$ |
| 2a. $X \subseteq Y \Rightarrow Gr_*(X) \subseteq Gr_*(Y)$ | 2b. $X \subseteq Y \Rightarrow Gr_b^*(X) \subseteq Gr_b^*(Y)$ |
| 3a. $Gr_*(\emptyset) = \emptyset$ 4a. $Gr_*(U) = U$ | 3b. $Gr_b^*(\emptyset) = \emptyset$ 4b. $Gr_b^*(U) = U$ |
| 5a. $Gr_*(Gr_*(X)) = Gr_*(X)$ | 5b. $Gr_b^*(X) = Gr_b^*(Gr_b^*(X))$ |
| 6a. $Gr_*(X \cap Y) \subseteq Gr_*(X) \cap Gr_*(Y)$ | 6b. $Gr_b^*(X \cap Y) \subseteq Gr_b^*(X) \cap Gr_b^*(Y)$ |
| 7a. $Gr_*(X) \cup Gr_*(Y) \subseteq Gr_*(X \cup Y)$ | 7b. $Gr_b^*(X) \cup Gr_b^*(Y) \subseteq Gr_b^*(X \cup Y)$ |
| 8a. $Gr_*(X) \subseteq Gr_b^*(Gr_*(X))$ | 8b. $Gr_*(Gr_b^*(X)) \subseteq Gr_b^*(X)$ |
| 9a. $Gr_*(X)' = Gr_b^*(X')$ | 9b. $Gr_b^*(X)' = Gr_*(X')$ |
| 10. $X \in Def_{Gr}(U) \Leftrightarrow X = Gr_*(X)$ 10'. $X \in Cri_{Gr}(U) \Leftrightarrow Gr_*(X) = Gr_b^*(X)$ | |
| 11. $X, Y \in Def_{Gr}(U) \Rightarrow X \cup Y \in Def_{Gr}(U)$ (it also implies "=" in 7a) | |
| 11'. $X, Y \in Cri_{Gr}(U) \Rightarrow X \cap Y, X \cup Y \in Cri_{Gr}(U)$ ("=" in 6a,6b,7a,7b,8a,8b) | |

Besides the previously mentioned equation $Gr_b^*(X) \cap NEG_{Gr}(X) = \emptyset$, let us emphasize property 5b. An issue with many rough set extensions is that the upper approximations are not idempotent. It is also the case of $Gr^*$, i.e. we may get $Gr^*(X) \subsetneq Gr^*(Gr^*(X))$ and further $Gr^*(Gr^*(...(Gr^*(X)))) = U$, when iterating enough many times. Introducing $Gr_b^*(X)$ instead of $Gr^*(X)$ eliminates this problem and provides far more regular properties. In particular, the structure of $Cri_{Gr}(U)$ is interesting to study from an algebraic point of view.

## 4 Illustration: Rough Sets and Tolerance Relations

Tolerance relations are used in information systems in various ways, in case of, e.g.: numeric attributes, missing values, and in indeterministic information systems [1,3,4,5,8,9,16]. In such cases, equivalence indiscernibility relations are often too strict and inadequate. Let $S = \langle Ob, At, Val_a \rangle$ be given. For a numeric attribute $a \in At$ ($Val_a \subseteq \mathbb{R}$) and some parameter $\varepsilon \geq 0$, we can define relation $\rho(a, \varepsilon) \subseteq U \times U$ as $(x, y) \in \rho(a, \varepsilon) :\Leftrightarrow |a(x) - a(y)| \leq \varepsilon$. As another example, if $a \in At$ is indeterministic, then we can consider, e.g., $sim(a) \subseteq U \times U$ defined as $(x, y) \in sim(a) :\Leftrightarrow a(x) \cap a(y) \neq \emptyset$. One can surely imagine further examples for different attribute types and parameter settings. Further, for $B \subseteq At$, one can define tolerance as intersection of $\rho(a, \varepsilon)$, $sim(a)$, $ind(a)$, etc., for all $a \in B$.

Given understanding of how tolerances can be defined in information systems, we follow with recalling two approaches. The first one was introduced in [16] within the framework of *tolerance approximation spaces*. Here we refer to its basic version: Let tolerance $\tau \subseteq U \times U$ be given. For any $X \subseteq U$ we put:

$$\tau_*(X) := \{x \in U : \tau(x) \subseteq X\} \qquad \tau^*(X) := \{x \in U : \tau(x) \cap X \neq \emptyset\}.$$

$\tau_*$ and $\tau^*$ do not follow the idea of definability and crispness in terms of the unions of granules, as they consist of the *centers* of granules only. When putting $NEG_\tau(X) := \tau_*(X')$, we avoid the problem of overlapping regions, i.e. $\tau^*(X) \cap NEG_\tau(X) = \emptyset$. Operators $\tau_*$ and $\tau^*$ have also quite valuable properties when compared with those in Table 3. However, we have neither 5a nor 5b, i.e. there

are cases when $\tau_*(\tau_*(X)) \subsetneq \tau_*(X)$ and $\tau^*(X) \subsetneq \tau^*(\tau^*(X))$. There is no analogy of *biting* procedure introduced in Section 3 to get 5a or 5b for $\tau_*$ and $\tau^*$.

The second method was proposed by Pawlak [14]. It follows a general model introduced in Section 3 for $Gr_P(U) := \{\tau(x) : x \in U\}$. For $X \subseteq U$ we define:

$$P_\tau(X) := \bigcup\{\tau(x) : \tau(x) \subseteq X\} \qquad P^\tau(X) := \bigcup\{\tau(x) : \tau(x) \cap X \neq \emptyset\}.$$

We put $POS_P(X) := P_\tau(X)$ and $NEG_P(X) := P_\tau(X')$ and we introduce

$$P_b^\tau(X) := P^\tau(X) \setminus NEG_P(X),$$

in purpose of avoiding overlapping regions and achieving the properties in Table 3. It is easy to note that $\bigcup Gr_P(U) = U$, hence Proposition 2 is applicable.

The third considered approach is more novel, though it could be referred to, e.g., [2,7,19], where the concepts/granules are defined from different perspectives as the sets of objects, which are all enough similar to each other. In the framework of tolerances $\tau \subseteq U \times U$, such sets correspond to *preclasses* (*cliques*) $Y \subseteq U$ such that for any $x, y \in Y$, there is $(x, y) \in \tau$. Preclasses maximal with respect to inclusion are called the *tolerance classes* of $\tau$. If a relation $\tau$ is infinite, then a statement that classes exist is equivalent to *the Axiom of Choice* (cf. [20]). The family of all classes of $\tau$ will be denoted by $\mathcal{H}_\tau$. Using the model described in Section 3 for $Gr_H(U) := \mathcal{H}_\tau$, we obtain the following, for any $X \subseteq U$:

$$H_\tau(X) := \bigcup\{Y \in \mathcal{H}_\tau : Y \subseteq X\} \qquad H^\tau(X) := \bigcup\{Y \in \mathcal{H}_\tau : Y \cap X \neq \emptyset\}.$$

For the same reason as before, we consider the *biting* procedure:

$$H_b^\tau(X) := H^\tau(X) \setminus NEG_H(X), \quad \text{where} \quad NEG_H(X) := H_\tau(X').$$

Since $\bigcup \mathcal{H}_\tau = U$, the operators $H_\tau$ and $H_b^\tau$ satisfy the properties in Table 3.

The last two approaches are quite comparable at the abstract level, by defining $Gr(U) \subseteq \mathcal{P}(U)$ as equal to $\{\tau(x) : x \in U\}$ and $\mathcal{H}_\tau$, respectively. One may claim that the tolerance classes seem to be closer to a general intuition behind *knowledge granules*, as it is easier to assign abstract descriptions to the sets of mutually *homogenous* elements, than to the neighborhoods $\tau(x) \subseteq U$ induced by their centers $x \in U$. Nevertheless, we would like to emphasize that both those methods of handling tolerances gain a lot by using the *biting* procedure. In particular, they become better comparable to each other:

**Proposition 3.** *For any $U$, tolerance $\tau \subseteq U \times U$, and $X \subseteq U$, we have:*

$$\tau_*(X) \subseteq P_\tau(X) \subseteq H_\tau(X) \subseteq X \subseteq H_b^\tau(X) \subseteq P_b^\tau(X) \subseteq \tau^*(X).$$

The above inclusions cannot be strengthened, as shown by the following:

*Example 1.* Let $S = \langle Ob, At, Val_a \rangle$ be the system defined in Table 1. Consider tolerance relations $\rho(\text{Height}, 5)$ and $sim(\text{Degrees})$. Since they are reflexive and symmetric, we interpret them in terms of undirected pairs of objects. Relation $\tau$

over $At$ takes the form $\rho(\text{Height}, 5) \cap sim(\text{Degrees})$, which corresponds to the following pairs: $\{x_1, x_8\}$, $\{x_2, x_3\}$, $\{x_2, x_4\}$, $\{x_2, x_5\}$, $\{x_3, x_5\}$, $\{x_4, x_6\}$, $\{x_4, x_{10}\}$, $\{x_6, x_9\}$, $\{x_7, x_8\}$, $\{x_8, x_9\}$. For $X := \{x_1, ..., x_6\}$ we obtain:

$$\begin{array}{ll} \tau_*(X) = \{x_2, x_3, x_5\} & H_b^\tau(X) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_{10}\} \\ P_\tau(X) = \{x_2, x_4, x_3, x_5\} & P_b^\tau(X) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_9, x_{10}\} \\ H_\tau(X) = \{x_2, x_3, x_4, x_5, x_6\} & \tau^*(X) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_8, x_9, x_{10}\} \end{array}$$

We can see that $\tau_*(X) \subsetneq P_\tau(X) \subsetneq H_\tau(X) \subsetneq X \subsetneq H_b^\tau(X) \subsetneq P_b^\tau(X) \subsetneq \tau^*(X)$. The operators $H_\tau$ and $H_b^\tau$ approximate concepts in a finest way, when comparing to $\tau_*$ and $\tau^*$, or $P_\tau$ and $P_b^\tau$. The flow of inclusions in Proposition 3 would not be so clear without *biting*, i.e. it is not easy to compare operators $H^\tau$, $P^\tau$, and $\tau^*$. On the one hand, for equivalence relations we obtain the following:

**Proposition 4.** *If tolerance $\tau \subseteq U \times U$ is transitive, then $\tau_*(X) = P_\tau(X) = H_\tau(X)$ and $H_b^\tau(X) = H^\tau(X) = P_b^\tau(X) = P^\tau(X) = \tau^*(X)$, for any $X \subseteq U$.*

On the other hand, for tolerances in general, the approach in Definition 2 provides a convenient framework for both theoretical and practical analysis.

## 5 Conclusions

We introduced a new approach to dealing with granular definability and crispness, as well as to extending the rough set approximations for information systems with non-equivalence relations. We analyzed the properties of new approximation operators and illustrated our general methodology within tolerance approximation spaces, adapting methods proposed in [14,16], as well as defining the approximation operators based on tolerance classes.

**Acknowledgements.** Research reported in this paper was supported by the research grant no. BST 1069/16 awarded to the second author from the Faculty of Psychology, Warsaw University. The second author wishes to thank Yiyu Yao and the Staff of the Department of Computer Science, University of Regina, for valuable discussions and for providing good working conditions and friendship atmosphere during his visit in the academic year 2005/06. Both authors wish to thank Andrzej Skowron for his valuable remarks.

## References

1. Bargiela, A., Pedrycz W.: Granular Computing: An Introduction. Kluwer Academic Publishers (2002).
2. Bazan, J.G., Skowron, A., Ślęzak, D., Wróblewski, J.: Searching for the Complex Decision Reducts: The Case Study of the Survival Analysis. In: Proc. of Foundations of Intelligent Systems, ISMIS'2003. Lecture Notes in Computer Science. **2871** (2003) 160–168.
3. Demri, S., Orłowska, E.: Incomplete Information: Structures, Inference, Complexity. Springer-Verlag (2002).

4. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets, International Journal of General Systems. **17(2-3)** (1990) 191–209.
5. Järvinen, J.: Knowledge representation and rough sets. Doctoral dissertation. University of Turku, Turku Center for Computer Science (1999).
6. Lin, T.Y.: Granular computing on binary relations I, II. In: L. Polkowski, A. Skowron, (Eds.), Rough Sets in Knowledge Discovery. Physica-Verlag (1998) 107–140.
7. Lin, T.Y.: A Roadmap from Rough Set Theory to Granular Computing. In: Proc. of Rough Sets and Knowledge Technology, RSKT'2006. Lectures Notes in Computer Science. **4062** (2006) 33–41.
8. Orłowska, E.: Logic for reasoning about knowledge. Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik. **35** (1989) 559–568.
9. Orłowska, E.: Reasoning with incomplete information: rough set based information logics. In: V. Algar, S. Bergler, F. Q. Dong (Eds.), Incompleteness and Uncertainty in Information Systems Workshop. Springer-Verlag (1993) 16–33.
10. Pawlak, Z.: Information Systems – theoretical foundation. Information systems. **6** (1981) 205–218.
11. Pawlak, Z.: Rough sets. International Journal of Computing and Information Sciences. **18** (1982) 341–356.
12. Pawlak, Z.: Rough sets. Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers (1991).
13. Pawlak, Z.: Elementary rough set granules: toward a rough set processor. In: S. K. Pal, L. Polkowski, A. Skowron (Eds.), Rough-Neural Computing: Techniques for Computing with Words. Springer-Verlag (2003) 5–13.
14. Pawlak, Z.: Some Issues on Rough Sets. Transactions on Rough Sets, I, Journal Subline, Lectures Notes in Computer Science. **3100** (2004) 1–58.
15. Polkowski, L.: Rough Sets: Mathematical Foundations. Physica-Verlag (2002).
16. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae. **27** (1996) 245–253.
17. Skowron, A., Stepaniuk, J.: Informational granules and rough-neural computing. In: S. K. Pal, L. Polkowski, A. Skowron (Eds.), Rough-Neural Computing: Techniques for Computing with Words. Springer-Verlag (2003) 43–84.
18. Skowron, A., Stepaniuk, J., Peters, J. F.: Towards discovery of relevant patterns from parametrized schemes of information granule construction. In: M. Inuiguchi, S. Hirano, S. Tsumoto (Eds.), Rough Set Theory and Granular Computing. Springer-Verlag (2003) 97–108.
19. Synak, P., Ślęzak, D.: Templates in Relational Information Systems. In: Proc. of Rough Sets and Knowledge Technology, RSKT'2007. Lecture Notes in Artificial Intelligence **4481** (2007).
20. Wasilewski, P.: On selected similarity relations and their applications into cognitive science (in Polish). Unpublished doctoral dissertation, Jagiellonian University: Department of Logic, Cracow, Poland (2004).
21. Yao, Y.Y.: Information granulation and rough set approximation. International Journal of Intelligent Systems. **16(1)** (2001) 87–104.
22. Yao, Y.Y.: A partition model of granular computing. Transactions on Rough Sets, I, Journal Subline, Lectures Notes in Computer Science. **3100** (2004) 232–253.

# Unusual Activity Analysis in Video Sequences

Ayesha Choudhary[1], Santanu Chaudhury[2], and Subhashis Banerjee[1]

[1] Department of Computer Science and Engineering
Indian Institute of Technology Delhi, New Delhi, India
{ayesha,suban}@cse.iitd.ernet.in
[2] Department of Electrical Engineering
Indian Institute of Technology Delhi, New Delhi, India
schaudhury@gmail.com

**Abstract.** We present a unique representation scheme for events in an area under surveillance, which provides a mechanism to analyze videos from multiple perspectives for unusual activity analysis. We propose clustering in event component spaces and define algebraic operations on these clusters to find co-occurrences of event components. A *usualness* measure for clusters is proposed that not only gives a measure on how usual or unusual an activity is, but also a basis for analyzing and predicting the possibly usual or unusual activities that can occur in the surveillance region.

**Keywords:** Clustering, Unsupervised Learning, Unusual Activity Analysis, Event Recognition.

## 1 Introduction

Automatic learning and detection of anomalous behavior from video sequences is an important area of research in computer vision, specially in the context of visual surveillance. Machine learning and probabilistic techniques are widely applied in this area. Most of the activity recognition systems predefine and model the anomalous activities so that the system can recognize whether the activities detected are anomalous or not [1]. Others learn the usual activity patterns either in supervised or unsupervised manner and then recognize unusual activities based on their dissimilarity from the usual ones. Supervised learning based methods not only need large volumes of training data, usually difficult to get for real world applications, they also suffer from the shortcoming that all activities in the real world cannot be predefined.

Given a long video sequence and no prior information of the scene, we propose a representation scheme for events that logically partitions the event feature vector. This representation allows us to apply different similarity measures on each of the components and cluster the event components rather than clustering the monolithic event vector. Therefore, it can be used for both video mining for similar events as well as unusual activity analysis. We propose a *usualness* measure on clusters that depends on the size of the cluster. As unusual activities are rare and dissimilar from normal, clusters with low *usualness* measure depict unusual

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 443–450, 2007.

activities. The novelty of our work also comprises of the algebraic operations defined on these clusters, which along with the *usualness* measure associated with each cluster allows us to explore the space of all clusters for detecting unusual events in the video. It gives us a tool for finding co-occurrences of event components, thus, allowing analysis of the video from multiple perspectives. Moreover, these algebraic operations on the clusters of event components allow us to get back clusters of the monolithic event vectors. Therefore, there is no loss of information by clustering in the event component spaces instead of clustering in the event space. The co-occurrence calculus for clusters is not present in the literature and therefore, is unique and a novel contribution of our work. This event representation and clustering scheme can also be used to develop applications like Intelligent Fast Forward [2], where given a event segment in a video the system is able to move to the next or all portions of the video where a similar event occurs.

In the next section, we discuss some of the main techniques that have been applied for activity recognition. In section 3 we present the event representation scheme. Section 4 defines the clustering framework. In section 5, we present the results and conclude in section 6.

## 2 Related Work

As mentioned above, most activity recognition systems model and learn known activities. The Hidden Markov Models (HMMs) and its variants are most widely used for this purpose, [3],[4], [5], [6], [7], [8], [9]. HMMs are used by Starner and Pentland [6] for modeling hand gestures. Variants of HMMs, Parameterized-HMM (PHMM) [10], Coupled-HMM(CHMM) [7] are used for recognizing complex activities like interaction between moving objects in the scene. In [11] stochastic context-free grammar is used for computing the probability of temporally consistent sequences of primitive actions that are recognized by a HMM model. In [12], a model of stochastic context-free grammar is proposed for recognizing semantically meaningful behavior over extended periods. The authors in [13] propose *propagation networks* for modeling temporal inter-leavings of low level events which may occur concurrently in multi-object activities. Bayesian networks is yet another popular technique used for activity recognition [14], [15], [1], [16], [4]. In [1], [16], multi-layered FSM model is proposed for activity recognition where supervised training using Bayesian formulation is used for estimation of the parameters of their model. In [17], a multi-layered FSM framework is used for unsupervised learning of usual activities. In this method those activities that are not recognized as usual are flagged as unusual. Usual activities are learnt using unsupervised clustering in [18]. Unlike our approach, these two methods learn usual activity patterns for detection of unusual activities. In our approach, we cluster all events and based on their *usualness* measure, events are flagged as usual or unusual.

## 3 Event Representation

Events in a long video sequence are characterized by the position of moving objects, through time. In general, it is observed that objects tend to move from one landmark to another. These landmarks include locations from which objects enter the scene, exit the scene and in general, locations where they stand and wait. In our terminology, these landmarks are referred to as *attractors* and a trajectory is then an *attractor* sequence.

Thus, an event feature vector is a high-dimensional vector that contains low-level information about the object in the scene, its positions through time and the time during which it is visible in the scene. This leads to the problem of clustering heterogeneous data in high dimensional vector space. Clusters in this space give a restricted view of similarity of events. For example, if a person $P_i$ traverses a landmark sequence $LS_j$ during a certain time interval and another person $P_j$ traverses $LS_j$ during another time interval, the event vectors will be dissimilar and shall not be clustered together. Thus, even if it is common for an object of category *individual* to traverse landmark sequence $LS_j$, clustering in the high dimensional event space leads to the loss of this information. We represent an event as a tuple,

$$T_i = (OID, OC, LS, TI)$$

where,

- OID: *Object ID* is the ID given to an object when it enters the scene.
- OC: *Object Category* is the category to which the object belongs, for example, individual or group.
- LS: *Landmark Sequence* is the sequence of *attractors* that the object visits during its presence in the scene.
- TI: *Time Interval* denotes the time during which the person is visible in the scene.

This representation logically partitions the event vector into semantically meaningful quantities. Each component is of a different data type, not necessarily numerical, and the components are not comparable among themselves. Therefore, different similarity measures can be applied on each component and clustering can be done in the component spaces instead of the event space.

## 4 Clustering Framework

We define the similarity measure for tuples and the *usualness* measure for clusters below:

**Definition 1:** *Similarity measure for tuples.* Assume that the data consists of tuples of the form $T_i = (t_{1_i}, t_{2_i}, \ldots, t_{m_i})$ where each component $t_k$ represents a numeric or semantic data type. The components $t_k$'s for all $k$ need not be comparable among themselves. Let $S_i$ be the similarity measure for the $i^{th}$ component,

$t_i$. Then, $S = (S_1, S_2, \ldots, S_m)$ defines the similarity measure between tuples $T_i$ and $T_j$ such that, $S(T_i) = T_j$ iff $S_1(t_{1_i}) = t_{1_j}, S_2(t_{2_i}) = t_{2_j}, \ldots, S_m(t_{m_i}) = t_{m_j}$. This similarity function defines an equivalence relation on the tuples.

**Definition 2:** *Size of a cluster.* The number of items belonging to a cluster defines the size of the cluster.

**Definition 3:** *Usualness measure associated with a cluster.* Let $\Omega$ be the set of all clusters, and $C \subset \Omega$ be a cluster of size $x$. The *usualness* measure function for a cluster $C$ is defined as:

$$p(C) = \begin{cases} 0 & x < Thres_1 \\ e^{-(x-Thres_2)^2/(2*\sigma^2)} & Thres_1 \leq x \leq Thres_2 \\ 1 & x > Thres_2 \end{cases} \quad (1)$$

where,
$\sigma = (Thres_2 - Thres_1)/3$, $Thres_1$ and $Thres_2$ are thresholds on the rate of growth of the *usualness* of a cluster.

A cluster represents an unusual activity if this measure is 0. If the measure is 1, the cluster represents a usual activity. All values of $p(C) \in (0, 1)$, denote the extent to which the cluster represents a usual phenomenon. This is similar to the membership function defined for a fuzzy set.

### 4.1 Clustering in Component Spaces

The clustering algorithm is a dynamic incremental clustering algorithm, which is applied to each component of the event tuple that is formed as the video is parsed. As the clusters are created, the values of the other components for that event vector are also stored. A component, denoted by $t$, is clustered as follows:

- Let $\Omega$ be the set of all clusters of a particular component. Initially, $\Omega = \phi$, the empty set.
- When the first tuple is encountered, create a cluster $C_1$, and assign $t_1$ to it. $p(C_1) = 0$.
- As the tuples are encountered, two possibilities exist:
  - If $S(t_k) = t_i \in C_i$, assign $t_k$ to cluster $C_i$ and update $p(C_i)$.
  - Otherwise, create a cluster $C_k$ and assign $t_k$ to it. $p(C_k) = 0$.
- Repeat until all the tuples are clustered.

Thus, event components can be clustered without knowing the number of clusters *a priori* and clusters for each component depicts how usual the occurrence of that component is. For example, in an airport the sequence of entering the airport and directly go to the airline desk is a commonly taken path depicting a usual event, whereas a person going from the entrance to a restricted area is a rarely traversed path depicting an unusual event. Thus, clustering in the event component space gives a flexible tool to evaluate the usualness of an event component without explicitly knowing which events occurred.

### 4.2 Properties of *usualness* Measure

The *usualness* measure defined on the clusters satisfy the following properties:

- $0 \leq p(C) \leq 1$
- $p(\phi) = 0$
- $p(A \cup B) = max\{p(A), p(B)\}$
- $p(A \cap B) \leq min\{p(A), p(B)\}$

where $C$ is any cluster, $\phi$ is an empty cluster, and $A$ and $B$ are clusters either from the same or different component spaces. The union of two cluster defines the *OR* operation and is well defined if both the clusters belong to the same component space. It defines a commutative monoid on the space of all clusters.

### 4.3 Composition of Clusters

When the event component clusters are created or updated, if the tuple information is also stored, then composition of clusters give an insight into the co-occurrence of two or more event components. Let $C_{x^*}$ be the cluster for the value $x^*$ of the first component of the event cluster and $C_{y^*}$ be the cluster for the value $y^*$ of the second component of the event cluster. Then, a composition of the clusters will be the set

$$C_{x^* \otimes y^*} = \{(x_i, y_j) | S_x(x_i) = x^*, (x_i, y^*) \in C_{y^*} \text{and} S_y(y_j) = y^*, (x^*, y_j) \in C_{x^*}\}$$

where, $S_x$ and $S_y$ are the similarity measures on the $x$ and $y$ components.

Thus, when the values of the complete tuples are stored while clustering in the component space, the composition operation gets back the cluster in a higher dimensional space. This gives a powerful mechanism for getting all the cluster combinations in higher dimensional spaces from one-dimensional clusters. The *usualness* measure of the composite cluster can then be computed from its size.

Composition of clusters across spaces provides a tool to find the *usualness* of co-occurrence of two component values. For example, it answers queries of the form "Is it usual that groups of people traverse landmark sequence $LS_1$, from the entrance of the airport to the airline desk?" While the clusters in each component space provide only the knowledge of which component value occurs often, the composition of clusters gives us a different perspective to the state of the usual and unusual activities in the system.

In relational databases, a join operation combines records from two or more tables. The composition of clusters can be seen as a join operation between clusters, instead of records. This technique of manipulating the clusters gives us an insight into the state of the system. Moreover, the bounds on the *usualness* measure of the resulting clusters gives us an idea of the usualness of the co-occurrence of two components.

In case, it is desired to find the *usualness* measure of the composition of $C_x$ and $C_y$, without considering whether the $(x, y)$ tuple actually occurred as an

event component, equation 2 gives the greatest lower bound and the least upper bound on the *usualness* measure of the set $C_{x^*} \cap C_{y^*} = \{(x_i, y_j) | (x^*, y_j) \in C_{x^*} \text{and} (x_i, y^*) \in C_{y^*}\}$

$$p(C_{x^* \otimes y^*}) \leq p(C_{x^*} \cap C_{y^*}) \leq min\{p(C_{x^*}), p(C_{y^*})\} \tag{2}$$

These bounds can be used to find the *usualness* of the event tuples, for event components that may not have co-occurred. Thus, this also gives an insight into the usualness of events that may occur in the scene.

The composition of clusters from all four component spaces give back the *usualness* of the event tuple in the database. For instance, if an event tuple $T = (P_1, P, LS_1, TI_1)$ occurred n times. Suppose that $P_1$ belonging to the category $P$ traversed through landmark sequence $LS_1$ many times later in the video. Thus, the clusters for $P_1$, $P$, $LS_1$ each have size $>$ n, while $TI_1$ has size n. Then,

$$p(C_T) = p(C_{P_1 \otimes P \otimes LS_1 \otimes TI_1}) \leq min\{p(P_1), p(P), p(LS_1), p(TI_1)\} = p(TI_1)$$

This shows that the composition operation gives back the actual *usualness* measure of an event.

Therefore, properties of the *usualness* measure allow us to define well-defined operations on the clusters. Without explicitly storing the clusters in different dimensions, the composition operation gives back the clusters and their true *usualness* measure. This allows the user to get the information required for analyzing the activities as well as predicting the possibly unusual events that can occur in the area under surveillance. Thus, our event representation technique is powerful enough to give a multi-perspective view of the usual and unusual events in the scene as well as to find similar events across a long video sequence.

## 5 Results

In our implementation, adaptive background subtraction is used for detecting moving objects in the scene and estimating the category to which the object belongs. *Landmark Sequences* are found by finding the *attractors* at which the object enters the scene and the *attractors* it visits while it is in the scene. Finally, when the object exits from the scene, we cluster the event components. We use equality of components as the similarity measure. Our input video consists of people walking in a long corridor of a building.The attractors are the entrances to the corridor and the doors of the various offices. Figure 1 shows frames taken from the result video. Figure 2 shows a log of the usual and unusual landmark sequences in the input video, which are consistent with the ground truth. The log in figure 3 shows composition of clusters for event components: object category and landmark sequences.

## 6 Conclusion

We proposed an event representation scheme where each component of the event vector is a logical entity. We cluster in the component spaces instead of clustering


**Fig. 1.** Frames from the input sequence

```
File  Edit  View  Terminal  Tabs  Help
The landmark sequence is:
Attractor: 4
Attractor: 9
Attractor: 1
This is a usual landmark sequence for the scene

The landmark sequence is:
Attractor: 5
Attractor: 1
This is a usual landmark sequence for the scene

The landmark sequence is:
Attractor: 6
Attractor: 9
This is an unusual landmark sequence for the scene
The details of the event are:
The object id is: P11
The object category is: Person
The frame interval is from frame no: 9048 to frame no: 9134

The landmark sequence is:
Attractor: 5
Attractor: 6
Attractor: 9
Attractor: 1
This is an unusual landmark sequence for the scene
The details of the event are:
The token id is: P12
The token category is: Person
The frame interval is from frame no: 13808 to frame no: 13901

The landmark sequence is:
Attractor: 1
Attractor: 5
```

**Fig. 2.** The log of the video after clustering in the landmark sequence space

```
File  Edit  View  Terminal  Tabs  Help
The co-occurrence of
Object Category: Person
 and
Landmark Sequence:
Attractor: 1
Attractor: 4
is 0.324652 usual.

The co-occurrence of
Object Category: Person
 and
Landmark Sequence:
Attractor: 9
Attractor: 4
 is 0.324652 usual.

The co-occurrence of
Object Category: Person
 and
Landmark Sequence:
Attractor: 4
Attractor: 9
Attractor: 1
 is usual

The co-occurrence of
Object Category: Person
 and
Landmark Sequence:
Attractor: 5
Attractor: 1
 is usual

The co-occurrence of
```

**Fig. 3.** The log for co-occurrence of event components

the monolithic event vector. The proposed *usualness* measure on the clusters along with the algebraic operations defined on these clusters provide a flexible and well-defined tool to predict the co-occurrences as well as *usualness* of events. This method can be used for a variety of video applications, including unusual activity analysis and indexing and mining of videos.

## References

1. Hongeng, S., Bremond, F., Nevatia, R.: Representation and optimal recognition of human activities. In IEEE Conference on Computer Vision and Pattern Recognition (2000) 1818–1825
2. Zelnik-Manor, L., Irani, M.: Event-based analysis of video. In IEEE Conference on Computer Vision and Pattern Recognition (2001) 123–130

3. Kettnaker, V.: Time-dependent HMMs for visual intrusion detection. In IEEE Workshop on Event Mining: Detection and Recognition of Events in Video (2003)
4. Medioni, G., Cohen, I., Bremond, F., Hongeng, S., Nevatia, R.: Event detection and analysis from video stream. IEEE Trans. on Pattern Analysis and Machine Intelligence, 23(8) (2001) 873–889
5. Moore, D., Essa, I., Hayes, M.: Exploiting human actions and object context for recognition tasks. In International Conference on Computer Vision (1999) 80–86
6. Starner, T., Pentland, A.: Real-time american sign language recognition from video using hidden Markov models. In SCV (1995) 265–270
7. Brand, M., Oliver, N., Pentland, A.: Coupled hidden Markov models for complex action recognition. In IEEE Conference on Computer Vision and Pattern Recognition (1997) 994–999
8. Oliver, N., Rosario, B., Pentland, A.: A Bayesian computer vision system for modeling human interactions. In International Conference on Computer Vision Systems (1999) 255–272
9. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden Markov model. In IEEE Conference on Computer Vision and Pattern Recognition (1992) 379–385
10. Wilson, A., Bobick, A.: Recognition and interpretation of parametric gesture. In International Conference on Computer Vision (1996) 329–336
11. Ivanov, Y., Bobick, A.: Recognition of visual activities and interactions by stochastic parsing. IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(8) (2000) 852–872
12. Moore, D., Essa, I.: Recognizing multitasked activities from video using stochastic context-free grammar. In AAAI (2002)
13. Shi, Y., Bobick, A.: Representation and recognition of activity using propagation nets. In 16th International Conference on Vision Interface (2003)
14. Buxton, H., Gong, S.: Advanced visual surveillance using Bayesian networks. In International Conference on Computer Vision (1995) 111–123
15. Madabhushi, A., Aggarwal, J.: A Bayesian approach to human activity recognition. In 2nd International Workshop on Visual Surveillance (1999) 25–30
16. Hongeng, S., Nevatia, R.: Multi-agent event recognition. In International Conference on Computer Vision (2001) 84–93
17. Mahajan, D., Kwatra, N., Jain, S., Kalra, P., Banerjee, S.: A framework for activity recognition and detection of unusual activities. In Indian Conference on Computer Vision, Graphics and Image Processing (2004)
18. Zhong, H., Shi, J., Visontai, M.: Detecting unusual activity in video. In IEEE Conference on Computer Vision and Pattern Recognition (2004) 819–826

# Task-Based Image Annotation and Retrieval

Dympna O'Sullivan[1], David Wilson[2], Michela Bertolotto[3], and Eoin McLoughlin[3]

[1] University of Ottawa, Ottawa, Canada
dympna@management.uottawa.ca
[2] University of North Carolina, Charlotte, United States
davils@uncc.edu
[3] University College Dublin, Dublin, Ireland
michela.bertolotto, eoin.a.mcloughlin@ucd.ie

**Abstract.** In order to address problems of information overload in digital imagery task domains we have developed an interactive approach to the capture and reuse of image context information. Our framework models different aspects of the relationship between images and domain tasks they support by monitoring the interactive manipulation and annotation of task-relevant imagery. The approach allows us to gauge a measure of a user's intentions as they complete goal-directed image tasks. As users analyze retrieved imagery their interactions are captured and an expert task context is dynamically constructed. This human expertise, proficiency, and knowledge can then be leveraged to support other users in carrying out similar domain tasks. We have applied our techniques to two multimedia retrieval applications for two different image domains, namely the geo-spatial and medical imagery domains.

## 1 Introduction

In the digital image domain, advances in techniques for the capture and storage of information have caused an explosion in the amount of available data. Domains that rely on digital image analysis (e.g. geo-sciences, military and medicine) now have to contend with organizing, indexing, storing, and accessing information from huge repositories of heterogeneous data. Current image retrieval systems are typically characterized by one of two main approaches; they either support keyword-based indexing or a content-based approach where low level visual features are automatically extracted. It is, however, recognized that neither approach is fully adequate for answering the complete range of user queries. The keyword-based approach depends on images being accompanied by textual descriptions but indexes for these descriptions are time consuming to create and maintain, particularly since entries are not grounded in how the collections are being used. Content-based retrieval also suffers from many disadvantages. Results can frequently be poor due to the semantic gap and the subjectivity of human perception. The first is the difference between the high-level concepts that users search for and the low-level features employed to retrieve the imagery. The latter addresses the fact that different people or the same person

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 451–458, 2007.

in different situations may judge visual content differently. In this research we have addressed these problems by developing a context-based approach to image retrieval. Context-based retrieval relies on knowledge about why image contents are important in a particular domain and how specific images have been used to address particular tasks. Our framework models different aspects of the relationship between images and the domain tasks that they support by monitoring the interactive querying, manipulation and annotation of task-relevant imagery. This contrasts with prevalent retrieval and annotation schemes that focus on what individual images contain but that provide no context for which, if any, of those aspects are important to users. This research attempts to capture the knowledge implicit in using imagery to address particular tasks by leveraging a measure of the user's intentions with regard to tasks they address. This is done by situating intelligent support for gathering important expert context inside a flexible task environment and by monitoring and recording user actions. The capture of task-specific knowledge enables us to infer why image contents are important in a particular context and how specific images have been used to address particular domain goals. Our algorithms for capturing and reusing contextual knowledge are implemented using a case-based reasoning (CBR) approach. The capture of task knowledge allows for the development of a case base of expert task experiences. These previous experiences form the basis of a knowledge management system that complements direct image retrieval by presenting it along with other relevant task-based information. The repository of task-based experiences may be exploited to improve the ability of the application to make pro-active context-based recommendations. This underlying case-based engine forms the fundamental architectural framework and is employed by two developed applications.

In this research we focus on developing case-based knowledge management support for libraries of digital imagery. This research draws on general work in case-based knowledge management [1], as well as knowledge management initiatives in medicine that promote the collection, integration and distribution of a single medical modality [2]. In this research we are working with large collections of experience and user context. As in [3], we believe that user interactions with everyday productivity applications provide rich contextual information that can be leveraged to support access to task-relevant information. All contextual knowledge is gathered by the system using implicit analysis so that users are shielded from the burden of relevance feedback or other such explicit interest indicators [4]. By situating intelligent tools and support within task environments we can unobtrusively monitor and interpret user actions concerned with rich task domains based on a relatively constrained task environment model. Our methods for annotating multimedia are related to annotation for the semantic web [5] and multimedia indexing [6] where the focus is on developing annotated descriptions of media content. Multimedia database approaches such as QBIC [7] provide for image annotation but use the annotations to contextualize individual images. In this work we are concerned with a task-centric view of the annotations, where

we employ annotations to tell us how an image relates to a current domain task by using image annotations to contextualize task experiences.

The rest of this paper is outlined as follows. In the next section we outline the two developed image retrieval applications and then continue with a description of our contextual framework for task-based image annotation and manipulation in Section 3. Section 4 explains how we combine image interaction information with more high-level user concepts to retrieve complete knowledge-based user sessions and in Section 5 we describe our similarity metrics. We conclude in Section 6 with a discussion.

## 2 Developed Applications for Task-Based Image Retrieval

We have applied our techniques for the retrieval and management of digital image data to two application domains. The MAGIK application is a task-based image retrieval environment that has been designed for the management of geo-spatial image resources. The MEDIC application integrates digital medical imagery with electronic patient records in an Electronic Health Record System (EHRS). The application can be used both as a replacement for paper-based patient records and to support interacting clinicians by providing clinical decision support at the point of care.

### 2.1 The MAGIK Application

In the MAGIK (Managing Geo-Spatial Imagery and Knowledge) application we have developed storage, indexing, and retrieval tools for geo-spatial image information. We describe the system in terms of a typical user task. For example, an organization that uses geo-spatial data to support architectural development projects may employ the system to assist in selecting the optimal location for a new airport servicing an urban area. When a user logs into the image interaction environment, they are directed to an interface that enables them to search directly for imagery corresponding to their current task needs. A typical task-based query to our image repository is a straightforward request to a geo-spatial image database and may consist of any combination of image metadata and free-text semantic task information. As the user specifies their query this information is captured by the system and added to their current context. For example, the urban planner interested in building the airport might wish to view recent images of possible construction sites. They could outline specific image metadata (location, recent dates) and also provide a textual description of the kind of imagery they would like returned. In this instance they could specify that they are interested in retrieving images of undeveloped land of low elevation with good infrastructure on the outskirts of the urban center. Retrieved image results are displayed as a ranked list with an associated percentage matching score and are added to the user's current task context. The user can browse the images retrieved and select relevant information for further manipulation and annotation. We will describe our task-based image annotation environment in Section 3.

### 2.2 The MEDIC Application

The MEDIC (MobilE Diagnosis for Improved Care) application is a mobile clinical decision support system that integrates medical imagery with other types of patient information in an EHRS. The application allows clinicians to efficiently input, query, update, and compare patient records including associated medical imagery on mobile and desktop devices. This system will also be described in terms of a typical interaction. A patient presents at the Emergency Department with a dislocated thumb. The patient has not previously attended the hospital and so a new patient record is created in the EHRS. The patient supplies different types of information including demographic data and previous medical history. The interacting clinician then enters clinical information about the patient such as the presenting symptoms and a diagnosis. The application may then be used to order an X-RAY for patient's dislocated thumb and the patient is added to the list of waiting patients in the radiography department. Once an X-Ray has been performed the patient's images are added to the patient's profile in the EHRS. The radiographer can analyze the imagery using a set of image annotation tools (described in section 3) and update the patient's status in the EHRS. The original clinician can then access this information on his or her mobile device and treat the patient accordingly.

## 3 Task-Based Image Annotation

Our research is focused on capturing contextual task knowledge to perform more efficient image retrieval by employing annotations for capturing and recording aspects of tasks in progress. To this end we have developed tools for direct image manipulation to assist the user in organizing information about task-relevant imagery. When executing a specified task the user needs to be able to tease out the particular information aspects that support their goal. Ideally, two work products emerge: first, the actual image information as applied to the task, and second, a record of the information gathering process that allows for incremental development and provides a reference for subsequent justification and refinement. As users often need to make notes and annotations in order to support the former, the latter can be supported in a natural way by integrating intelligent annotation tools tailored to the information gathering environment. The environment supports the user in constructing the most on-point information kernels by allowing the user to locate and define regions of interest in the images. From the user's perspective this supports efficient interaction, as it minimizes the need to divert attention from the information source. The environment forms a lucid and well-structured foundation for users to report verdicts and conclusions as may be required of them in a typical work-related situation. However from a system perspective, the user-defined regions can then be linked to clarifications and rationale. These insights capture high level user concepts which allows associations or relations between images to be made without the need for content-based

or keyword-based analysis. It is considered imperative that task information be collected implicitly to shield users from the burden of explicit knowledge engineering. Therefore our techniques for gathering task context employ implicit methods for knowledge capture. The tools allow us to infer from inherent user actions and to capture fine-grained task knowledge that subsequently improves the ability of the application to make pro-active context-based recommendations to users with similar task goals. We identify three main advantages of our approach. First by reusing collective knowledge in support of similar tasks the time required to carry out new tasks can be significantly reduced. Second, the approach facilitates knowledge sharing by incorporating potentially relevant knowledge from other experiences. Finally from a knowledge management perspective, contextual expert knowledge relating to particular tasks may now be stored and reused as an additional resource for support, training and preserving organizational knowledge assets.

The tools for direct image manipulation include filters, transformation, highlighting, sketching, post-it type and multimedia annotation tools. We have selected the kinds of manipulations that would be most useful in helping to analyze and focus on image features. All sketching manipulations can be performed in a variety of colors and brush styles. The user can add personal media annotations to the image as a whole or to particular highlighted image aspects. Currently, the system supports annotation by text, audio and video, though retrieval is focused on text. The system integrates real-time audio and video capture as well as compression. A facility is in place that allows users to upload web documents as annotations which allows further context to be extracted by following HTML links. The system also supports annotation by cut, copy and paste between a given image and other images in the dataset, as well as images in any application that supports clipboard functionality for the given operating system. Returning to one of the sample interactions, the radiologist has received the sample patient's medical imagery and is viewing it through MEDIC's image viewer component. The radiologist may immediately annotate the image with a diagnosis and provide a recommendation for followup treatment. Or they may invoke the decision support model of the MEDIC application if they require extra information about the particular patient or injury. For example the radiologist may be having difficulty in diagnosing the problem from the particular image. The X-Ray, however may remind him of a similar image he viewed previously and he may remember some of the details of the previous patient. In this scenario the radiologist could input the details of the previous patient as search parameters to the application. The application will then filter this patient's profile as well as any similar profiles from the EHRS allowing for comparison between the current image and images from these previous case histories. If any of the similar images have been annotated the radiologist may study these notes for extra information regarding the specific injury. By accessing these additional resources that are not normally available in the hospital setting the radiologist is able to make a more informed and confident diagnosis. The radiologist may go on to annotate or manipulate the image as shown in Figure 1.

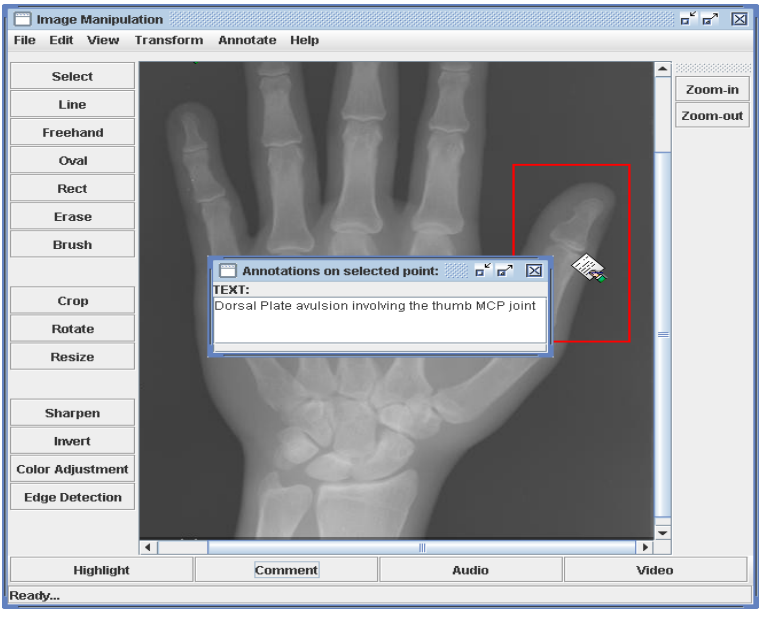



**Fig. 1.** Medical Image Annotation

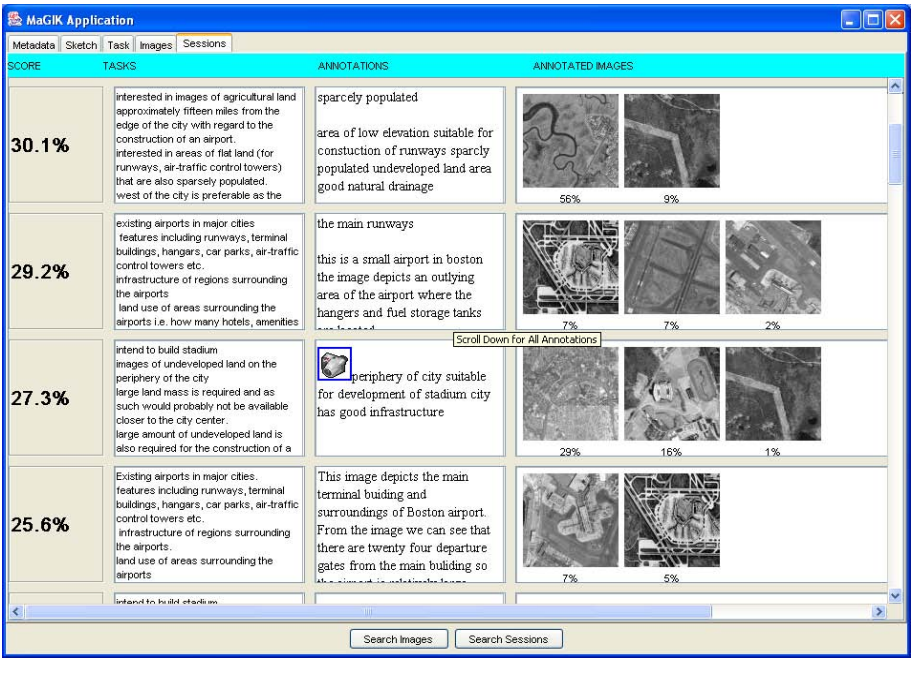



**Fig. 2.** Previous Geo-Spatial Task Contexts

## 4 Retrieval of Previous User Contexts

As the systems builds up encapsulated contextualized user interactions, a case base of previous sessions consisting of both high-level user concepts and actual image interactions is continuously updated. This knowledge base improves context-based query processing by enabling the retrieval of entire previous task-based sessions. This allows a current user to look for previous image analysis tasks that are most similar to the current task context, both to find relevant imagery and to examine decisions and rationale that went into addressing earlier tasks. Figure 2 shows an example of retrieved sessions for the sample user addressing the urban planning task with the MAGIK application. The interacting user can study previous geo-spatial tasks that bear similarity to their own. They can access previous user sessions that address planning or construction developments in urban areas and can study the entire work product of the previous user. The current user can analyze decisions and expertise that went into planning the previous development and if any of these interactions are appropriate or if any imagery and/or annotations are relevant to the current task, then these may be incorporated and integrated into the current task context.

## 5 Calculating Task Similarity

Our retrieval metrics are focused on text (textual queries and image annotations), using Information Retrieval metrics (e.g. [8]), specifically the Vector Space Model and Term Frequency-Inverse Term Frequency (TF-IDF) as a basis for similarity. Retrieval within both applications is taking place in the context of an overall workflow. In a geo-spatial context this workflow can include outlining task queries using a combination of metadata and semantic task descriptions, retrieving relevant imagery, manipulating and annotating appropriate imagery, retrieving similar user sessions and incorporating previous contexts into the new work product. Our task-based image retrieval employs textual indexes in two separate spaces, an annotation index and an image index. Each retrieved image must firstly pass a metadata filter using terms extracted from the image metadata query, and if it does so successfully, a final image score is computed as the average matching score of overall image and individual annotation similarities when matched against the semantic task description. A previous session score is calculated in a vector space across all retrieved sessions, where the text for each session is composed of the metadata, task query descriptions and applied annotations for relevant imagery. The total number of images annotated and browsed in each similar session as a fraction of the total number of images returned is then computed. The final session score is a weighted sum of session similarity and the proportion of annotated and browsed images. Some important steps in the context of a medical workflow are: entering preliminary patient details, recording results of an initial examination, inputting presenting conditions, uploading and annotating medical imagery, recording diagnoses and recommending treatments. Most patient profiles will consist of some if not most of the information described

above. Given a textual vector space for each constituent segment of a patient profiles we can match textual queries imputed by a clinician about a current patient to previous patient contexts. The task-based retrieval system employs indexes in separate spaces for the constituent segments of the patient profile. When searching for previous patient imagery or profiles, the clinician is required to enter a query using a combination of patient information in terms of these constituent profile segments. Their query and any specified weights are combined and compared to previous patient profiles and a weighted average is used to compute similarity between the current patient and other patients from the medical database. These indices are used to calculate similarity in both retrieval of medical images and retrieval of patient case histories.

## 6 Conclusions

We have introduced a case-based approach to developing a context-based retrieval system for digital imagery. The research emphasizes a task-based approach to the capture of human expertise and domain knowledge. New image requests may then be grounded in such knowledge which is reused to support other system users by leveraging previous task-based context, both in terms of individual images as well as entire previous task-based sessions. The development of two separate applications shows that our task-based approach to image retrieval is a general one and that our techniques can be scaled to different fields that rely on image analysis as well as different types of image datasets.

## References

1. Becerra-Fernandez, I., Aha, D.: Case-based problem solving for knowledge management systems. (1999) 219–223
2. Jadad, A., Haynes, R., Hunt, D., Browman, G.: The internet and evidence-based decision making: A needed synergy for efficient knowledge management in health care. Canadian Medical Association Journal **162** (2000)
3. Budzik, J., Hammond, K.: User interactions with everyday applications as context for just-in-time information access. (2000)
4. Claypool, M., Le, P., Waseda, M., Brown, D.: Implicit interest indicators. ACM Intelligent User Interfaces Conference, IUI 2001 (2001) 33–40
5. Hollink, L., Schreiber, A., Wielinga, B., Worring, M.: Classification of user image descriptions. International Journal of Human Computer Studies **66** (2004)
6. Worring, M., Bagdanov, A., Gemerr, J., Geusebroek, J., Hoang, M., Schrieber, A., Snoek, C., Vendrig, J., Wielemaker, J., Smuelders, A.: Interactive indexing and retrieval of multimedia content. (2002) 135–148
7. Flickner, M., Sawhney, H., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: The QBIC system. IEEE Computer **28** (1995) 23–32
8. Salton, G., McGill, M.: Introduction to modern information retrieval. (1983)

# Improvement of Moving Image Quality on AC-PDP by Rough Set Based Dynamic False Contour Reduction

Gwanggil Jeon[1], Marco Anisetti[2], Kyoungjoon Park[1], Valerio Bellandi[2], and Jechang Jeong[1]

[1] Department of Electronics and Computer Engineering, Hanyang University, 17 Haengdang-dong, Seongdong-gu, Seoul, Korea
{windcap315,joony79,jjeong}@ece.hanyang.ac.kr
[2] Department of Information Technology, University of Milan, via Bramante, 65 – 26013, Crema (CR), Italy
{anisetti,bellandi}@dti.unimi.it

**Abstract.** Plasma display panel (PDP) has become popular as high-end television monitors. In PDPs, gray levels are expressed by the pulse-number modulation technique to generate gradation display. Although this method can display still image faithfully, it can yield annoying distortions when displaying moving images. In order to reduce the dynamic false contour and develop moving picture quality, we propose a rough set based effective subfield optimization technique. Simulation results show that the dynamic false contour can be effectively reduced by rough set based subfield optimization algorithm.

**Keywords:** Plasma display panel (PDP), motion picture distortion, dynamic false contour (DFC), temporal artifact, gray scale, rough set theory.

## 1 Introduction

Although the mature cathode-ray tube (CRT) technology has provided high quality image displays at low costs for the past decades, CRTs are inherently bulky and heavy. Thus, a lot of studies have been made to develop various flat-panel displays, such as liquid-crystal display (LCD), organic light emitting device (OLED), and PDP. On the display device market, the PDP becomes the best candidate in the competition for large size (above 40 inches) flat screen, thin wall hanging displays. Furthermore, it is assumed that PDP is expected to be the next generation of TV displays, offering the possibility of bigger than 70 inches diagonal. Therefore, it is assumed that the PDP will quickly gain acceptance for home use replacing the traditional CRT displays. However, PDP is facing challenge of LCD, which is about to be expanded to 40~60 inches. In order to enter and hold the mainstream display market share occupied by CRT and LCD, technical advance is highly required not only to improve the picture quality but also to lower the power consumption. In order to win and survive in this competition, PDP should solve some old problems such as obtaining higher discharge efficiency, high picture quality, and low cost. To be honest, compare with LCD, picture quality is the most apparent weakness of PDP. In order to improve the picture quality, it's very important to get rid of the dynamic false contour when moving picture is reproduced and to increase the contrast ratio and the luminance of the

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 459–466, 2007.

picture reproduced. In order to solve those problems, the picture quality must be improved. False contour has been a particular concern for moving picture, and it referred to as the dynamic false contour (DFC).

Many techniques have been presented to solve DFC problem as answering following issues [1-3], such as,

*How to dispose the divided subfield?*,
*How to divide and partition the frame into multiple subfields?*,
*How to select the subfield codeword for each gray level*?

In this paper, finding the optimal subfield pattern and arrangement is beyond our scope. And we focus on the last issue which can be represented as "subfield optimization." In fact, selecting the optimal codeword among several candidates is the hardest problem.

The main purpose of this article is designing optimal subfield system, by means of rough set theory. Rough set theory is defined by equivalence relations in an information system described as a database. A method of reducing attributes in the given information system has already been developed by equivalence relations with regard to attributes. This paper is organized as follows. In Section 2, we explain the DFC briefly. Rough set based subfield optimization technique will be discussed in Section 3. Applied example is provided in Section 4. Experiment results will be given in Section 5. Finally, we conclude the paper in Section 6.

## 2 Dynamic False Contour

The PDP employs a driving scheme using subfields combination method for gray scale representation. Gray scales are represented by modulation of a total number of the light radiation pulses of each pixel within a NTSC TV field time of 16.7ms (if the TV system is PAL or SECAM, 20.0ms). In order to express 256 gray scale levels, PDP utilizes the binary coded light-radiation-period scheme as shown in Fig. 1. Each TV frame may be divided into $n$ subfields (in this Section, 8 is assigned as $n$).

Each subfield consists of a reset period, an address period and sustain (display) period. The widths of the display periods are assigned according to the binary sequence $2^0$, $2^1$,…, $2^{n-1}$, $2^n$. 256 gray scale levels can be organized by combining these subfields into emit light.

The most common technique is to divide the subfields having relatively heavier weight of luminance information and disperse them in the field. However, if the number of subfields becomes larger, then the maximum available brightness of the image reproduced decreases. Because the address period is usually fixed depending on the number of scan lines. Moreover, the percentage of the sustain period occupied in a TV field decreases as well. Therefore, in order to obtain enough peak brightness, the address pulse width needs to be shortened. However, shortening the address pulse width narrows the voltage margin of the address pulse, and then imperfect address will degrade the picture quality.

Motion picture distortion (MPD) is often illustrated by disorder of average light perception over an image, as shown in Fig. 2. This method is quite appropriate to

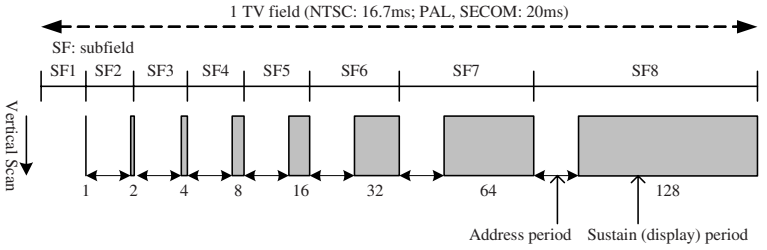



**Fig. 1.** Binary subfield sequence for [1,2,4,8,16,32,64,128], 8-bit 256 gray scales

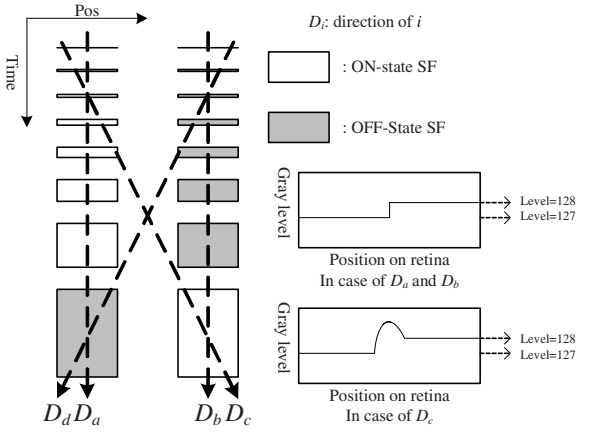



**Fig. 2.** An example of dynamic false contour: perception of disturbance

express still images. However, if a picture with motion needs to be reproduced, an unwanted distortion looking similar to false contour generated across adjacent pixels. $D_a$ and $D_b$ indicate the cases for still image with no gray scale disturbance, while $D_c$ or $D_d$ is the path (direction) along which the radiation is perceived by the trace of the eyes in a motion picture. The average light radiation which perceived along the path $D_c$ or $D_d$, becomes quite different from the original perception along $D_a$ or $D_b$. The perception along the path $D_c$ (or $D_d$) is almost doubled (or halved). Therefore, we may feel brighter or darker pixels than the original picture at the boundaries between the $127^{th}$ level and the $128^{th}$ level due to the effect of DFC.

## 3 Proposed Subfield Optimization Algorithm

A gray level expression of PDP is implemented by a time division of a picture signal of one field into a plurality of subfields. Almost all techniques on subfield optimization have been based on individual intuition. A subfield pattern (1,2,4,7,11, 16,32,42,60,80) was proposed in [3], which based on the assumption on the behavior of human visual system. In [4], the modified binary code was obtained by dividing major subfields 64, 128 of the conventional (1,2,4,8,16, 32,64,128) into four subfields

of equal length of 48 as (1,2,4,8,16,32,48,48,48,48). And in [5], the subfield pattern (1,2,4,8,16,32,42,44,52,54) was obtained using genetic algorithm. In [6], subfield sequence based on twelve-subfield Fibonacci sequence (1,2,3,5,8,13,19,25,32,40,49, 58) was proposed. This sequence is well employed by the most PDP makers today. The numbers inform the subfield weightings, which are used to judge any video level independently of the picture content and energy. Table 1 shows the number of possible codeword, according to the gray scale value.

**Table 1.** Fibonacci sequence based subfield and MPD distribution; Left: gray level value, Right: number of possible codeword

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 32 | 6 | 64 | 15 | 96 | 24 | 128 | 30 | 160 | 27 | 192 | 15 | 224 | 4 |
| 1 | 1 | 33 | 7 | 65 | 18 | 97 | 28 | 129 | 31 | 161 | 26 | 193 | 17 | 225 | 6 |
| 2 | 1 | 34 | 5 | 66 | 15 | 98 | 28 | 130 | 32 | 162 | 24 | 194 | 14 | 226 | 5 |
| 3 | 2 | 35 | 8 | 67 | 19 | 99 | 28 | 131 | 32 | 163 | 27 | 195 | 15 | 227 | 5 |
| 4 | 1 | 36 | 6 | 68 | 19 | 100 | 30 | 132 | 31 | 164 | 24 | 196 | 15 | 228 | 6 |
| 5 | 2 | 37 | 6 | 69 | 15 | 101 | 27 | 133 | 32 | 165 | 25 | 197 | 13 | 229 | 4 |
| 6 | 2 | 38 | 8 | 70 | 19 | 102 | 28 | 134 | 31 | 166 | 25 | 198 | 13 | 230 | 4 |
| 7 | 1 | 39 | 5 | 71 | 17 | 103 | 29 | 135 | 31 | 167 | 22 | 199 | 12 | 231 | 5 |
| 8 | 3 | 40 | 9 | 72 | 18 | 104 | 27 | 136 | 31 | 168 | 24 | 200 | 11 | 232 | 3 |
| 9 | 2 | 41 | 9 | 73 | 22 | 105 | 30 | 137 | 30 | 169 | 24 | 201 | 13 | 233 | 4 |
| 10 | 2 | 42 | 7 | 74 | 18 | 106 | 29 | 138 | 31 | 170 | 22 | 202 | 11 | 234 | 4 |
| 11 | 3 | 43 | 11 | 75 | 21 | 107 | 30 | 139 | 32 | 171 | 23 | 203 | 10 | 235 | 2 |
| 12 | 1 | 44 | 7 | 76 | 21 | 108 | 31 | 140 | 30 | 172 | 22 | 204 | 12 | 236 | 4 |
| 13 | 3 | 45 | 9 | 77 | 19 | 109 | 28 | 141 | 30 | 173 | 22 | 205 | 10 | 237 | 3 |
| 14 | 3 | 46 | 11 | 78 | 21 | 110 | 29 | 142 | 30 | 174 | 23 | 206 | 11 | 238 | 2 |
| 15 | 2 | 47 | 8 | 79 | 20 | 111 | 30 | 143 | 29 | 175 | 21 | 207 | 12 | 239 | 4 |
| 16 | 4 | 48 | 12 | 80 | 21 | 112 | 29 | 144 | 30 | 176 | 20 | 208 | 8 | 240 | 2 |
| 17 | 2 | 49 | 11 | 81 | 23 | 113 | 30 | 145 | 29 | 177 | 21 | 209 | 11 | 241 | 3 |
| 18 | 3 | 50 | 10 | 82 | 22 | 114 | 30 | 146 | 28 | 178 | 19 | 210 | 9 | 242 | 3 |
| 19 | 4 | 51 | 12 | 83 | 22 | 115 | 30 | 147 | 31 | 179 | 21 | 211 | 7 | 243 | 1 |
| 20 | 2 | 52 | 10 | 84 | 23 | 116 | 32 | 148 | 30 | 180 | 21 | 212 | 11 | 244 | 3 |
| 21 | 4 | 53 | 11 | 85 | 22 | 117 | 31 | 149 | 29 | 181 | 18 | 213 | 7 | 245 | 2 |
| 22 | 4 | 54 | 13 | 86 | 24 | 118 | 30 | 150 | 30 | 182 | 22 | 214 | 9 | 246 | 2 |
| 23 | 3 | 55 | 11 | 87 | 24 | 119 | 31 | 151 | 27 | 183 | 18 | 215 | 9 | 247 | 3 |
| 24 | 5 | 56 | 12 | 88 | 22 | 120 | 31 | 152 | 29 | 184 | 17 | 216 | 5 | 248 | 1 |
| 25 | 4 | 57 | 13 | 89 | 25 | 121 | 31 | 153 | 28 | 185 | 19 | 217 | 8 | 249 | 2 |
| 26 | 4 | 58 | 13 | 90 | 25 | 122 | 32 | 154 | 27 | 186 | 15 | 218 | 6 | 250 | 2 |
| 27 | 6 | 59 | 15 | 91 | 24 | 123 | 31 | 155 | 30 | 187 | 19 | 219 | 6 | 251 | 1 |
| 28 | 5 | 60 | 15 | 92 | 27 | 124 | 32 | 156 | 28 | 188 | 19 | 220 | 8 | 252 | 2 |
| 29 | 5 | 61 | 14 | 93 | 24 | 125 | 32 | 157 | 28 | 189 | 15 | 221 | 5 | 253 | 1 |
| 30 | 6 | 62 | 17 | 94 | 26 | 126 | 31 | 158 | 28 | 190 | 18 | 222 | 7 | 254 | 1 |
| 31 | 4 | 63 | 15 | 95 | 27 | 127 | 30 | 159 | 24 | 191 | 15 | 223 | 6 | 255 | 1 |

Above twelve bit gray level data need to be converted to codeword, in order to be used to display image data on PDP. We propose hamming distance (HD), which measures the amount of MPD between arbitrary two gray levels focusing on codeword. Two parameters for gray level expression, subfield vector, and a set of codeword should be determined before applying MPD distance for the two gray levels. The subfield vector is a set of ratios of sustain pulses for each subfield. A subfield vector $SF_{12}$ which has 256 gray levels and twelve subfields. $SF_{12}$ should satisfy following conditions: (1) $SF_{12}=(f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11})$, (2) each intensity $f_i \in [0,255]$, (3) $for(i=0, sum=0; i<12; i++)\ sum+=f_i$, where $sum=255$. Let us consider there exists at least one codeword $C_k=[c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}]$ such that $k=C_\gamma \cdot SF_{12}$ is true where $c_\gamma \in \{0,1\}$. Then HD is determined between $\alpha$ and $\beta$ is given by $HD=(C_\alpha \text{ XOR } C_\beta)\cdot SF_{12}$. We can calculate the HD value of each

candidate codeword. It is assumed that the codeword with smallest HD will be chosen as the best codeword. Above process are conducted with five images (Airplane, Baboon, Barbara, Finger, and Lena). Some gray level values can be represented in several codeword. For example, gray level values 225, 226, 227, 228, 229, and 230 are represented in six, five, five, six, four, and four codewords as can be seen in Table 2. In order to make rough set based codeword decision rule, we classify the codewords into dominant codewords and unnecessary codewords. Table 2 shows all codewords ranging from gray scale value 225 to 230. In order to select dominant codeword, three rules ($R^1$, $R^2$, and $R^3$) are made.

$R^1$: If the percentage of any codeword in each gray scale value is bigger than 75%, then the codeword is chosen as dominant codeword, and rough set based technique is not employed. If this condition is not satisfied, moves to Rule 2.

$R^2$: If the sum percentage of any of two codewords is bigger than 75%, and bigger one is less than two times of smaller one, then both codewords are chosen as the dominant codewords.

$R^3$: Otherwise, the three major codewords are selected as dominant codewords.

**Table 2.** All codewords ranging from gray scale value 225 to 230. The 1st column is gray level; the 2nd column is the number of possible codewords. The other columns are possible codewords.

| ... | | Codeword candidate 1 | Codeword candidate 2 | Codeword candidate 3 | Codeword candidate 4 | Codeword candidate 5 | Codeword candidate 6 |
|---|---|---|---|---|---|---|---|
| 225 | 6 | 111011101111<br>88 **(50.3%)** | 100111101111<br>79 **(45.1%)** | 110101011111<br>0 (0%) | 001101011111<br>2 (1.1%) | 000011011111<br>5 (2.9%) | 010000111111<br>1 (0.6%) |
| 226 | 5 | 010111101111<br>55 **(79.7%)** | 101101011111<br>1 (1.4%) | 100011011111<br>6 (8.7%) | 110000111111<br>5 (7.2%) | 001000111111<br>2 (2.9%) | |
| 227 | 5 | 110111101111<br>21 **(45.7%)** | 001111101111<br>18 **(39.1%)** | 011101011111<br>3 (6.5%) | 010011011111<br>1 (2.2%) | 101000111111<br>3 (6.5%) | |
| 228 | 6 | 101111101111<br>49 **(89.1%)** | 111101011111<br>0 (0.0%) | 110011011111<br>4 (7.3%) | 001011011111<br>0 (0.0%) | 011000111111<br>1 (1.8%) | 000100111111<br>1 (1.8%) |
| 229 | 4 | 011111101111<br>10 **(45.5%)** | 101011011111<br>11 **(50.0%)** | 111000111111<br>1 (4.5%) | 100100111111<br>0 (0.0%) | | |
| 230 | 4 | 111111101111<br>9 **(45.0%)** | 011011011111<br>0 (0.0%) | 000111011111<br>10 **(50.0%)** | 010100111111<br>1 (5.0%) | | |
| ... | | | | | | | |

## 4 Applied Example

Fig. 3 shows an example to assign a best codeword among many possible candidates. If the gray scale value 227 is coded, we may select just one optimal codeword among following five candidates; [110111101111], [001111101111], [011101011111], [010011011111], or [101000111111]. According to above condition, Table 2 can be reduced as Table 3. Let us consider the value of adjacent pixels are 225 (mid left), 225 (upper left), 228 (upper center), and 230 (upper right). And those adjacent pixels have been coded as [111011101111], [100111101111], [101111101111], and [111111101111], respectively. If the candidate codeword is just one, then the codeword is used to represent the gray value. But if the number of candidate codeword is two or more, then the number of occasion is counted and memorized. In order to reduce the requirements of computational burden, a lookup table which based on rough set theory is designed. Table 4 shows an example of codeword decision problem.

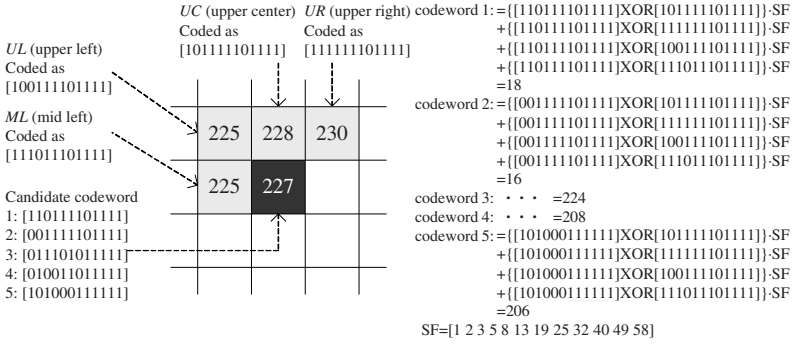



**Fig. 3.** An example of codeword decision process

**Table 3.** Dominant codewords in each gray scale value

| ... | # of candidate | Codeword 1 | Codeword 2 |
|---|---|---|---|
| 225 | 2 | 111011101111 | 100111101111 |
| 226 | 1 | 010111101111 | |
| 227 | 2 | 110111101111 | 001111101111 |
| 228 | 1 | 101111101111 | |
| 229 | 2 | 011111101111 | 101011011111 |
| 230 | 2 | 111111101111 | 000111011111 |
| ... | | | |

**Table 4.** Set of the selected method corresponding to each pattern. This is an example of codeword decision, in case of gray scale value 227.

| *U* | *a* | *b* | *c* | *d* | *m* | *U* | *a* | *b* | *c* | *d* | *m* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Small | Small | Small | Small | 1 | 9 | Big | Small | Small | Small | 2 |
| 2 | Small | Small | Small | Big | 1 | 10 | Big | Small | Small | Big | 1 |
| 3 | Small | Small | Big | Small | 1 | 11 | Big | Small | Big | Small | 1 |
| 4 | Small | Small | Big | Big | 1 | 12 | Big | Small | Big | Big | 1 |
| 5 | Small | Big | Small | Small | 2 | 13 | Big | Big | Small | Small | 2 |
| 6 | Small | Big | Small | Big | 1 | 14 | Big | Big | Small | Big | 2 |
| 7 | Small | Big | Big | Small | 1 | 15 | Big | Big | Big | Small | 2 |
| 8 | Small | Big | Big | Big | 1 | 16 | Big | Big | Big | Big | 2 |

where

*Univese of discourse U*={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}
*Conditional attribute a*=([110111101111]XOR[Codeword$_{ML}$])·$SF_{12}$
*Conditional attribute b*=([110111101111]XOR[Codeword$_{UC}$])·$SF_{12}$
*Conditional attribute c*=([001111101111]XOR[Codeword$_{ML}$])·$SF_{12}$
*Conditional attribute d*=([001111101111]XOR[Codeword$_{UC}$])·$SF_{12}$
*Decision value* $D_a$=$D_b$=$D_c$=$D_d$={S, B} here, Small≤80, Big>80.
*Evaluation attribute m*={1: [110111101111], 2: [001111101111]}

By the means of [7], Table 4 can be written as minimal decision rule. We can make lookup table which based on rough set based decision rule in each gray level value.

## 5 Experimental Results

The proposed techniques for subfield optimization are evaluated using five 512×512 test images. Fig. 4 shows the simulation results on the Lena image. The proposed

technique provides more pleasing visual quality and significantly improves edge and noise reduction by accurately optimizing codewords. Fig. 4(a) shows the original Lena image. Fig. 4(b) is obtained by the conventional binary coding that uses the subfield vector (1,2,4,8,16,32,48,48,48,48), while Fig. 4(c) is obtained by the subfield vector (1,2,4,8,16,32,42,44,52,54). In the proposed simulation, it is assumed that the subfield pattern and arrangement is given as Fibonacci sequence (1,2,3,5,8,13,19,2532,40,49,58). The results image is shown in Fig. 4(d). Table 5 compares the PSNR performances of the several algorithms.

(a) Original 256*256 "Lena" image (b) Method [4]

(c) Method [5] (d) Proposed method

**Fig. 4.** DFC evaluation results for three method with motion of 3 pixels/field: (a) Original image "Lena"; (b) Method [4]; (c) Method [5]; (d) Proposed method

**Table 5.** PSNR (dB) results of different subfield optimization methods for various images

| | SF=[1,2,4,8, 16,32,64,128] | Method [3] | Method [4] | Method [5] | Proposed Method |
|---|---|---|---|---|---|
| Airplane | 16.913659 dB | 17.523938 dB | 17.546208 dB | 17.572015 dB | 18.443768 dB |
| Baboon | 15.241433 dB | 18.661907 dB | 18.956357 dB | 19.089382 dB | 19.856587 dB |
| Barbara | 16.928961 dB | 19.693254 dB | 19.714049 dB | 19.864809 dB | 19.971561 dB |
| Finger | 13.397256 dB | 15.949619 dB | 16.221132 dB | 16.348113 dB | 16.943683 dB |
| Lena | 17.337039 dB | 19.768283 dB | 20.006524 dB | 20.219052 dB | 20.296184 dB |

## 6 Conclusion

In this work, we developed a rough set-based DFC reduction model for gray level disturbances in PDPs and showed that the disturbances can be avoided if the codeword of every gray level selected adaptively by lookup table. Given the rough set based optimal subfield lookup table, which is optimized for each gray level, dynamic false contour reduced significantly. Simulation results showed that the dynamic false contour can be effectively reduced by rough set based subfield optimization algorithm.

## Acknowledgment

This research was supported by Seoul Future Contents Convergence (SFCC) Cluster established by Seoul R&BD Program.

## References

1. B. L. Xu, Z. C. Xie, J. S. Tian, and H. B. Niu, “Improvement in PDP image quality by suppressing dynamic false contour while maintaining high brightness,” in SID Dig., 2003, pp. 455-457
2. Z. -J. Liu, C. -L. Liu, and Z. -H. Liang, “An adaptive subfield coding method for driving AC PDPs,” IEEE Trans. Plasma Science, vol. 34, pp. 397-402, April, 2006
3. S. Weitbruch, R. Zwing, and C. Correa, “PDP picture quality enhancement based on human visual system relevant features,” in Proc. IDW’00, 2000, pp. 699-702
4. T. Yamaguchi and S. Mikoshiba, “An improvement of PDP picture quality by using a modified-binary-coded scheme with a 3D scattering of motional artifacts,” IEICE Trans. Electron., vol. E80-C, no. 8, pp. 1079-1085, Aug. 1997
5. S, -H, Park and C. -W. Kim, “An optimum selection of subfield pattern for plasma displays based on genetic algorithm,” IEICE Trans. Electron., vol. E84-C, no. 11, pp. 1659-1666, No.v 2001
6. G. Odor, A. Krikelis, G. Vesztergombi, F. Rohrbach, “Effective Monte Carlo simulation on System-V massively parallel associative string processing architecture,” in Proc. PDP '99, 1999, pp. 281 - 288
7. Z. Pawlak - “Rough Sets - Theoretical Aspects of Reasoning about Data,” Kluwer Academic Publishers, 1991

# Image Digital Watermarking Technique Based on Kernel Independent Component Analysis*

Yuancheng Li, Kehe Wu, Yinglong Ma, and Shipeng Zhang

School of Computer Science and Technology,
North China Electric Power University, Beijing, 102206, China
{drlyc,wkh,ylma,spzh}@163.com

**Abstract.** In this paper, we present a novel image digital watermarking technique based on Kernel Independent Component Analysis (KICA). Use the nice characteristic of the KICA, which can results the blind separation of nonlinearly mixed signals, the imperceptibility and robustness requirements of watermarks are fulfilled and optimized. In the proposed scheme, the watermark image is first transformed by Arnold method, and then embedded into the lowest frequency subband in DWT domain. The recovery of owner's image is turning the watermarked image into DWT domains then use KICA to extract the watermark. Finally the watermark is transformed by Arnold method again, so we can get the original watermark image. Experimental results show that the proposed method gains better performance in robustness than that of ICA with respect to traditional image processing including cropping, filtering, add noise and JPEG image compression.

**Keywords:** image digital watermarking, Kernel Independent Component Analysis, wavelet transform.

## 1 Introduction

With the development of network and multimedia techniques, the transmission and access of digital media production (such as image, audio, etc) become more convenient. Although these techniques have brought us lots of benefits, the unrestricted copying and manipulation brings new issue of intellectual property rights. So the digital watermarking techniques which can solve these problems become very important. Generally, the digital watermarking is a technique which the owner via embedding a digital marking into a host media orders to verifying the ownership.

Recent years many techniques have been proposed, in which digital watermarking is quite efficient and promising. Watermarking methods operating in the wavelet domain [1] and in the discrete cosine transform (DCT) domain have been proposed [2], [3]. The watermark is embedded in the frequency domain more robust. Extensive experiment show that the watermark embeds in the discrete wavelet transform (DWT) domain is better than the DCT. In [4], the watermark is embedded in DCT domain and is obtained by applying Independent Component Analysis (ICA). In [5], digital

* Supported by the Doctor Degree Teacher Research Fund in North China Electric Power University.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 467–474, 2007.

image watermarking based on the combination of DWT and ICA is proposed. Francis R.Bach proposed Kernel Independent Component Analysis (KICA) theory [6]. The KICA algorithm outperforms many of presently known algorithms. Recently, Kezhong has proposed a method on KICA-based face recognition. He uses KICA to extract nonlinear independent feature. The result shows that KICA algorithm outperforms ICA algorithm in face recognition application [7].

Based on the watermarking techniques above, this paper proposes an image digital watermarking based on KICA algorithm. In this scheme, the watermark image is transformed by Arnold method then embedded into the lowest frequency subband DWT domain. The recovery of owner's image is turning the watermarked image into DWT domains then use KICA to extract the watermark. At last, the watermark is transformed by Arnold method again, so we get the original watermark image. Experimental results show that the proposed method gains better performance in robustness than that of ICA algorithm with respect to traditional image processing including cropping, filtering, add noise and JPEG image compression.

## 2 Kernel ICA

The ICA is based on a linear model; this theory is failed in the case of non-linearity. So the KICA was proposed in [6]. The main idea of KICA[8] is to map the input data into an implicit feature space $\phi : x \in R^n \to \phi(x) \in F$ Via nonlinear mapping , the data is converted easy data to analyses. Then we can use kernel function k (), which maps the training samples to a higher dimensional feature space.

$$\left\langle \Phi(X_i), \Phi(X_j) \right\rangle = K(x_i, x_j) \tag{1}$$

Several kinds of kernel functions are commonly adopted in kernel method. In this paper , we use Gaussian kernel in the following

$$K(x, y) = \exp\left(-\|y - x\|^2 / \delta^2\right) \tag{2}$$

The method can be summarized as follows:

Given image data $X = [x_1, x_2]$

1. Use kernel function $K_{i,j} = k(x_i, x_j)$

2. Calculate the eigenvectors matrix $\alpha$ and eigenvalues matrix $\Lambda^{\Phi}$ of kernel matrix K.

3. Compute whitened data

$$X_w^{\Phi} = \left(\Lambda^{\Phi}\right)^{-1} \alpha^T K \tag{3}$$

4. Compute separating matrix $W_I^{\Phi}$ by whitening transform

5. For the image y, the feature s can be calculated as

$$s = W_I^{\Phi} \left(\Lambda^{\Phi}\right)^{-1} \alpha^T K(x, y) \tag{4}$$

where K is a kernel function.

In the above algorithm, we only need to use the kernel function K instead of $\Phi$, but selecting an appropriate kernel function for a particular application is difficult. In general, the RBF network is preferred to train the classifier, because it is more powerful and more efficacious than Polynomial and Two-layer. So, we used Gauss kernel function in this paper. Use the nice characteristic of the KICA, which can results the blind separation of nonlinearly mixed signals, the imperceptibility and robustness requirements of watermarks are fulfilled and optimized.

## 3 The Proposed Scheme

The overview of the proposed watermarking embed and extract process is shown in Fig. 1 and Fig.2. The details of the scheme will be described in the following.

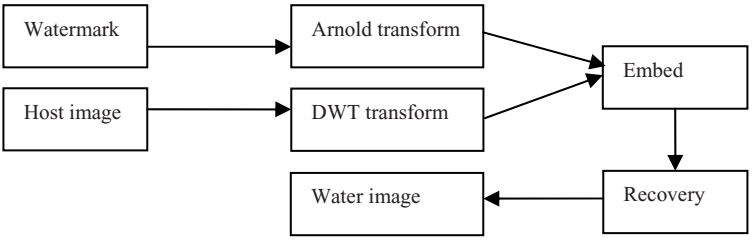



**Fig. 1.** The diagram of the watermarking embed

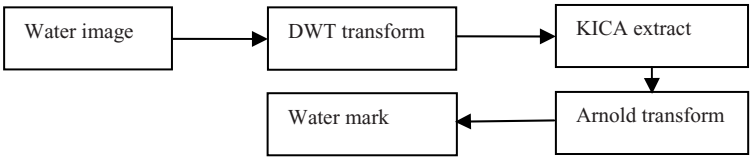



**Fig. 2.** The diagram of the watermarking extract

### 3.1 The Embedding Scheme

1. Digital image scrambling: In order to improve the security and robustness of the watermark, this paper use Arnold transform to scrambling the watermark. The transform can be shown as follows

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} (\mathrm{mod} N) \tag{5}$$

x,y ∈ 0,1,2,...,N-1. Because the Arnold transform is cycle, so if the watermark image transformed n (n<N) times we can get the scrambled image. The scrambled image is turned N-n times we can get the original watermark image.

2. Host image three levels DWT: the decomposition of an image using DWT comprises of a chosen low pass and a high pass filter. The low pass and high pass filters are applied to each row of data to separate the low frequency and the high frequency components. After an one-scale wavelet transform we can decompose an image into four finer scale subbands, labeled with LL1, HL1, LH1 and HH1. Continuous decomposing the lowest frequency subband, LLk, of each scale, we get four coarser scale subbands, LLk+1, HLk+1, LHk+1 and HHk+1. Because the details parts can be affect easily by noise so we can embed the watermark into the low-frequency domain (LL3).

3. The process of watermark embeds: we use 256x256 images and applied three-level DWT transform in this paper. We use a gray watermark image 'ncepu.tif' embedding the 32x32 subbands. The formula (6) is as model of mixture:

$$W = a \cdot W_i + W_y \tag{6}$$

where a is a parameter controlling the watermark strength, $W_i$ presents watermark, $W_y$ presents DWT transform domain, and $W$ Presents the watermarked image. We use three levels IDWT on image, so we can get the watermarked image.

### 3.2 Watermark Extraction Scheme

We use KICA to extract the original signal and watermark from the watermarked image. Before using KICA, we have to do some work to the watermarked image.

1. Get average of image signal: First we get the average of the image signal; there is an advantage to the watermark extraction. The method as this: X as image signal vector. At first, we use formula $m=E\{X\}$ to get average, then use $k=std(X-m)$ to get the result K. last we get the vector $X=X/K$. The same process is to the host image, so we got two signals.

2. Using two signals above to do KICA extraction, we got the watermark signal. Then we can use Arnold transform on the watermark signal to get the mark image.

## 4 Experimental Results

In order to validate the proposed method, we compared the performance of KICA with ICA method in this paper. We use four attack methods to test the two methods above. In the following experiments, a set of gray images of 256x256, "lena", "rice", "woman", "cameraman" are used for host images. The gray image "ncepu.tif" of 32x32, is used as watermark shown in Fig4. Due to the limitation of paper space, we will exhibit here only the experimental results obtained on the standard image

"lena.tif" of 256x256 shown in Fig.3, but similar results have been obtained with other images. In measure exactly, we use Peak Signal to Noise Ratio (PSNR) as a criterion which can evaluate the watermark strength. The normalized correlation (NC), which describes the correlation between the original watermark and the extracted watermark, can evaluate the performance of the proposed method.

**Fig. 3.** Original image

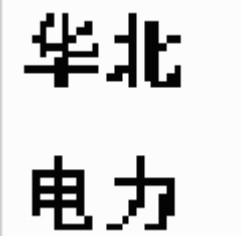


**Fig. 4.** Watermark

**Test 1:** attack free and parameter 'a' is equal to 0.26

**Fig. 5.** Watermarked image PSNR=24.7677

**Fig. 6.** Extracted watermark used by KICA, NC=0.9991

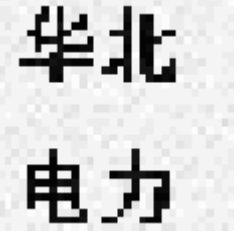


**Fig. 7.** Extracted watermark used by ICA, NC=0.9972

**Test 2:** Gaussian noise attack: strength parameter a is equal to 0.26 and attack parameter is 0.01

**Fig. 8.** Attackted image PSNR=17.8694

**Fig. 9.** Extracted used by KICA, NC=0.9610

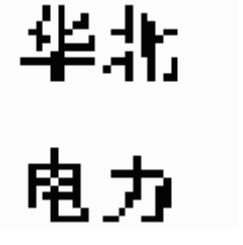


**Fig. 10.** Extracted used by ICA, NC=0.9492

**Test 3:** Gauss filter attack: strength parameter a=0.26 and attack parameter is [3, 1]

**Fig. 11.** Attackted image PSNR=21.0362

**Fig. 12.** Extracted used by KICA, NC=0.8472

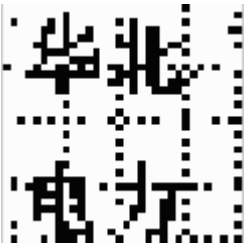


**Fig. 13.** Extracted used by ICA, NC=0.7202

**Test 4:** JEPG compression attack: strength parameter a=0.26 and compression parameter is 70%.

**Fig. 14.** Attackted image PSNR=22.5579

**Fig. 15.** Extracted used by KICA, NC=0.9961

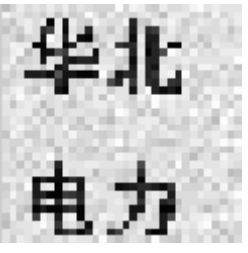


**Fig. 16.** Extracted used by ICA, NC=0.9758

**Test 5:** Image cropping attack: strength parameter a is equal to 0.26 image is cropped into 3/4

**Fig. 17.** Attackted image PSNR=5.7951

**Fig. 18.** Extracted used by KICA, NC=0.8603

**Fig. 19.** Extracted used by ICA, NC=0.0258

According to the Table 1, when the watermarked image is attacked, the NC shows that the KICA method is better than ICA method. Especially when the watermarked image is cropped 1/4, the ICA can not extracted the watermark.

**Table 1.** PSNR(dB) and NC results of above attack experiment

| | Attack free | Gaussian noise | Gauss filter | JEPG compression | Image crop |
|---|---|---|---|---|---|
| PSNR | 24.7677 | 17.8694 | 21.0362 | 22.5579 | 5.7951 |
| KICA NC | 0.9991 | 0.9610 | 0.8472 | 0.9961 | 0.8603 |
| ICA NC | 0.9972 | 0.9492 | 0.7202 | 0.9758 | --------- |

In order to compare the performance of KICA and ICA completely, we draw curves as show in Fig.20, Fig.21, Fig.22.

1. Gaussian noise attack: we use Gaussian noise to attack the watermarked image, and the parameter is from 0.024 to 0.06. The result is shown as Fig 20. Obviously, the NC value of the KICA is higher than ICA after a series Gaussian noise attacks.

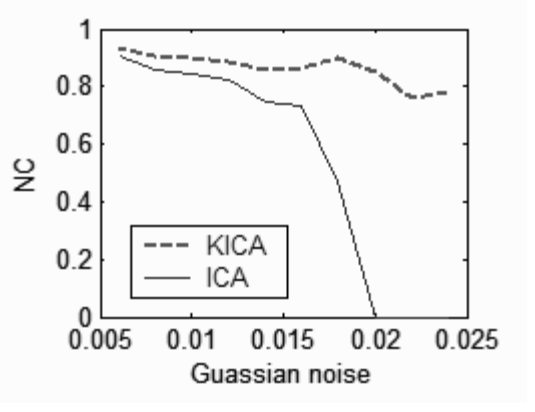



**Fig. 20.** Gaussian noise attack

2. Gaussian filter attack: the parameters are in range from [3, 0.5] [3, 1] to [3, 5]. The result is shown as fig 21. From the curve, we can draw concludes that the KICA is better than ICA after a series Gaussian filter attacks.

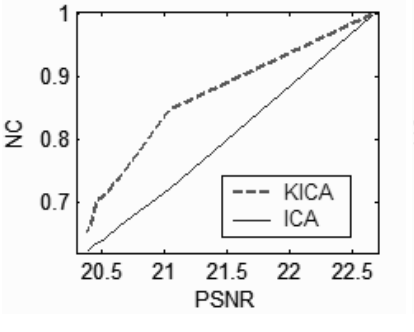



**Fig. 21.** Gaussian filter attack

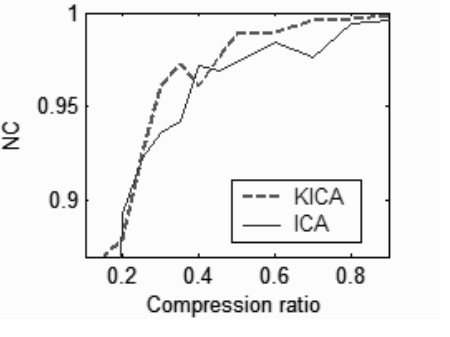



**Fig. 22.** JPEG Compression attack

3. JPEG compression attack: the parameters are in range 15% to 90%.The result is shown as Fig 22. As curves are shown above, there are two points the ICA better than KICA. But on the whole the KICA is better than ICA. When compression ratio is less than 10%, the ICA can not extract the watermark but KICA can do it.
4. Image cropped attack: If the watermarked image is cropped more than 1/6, the watermark can not be extracted by ICA, whereas the KICA can extract the watermark which is cropped by 1/4. When the watermarked image is cropped, it is obvious that the performance of the KICA method is better than the ICA.

## 5 Conclusion

This paper proposed a KICA based and DWT domain digital watermarking scheme. Extensive experimental results show that the proposed method is comparative with the ICA in extracting watermark, but gains better performance in robustness than that of ICA algorithm with respect to traditional image processing including cropping, filtering, add noise and JPEG image compression. The reason of the results seems that the KICA is a nonlinear kernel method, and some of image process attacks are nonlinear transform, KICA have the advantage of the ability of extracting nonlinear signals compared with the ICA. The shortcoming of the proposed scheme is that the KICA spend more time to extract the watermark than the ICA. It will be the work in the future. Besides, the paper focuses on applying the presented method to digitized images although the same approach can be used for other media, such as music or video.

## References

1. Kundur, D., Hatzinakos, D.: Digital Watermarking Using Multiresolution Wavelet Decomposition. In proc. ICAssp, (1998)2969–2972.
2. Cox.J.kilian, I.J., Leighton, F.T., shamoon, T.: Secure Spread Spectrum Watermarking for Multimedia. IEEE Trans Image Processing, Vol.6(1997)1673–1687
3. Barni, M., Bartolini, F., Cappellini, V.: A DCT-domian System for Robust Image Watermarking. In Signal Processing, Vol.66(1998)357–372.
4. Gonzalez-Serrano, F.j., Molina-Bulla,H.Y., Murillo-Fuente,J.Js: Independent Component Analysis Applied to Digital Image Watermarking. In Proc.IEEE,Int Conference on Acoustics, Speech, and Blind Signal Separation,(ICASSP'01),Vol.3(2001)1997–2000
5. JuLiu, XiangangZhang, Jiande Sun: A New Image Watermarking Scheme Based on DWT and ICA. In IEEE International Conference on Neural Networks & Signal processing (ICNNSP'03), (2003)1489–1492
6. Francis R.Bach: Kernel Independent Component Analysis: Journal of Machine Learning Research,Vol.3(2002) 1–48
7. Zhang Yan-Kun, Liu Chong-qing: Face Recognition Based on Kernel Independent Components Analysis. Optical Technique ,Vol.30 No.5(2004)613–622
8. Scholkopf ,B. , Smola ,A.: Learning with Kernels, Gambridge,Mass:MIT Press(2002)
9. Jian Cheng, Qingshan Liu, hanqing Lu: Texture Classification Using Kernel Independent Component Analysis. ICPR, Vol.17 (2004) 1051–4651

# Image Pattern Recognition Using Near Sets*

Christopher Henry and James F. Peters

Department of Electrical and Computer Engineering,
University of Manitoba
Winnipeg, Manitoba R3T 5V6 Canada
{chenry,jfpeters}@ee.umanitoba.ca

**Abstract.** The problem considered in this paper is how to recognize similar objects based on the detection of patterns in pairs of images. This article introduces a new form of classifier based on approximation spaces in the context of near sets for use in pattern recognition. By way of introducing the basic approach, nonlinear diffusion is used for edge detection and object contour extraction. This form of image transformation makes it possible to compare the contours of objects in pairs of images. Once the contour of an image has been identified, it is then possible to construct approximation spaces based on vectors of probe function measurements associated with selected image features. In this article, the only feature considered is *contour*, which leads to many contour probe functions. The contribution of this article is a new form of classifier, based on approximation spaces, for use in image pattern recognition.

**Keywords:** Approximation space, image, feature extraction, near sets, nonlinear diffusion, pattern recognition, rough sets.

## 1 Introduction

The problem considered in this paper is how to recognize similar objects based on the detection of patterns in pairs of images. The proposed solution to this problem utilizes approximation spaces introduced by Zdzisław Pawlak (see, *e.g.*, [1,2]), later generalized in [3], and further refined in [4]. In this paper, the approach to approximation space-based image pattern recognition is strictly limited to discovering similar objects in images based on object contours. Specifically, a user creates a template image in the form of a "sketch." The goal is then to recognize all images within a set of samples that match the template. The results reported in this article are limited to three known objects, two that match the template, and one that does not. Nonlinear diffusion is used for image smoothing and object contour extraction. The traditional approach suggested in [5], for

* The authors thank the anonymous reviewers for their very helpful suggestions. This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant 185986 and Manitoba Hydro grant T277.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 475–482, 2007.

recognition of an object in an image $\mathcal{I}$ with a suspected match in an image $I_1$ is performed by comparing probe function values in

$$\mathcal{I} \approx (I_1)T \Leftrightarrow |f(\mathcal{I}) - f(I_1)| < \varepsilon, \forall f, \varepsilon \in [0, 1]$$

where $\mathcal{I}$ is approximately the same as $I_1$ after some transformation $T$, iff $|f(\mathcal{I}) - f(I_1)| < \varepsilon$ for all $f$ associated with, *e.g.*, the *contour* of an object in an image. In contrast, the approach taken in this article is to match a sketch drawn by a user with an object contained in an image by recording contour probe function values of both objects in a data table and constructing an approximation space. Lower rough coverage values are then used to determine if the template image is a match to the unknown image. The contribution of this article is a new form of approximation space-based classifier for use in image pattern recognition.

This article is organized as follows. An approach to edge detection is briefly presented in Section 2. Sections 3 and 4 briefly present the fundamentals of approximation spaces with respect to near sets and their application to pattern recognition, respectively. Finally, sample results of the proposed approach are presented in Section 5.

## 2 Edge Detection

Sketches inherently represent edges of the objects we are trying to match. Consequently, a natural place to start is with image segmentation, which is the process of partitioning an image into regions that are representative of the objects within the image [6]. This can be accomplished by identifying the edges which are high contrast regions of an image. This article uses nonlinear diffusion image filtering to achieve segmentation (and subsequently perform edge detection). This method is based on actual physical processes such as the diffusion of heat in a metal bar [7,8,9]. The process is considered nonlinear because the diffusivity becomes a decreasing function of the magnitude of the gradient, since the gradient will produce a large value in areas of large contrast (edges within the image) [8]. The result is that uniform (low gradient magnitude) areas within the image undergo more diffusion than areas with high contrast (high gradient magnitude). An example of nonlinear diffusion is given in Fig. 1 using the nonlinear diffusion toolbox for Matlab [7].

## 3 Approximation Spaces

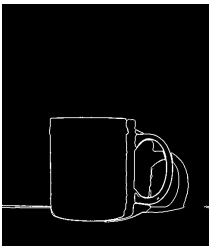
This section introduces a view of approximation spaces defined in a slightly modified manner in comparison with the original definition in [3]. Any generalized approximation space (GAS) is a tuple

$$GAS = (U, A, N_r, \nu_B),$$

where $U$ is the universe (elements of $U$ may be, for example, objects, behaviours, or perhaps states), $A$ is a set of probe functions (such that $x \in U$ and $f(x) \in A$),

1.1: Original image

1.2: Segmentation using nonlinear diffusion



1.3: Binary contour using nonlinear diffusion

**Fig. 1.** Results of nonlinear diffusion on an image

$N_r$ is a neighbourhood family function and $\nu_B$ is an overlap function defined by (1).

$$\nu_B : \mathcal{P}(U) \times \mathcal{P}(U) \longrightarrow [0,1], \tag{1}$$

where and $\mathcal{P}(U)$ is the powerset of $U$ [4]. Eq. 1 maps a pair of sets to a number in $[0,1]$ representing the degree of overlap between the sets of objects with features defined by $B \subseteq A$ [3]. For each subset $B \subseteq A$ of probe functions, define the binary relation $\sim_B = \{(x, x') \in U \times U : \forall f \in B, f(x) = f(x')\}$. Since each $\sim_B$, is an equivalence relation (*i.e* the $Ind_B$ indiscernibility relation), for $B \subset A$ and $x \in U$ let $[x]_B$ denote the equivalence class, or *block*, containing $x$, that is,

$$[x]_B = \{x' \in U : \forall f \in B, f(x') = f(x)\} \subseteq U.$$

If $(x, x') \in \sim_B$ (also written $x \sim_B x'$) then $x$ and $x'$ are said to be *B-indiscernible*. Then define a family of neighborhoods $N_r(A)$, *i.e.*,

$$N_r(A) = \bigcup_{B \subseteq P_r(A)} [x]_B,$$

where $P_r(A) = \{B \subseteq A : |B| = r\}$ for any $r$ such that $1 \leq r \leq |A|$. That is, $r$ denotes the number of features used to construct families of neighborhoods. Information about a sample $X \subseteq U$ can be approximated from information contained in $B$ by constructing a $N_r(B)$-lower approximation

$$N_r(B)_* X = \bigcup_{x:[x]_B \subseteq X} [x]_B,$$

and a $N_r(B)$-upper approximation

$$N_r(B)^* X = \bigcup_{x:[x]_B \cap X \neq \emptyset} [x]_B.$$

Then $N_r(B)_* X \subseteq N_r(B)^* X$ and the boundary region $BND_{N_r(B)}(X)$ between upper and lower approximations of a set $X$ is defined to be the complement of $N_r(B)_* X$, *i.e.*

$$BND_{N_r(B)}(X) = N_r(B)^*X \backslash N_r(B)_*X = \{x \in N_r(B)^*X \mid x \notin N_r(B)_*X\}.$$

A family of neighborhoods $N_r(B)$ is near a set $X$ *iff* $|BND_{N_r(B)}(X)| \geq 0$. This means every rough set is a near set but not every near set is a rough set. Lastly, use the notation $B_j(x)$ to denote a subset of $N_r(B)$, where $j \in B$. Put

$$\nu_j(B_j(x), N_r(B)_*X) = \begin{cases} \frac{|B_j(x) \cap N_r(B)_*X|}{|N_r(B)_*X|}, & \text{if } N_r(B)_*X \neq \emptyset, \\ 1, & \text{if } N_r(B)_*X = \emptyset, \end{cases}$$

where $\nu_j$ is a specialized form of rough coverage (see, *e.g.*, [10]).

## 4 Approximation Spaces and Pattern Recognition

It is now possible to formulate a basis for object recognition, which parallels the traditional formulation of pattern recognition. Let $X = D$ represent a decision class containing all elements of $U$ obtained from the template image using probe functions from $B$. $D$ represents a standard for classifying images. Observe that a non-zero rough coverage value $\nu_j$ means that $B_j(x)$ contains elements that are members of the decision class $D$. Further, a larger number of non-zero coverage values implies that a significant number of blocks contain elements that are part of the decision class (the template image). Consequently, the ratio of non-zero coverage values to total coverage values can be used as a new form of image classifier. Put,

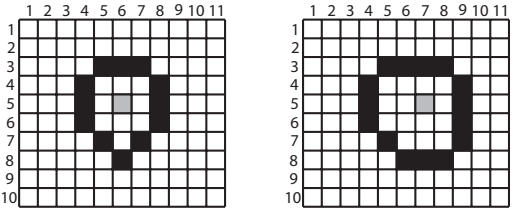
$$C_\nu(GAS) = \frac{|\{\nu_j : \forall B_j(x) \in N_r(B), \nu_j > 0\}|}{|\{\nu_j : \forall B_j(x) \in N_r(B)\}|},$$

where $C_\nu(GAS)$ is the ratio of non-zero coverage values to total coverage values obtained from a specific $GAS$ (for convenience we simply write $C_\nu$). Then recognition of objects that are approximately the same is defined by comparing non-zero coverage ratios using

$$\mathcal{O} \approx (\mathcal{O}_{id})T \Leftrightarrow C_\nu > \varepsilon,$$

where $\varepsilon \in [0, 1]$. That is to say, the object $\mathcal{O}$ is approximately the same as $\mathcal{O}_{id}$ after some transformation $T$ whenever $C_\nu$ is greater than some $\varepsilon$.

By way of an illustration of the utility of approximation spaces, a near set approach to pattern recognition is briefly considered here. Recall that the goal of this process is to match a template with an unknown image. Let us define a decision system as a data table $(U, A)$ such that $A$ contains a distinguished probe function $d$ representing a decision. Thus the set $D \subseteq U$ consists of all the elements for which $d(x) = 1$. The first step in creating a decision system is to create the data table. Such tables will then be used to set up approximation spaces to determine the degree that an object in an image resembles the template.



2.1: Template image 2.2: Unknown image

**Fig. 2.** Contour comparison

The approach used in this article is to create a data table from two images where all elements associated with the template make up the decision class $D$. Two such tables are given in Tables 1 and 2 created from the images shown in Fig. 2. Table 1 represents the ideal case in which the template in Fig. 2.1 is compared with itself. Similarly, Table 2 contains data obtained from comparing the template in Fig. 2.1 with the unknown image given in Fig. 2.2.

**Table 1.** Decision system for Fig. 2.1

| $x_i$ | Probe functions $f_0 \cdots f_{10}$ | | | | | | | | | | | $d$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{x_0}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $\mathbf{x_1}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $\mathbf{x_2}$ | 0 | 0 | 0 | 0 | 3 | 2 | 3 | 0 | 0 | 0 | 0 | 1 |
| $\mathbf{x_3}$ | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 1 |
| $\mathbf{x_4}$ | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 1 |
| $\mathbf{x_5}$ | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 1 |
| $\mathbf{x_6}$ | 0 | 0 | 0 | 0 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 1 |
| $\mathbf{x_7}$ | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 |
| $\mathbf{x_8}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $\mathbf{x_9}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $x_{10}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_{11}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_{12}$ | 0 | 0 | 0 | 0 | 3 | 2 | 3 | 0 | 0 | 0 | 0 | 0 |
| $x_{13}$ | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| $x_{14}$ | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
| $x_{15}$ | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| $x_{16}$ | 0 | 0 | 0 | 0 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| $x_{17}$ | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_{18}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_{19}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 2.** Dec. sys. for Figs. 2.1 and 2.2

| $x_i$ | Probe functions $f_0 \cdots f_{10}$ | | | | | | | | | | | $d$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{x_0}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $\mathbf{x_1}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $\mathbf{x_2}$ | 0 | 0 | 0 | 0 | 3 | 2 | 3 | 0 | 0 | 0 | 0 | 1 |
| $\mathbf{x_3}$ | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 1 |
| $\mathbf{x_4}$ | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 1 |
| $\mathbf{x_5}$ | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 1 |
| $\mathbf{x_6}$ | 0 | 0 | 0 | 0 | 3 | 0 | 3 | 0 | 0 | 0 | 0 | 1 |
| $\mathbf{x_7}$ | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 |
| $\mathbf{x_8}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $\mathbf{x_9}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $x_{10}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_{11}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_{12}$ | 0 | 0 | 0 | 0 | 4 | 3 | 2 | 3 | 0 | 0 | 0 | 0 |
| $x_{13}$ | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| $x_{14}$ | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| $x_{15}$ | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| $x_{16}$ | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| $x_{17}$ | 0 | 0 | 0 | 0 | 0 | 4 | 3 | 4 | 0 | 0 | 0 | 0 |
| $x_{18}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $x_{19}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Moreover, to populate the tables, the coordinates of the centroid of each image are calculated to find the geometric centre of the image (the grey pixels in the centres of the contours). Next, the distances from the centroid are calculated using the taxicab metric for each point on the contour of each image. Note, distances are only reported for points on the contour. Lastly, what follows is an example showing how to obtain $C_\nu = 1$ for Table 1. Similar calculations produce a value of $C_\nu = 0.592593$ for Table 2. Observe that Table 2 produces a lower value of $C_\nu$ since Fig. 2.1 and Fig. 2.2 are not identical.

Decision class: $D = \{x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$

$B_j(x)$ : $\{\nu_j\}$
$B_{f_0}(x_0) = \{x_0, x_1, x_2,$
$x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}\}$ : $\{1.0000\}$
$B_{f_1}(x_0) = \{x_0, x_1, x_2,$
$x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}\}$ : $\{1.0000\}$
$B_{f_2}(x_0) = \{x_0, x_1, x_2,$
$x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}\}$ : $\{1.0000\}$
$B_{f_3}(x_0) = \{x_0, x_1, x_2, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{16}, x_{17}, x_{18}, x_{19}\}$ : $\{1.0000\}$
$B_{f_3}(x_3) = \{x_3, x_5, x_{13}, x_{15}\}$ : $\{1.0000\}$
$B_{f_3}(x_4) = \{x_4, x_{14}\}$ : $\{1.0000\}$
$B_{f_4}(x_0) =$
$\{x_0, x_1, x_3, x_4, x_5, x_7, x_8, x_9, x_{10}, x_{11}, x_{13}, x_{14}, x_{15}, x_{17}, x_{18}, x_{19}\}$ : $\{1.0000\}$
$B_{f_4}(x_2) = \{x_2, x_6, x_{12}, x_{16}\}$ : $\{1.0000\}$
$B_{f_5}(x_0) =$
$\{x_0, x_1, x_3, x_4, x_5, x_6, x_8, x_9, x_{10}, x_{11}, x_{13}, x_{14}, x_{15}, x_{16}, x_{18}, x_{19}\}$ : $\{1.0000\}$
$B_{f_5}(x_2) = \{x_2, x_{12}\}$ : $\{1.0000\}$
$B_{f_5}(x_7) = \{x_7, x_{17}\}$ : $\{1.0000\}$
$B_{f_6}(x_0) =$
$\{x_0, x_1, x_3, x_4, x_5, x_7, x_8, x_9, x_{10}, x_{11}, x_{13}, x_{14}, x_{15}, x_{17}, x_{18}, x_{19}\}$ : $\{1.0000\}$
$B_{f_6}(x_2) = \{x_2, x_6, x_{12}, x_{16}\}$ : $\{1.0000\}$
$B_{f_7}(x_0) = \{x_0, x_1, x_2, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{16}, x_{17}, x_{18}, x_{19}\}$ : $\{1.0000\}$
$B_{f_7}(x_3) = \{x_3, x_5, x_{13}, x_{15}\}$ : $\{1.0000\}$
$B_{f_7}(x_4) = \{x_4, x_{14}\}$ : $\{1.0000\}$
$B_{f_8}(x_0) = \{x_0, x_1, x_2,$
$x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}\}$ : $\{1.0000\}$
$B_{f_9}(x_0) = \{x_0, x_1, x_2,$
$x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}\}$ : $\{1.0000\}$
$B_{f_{10}}(x_0) = \{x_0, x_1, x_2,$
$x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}\}$ : $\{1.0000\}$

$N_r(B)_*X = \{\emptyset\}$
$C_\nu = 1$

## 5 Results

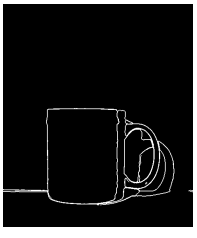
Again by way of illustration of the approach to recognizing similar objects in images, template images of tea cups (see Fig. 3) were compared to unknown sample image contours (see Fig. 4) obtained by nonlinear diffusion. The goal was to obtain a higher value of $C_\nu$ when comparing a sketch of a tea cup with that of a contour obtained from an image of a tea cup. As shown in Table 3, the template image in both cases produces a higher ratio of non-zero lower coverage vales when compared to the contour of a tea cup than that of the fire hydrant.

These results are promising, since they show that a lower approximation space in the context of near sets can be used for pattern recognition. However, there is still much to be investigated. For instance, observing the effects of translation and

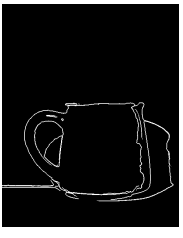
3.1: Sketch of 1st tea cup

3.2: Sketch of 2nd tea cup

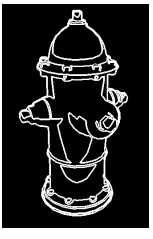
**Fig. 3.** Sample sketches (template images)



4.1: First tea cup



4.2: Second tea cup



4.3: Fire hydrant

**Fig. 4.** Sample image contours

rotation on the sample images. This method should be translation and rotation independent (within some small $\varepsilon$) due to the fact that centroid distances that should not change on rotation or translation of the image as long as the entire object is still present.

Other problems should be investigated as well. For example, a comparison of other edge detection techniques and the nonlinear diffusion process is required. This method was selected because it had already been implemented. However, it may not be best suited to the task at hand. Also, other edge detection methods may be more attractive in terms of timing. Currently, the proposed method takes several minutes to obtain the gradient. This is fine when comparing two images, but is unrealistic when searching through an archive containing thousands of them. Similarly, other forms of feature extraction should be explored as well. At present, only one feature, namely, *contour* has been considered. Contour probe function measurements constituting the top five distances from the centroid are used. It may be that there are better features or a combination of multiple features that can be used to provide better results. Also, both ratios were higher for the tea cup images than the fire hydrant, however, there some difference between the results obtained for both tea cups. Consequently, thresholding techniques (such as neural networks) need to also be investigated to determine when it is sufficient to say a sample image being considered "matches" the sketch drawn

**Table 3.** Sample Results

| Decision Systems | Lower coverage ratios |
|---|---|
| Template image Fig. 3.1 vs. tea cup contour Fig. 4.1 | 0.521186 |
| Template image Fig. 3.1 vs. fire hydrant contour Fig. 4.3 | 0.437100 |
| Template image Fig. 3.2 vs. tea cup contour Fig. 4.2 | 0.515041 |
| Template image Fig. 3.2 vs. fire hydrant contour Fig. 4.3 | 0.413616 |

by a user. Finally, a direct comparison to current image classifiers is needed to determine if this method is an improvement.

## 6 Conclusion

This article introduces an approximation space-based classifier for use in image pattern recognition. Initial results are promising inasmuch as templates (obtained from sketches) of target objects (*e.g.*, tea cups) produce higher non-zero coverage ratios when compared to objects in test images and low coverage ratios when compared to other objects (*e.g.*, fire hydrants). However, further instigation is required before definite conclusions can be made about the proposed image classifier.

## References

1. Pawlak, Z.: Classification of objects by means of attributes. Polish Academy of Sciences, Technical Report PAS 429 (1981)
2. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences **177** (2007) 3–27
3. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae **27** (1996) 245–253
4. Peters, J.: Near sets. special theory about nearness of objects. Fundamenta Informaticae **75** (2007) 407–433
5. Pavel, M.: Fundamentals of Pattern Recognition. Marcel Dekker, Inc., NY (1993)
6. Gonzalez, R., Woods, R.: Digital Image Processing. 2nd edn. Prentice-Hall, Toronto (2002)
7. D'Almeida, F.: Nonlinear diffusion toolbox (2003)
8. Weickert, J.: Anisotropic Diffusion in Image Processing. Ph.d. dissertation, University of Kaiserslautern (1996)
9. Weickert, J.: Applications of nonlinear diffusion in image processing and computer vision. Acta Mathematica Universitatis Comenianae **70** (2001) 33–50
10. Peters, J., Henry, C.: Reinforcement learning with approximations spaces. Fundamenta Informaticae **71** (2006) 323–349

# Robotic Target Tracking with Approximation Space-Based Feedback During Reinforcement Learning*

Daniel Lockery and James F. Peters

Department of Electrical and Computer Engineering,
University of Manitoba
Winnipeg, Manitoba R3T 5V6 Canada
dlockery@ee.umanitoba.ca, jfpeters@ee.umanitoba.ca

**Abstract.** This paper presents a method of target tracking for a robotic vision system employing reinforcement learning with feedback based on average rough coverage performance values. The application is for a line-crawling inspection robot (ALiCE II, the second revision of Automated Line Crawling Equipment) designed to automate the inspection of hydro electric transmission lines and related equipment. The problem considered in this paper is how to train the vision system to track targets of interest and acquire useful images for further analysis. To train the system, two versions of Watkins' Q-learning were implemented, the classical single-step version and a modified strain using an approximation space-based form of what we term *rough feedback*. The robot is briefly described along with experimental results for the two forms of the Q-learning control algorithm. The contribution of this article is an introduction to a modified version of Q-learning control with rough feedback to monitor and adjust the learning rate during target tracking.

**Keywords:** Approximation space, target tracking, monocular vision, reinforcement learning, rough sets, Q-learning.

## 1 Introduction

The problem considered in this paper is how to influence a system with a control algorithm that learns from past experience of actions for any given situation. The system consists of a robotic platform with a monocular vision apparatus used to track and acquire images of desired targets. Control of the vision system is accomplished via the Q-learning algorithm [16,18]. Two versions of this method are included, Watkins' single-step approach and a modified version of Watkins' algorithm that employs a rough set approach.

* The author gratefully acknowledges the suggestions and insights by Andrzej Skowron, David Gunderson, Maciej Borkowski, Christopher Henry concerning topics in this paper. This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant 185986.

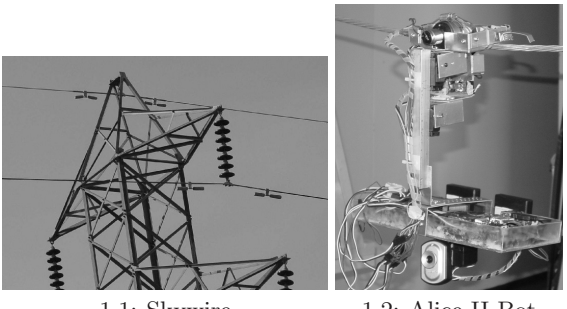
A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 483–490, 2007.

Several examples of recent work have been reported for the target tracking control problem, including [1,2,6,11,12,13]. The robotic platform as well as reinforcement learning are a common thread. The contribution of this article is the implementation and comparison of a rough coverage feedback value for adjusting the learning rate of a control algorithm versus the classical implementation.

This article is organized as follows. A brief introduction to the robotic platform is provided in Section 2. The rough set and approximation space theory employed in the control algorithm are briefly discussed in Section 3. Section 4 includes a look at the reinforcement learning control algorithms, followed by the experimental results in Section 5.

## 2 The ALiCE II Robot

The ALiCE II device is an inspection robot designed to crawl along a 9mm sky wire that exists at the uppermost part of large transmission and distribution hydro power lines (see Fig. 1). The bot must function in a harsh environment buffeted by wind, hampered by electromagnetic fields, rain, and huge swings in temperature.



1.1: Skywire 1.2: Alice II Bot

**Fig. 1.** Environment for Target Tracking System

A camera system is attached to the base of ALiCE II to allow for a complete view of all conductors and the tower structure beneath it partially shown in Fig. 1.1. Through low-level control, the robot is able to navigate back and forth along the sky wire and pause as required to acquire images of interest from the camera. Fig. 1.2 provides a front overhead view and a rear view of the experimental setup demonstrating the line crawling robot. There are a number of different types of targets of interest including insulators, pins, conductors, and tower structures. Once a target has been sighted, tracking is used to help gather the best images by maximizing the surface area and compensating for external influences like wind speed. The camera mounted on the underside of the robot's platform consists of two servo motors that pan and tilt the camera as required for positioning.

## 3 Rough Sets and an Example Approximation Space

This section includes a brief introduction to rough sets followed by a description of an approximation space and an associated example. Representation of objects and features occur in the form of data tables to simplify the processing steps [12]. Let $U$ denote a non-empty finite set called a *universe* and let $\mathcal{P}(U)$ be the power set of $U$ (*i.e.* the family of all subsets of $U$). In this paper, elements of $U$ correspond to observed behaviors. A *feature* $\mathcal{F}$ of elements in $U$ is measured by an associated probe function $f = f_{\mathcal{F}}$ whose range is denoted by $\mathcal{V}_f$, called the *value set* of $f$; that is, $f : U \to \mathcal{V}_f$. There may be more than one probe function for each feature. For example, a feature of a behavior may be the reward obtained for performing an action. The similarity or equivalence of objects can be investigated quantitatively by comparing a sufficient number of object features by means of probes [10]. For present purposes, we identify the set of features with the set of associated probe functions, and hence we use $f$ rather than $f_{\mathcal{F}}$ and call $\mathcal{V}_f = \mathcal{V}_{\mathcal{F}}$ a set of feature values. If $F$ is a finite set of probe functions for features of elements in $U$, the pair $(U, F)$ is called a *data table*, or *information system* (IS).

For each subset $B \subseteq F$ of probe functions, define the binary relation $Ind_B = \{(x, x') \in U \times U : \forall f \in B, f(x) = f(x')\}$. Since each $Ind_B$ is an equivalence relation, for $B \subset F$ and $x \in U$ let $[x]_B$ denote the equivalence class, or *block*, containing $x$, that is,

$$[x]_B = \{x' \in U : \forall f \in B, f(x') = f(x)\} \subseteq U.$$

If $(x, x') \in [x]_B$, then $x$ and $x'$ are said to be *indiscernible* with respect to all feature probe functions in $B$, or simply, $B$-*indiscernible*. Information about a sample $X \subseteq U$ can be approximated from information contained in $B$ by constructing a $B$-lower approximation

$$B_*X = \bigcup \{[x]_B \mid [x]_B \subseteq X\},$$

and a $B$-upper approximation

$$B^*X = \bigcup \{[x]_B \mid [x]_B \cap X \neq \emptyset\},.$$

The $B$-lower approximation $B_*X$ is a collection of blocks of sample elements that can be classified with full certainty as members of $X$ using the knowledge represented by features in $B$. By contrast, the $B$-upper approximation $B^*X$ is a collection of blocks of sample elements representing both certain and possibly uncertain knowledge about $X$. Whenever $B_*X \subsetneq B^*X$, the sample $X$ has been classified imperfectly, and is considered a rough set. In this paper, only $B$-lower approximations are used.

### 3.1 Approximation Spaces

The original discussion of approximation spaces was introduced by Pawlak [7] and has since been expanded to a generalized version [14,15]. Approximation

spaces provide the basis for a new form of reinforcement learning based on acceptable patterns of behaviour viewed in the context of a line-crawling robot. This section briefly introduces approximation spaces and the generalized version with rough coverage.

The original definition of an approximation space provided by Pawlak [7] contained the pair $(U, Ind)$. Where $U$ corresponds to a non-empty finite set and $Ind$ represents an indiscernibility relation on subsets of $U$ ($Ind \subset U \times U$) [7]. More recently, a generalized approximation space was introduced by Skowron and Stepaniuk [14], [15] represented by a triple, $(U, I, \nu)$.

- $U$ is a non-empty set of objects, and $\mathcal{P}(U)$ is the powerset of $U$,
- $I : U \rightarrow \mathcal{P}(U)$ is such that $x \in I(x)$ for any $x \in U$,
- $\nu : \mathcal{P}(\mathrm{U})$ x $\mathcal{P}(\mathrm{U}) \rightarrow [0, 1]$ is an overlap function (inclusion or coverage).

Similar to the classical description, $U$ corresponds to a non-empty, finite set. The uncertainty function $I$ maps each $x \in U$ to a neighbourhood such that a given object $x$ is associated with a set of objects that are similar in some respects. This function can also be used to help define a covering of $U$ [12] specialized relative to $B_*X$ and blocks in the $Ind_B$-partition of $U$. Pertaining to coverage of sets, $\nu$ is a measure of overlap and is referred to as *inclusion* or *coverage* depending upon the configuration of the expression. In this paper, *rough coverage* is used as a measure of set overlap as the basis for a measure of Q-learning performance.

$$\nu(X, Y) = \begin{cases} \frac{|X \cap Y|}{|Y|}, & \text{if} \quad \mathrm{Y} \neq \emptyset, \\ 1 \quad , & \text{if} \quad \mathrm{Y} = \emptyset. \end{cases} \tag{1}$$

The value of $\nu$ represents the degree of coverage, ranging from 1, when the sets are equal to one another ($X = Y$) to the minimum value of 0, when there are no common elements in $X$ and $Y$ ($X \cap Y = \emptyset$) [15]. Anything in between 0 and 1 represents at least some degree of overlap between the two sets in question.

We used the average rough coverage as a performance metric. Let $d$ denote a partial function that represents a decision about an object based on evaluation of $D$ denote the set $D = \{x \in U \mid d(x) = 1(accept)\}$. $\nu$ is computed by substituting $B_*D$ or known accepted tracking behaviours for $Y$ and each $[x]_B$ substituted for $X$ in (1). $\bar{\nu}$ (average coverage) is computed by averaging individual $\nu$ values, and provides the backbone for a new form of Q-learning represented by Alg. 1.

## 4 Reinforcement Learning Control Algorithms

The line crawling robot uses reinforcement learning to discover how to control the movements of a digital camera used to track randomly moving targets. The control algorithm selected was single-step Q-learning and it was modified to incorporate rough coverage as a feedback performance metric and then compared to the classical algorithm. This section includes a brief overview of Q-learning and the modification made to provide rough feedback followed by the formal algorithm.

The Q-learning algorithm learns based on the action (or $Q$) value associated with each state as opposed to using the value function associated with being in a given state. Q-learning was developed by Watkins, formally reported in 1989 [18]. The Q-learning method in its simplest form is a single step temporal difference learning method that is capable of maximizing the action value of an agent regardless of the policy being followed [3,16,18]. The concept of a single step algorithm is that it looks into the future one step in advance when estimating the best course of action to take from the current state. The best action is generally the choice that maximizes the future discounted reward available from all actions pertaining to the current state. It is important to note that for Q-learning it does not matter what policy is being followed, it will always maximize the action value [18]. Q-learning falls into the category of off-policy algorithms [16]. This implies that the decisions made for selecting a course of action do not necessarily follow the policy that is exploring the state space [16].

During the initialization of the Q-learning algorithm, a policy must be established to determine a preliminary (most likely sub-optimal) mapping from states to actions. As indicated in Alg. 1, this policy is not greedy, which means that there is a possibility that it will not always follow the action with the highest immediate reward. There must be at least a small chance that this policy will explore alternate actions providing potential visits to sub-optimal actions that may return greater long term rewards. Throughout the learning process of single step Q-learning, state-action pairs are examined one step ahead. Rather than following the original non-greedy policy that is selecting actions, a greedy policy is used to determine the best action to take from the current state.

$$Q(s,a) \longleftarrow Q(s,a) + \alpha[r + \gamma \cdot max_{a'} Q(s',a') - Q(s,a)] \tag{2}$$

From Eq. 2, we see that although the non-greedy policy is selecting actions, the update is affected by the maximizing greedy policy inherent in the Q-learning update rule. The algorithm parameters that can be found in Q-learning are $\gamma$, and $\alpha$ which correspond to the discount factor and the learning rate step size adjustment, respectively.

The term *episode* refers to a length or the amount of time steps present in an episode of analysis for the reinforcement learning process. The difference from classical Q-learning for the rough feedback version lies in the update rule presented in Eq. 2, it was modified to include the average rough coverage value to adjust the value of $\alpha$. As the average rough coverage value increases, the value of $\alpha$ decreases, reducing the step size change to prevent overshoot when approaching a desirable policy. Conversely, for situations where the average rough coverage is low, the value of $\alpha$ will increase, causing the adjustment to be greater in the hopes of finding more suitable behaviours.

## 5 Experimental Results

The experimental work was done using a camera system identical to that found on the line crawling robot (seen in Fig. 1. Noise was introduced into the

**Algorithm 1.** The Q-Learning Method With Rough Feedback

**Input :** States, $s \in S$, Actions $a \in A(s)$, Initialize Q(s,a), $\bar{\nu}$, $\alpha$, $\gamma$, $\pi$ to an arbitrary policy (non-greedy);
**Output :** Optimal action value Q(s,a) for each state-action pair;
**while** *True* **do**
  **for** $(i = 0; i \leq \#ofepisodes; i++)$ **do**
    Initialize $s$ and data table;
    Choose a from $s$, using policy derived from Q;
    **for** *Repeat(for each step of episode):* **do**
      Take action $a$; observe reward, $r$, and next state, $s'$;
      Record state, action, and associated reward in data table;
      Q(s,a) $\longleftarrow$ Q(s,a) + $(1 - \overline{\nu})\alpha[r + \gamma max_a Q(s', a') - Q(s, a)]$;
      s $\longleftarrow$ s'; a $\longleftarrow$ a';
      until $s$ is terminal;
    **end**
    Generate $\bar{\nu}$ from results recorded in data table for current episode;
    Update new value of $\bar{\nu}$;
    Clear data table;
  **end**
**end**

environment in the form of random movements of the platform, similar to what would be experienced in light wind conditions. This section includes preliminary results and a brief discussion of their implications.

The parameters of the Q-learning algorithms were selected as 0.1 for both the learning and discount rate ($\alpha$ and $\gamma$ respectively). The length of the experiments were 5 minutes each and a sample result is included (See Fig. 2). The number of samples differs between the two methods since the rough feedback method requires more calculations to generate the rough coverage values. The trade-off of spending more time processing data is improved results as seen in the results. The rough-feedback implementation is able to adjust to the movements of the camera and this can be seen with the reduced RMS error pertaining to the target location. As the target moves, the algorithms attempt to learn the best movements in any given situation for tracking. The adjustable learning rate allows the rough-feedback method to react more quickly or more slowly depending on how well it performs. The classical method maintained a reasonable rate of performance but since the environment was somewhat noisy, it was unable to adjust as quickly and provided a stable error rate around 1 pixel greater than that of the rough feedback method.

## 6 Conclusions

The preliminary results are promising for the rough feedback implementation of the Q-learning algorithm. The error rate recorded was a significant improvement over the classical implementation. The difference in the number of samples over

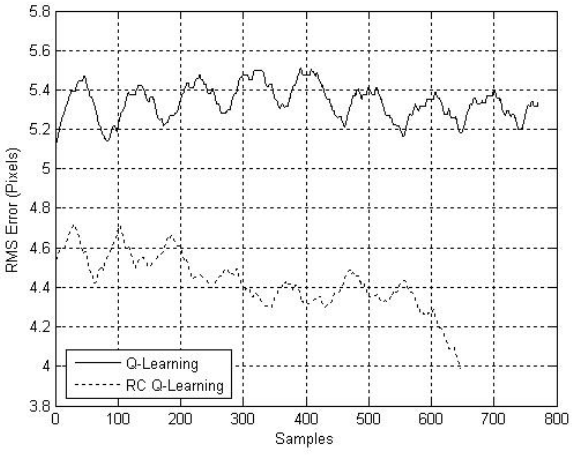



**Fig. 2.** Plotted results, Samples (x-axis) vs. RMS pixel error (y-axis)

a five minute period was 125 extra samples for the classical method over the rough feedback method. This was expected since creating the data tables and the required processing to generate the average rough coverage values is more work.

The performance boost is significant enough that it is worth considering as an alternative approach to target tracking in our environment. The only drawback is the amount of processing and extra time required which necessitates more battery power for an autonomous robot and the extra time weakens the tracking capabilities, since it pauses during heavy computations to preserve power. An attempted possible solution was to reduce the episode sizes when generating data tables. This resulted in less computational power and time but at the cost of reducing the sample size of behaviours drawn upon when revising the learning rate. The extra computational cost reduced the number of samples by 16% and improved the accuracy by an average of 20% over the classical method. Reducing the sample size did not adversely affect the learning process as the experimental environment did not have many variables. However, in more complex environments, reducing the sample size could introduce poorer results if the selected samples did not reflect the general behaviour.

In conclusion, the preliminary results the new form of Q-learning demonstrate that adjusting the approximation space-based learning rate based on average rough coverage provides improved accuracy for the target tracking process. Further experimental work is required to verify consistency and an optimal episode size as well as the time required to generate the rough coverage values. Future work on Q-learning will include ways to improve the proposed method. This is definitely possible by hearkening back to the run-and-twiddle adaptive learning method introduced by Oliver Selfridge in 1984.

## References


1. Distante, C., Anglani, A. and Taurisano, F.: Target Reaching by Using Visual Information and Q-learning Controllers, *Autonomous Robots*, vol.9, no. 1, pp. 41-50, Springer Netherlands, August 2000.
2. Gaskett, C.: *Q-Learning for Robot Control*, Ph.D. Thesis, Department of Systems Engineering, Supervisor: A. Zelinsky, The Australian National University, 2002.
3. Kaelbling, L.P, Littman, M.L. and Moore, A.W.: Reinforcement Learning: A Survey, *Journal of Artificial Intelligence Research*, vol. 4, pp. 237-248, May 1996.
4. Komorowski J., Polkowski, L. and Skowron, A.: Rough Sets: A Tutorial, In S.K. Pal and A. Skowron, editors, Rough-Fuzzy Hybridization: A New Method for Decision Making, Springer-Verlag, Singapore(in Print), 1998.
5. Munakata, T. and Pawlak, Z.: Rough Control Application of Rough Set Theory to Control, *Fourth European Congress on Intelligent Techniques and Soft Computing*, vol. 1, pp. 209-218, Aachen, Germany, September 1996.
6. Nakamura, T. and Asada, M.: Motion Sketch: Acquisition of Visual Motion Guided Behaviors, *Fourteenth International Joint Conference on Artificial Intelligence*, vol. 1, pp. 126-132, Montreal, Canada, 1995.
7. Pawlak, Z.: Rough Sets, *International Journal of Computer and Information Sciences* vol. 11, no. 5, pp. 341-356, 1982.
8. Pawlak, Z.: *Rough Sets. Theoretical Reasoning about Data*, Theory and Decision Library, Series D: System Theory, Knowledge Engineering and Problem Solving, vol. 9, Kluwer Academic Pub., Dordrecht, 1991.
9. Pawlak, Z.: Rough Set Theory and its applications, *Journal of Telecommunications and Information Technology*, 3, 2002.
10. Peters, J.F.: Classification of objects by means of features. In: Fogel, D., Mendel, J., Yao, X., Omori, T. (Eds.), Proc. First IEEE Symposium on Foundations of Computational Intelligence (FOCI'07), 1-5 April 2007, *in press*.
11. Peters, J.F., Henry, C., Lockery, D., Borkowski, M., D. Gunderson, Ramanna, S.: Line-Crawling Bots That Inspect Electric Power Transmission Line Equipment, *The 3rd International Conference on Autonomous Robots and Agents (ICARA)*, Palmerston North, New Zealand, December 2006.
12. Peters, J.F., Lockery, D.A., and Ramanna, S.: Monte Carlo Off-Policy Reinforcement Learning: A Rough Set Approach, *Proceedings of The Fifth International Conference on Hybrid Intelligent Systems*, November 2005.
13. Peters, J.F., Borkowski, M., Henry, C., Lockery, D.: Monocular vision system that learns with approximation spaces. In: Ella, A., Lingras, P., Slezak, D., Suraj, Z.: Rough Set Computing: Toward Perception Based Computing. Idea Group Publishing, Hershey, PA (2006), 1-22.
14. Skowron, A. and Stepaniuk, J.: Tolerance Approximation Spaces, *Fundamenta Informaticae*, vol. 27, no. 2/3, pp. 245-253, 1996.
15. Stepaniuk, J.: Approximation Spaces, Reducts and Representatives, in L. Polkowski and A. Skowron (Eds.), *Rough Sets in Knowledge Discovery 2, Studies in Fuzziness and Soft Computing*, 19, pp. 109-126, Heidelberg: Springer, 1998.
16. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*, Cambridge, MA: The MIT Press, 1998.
17. Swiniarski, R.W. and Skowron, A.: Rough set methods in feature selection and recognition, *Pattern Recognition Letters 24*, pp. 833-849, 2002.
18. Watkins, C.J.C.H.: Learning from Delayed Rewards, *Ph.D. Thesis, supervisor: Richard Young, King's College*, University of London, UK, pp. 25-36, May 1989.

# Web Based Health Recommender System Using Rough Sets, Survival Analysis and Rule-Based Expert Systems

Puntip Pattaraintakorn[1], Gregory M. Zaverucha[2], and Nick Cercone[3]

[1] Department of Mathematics and Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand 10520
kppuntip@kmitl.ac.th
[2] School of Computer Science, University of Waterloo, Ontario, Canada N2L 3G1
gzaveruc@cs.uwaterloo.ca
[3] Faculty of Science and Engineering, York University, Ontario, Canada M3J 1P3
ncercone@yorku.ca

**Abstract.** We propose a health recommendation system architecture using rough sets, survival analysis approaches and rule-based expert systems. Our main goal is to recommend clinical examinations for patients or physicians from patients' self reported data. Such data will be treated as condition attributes, while survival time from a follow-up study will be treated as the target function. We have amalgamated rough set theory, relational databases, statistics, soft computing and several pertinent techniques to generate a hybrid intelligent system for survival analysis. This study represents the completion of our system by adding a recommendation module.

**Keywords:** Rough sets, Survival analysis, Recommender system.

## 1 Introduction

Given user profile information, recommender systems attempt to predict items (e.g., music, books, web pages) in which a user might be interested. Thus, general recommender systems were developed for e-commerce. Several studies have shown that systems predict the target user's requirement accurately (e.g., movies [1] and hardware retail [2]).

**Motivation and Applications.** Our recommendation system was designed with the goal of providing accurate, low-cost medical recommendations. In countries where health care costs are prohibitively expensive, this system can provide a free alternative. While not seeking to be a drop-in replacement or perfect substitute for professional medical advice, there are many cases where some information is better than nothing. Consider the following examples. A patient can only afford a limited number of tests, but cannot determine which ones should take priority. There may be an inexperienced, or no doctor available and the patient would like a second opinion.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 491–499, 2007.

Regular check-ups allow doctors to diagnose diseases early, allowing wider options for treatment. Many patients cannot afford these regular visits, and instead only are examined when there is a problem. A free recommendation from our system may allow earlier diagnosis. Rural areas with limited access to medical professionals can also benefit from more frequent medical advice. To provide this service, we aim to eventually deploy this system in low-cost public kiosks. In this paper, we describe the design of a web-based prototype.

We introduce in Sect. 2 preliminaries and notation of some survival analysis, rough sets, recommendation rules, new measurements and recommendation systems. In Sect. 3 we propose a web based health recommendation system architecture. We demonstrate the applicability of a part of our proposed system on geriatric data set in Sect. 4. In Sect. 5 we provide conclusion and add some general remarks of what next steps will be taken.

## 2 Preliminaries and Notation

### 2.1 Survival Analysis

Survival analysis [3] is a branch of statistics that studies time-to-event data. Death or failure is called an *event* in the survival analysis literature. Survival analysis is called *reliability analysis* in engineering, and *duration analysis* in economics. Bazan et al. [4] applied the Kaplan-Meier method and the Prognostic Index to head and neck cancer patients, then used rough sets generate decision rules. They illustrated that rough sets can contribute to a medical expert system. Zaluski et al. applied rough sets to construct decision rules that classify a binary target function: cancer recurrence [5]. The authors provided a comparison of several approaches compared to rough sets. Song et al. analyzed the same geriatric data (Sect. 4) to evaluate the potential of rough sets, artificial neural networks and the frailty index in predicting survival time [6]. As reported, the prediction performance of rules induced by rough sets approaches were comparable with results of using other learning systems [5,6,11]. Survival analysis (see our previous studies [7,8,9,10,11]) attempts to answer questions such as:

"Is diabetes (or others) a significant risk factor for geriatric patients?"
"What are the rules for survival time predictions of geriatric patients?"

In this paper, we present the novel approach to answer this question:

"According to the analysis of risk factors and survival time prediction, what are the recommending clinical examinations for prolonging the survival time?"

**Kaplan-Meier Survival Analysis.** [3] The proportion of the population of patients who would survive a given length of time under the same circumstances is given by the Kaplan-Meier method as shown in (1). $S$ is based on the probability that each patient survives at the end of a time interval, on the condition that the patient was present at the start of the time interval.

$$\hat{S}(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right) \tag{1}$$

where $t_i$ is the period of study at point $i$, $d_i$ is number of events up to point $i$ and $n_i$ is number of patients at risk prior to $t_i$.

While the Kaplan-Meier method focused on a single risk factor, the Cox proportional hazard model is used for multiple attributes. Our previous study applied the Cox proportional hazard model to analyze multiple attributes [10].

### 2.2 Rough Set Theory

In the early 1980s, rough set theory was developed by Pawlak [12]. Rough sets were redefined using database operations, the computing times were improved remarkably by using the database system directly [7,8,9,10,11]. This is one reason why rough set theory is a leading approach in *soft computing*. In addition to the soft computing ability, rough sets can derive decision rules from a decision table efficiently and effectively. In general, a decision table $S$ consists of rows labelled by *objects* and columns labelled by *attributes*. The entries in the table are described by *attribute values*. There are several ways to represent our knowledge. If our goal is to express what must happen or what does happen when certain conditions are met, then we can use decision rules. We usually express each decision rule as an IF... THEN... statement. For example, *"IF $C$ is $c_1$ THEN $D$ is $d_1$"*, where $c_1$, $c_2$ and $d_1$, $d_2$ are values that correspond to attribute $C$ and $D$, respectively. Issues related to preprocessing and derive decision rules from survival analysis data can be found in [4,7,8,10,11]) and postprocessing in [9]. In this article, we further analyze the postprocessing step of the acquired rules.

### 2.3 Recommendation Rules

In general, a decision rule can have more than one antecedent (combined either by AND or OR logical operations). Similarly, a decision rule can have more than one consequent. Antecedents or consequents can describe unary relations, e.g.,

Bangkok is raining. Bangkok has a lot of traffic.

Antecedent or consequent can describe binary relations, e.g.,

Bangkok has more traffic than Toronto.

Decision rules can describe relations, e.g.,

IF Bangkok has a lot of traffic THEN travel on the subway.
IF Bangkok is raining THEN bring an umbrella.

Here, we analyzed the relationship within decision rules and added expert knowledge to generate a *recommendation rule* [13]. The following is an example of a recommendation rule that takes a set of inputs and gives advice as a result.

IF Bangkok has a lot of traffic AND Bangkok is raining
THEN travel on the subway AND bring an umbrella.

We introduce two measurements that will be used in our system with our recommendation rules. Given $n$, the total number of facts in the database, $m$

the total number of rules, $R_i$ is rule number $i$ and $p_{ij}$ is the priority of each fact $j$ for rule $i$, the *rule priority* is calculated from the value-pair condition attribute that matched the fact as follows:

$$RULE\_PRIORITY(R_i) = \frac{1}{n}\sum_{j=1}^{n} p_{ij} \tag{2}$$

where $i$ runs through $1, 2, \ldots, m$.

In some cases, a patient's input might match more than one recommendation rule and several rules are triggered. Our system will calculate the *recommendation score* from the rule quality [14], rule cover and rule priority (2) as follows:

$$RECOMMENDATION\_SCORE(R_i) = (RULE\_QUALITY(R_i) + RULE\_PRIORITY(R_i)) + \left(\frac{RULE\_COVER(R_i)}{n}\right). \tag{3}$$

Only one recommendation rule will be fired if the highest recommendation score exceeds the *recommendation threshold*, $\alpha$. Otherwise, multiple recommendation rules will be fired.

### 2.4 Recommender Systems

A recommender system is a decision support system that provides a personalized solution in a brief and clear form from the user's given information. In this study, we construct a recommender system based on rule-based systems. A recommender system consists of three components: (i) a database of rules, (ii) a database of facts and (iii) an inference engine [13]. First, the knowledge base contains a set of rules that represent the knowledge possessed by the system (e.g., from previous analysis). Second, the database of facts represents inputs to the system that are used to cause actions or derive recommendations. Finally, the inference engine is the part of the system that generates a recommendation. The inference engine uses the rules and facts when inferring recommendations.

Our system uses deduction to reach a recommendation from a set of antecedents, this is called *forward chaining*. This approach begins with a set of facts and rules, and tries to find a way of using those rules and facts to deduce a recommendation or a suitable action. To apply forward chaining, the first step is to take the facts from the fact database and check if any (combination) of these matches all the antecedents of the rules in the rule database. When all the antecedents of a rule are matched by facts in the database, then this rule is *triggered*. The rule is then *fired*, which means its conclusion is added to the facts database. If the conclusion of the rule that has fired is an action or a recommendation, then the system makes recommendations or actions take place. In our study, the only action is to provide recommendations.

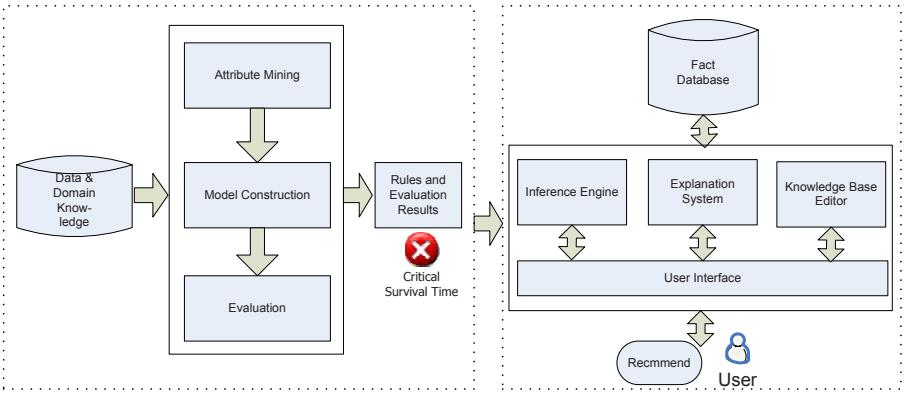



**Fig. 1.** Our proposed web based recommender system

## 3 Methodology

### 3.1 System Design

Our health recommender system architecture is depicted in Fig. 1. In the left half of Fig. 1, we show the data analysis techniques described and evaluated in previous work [7,8,9,10,11]. The output is a set of rules, which form the basis for recommendations. These rules create the fraction of the fact database which supports the Inference Engine; tasked with providing recommendations based on the user's input and the known facts. The fact database also contains the information required by the explanation system, the part of the system which helps users understand their recommendation. This user support can take the form of explanation of symptoms and terminology, side effects of treatments or contact information of local resources.

Once the rules have been created and the fact database populated, the entire system can be deployed on a single machine (in the case of a kiosk) or a client-server model may be used (in the case of web-based deployment). Our first prototype will use a web-based user interface because of the simplicity of implementation and deployment. The recommendation system can continue to be provided from a web interface, but public kiosks are required for maximum accessibility (many of the target users do not own personal computers).

### 3.2 Privacy and Data Collection

In the prototype phase, collection of data will provide important feedback on the system's use, allowing for improvements. Collection of personal data creates privacy issues, which are magnified when the data describes one's health. Users may avoid a system if they are not confident that it protects their private information. For this reason, privacy (as well as security) should be considered during the development of any system which stores personal information.

The exact mechanism used to provide privacy depends on the details of deployment, however the basic strategy will be to collect data *anonymously*. When collecting information from the user, the system will not require or store, Identifying information. Stronger privacy assurance [15] requires the development of a model, and is outside the scope of this work.

## 4 A Case Study

In this section, we provide a case study based on a geriatric data set [7,8,9,10,11]. In our previous studies, rough sets approaches and several pertinent techniques were used to generate reducts and dispensable attributes [7,8,10,11]. Table 1 displays the decision rules, their rule qualities [14] and rule cover [14] from [7,8,9,10,11]. Attributes {takemed} and {walk} are dispensable attributes [8].

**Table 1.** Geriatric survival prediction rules database

| Decision Rules | Rule Quality | Rule Cover |
|---|---|---|
| IF (edlevel!=2 or 4) and (0<shopping≤0.5) and (meal≤0)and (trouble>0) and (livealo>0) and (sneeze≤0) and (hbp≤0) and (eyetrou≤0) and (feet≤0) and (nerves≤0) and (sex>1)<br>THEN (survival time = 7–18 months) | 1.969635 | 13 |
| IF (housew>0) and (cough≤0) and (tired≤0) and (hbp≤0) and (eyetrou≤0) and (kidney>0) and (bowels≤0) and (nerves≤0) and (2<age6≤4)and (sex>1)<br>THEN (survival time = 7–18 months) | 1.935979 | 12 |
| ... | ... | ... |
| IF (edlevel!=2) and (eyesi≤0) and (health>0) and (trouble≤0) and (sneeze≤0) and (heart≤0) and (arthriti≤0) and (eyetrou>0) and (dental≤0) and (chest≤0) and (kidney≤0) and (bladder≤0) and (feet≤0) and (skin≤0) and (age6≤1)<br>THEN (survival time = 7–18 months) | 1.012614 | 1 |

**Table 2.** Fact database of geriatric

| Facts | Priority | Facts | Priority | Facts | Priority | Facts | Priority |
|---|---|---|---|---|---|---|---|
| eyesi < 0.25 | 0.3 | hear < 0.25 | 0.3 | eat = 0 | 0.1 | cough = 0 | 0.2 |
| tired = 0 | 0.1 | sneeze = 0 | 0.2 | hbp = 0 | 0.5 | heart = 0 | 1.0 |
| arthriti = 0 | 1.0 | stroke = 0 | 0.8 | parkinso = 0 | 1.0 | eyetrou = 0 | 0.2 |
| eartrou = 0 | 0.2 | dental = 0 | 0.4 | chest = 0 | 1.0 | stomac = 0 | 0.9 |
| kidney = 0 | 0.9 | bladder = 0 | 0.8 | bowels = 0 | 0.8 | diabet = 0 | 1.0 |
| feet = 0 | 0.1 | nerves = 0 | 0.9 | skin = 0 | 0.6 | fracture = 0 | 0.9 |
| age6 > 3 | 0.7 | | | | | | |

The Kaplan-Meier method, $p$-value, Log-rank, Brewslow and Tarone-Ware tests [8] were used to generate the life time table and Kaplan-Meier survival curves. Our system then analyzed the curves together with the results from rough sets to obtain the risk factors [8,10,11]. ELEM2 by An and Cercone [14] was successively used to derive survival prediction rules with its heuristic approaches. Only the rules that have critical survival time (7-18 months) are selected for to provide recommendations. Table 2 shows our fact database. The priority range is $[0, 1]$ (where 1 is the highest priority). Please note that, {age} is included in the fact database but we will not recommend any test for this fact. We then use

**Table 3.** Recommendation rules of geriatric in the knowledge base

| Recommendation Rules | Rule Quality | Rule Cover | Rule Priority |
|---|---|---|---|
| IF (edlevel!=2 or 4) and (0<shopping≤0.5) and (meal≤0) and (trouble>0) and (livealo>0) and (sneeze≤0) and (hbp≤0) and (eyetrou≤0) and (feet ≤0) and (nerves≤0) and (sex>1)<br>THEN (test sneeze) and (test hbp) and (test eyetrou) and (test feet) and (test nerves) | 1.969635 | 13 | 0.076 |
| IF (housew>0) and (cough≤0) and (tired≤0) and (hbp≤0) and (eyetrou ≤0) and (kidney>0) and (bowels≤0) and (nerves≤0) and (2<age6≤4) and (sex>1)<br>THEN (test cough) and (test tired) and (test hbp) and (test eyetrou) and (test bowel) and (test nerve) | 1.935979 | 12 | 0.136 |
| ... | ... | ... | ... |
| IF (edlevel!=2) and (eyesi≤0) and (health>0) and (trouble≤0) and (sneeze ≤0) and (heart≤0) and (arthriti≤0) and (eyetrou>0) and (dental≤0) and (chest≤0) and (kidney≤0) and (bladder≤0) and (feet≤0) and (skin≤0) and (age6≤1)<br>THEN (test eyesi) and (test sneeze) and (test heart) and (test arthriti) and (test dental) and (test chest) and (test kidney) and (test bladder) and (test feet) and (test skin) | 1.012614 | 1 | 0.252 |

**Table 4.** Example input and output

| Example input | Example output |
|---|---|
| IF (edlevel!=2 or 4) and (0 < shopping < 0.5) and (meal≤0) and (trouble≥0) and (livealo ≥0) and (sneeze≤0) and (hbp≤0) and (eyetrou≤0) and (feet≤0) and (nerves≤0) and (sex>1)<br>THEN (survival time = 7-18 months) | Recommended clinical examinations: sneeze, high blood pressure, eye trouble, feet, nerves |

the facts from Table 2 to calculate the rule priority (Sect. 2.3) and add it to the rules in Table 1. The rules in Table 1 are transformed to decision rules to recommend tests and stored in our knowledge base (Table 3).

For an example of how the rule priority is calculated, take the first rule in Table 1. Its rule priority is equal to $(0.2 + 0.5 + 0.2 + 0.1 + 0.9)/25 = 0.076$ where $n = 25$. The conclusion (action) of the rule are the tests: *(test sneeze), (test hbp), (test eyetrou), (test feet)* and *(test nerves)* that match the facts in Table 2. When a user inputs their data into our system, if the prediction is a critical case (survival time 7–18 months), our recommendation system (right part of Fig. 1) will start its analysis. For example, suppose input was as in Table 4. This patient's input matches the first recommendation rule and does not match any other rule. We trigger the first recommendation rule and fire the action of first recommendation rule. The tests recommended to the user are shown in Table 4.

## 5 Concluding Remarks and Future Works

We have proposed a health recommendation system architecture using rough sets, survival analysis and rule-based expert systems. Our system was designed with the goal of providing accurate, low-cost clinical examination recommendations given patients' self reported data. In countries where health care costs

are prohibitively expensive, this system can provide a free alternative. Our system generates not only decision rules but also applicable recommendations for patients. Our future works will complete the implementation of the prototype system. Heuristics to include clinical examination costs will also be investigated.

## Acknowledgements

This research was supported by NSERC, Canada and KMITL research fund, Thailand. Thanks also to Arnold Mitnitski and Kanlaya Naruedomkul and the anonymous reviewers for their helpful comments.

## References

1. Blatter, M., Zhang, Y., Maslow, S.: Exploring an Opinion Network for Taste Prediction: An Empirical Study. Physica A **373** (2007) 753–758
2. Liu, D., Shih, Y.: Hybrid Approaches to Product Recommendation Based on Customer Lifetime Value and Purchase Preferences. J. Syst. Software **77** (2005) 181–191
3. Elisa, L.T., John, W.W.: Statistical Methods for Survival Data Analysis. 3rd edn. John Wiley & Sons New York (2003)
4. Bazan, J., Skowron, A., Slezak, D., Wroblewski, J.: Searching for the Complex Decision Reducts: The Case Study of the Survival Analysis. In: Bazan, J., Skowron, A., Slezak, D., Wroblewski, J. (Eds.): Foundations of Intelligent Systems. Lecture Notes in Artificial Intelligence, Vol. 2871. Springer-Verlag, Berlin Heidelberg (2003) 160–168
5. Zaluski, J., Szoszkiewicz, R., Krysinski, J., Stefanowski, J.: Rough Set Theory and Decision Rules in Data Analysis of Breast Cancer Patients. In: Peters, J.F., Skowron, A., Grzymala-Busse, J.W., Kostek, B., Swiniarski, R.W., Szczuka, M.S. (Eds.): Transactions on Rough Sets I. Lecture Notes in Computer Science, Vol. 3100. Springer-Verlag, Berlin Heidelberg (2004)
6. Song, X., Mitnitski, A., MacKnight, C., Rockwood, K.: Assessment of Individual Risk of Death Using Self-report Data: An Artificial Neural Network Compared to a Frailty Index. J. Am. Geriatr. Soc. **52** (2004) 1180–1184
7. Pattaraintakorn, P., Cercone, N., Naruedomkul, K.: Hybrid Intelligent Systems: Selecting Attributes for Soft-Computing Analysis. Proc. 29th IEEE Computer Society International Computer Software and Applications Conference (COMPSAC 2005), IEEE Press (2005) 319–325
8. Pattaraintakorn, P., Cercone, N., Naruedomkul, K.: Selecting Attributes for Soft-computing Analysis in Hybrid Intelligent Systems. In: Slezak, D., Yao, J.T., Peters, J.F., Ziarko, W., Hu, X. (Eds.): Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing. Part II. Lecture Notes in Artificial Intelligence, Vol. 3642. Springer-Verlag, Berlin Heidelberg (2005) 698–708
9. Pattaraintakorn, P., Cercone, N., Naruedomkul, K.: Rule Analysis with Rough Sets Theor. Proc. 2006 IEEE International Conference on Granular Computing (GrC 2006), IEEE Press (2006) 582–585
10. Pattaraintakorn, P., Cercone, N.: Hybrid Rough Sets-Population Based System. In: Peters, J.F. (et al.) (Eds.): Transactions on Rough Sets VII. Lecture Notes in Computer Science, Vol. 4400. Springer-Verlag, Berlin Heidelberg (2007) 190–205

11. Pattaraintakorn, P., Zaverucha, G., Cercone, N., Naruedomkul, K.: A Foundation of Rough Sets Theoretical and Computational Hybrid Smart System for Survival Analysis. Artif. Intell. Med. (under submission)
12. Pawlak, Z.: Rough sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
13. Coppin, B.: Artificial Intelligence Illuminated. Jones and Bartlett Publishers, Inc., Sudbury, Mass (2004)
14. An, A., Cercone, N.: ELEM2: A Learning System for More Accurate Classifications. In: Mercer, R.E., Neufeld, E. (Eds.): Advances in Artificial Intelligence. Lecture Notes in Computer Science, Vol. 1418, Springer-Verlag, Berlin Heidelberg (1998) 426–441
15. Sweeney, L.: k-Anonymity: A Model for Protecting Privacy. Int. J. Uncertain. Fuzz. **10(5)** (2002) 557–570

# RBF Neural Network Implementation of Fuzzy Systems: Application to Time Series Modeling

Milan Marček[1] and Dušan Marček[2]

[1] Faculty of Philosophy and Science, Silesian University, 746 01 0pava, Czech Republic & MEDIS Nitra, Ltd., Pri Dobrotke 659/81, 949 01 Nitra-Dražovce, Slovak Republic
marcek@fria.utc.sk
[2] Faculty of Philosophy and Science, Silesian University, 746 01 0pava, Czech Republic & Faculty of Management Science and Informatics, University of Zilina 010 26 Zilina, Slovak Republic
dusan.marcek@fpf.slu.cz, dusan.marcek@fri.utc.sk

**Abstract.** At first, we discuss the basic structure of the fuzzy system as a simple yet powerful fuzzy modeling technique. Neural networks and fuzzy logic models are based on very similar underlying mathematics. The similarity between RBF networks and fuzzy models is noted in detail. Then, we propose the extension of RBF neural networks by the cloud model. Time series approximation and prediction by applying RBF neural networks or fuzzy models and comparisons between the various types of RBF networks and statistical models are discussed at length.

**Keywords:** Probabilistic time-series models, fuzzy system, classic and soft RBF network, cloud models, granular computing.

## 1 Introduction

In most studies of identification of processes by using input-output data, it is assumed that there exists a functional structure between the input and the output. It is, however, very unrealistic to substitute a non-linear process by a simple linear mapping. More sophisticated approaches are frequently considered. The fuzzy systems and the fuzzy controllers are among them. The fuzzy system consists of series of fuzzy rules each of which takes the form of a "if ... then ..." sentence.

Basically there are two ways for automatic formation of fuzzy relations. The first one is based on clustering methods [7]. The second is based on neural networks [4]. Neural networks can adaptively generate the fuzzy rules in a fuzzy system by supervised or unsupervised competitive learning, which is also in fact the product-space clustering technique. We have illustrated this approach in [6]. A class of neural networks, i. e. the feed-forward networks have been proven to be capable of representing various complex nonlinear input-output mappings.

In this paper, we consider the approximation ability of ARMA models and models based on fuzzy systems to "explain" the behaviour of time-series variables. In addition, we explore some of the more important specifications associated with approximation of time-series variables using RBF networks.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 500–507, 2007.

The paper is organised into 5 Sections. The main constructs of fuzzy system architecture will be briefly introduced in the next Section. In Section 3 we introduce the architecture of RBF neural networks with the aim to highlight its mathematical similarity to the fuzzy system. In Section 4, we demonstrate the approximation abilities of RBF neural networks on an application and compare them with the statistical approach. Conclusions are offered in Section 5.

## 2 Fuzzy Systems

Fuzzy systems are structures to estimate input-output functions of modelled systems from sample data. This section concentrates on the basic principles of identifying input-output functions of systems using fuzzy systems. Fuzzy systems theory have been recently consolidated and presented by B. Kosko [5].

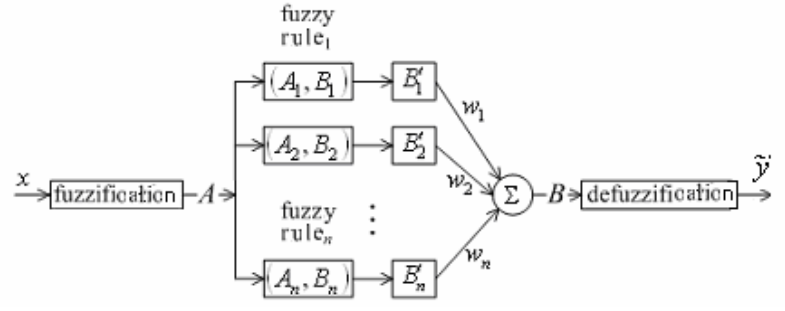



**Fig. 1.** Fuzzy system architecture

The basic fuzzy system architecture is shown in Fig. 1. In this architecture the fuzzy system maps input fuzzy sets $A$ to output fuzzy sets $B$. The fuzzy inference computes the output fuzzy sets $B_i'$, weights them with the weights $w_i$, and sums to produce the output fuzzy set $B$, i.e.

$$B = \sum_i w_i B_i' \tag{1}$$

The fuzzy system is distributed and consists of a series of a separate fuzzy rules (relations) of the type of *if* $A_i$ *then* $B_i$.

Centroidal output converts fuzzy sets vector $B$ to a scalar. The most popular centroidal defuzzification technique uses all the information in the fuzzy distribution $B$ to compute the crisp $y$ value as the centroid $\tilde{y}$ or centre of mass of $B$, i. e.

$$\tilde{y} = \int_{-\infty}^{\infty} y\mu_B(y)dy \Big/ \int_{-\infty}^{\infty} \mu_B(y)dy \tag{2}$$

where $\mu_B$ represents the union of all clipped output fuzzy sets. When the output membership functions are singletons, then, in the case of an $\Re^k \to \Re$ function, Eq. (2) becomes

$$\tilde{y} = \sum_{j=1}^{n} y_j \mu_j(x) / \sum_{j=1}^{n} \mu_j(x) \tag{3}$$

where $y_j$ stands for the centre of gravidity of the *j*th output singleton, the notation $\mu$ is used for a membership function and *n* denotes the number of rules.

As mentioned earlier the output fuzzy sets can be calculated if all the separated fuzzy rules are known and the weights are determined. As in fuzzy logic systems all operations involve sets, the amount of calculation per inference rises dramatically. In a fuzzy system, powerful tools for generating fuzzy rules purely from data are neural networks. In next section we show, how to obtain fuzzy rules and how to determine the weights $w_i$ for fuzzy system using RBF networks.

## 3 RBF Neural Network Implementation of Fuzzy Logic

As shown above, fuzzy systems offer methodologies for managing uncertainty in a rule-based structure. In this section, RBF neural network structures are used (see Fig. 2) as a tools of performing fuzzy logic inference for fuzzy system depicted in Fig. 1. We propose the neural architecture according to the Fig. 2 whereby the a priori knowledge of each rule is embedded directly into the weights of the network.

The structure of a neural network is defined by its processing units and their interconnections, activation functions, methods of learning and so on. In Fig. 2, each circle or node represents the neuron. This neural network consists an input layer with input vector $\mathbf{x}$ and an output layer with the output value $\hat{y}_t$. The layer between the input and output layers is normally referred to as the hidden layer. Here, the input layer is not treated as a layer of neural processing units. One important feature of RBF networks is the way how output signals are calculated in computational neurons. The output signals of the hidden layer are

$$o_j = \psi_2(\|\mathbf{x} - \mathbf{w}_j\|) \tag{4}$$

where $\mathbf{x}$ is a *k*-dimensional neural input vector, $\mathbf{w}_j$ represents the hidden layer weights, $\psi_2$ are radial basis (Gaussian) activation functions. Note that for an RBF network, the hidden layer weights $\mathbf{w}_j$ represent the centres $\mathbf{c}_j$ of activation functions $\psi_2$.

The output layer neuron is linear and has a scalar output given by $\hat{y} = \sum_{j=1}^{s} v_j o_j$ where $v_j$ are the trainable weights connecting the component of the output vector $\mathbf{o}$. Then, the output of the hidden layer neurons are the radial basic functions of a proximity of weights and input values. A serious problem is how to determine the number of hidden layer (RBF) neurons. The most used selection method is to preprocess training (input) data by some clustering algorithm. After choosing the cluster centres, the shape parameters $\sigma_j$ must be determined. These parameters

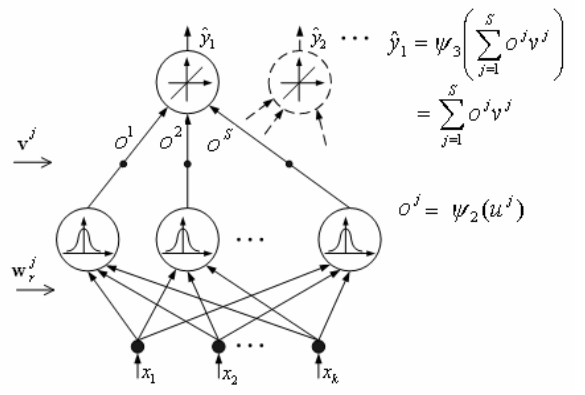



**Fig. 2.** RBF neural network architecture

express an overlapping measure of basis functions. For Gaussians, the standard deviations $\sigma_j$ can be selected, i. e. $\sigma_j \sim \Delta c$ where $\Delta c$ denotes the average distance among the centres.

To show the similarity of the RBF neural network and the fuzzy system, consider again the scalar output $\hat{y}$. The RBF network computes the output data set as

$$\hat{y}_t = G(\mathbf{x}_t, \mathbf{c}, \mathbf{v}) = \sum_{j=1}^{s} v_{j,t}\, \psi_2(\mathbf{x}_t, \mathbf{c}_j) = \sum_{j=1}^{s} v_j o_{j,t}\,, \qquad t = 1, 2, ..., N \tag{5}$$

where $N$ is the size of data samples, $s$ denotes the number of the hidden layer neurons. The hidden layer neurons receive the Euclidian distances ($\|\mathbf{x} - \mathbf{c}_j\|$) and compute the scalar values $o_{j,t}$ of the Gaussian function $\psi_2(\mathbf{x}_t, \mathbf{c}_j)$ that form the hidden layer output vector $\mathbf{o}_t$. Finally, the single linear output layer neuron computes the weighted sum of the Gaussian functions that form the output value of $\hat{y}_t$.

If the scalar output values $o_{j,t}$ from the hidden layer will be normalised, where the normalisation means that the sum of the outputs from the hidden layer is equal to 1, then the RBF network will compute the "normalised" output data set $y_t$ as follows

$$y_t = G(\mathbf{x}_t, \mathbf{c}, \mathbf{v}) = \sum_{j=1}^{s} v_{j,t} \frac{o_{j,t}}{\sum_{j=1}^{s} o_{j,t}} = \sum_{j=1}^{s} v_{j,t} \frac{\psi_2(x_t, c_j)}{\sum_{j=1}^{s} \psi_2(x_t, c_j)}, \; t = 1, 2, ..., N. \tag{6}$$

The similarity of approximation schemes (6) and (3) is obvious. From these schemes is shown that the weights $v_{j,t}$ in Eq. (6) to be learned correspond to $w_i$ in Eq. (1), and $\psi_2(./.)$ to $\mu_j(x)$ in Eq. (3). Thus, the adaptive fuzzy system depicted in Fig. 1 uses neural techniques to abstract fuzzy principles and to choose the weights $w_i$, and gradually refine those principles as the system samples new cases. These

properties were firstly recognised by V. Kecman [3]. In Fig. 2, the network with one hidden layer and normalised output values $o_{j,t}$ is the fuzzy logic model or the soft RBF network.

Next, to improve the abstraction ability of soft RBF neural networks with architecture depicted in Fig. 2, we replaced the standard Gaussian activation (membership) function of RBF neurons with functions based on the normal cloud concept.

Definition: Let *U* be the universe of discourse. *A* is a qualitative concept valued on *U*. The certainty degree $\mu_A(x)$ of a random sample *x* of *A* in *U* to the concept *A* is a random number with a stable tendency. Then the distribution of *x* on *U* is called a cloud model and *x* is called a cloud drop.

Cloud models are described by three numerical characteristics [2]: Expectation (*Ex*) as most typical sample which represents a qualitative concept, Entropy (*En*) as the uncertainty measurement of the qualitative concept and Hyper Entropy (*He*) which represents the uncertain degree of entropy. *En* and *He* represent the granularity of the concept, because both the *En* and *He* not only represent fuzziness of the concept, but also randomness and their relations. This is very important, because in economics there are processes where the inherent uncertainty and randomness are associated with different time. Then, in the case of soft RBF network, the Gaussian membership function $\psi_2(./.)$ in Eq. (6) has the form

$$\psi_2(\mathbf{x}_t, \mathbf{c}_j) = \exp\left[-(\mathbf{x}_t - E(\mathbf{x}_j)/2(En')^2\right] = \exp\left[-(\mathbf{x}_t - \mathbf{c}_j)/2(En')^2\right] \tag{7}$$

where $En'$ is a normally distributed random number with mean $En$ and standard deviation $He$, $E$ is the expectation operator. In order to keep the paper at a desirable length, the reader should refer to the references cited in this section for more profound theoretical background.

## 4 An Application

We illustrate the classic, fuzzy logic (soft) and cloud (granular) RBF neural networks on the input – output function estimation of a sales process. The time plot of the data set used in this application (the 724 daily sales for Hansa Flex company, 2004-2005) is shown in Fig. 3.

Statistical models chosen after some experimentation using the Statgraphics procedures were

$$y_t = \phi_1 y_{t-7} + \varepsilon_t \quad \text{or} \tag{8}$$

$$y_t - y_{t-7} = \theta_1 \varepsilon_{t-7} + \varepsilon_t . \tag{9}$$

Both statistical models have typical seasonal behavior with the seventh lag. Fitted models have the following forms: $\hat{y}_t = -0.1248 y_{t-7}$ or $y_t - y_{t-7} = 0.93868\varepsilon_{t-7}$ respectively. The usual diagnostic checking procedures according to Box & Jenkins [1] do not reveal any inadequacies in these models. The Box-Jenkins theory was also

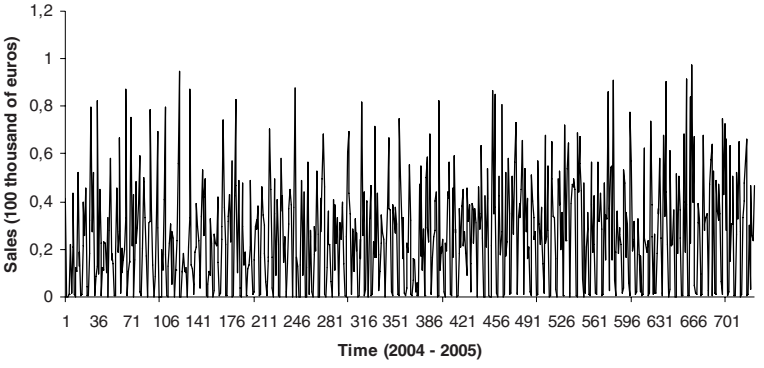



**Fig. 3.** Daily sales from January 2004 to December 2005

used to specify the neural input variables. As shown from Eq. (8) and (9), these variables are here $y_{t-7}$ and $\varepsilon_{t-7}$ respectively.

In the RBF neural network framework, the non-linear function $f(\mathbf{x})$ was estimated according to the expressions in Eq. (5). In the case of RBF fuzzy logic network, the non-linear input – output approximation function was estimated according to the formula (6). Next, the fuzzy logic RBF neural network was extended towards estimation with (a priori known) noise levels of the entropy. Noise levels are indicated by hyper entropy. It is assumed that the noise level is constant over time. We select, for practical reasons, that the noise level is a multiple, say 0.015, of entropy. In Table 1, we give the achieved results of approximation ability in dependence on various number of RBF neurons. The mean square error (MSE) was used to measure the approximation ability.

The mean (centre), standard deviation of the clusters (RBF neurons) are computed using K-means algorithm. The data used are the same as used in the previous

**Table 1.** The MSE´s measures of approximation accuracy of various RBF networks related to the different number of clusters (RBF neurons)

| Numb. of RBF Neurons | NNW Architecture: | Gausian Classic RBF | Soft RBF | Classic with Normal Cloud Concept | Soft with Normal Cloud Concept |
|---|---|---|---|---|---|
| **RBF network representations for model (8):** | | | | | |
| 3 | | 1.439 | 0.698 | 1.503 | 0.729 |
| 5 | | 0.729 | 0.693 | 0.817 | 0.716 |
| 10 | | 0.687 | 0.675 | 0.671 | 0.678 |
| 15 | | 0.697 | 0.681 | 0.681 | 0.678 |
| **RBF network representations for model (9):** | | | | | |
| 3 | | 0.783 | 0.646 | 0.786 | 0.647 |
| 5 | | 0.810 | 0.632 | 0.803 | 0.630 |
| 10 | | 0.607 | 0.571 | 0.607 | 0.571 |
| 15 | | 0.582 | 0.563 | 0.582 | 0.563 |

statistical models. As shown in Table 1, models that generate the "best" *MSE´*s are soft RBF networks.

Comparing both approaches, i. e. the models based on the Box-Jenkins methodology (the MSE for model expressed by Eq. (8) is 0.7793 and by Eq. (9) is 0.74606 respectively), and the models based on RBF networks approaches, we clearly see that models based on RBF networks are better approximation models because the estimated values are close to the actual values.

**Table 2.** The MSE´s measures of ex post forecast accuracy of various RBF networks related to the different number of clusters (RBF neurons)

| Numb. of RBF Neurons | NNW Architecture: | Gausian Classic RBF | Soft RBF | Classic with Normal Cloud Concept | Soft with Normal Cloud Concept |
|---|---|---|---|---|---|
| **RBF network representations for model (8):** | | | | | |
| 3 | | 1.6634 | 0.8602 | 1.6092 | 0.8488 |
| 5 | | 0.8509 | 0.8377 | 0.8489 | 0.8338 |
| 10 | | 0.8055 | 0.8359 | 0.8051 | 0.8346 |
| 15 | | 0.8433 | 0.8480 | 0.8391 | 0.8026 |
| **RBF network representations for model (9):** | | | | | |
| 3 | | 0.8452 | 0.6869 | 0.8451 | 0.6879 |
| 5 | | 0.8806 | 0.6548 | 0.8801 | 0.6549 |
| 10 | | 0.6600 | 0.6241 | 0.6649 | 0.6245 |
| 15 | | 0.6307 | 0.6248 | 0.8795 | 0.6052 |

Next, a forecast model was produced. Forecasts are provided during the ex post forecast period ( $y_{525}$ , …, $y_{724}$ , i. e. the sample period ends with observation $y_{524}$ ). Table 2 presents the MSE´s measures of ex post forecast accuracy. As can be seen from Table 2, the soft RBF networks have indeed a forecasting power: if anything, it seems that they manage to forecast better than other RBF network architectures.

## 4 Conclusion

In this article, we have extended RBF neural network methodology to approximate the non-linear time series data using normal cloud models in the role of standard Gaussian activation (membership) function for RBF neurons. This was done by formulating a hyper entropy of standard deviation (entropy) of the Gaussian cloud model.

To approximate the input-output function of a business process, the RBF neural network approach was applied on the daily sales data of the Hansa Flex company and compared with an approach based on statistical procedures. For the sake of approximation abilities we evaluated 34 models. Two models are based on the Box-Jenkins time series analysis approach, and 32 models are based on the neural (fuzzy logic) methodology. Using the disposable data a very appropriate model is the soft RBF network with activation functions based on the granular concept. It is also

interesting to note that the most computationally intensive models, the model based on the Box-Jenkins methodology, is newer considered “best”.

**Acknowledgement.** This work was supported by the grants VEGA 1/2628/05 and GAČR 402/05/2768. The authors would like to express their sincere to the anonymous referees for their careful reading of the paper and their constructive comments which resulted in the improvement of the paper.

## References

1. Box, G. E. P. and Jenkins, G. M.: Time Series Analysis, Forecasting and Control. San Francisco, CA : Holden-Day, (1970
2. Changyu Liu, Deyi Li, Yi Du , Xu Han: Normal Cloud Models and Their Interpretation. The 11th World Congress of International Fuzzy Systems Association (IFSA 2005). Beijing China, July 28-31, 2005, Springer, Volume III, 1540–1543
3. Kecman, V: Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy logic Models. Massachusetts Institute of Technology, The MIT Press, (2001)
4. Keller, J., M., Yager, R., R., Tahani, H.: Neural network implementation of fuzzy logic. Fuzzy Sets and Systems 45 (1992), 1–12
5. Kosko, B.: Neural networks and fuzzy systems a dynamic approach to machine intelligence. Prentice Hall, Inc., 1992
6. Marček, D.: Determination of fuzzy relations for economic fuzzy time series models by SCL techniques. The 11th World Congress of International Fuzzy Systems Association (IFSA 2005). Beijing China, July 28-31, 2005, Tsinghua University Press, Springer, Volume III, 1419–1424
7. Yoshinari, Z., Pedritz, W., Hirota, K.: Construction of fuzzy models through clustering techniques. Fuzzy Sets and Systems 54 (1993), 157–165

# Selecting Samples and Features for SVM Based on Neighborhood Model

Qinghua Hu, Daren Yu, and Zongxia Xie

Harbin Institute of Technology, Harbin 150001, P.R. China
huqinghua@hcms.hit.edu.cn

**Abstract.** Support vector machine (SVM) is a class of popular learning algorithms for good generalization. However, it is time-consuming in training SVM with a large set of samples. How to improve learning efficiency is one of the most important research tasks. It is known although there are many candidate training samples in learning tasks only the samples near decision boundary have influence on classification hyperplane. Finding these samples and training SVM with them may greatly decrease time and space complexity in training. Based on the observation, we introduce neighborhood based rough set model to search boundary samples. With the model, we divide a sample space into two subsets: positive region and boundary samples. What's more, we also partition the features into several subsets: strongly relevant features, weakly relevant and indispensable features, weakly relevant and superfluous features and irrelevant features. We train SVM with the boundary samples in the relevant and indispensable feature subspaces, therefore simultaneous feature and sample selection is conducted with the proposed model. Some experiments are performed to test the proposed method. The results show that the model can select very few features and samples for training; and the classification performances are kept or improved.

**Keywords:** neighborhood rough sets, feature selection, sample selection, SVM.

## 1 Introduction

In last decade, we are witnessing great success of support vector machines (SVM) in a lot of theoretic research and practical applications. However, SVM learning algorithms suffer from exceeding time and memory requirements if training pattern set is very large because the algorithm requires solving a quadratic programming (QP) with time complexity $O(M^3)$ and space complexity $O(M^2)$, where $M$ is the number of training samples [1]. In order to deal with the large scale quadratic programming, one major method is the decomposition based techniques, which decompose the large QP problem into a set of smaller problems so that the memory difficulty is avoided. However, for huge problems with many support vectors, the method still suffers from slow convergence.

In [2], Cortes and Vapnik showed that the weights of optimal classification hyperplane in feature space can be written as linear combination of support vectors, which shows optimal hyperplane is independent of other training samples except

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 508–517, 2007.

support vectors. One can select a part of the samples, so-called support vectors, to train SVM, rather than the whole training set. In this way, the learning time and space complexity may be greatly reduced [3, 4]. Based on this observation, some researches were reported to select patterns for SVM. Lee and Mangasarian [5] chose a random subset of the original samples and then learning classification plane with the subset. However, it is not clear how many samples should be included in the random subset. Almeida et al. [6] grouped the training samples into some clusters with k-means clustering, and the clusters with homogeneous class are replaced with the centroids of the clusters. Obviously, it is difficult to specify the number of clusters with a complex learning task. Koggalage and Halgamuge [7] gave a clustering based sample selection algorithm for SVM, where they assumed that the cluster centers were known in advance. In real-world applications, it is not the case. Shin proposed a neighborhood entropy based samples selection algorithm, which uses local information to identify those patterns likely to be located near decision boundary. They associated each samples with k nearest neighbors, then checked whether the neighbors came from multiple classes based on entropy measure [3]. Furthermore, they gave the proof that neighborhood relation between training samples in input space is preserved in feature space [4].

In fact, neighborhood relations were used to extend Pawlak's rough set model about twenty years ago [8, 9, 11]. Each object is assigned a subset of objects which are near the center object. This subset is called a neighborhood information granule. The family of neighborhood granules forms a cover of the object space. Arbitrary subset of the universe can be approximated with part of the neighborhood granules. Connecting the definition of boundary in neighborhood model and that presented in [3], we can find that they refer to the same nature but in different forms. Neighborhood rough sets present a more sound and systematical framework about this problem. Feature subset selection is an efficient technique to improve generalization and reduce classification cost [10, 14]. In this paper, we will introduce neighborhood rough set model to simultaneous select features and samples for training support vector machines.

## 2 Neighborhood Based Rough Set Model

Both rough sets and SVM deal with learning problems with structural data. Formally, the data can be written as a tuple $IS =< U, A, V, f >$, where, $U$ is the nonempty set of samples $\{x_1, x_2, \cdots, x_n\}$, called a universe, $A$ is the nonempty set of variables $\{a_1, a_2, \cdots, a_m\}$, $V_a$ is the value domain of attribute $a$; $f$ is the information function: $f : U \times A \to V$. More specially, $< U, A, V, f >$ is also called a decision table if $A = C \bigcup D$, where $C$ is the set of condition attributes, $D$ is the decision.

**Definition 1.** Given arbitrary $x_i \in U$ and $B \subseteq C$, the neighborhood $\delta_B(x_i)$ of $x_i$ in the subspace $B$ is defined as

$$\delta_B(x_i) = \{x_j \mid x_j \in U, \Delta(x_i, x_j) \le \delta\},$$

where $\Delta$ is a metric function.

A neighborhood relation N over the universe can be written as a relation matrix $M(\mathrm{N}) = (r_{ij})_{n\times n}$ where

$$r_{ij} = \begin{cases} 1, & \Delta(x_i, x_j) \le \delta \\ 0, & otherwise \end{cases}.$$

It is easy to show that N satisfies 1) reflexivity: $r_{ii} = 1$; 2) symmetry: $r_{ij} = r_{ji}$.

**Definition 2.** Consider a metric space $<U, \Delta>$, N is a neighborhood relation on $U$, $\{\delta(x_i) \mid x_i \in U\}$ is the family of neighborhood granules. Then we call $<U, \Delta, \mathrm{N}>$ a neighborhood approximation space.

**Definition 3.** Given neighborhood approximation space $<U, \Delta, \mathrm{N}>$, $X \subseteq U$, two subsets of objects, called lower and upper approximations of *X*, are defined as

$$\underline{\mathrm{N}}X = \{x_i \mid \delta(x_i) \subseteq X, x_i \in U\},\ \overline{\mathrm{N}}X = \{x_i \mid \delta(x_i) \bigcap X \neq \varnothing, x_i \in U\}.$$

The boundary region of $X$ in the approximation space is formulated as

$$BNX = \overline{\mathrm{N}}X - \underline{\mathrm{N}}X$$

**Definition 4.** Given a decision table $NDT = <U, C \bigcup D, V, f>$, $X_1, X_2, \cdots, X_N$ are the object subsets with decisions 1 to *N*, and $\delta_B(x_i)$ is the neighborhood information granules including $x_i$ and generated by attributes $B \subseteq C$, Then the lower and upper approximations of the decision *D* with respect to attributes *B* are defined as

$$\underline{\mathrm{N}_B}D = \bigcup_{i=1}^{N} \underline{\mathrm{N}_B}X_i,\ \overline{\mathrm{N}_B}D = \bigcup_{i=1}^{N} \overline{\mathrm{N}_B}X_i.$$

The decision boundary region of *D* with respect to attributes *B* is defined as

$$BN(D) = \overline{\mathrm{N}_B}D - \underline{\mathrm{N}_B}D.$$

Decision boundary is the object subset whose neighborhoods come from more than one decision class and the lower approximation of decision, also called positive region of decision, denoted by $POS_B(D)$, is the subset of objects which neighborhoods consistently belong to one of the decision classes. It is easy to show $POS_B(D) \bigcup BN(D) = U$. Therefore, the neighborhood model divides the samples into two groups: positive region and boundary.

**Definition 5.** Dependency of *D* to *B* is defined as the ratio of consistent objects:

$$\gamma_B(D) = \frac{|POS_B(D)|}{|U|}.$$

**Definition 6.** Giving $< U, A \bigcup D, V, f >, B \subseteq A$, we say attribute subset $B$ is a relative reduct if
1) $\gamma_B(D) = \gamma_A(D)$ ; 2) $\forall a \in B, \gamma_B(D) > \gamma_{B-a}(D)$ .

The first condition guarantees that $POS_B(D) = POS_A(D)$ . The second condition shows there is no superfluous attribute in the reduct. Therefore, a reduct is the minimal subset of attributes which has the same approximating power as the whole attribute set. This definition presents a feasible direct to find optimal feature subsets.

Let $< U, A \bigcup D, V, f >$ be a decision table and $\{B_j \mid j \le r\}$ is the set of reducts, we denote the following attribute subsets:

$$Core = \bigcap_{j \le r} B_j, \; K = \bigcup_{j \le r} B_j - Core, \; K_j = B_j - Core, \; I = A - \bigcup_{j \le r} B_j .$$

**Definition 7.** *Core* is the attribute subset of *strong relevance*, which cannot be deleted from any reduct, otherwise the prediction power of the system will decrease. Namely, $\forall a \in Core$, $\gamma_{A-a}(D) < \gamma_A(D)$**.** Therefore the core attributes will be in all of the reducts. $I$ is the *completely irrelevant* attribute set. The attribute in $I$ will not be included in any reduct, which means $I$ is completely useless in the system. $K_j$ is a *weak relevant attribute set*. The union of $Core$ and $K_j$ forms a reduct of the information system. Given a feature subset $B = core \bigcup k_i$, then $\forall a \in k_j$, $j \ne i$, is said to be redundant.

Training SVM just with the boundary samples in the reduced attribute subspace will speedup the learning process, improve generalization power of trained classifiers and reduce the cost in measuring and storing data. The following section will present the algorithms to search reducts and discover boundary samples.

## 3 Algorithm Design

In this section we will construct two algorithms for feature selection and boundary sample discovery, respectively. First we find a feature subset based on the neighborhood rough set model with the proposed algorithm. Then we search boundary samples in the reduced subspaces.

The motivation of rough set based feature selection is to select a minimal attribute subset, which has the same characterizing power as the whole attribute set, and without any redundant attribute.

**Definition 8.** Given $< U, A, D >$, $B \subseteq A$, $a \notin B$, the significance of an attribute is defined as

$$SIG(a, B, D) = \gamma_{B \bigcup a}(D) - \gamma_B(D) .$$

Considering time complexity, we introduce the forward search strategy to find a reduct.

**Algorithm: Forward Attribute selection based on neighborhood model**
**Input:** $< U, A, d >$ and $\delta$ // $\delta$ is the threshold to control the size of neighborhood
**Output:** reduct $red$
Step 1: $\varnothing \to red$ ; // $red$ is the pool to contain the selected attributes
Step 2: For each $a_i \in A - red$ , compute $SIG(a_i, B, D) = \gamma_{red \cup a_i}(D) - \gamma_{red}(D)$ ,
Step 3: select the attribute $a_k$ which satisfies:

$$SIG(a_k, B, D) = \max_i (SIG(a_i, red, B))$$

Step 4: if $SIG(a_k, B, D) > 0$ ,

$red \bigcup a_k \to red$
go to step2
else
return $red$

Here the algorithm adds an attribute with the great increment of dependence into the reduct in each circle until the dependence does not increase, namely, adding any new attribute will not increase the dependence in this case. The time complexity of the algorithm is $O(N \times N)$ , where N is the number of candidate attributes.

This algorithm finds the positive region samples for evaluating the significance of attributes in step 2. According to the property showed in section 2, we know $BN(D) = U - POS_B(D)$ . So, boundary samples can be computed in this algorithm. However, the aim of attribute reducion is to find feature subset which can distinguish the samples. It is different from discovering boundary samples. To support separating hyper-plane, one requires a set of boundary samples with an appropriate size. Too few boundary samples are not enough to support the optimal hyper-plane. Therefore, on one hand, we should delete most of the samples in the positive region; on the other hand, we should keep enough samples near the decision boundary to support the optimal hyper-plane. The value of $\delta$ depends on applications. Generally speaking, if the inter-class distance of a learning sample set is large, we should assign $\delta$ with a large value to get enough boundary samples to support the optimal hyperplane.

## 4 Experimental Analysis

First, let's see two toy examples in figure 1. There are two typical classification problems. The first one is a binary classification problem with circle classification plane. The second one is $4 \times 4$ checkerboard problem. Figures 1-1 and 1-5 show the raw sample set. Figures 1-2 and 1-6 show the optimal classification planes trained with the raw data. While figures 1-3 and 1-7 show the boundary samples found with neighborhood rough set model. Finally, 1-4 and 1-8 present the optimal planes trained with the boundary samples only. We can see that two kinds of separating planes are quite similar although most of learning samples don't take part in training process in the second algorithm.

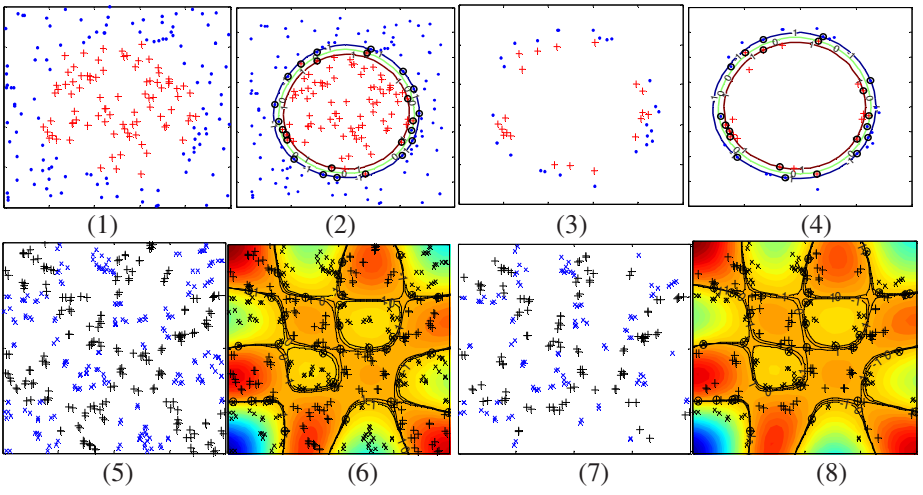



**Fig. 1.** Illustrative examples

In order to test the proposed algorithms, some data sets are collected, outlined in table 1.

**Table 1.** Data description

| | Data set | Abbreviation | Samples | features | Classes |
|---|---|---|---|---|---|
| 1 | Ionosphere | Iono | 351 | 34 | 2 |
| 2 | Sonar, Mines vs. Rocks | Sonar | 208 | 60 | 2 |
| 3 | Small Soybean | Soy | 47 | 35 | 4 |
| 4 | Diagnostic Breast Cancer | WDBC | 569 | 31 | 2 |
| 5 | Prognostic Breast Cancer | WPBC | 198 | 33 | 2 |
| 6 | Wine recognition | Wine | 178 | 13 | 3 |

First, we compare the feature selection algorithms based on neighborhood model with other existing methods reported in literatures. Table 2 shows the numbers of selected features and classification accuracies based on neighborhood rough set model with different distance metrics. Before conduct the reduction, all the numerical

attributes are normalized into interval [0, 1]. We use the selected features to train RBF-SVM, and find that average classification accuracies of infinite norm neighborhood model are better than the other two, and then is 1-norm neighborhood model. However, the numbers of features based on 1-norm are half of the features selected with $\infty$-norm. If we consider the cost of decision in measuring and storing the features, sometimes we maybe prefer the solution found with 1-norm model; especially, the average number of features in the raw data is 34.17, while there are just 4.67 features in the reduced data.

**Table 2.** Feature numbers with three definitions of neighborhoods, $\delta = 0.125$

| | 1-norm | | 2-norm | | Infinite-norm | |
|---|---|---|---|---|---|---|
| | **N** | **accuracy** | **N** | **accuracy** | **N** | **accuracy** |
| Iono | 6 | 0.91±0.05 | 9 | 0.93±0.05 | 12 | 0.93±0.06 |
| Sonar | 5 | 0.78±0.11 | 6 | 0.75±0.13 | 7 | 0.84±0.08 |
| Soy | 2 | 1.00±0.00 | 2 | 1.00±0.00 | 2 | 1.00±0.00 |
| WDBC | 6 | 0.96±0.03 | 8 | 0.97±0.02 | 21 | 0.98±0.02 |
| WPBC | 5 | 0.76±0.03 | 6 | 0.76±0.03 | 11 | 0.78±0.08 |
| Wine | 4 | 0.96±0.03 | 5 | 0.95±0.04 | 6 | 0.98±0.04 |
| **Aver**. | 4.67 | 0.8969 | 6 | 0.8933 | 9.83 | 0.9187 |

Table 3 shows the comparison of numbers of selected features and accuracies with the reduced data, where, the first two columns present the numbers of features in the raw data and accuracies; then the second two columns are the numbers of selected features with classical rough set algorithm proposed in [15] and the corresponding classification accuracies with the reduced data; consistency based algorithm was proposed in [16]; while fuzzy entropy based method was introduced in [10]. Comparing table 2 and table 3, we can see that the performance of all the feature subset selection algorithms is comparable. Although fuzzy entropy based method get the best classification accuracy, it requires the most features in these algorithms.

**Table 3.** Numbers of features and accuracies with different feature selection algorithms

| | **Raw data** | | **Classical rough sets** | | **consistency** | | **Fuzzy entropy** | |
|---|---|---|---|---|---|---|---|---|
| Data | **N** | **Accuracy** | **N** | **Accuracy** | **N** | **Accuracy** | **N** | **Accuracy** |
| Iono | 34 | 0.94±0.05 | 10 | 0.93±0.05 | 9 | 0.95±0.04 | 13 | 0.95±0.04 |
| Sonar | 60 | 0.850±0.09 | 6 | 0.71±0.10 | 6 | 0.78±0.07 | 12 | 0.83±0.09 |
| Soy | 35 | 0.93±0.11 | 2 | 1.00±0.00 | 2 | 1.0±0.00 | 2 | 1.00±0.00 |
| Wdbc | 30 | 0.98±0.02 | 8 | 0.96±0.02 | 11 | 0.96±0.02 | 17 | 0.97±0.02 |
| Wpbc | 33 | 0.78±0.04 | 7 | 0.78±0.05 | 7 | 0.76±0.03 | 17 | 0.81±0.06 |
| **Wine** | 13 | 0.99±0.02 | 4 | 0.95±0.05 | 4 | 0.95±0.05 | 9 | 0.98±0.03 |
| **Aver.** | **34** | **0.9111** | **6.17** | **0.8899** | **6.5** | **0.9010** | **11.67** | **0.9226** |

**Table 4.** Classification results based on 10-fold cross validation

| Data | 1-norm feature | | | 1-norm feature +1-norm boundary | | | 1-norm feature +2-norm boundary | | |
|---|---|---|---|---|---|---|---|---|---|
| | B | SV | accuracy | B | SV | accuracy | B | SV | accuracy |
| Iono | 351 | 111 | 0.91±0.05 | 217 | 101 | 0.92±0.05 | 171 | 91 | 0.91±0.05 |
| Sonar | 208 | 130 | 0.78±0.11 | 120 | 113 | 0.75±0.11 | 142 | 122 | 0.78±0.11 |
| Wdbc | 569 | 104 | 0.96±0.03 | 95 | 89 | 0.96±0.03 | 128 | 95 | 0.96±0.03 |
| Wpbc | 198 | 93 | 0.76±0.03 | 59 | 51 | 0.76±0.03 | 88 | 73 | 0.75±0.03 |
| Wine | 178 | 70 | 0.94±0.05 | 86 | 65 | 0.94±0.05 | 73 | 61 | 0.94±0.06 |

In table 4, as to wdbc, wpbc and wine, only a minority of the raw samples are selected as boundary (denoted by B), and most of the samples are not involved in training. The training process will be greatly speeded up with the reduced data. At the same time, we can find that average classification accuracies don't decrease compared with the results trained with the whole sample set, which shows that the boundary samples selected with neighborhood model are able to support the optimal classification hyperplane.

## 5 Conclusion

In this paper, we show a neighborhood rough set based algorithm to segment samples set into positive region and boundary. And we collect boundary samples to train SVM. What's more, neighborhood model also divides features into four subsets. We train SVM with the selected sample subset in the reduced feature subspaces. Experimental results show that the proposed method can exactly discover boundary samples of complex classification problems and the attribute reduction algorithm based on neighborhood rough sets is able to select minority of features and keep the similar classification power. So the proposed method reduces the data in terms of samples as well as features.

## References

1. Burges C. J. C.: A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (1998) 121–167
2. Cortes C., Vapnik V.: support vector networks. Machine learning 20 (1995) 273-297
3. Shin H., Cho S.: Fast pattern selection for support vector classifiers, Lecture Notes in Artificial Intelligence 2637(2003) 376–387
4. Shin H., Cho S.: Invariance of neighborhood relation under input space to feature space mapping, Pattern recognition letters 26 (2005) 707-718
5. Lee Y. J., Mangasarian O. L.: RSVM: Reduced Support Vector Machines, Data Mining Institute Technical Report 00-07, July, 2000, First SIAM International Conference on Data Mining, Chicago, 2001
6. Almeida M. B., Braga A., Braga J. P.: SVM-KM: Speeding SVMs learning with a priori cluster selection and k-means. 6th Brazilian Symposium on Neural Networks, 162–167, 2000
7. Koggalage R., Halgamuge S.: Reducing the number of training samples for fast support vector machine classification, Neural information processing 2 (2004) 57-65
8. Lin T. Y.: Neighborhood systems and relational database. In proceedings of 1988 ACM sixteenth annual computer science conference, Feb. 23-25, 1988
9. Lin T. Y.: Neighborhood systems- A qualitative theory for fuzzy and rough sets. In: advances in machine intelligence and soft computing, P. Wang (Eds), 132-155, 1997
10. Hu Q., Yu D., Xie Z.: Information-preserving hybrid data reduction based on fuzzy-rough techniques, Pattern Recognition Letters 27 (2006) 414-423
11. Yao Y. Y.: Relational interpretations of neighborhood operators and rough set approximation operators, Information Sciences 111 (1998) 239–259
12. Guyon I., Weston J., Barnhill S., et al.: Gene selection for cancer classification using support vector machines, Machine Learning 46 (2002) 389-422

13. Lin K. M., Lin C. J.: A study on reduced support vector machines, IEEE transactions on neural networks 14 (2003) 1149-1159
14. Lyhyaoui A., Martinez M., Mora I., Vazquez M., J. et al.: Sample selection via clustering to construct support vector-like classifiers, IEEE Transactions on neural networks 10 (1999) 1474-1481
15. Zhong N., Dong J., Ohsuga S.: Using rough sets with heuristics for feature selection, Journal of Intelligent Information Systems 16 (2001)199-214
16. Dash M., Liu H.: Consistency-based search in feature selection, Artificial intelligence 151 (2003) 155-176

# Intelligent Decision Support Based on Influence Diagrams with Rough Sets

Chia-Hui Huang[1,2], Han-Ying Kao[1,⋆], and Han-Lin Li[2]

[1] Department of Industrial and Operations Engineering, University of Michigan
IOE Building, 1205 Beal Avenue, Ann Arbor, MI 48109-2117
{leohkkimo,teresak_hk}@yahoo.com.tw, teresahk@umich.edu
[2] Institute of Information Management, National Chiao Tung University
Management Building 2, No. 1001 Ta Hsueh Road, Hsinchu 300, Taiwan
leohkkimo@yahoo.com.tw, hlli@cc.nctu.edu.tw

**Abstract.** Influence diagrams have been widely used as knowledge bases in business and engineering. In conventional influence diagrams, the numerical models of uncertainty are probability distributions associated with chance nodes and value tables for value nodes. However, when imprecise knowledge from large-scaled data set is involved in the systems, the suitability of probability distributions is questioned. This study proposes an alternative numerical model for influence diagrams: rough sets. In the proposed framework, the causal relationships among the nodes and the decision rules are expressed with rough sets from information systems. This study develops rough set-based framework in influence diagrams with an illustrative example.

**Keywords:** Rough sets, decision rules, Bayes' theorem, influence diagrams.

## 1 Introduction

Influence diagrams are a graphical technique for a decision problem under uncertainty [1,3,12], which have been widely used as knowledge representation and decision models [2,3,4,11,12]. Influence diagrams were originally proposed as a compact representation of decision trees for symmetric decision scenarios, and now regarded more as an extension of Bayesian networks [10]. Various methods have been developed for learning or evaluating influence diagrams [2,3,4,11,12]. In previous investigations, the numerical models of the influence diagrams used to be limited in probability distributions [1,11].

However, when imprecise knowledge from large-scaled data set is involved in the systems, how to reason from approximate information becomes a core issue in evaluating influence diagrams effectively. This study proposes an alternative numerical model for the knowledge in influence diagrams, rough sets. Rough set

⋆ Corresponding author.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 518–525, 2007.

theory was first introduced by Pawlak [5] as a tool dealing with risk and impreciseness in decision-making. The probabilistic approaches have been previously applied to rough set theory [7,8,9,13].

The purposes of this study are (1) describe how rough sets theory can be applied to express the dependency in influence diagrams, and (2) develop a rough set-based influence diagrams which combine rough set decision rules with the graphical structure of the influence diagrams.

This paper is organized as follows. Section 2 defines the notations and the framework of influence diagrams. Section 3 describes the concept of rough sets and decision rules. In section 4 we show how rough set theory establishes the numerical model and decision support in influence diagrams. A numerical example will be demonstrated. Finally, section 5 gives the concluding remarks.

## 2 Influence Diagrams

Before illustrating rough set-based influence diagrams, we first present the basic concept of influence diagrams. Influence diagrams were originally introduced by Howard and Matheson [1] as a compact representation of decision models. They may also be thought of as an extension of Bayesian networks [2,10].

An influence diagram is a directed acyclic graph (DAG) with three types of nodes, decision, chance, and value. A decision node, drawn as a rectangle, represents choices available to the decision makers. A chance node, shown as a circle, represents random variables or uncertain quantities. Finally, a value node or utility node, shown as a diamond, represents the utility or the objective to be maximized.

An influence diagram (ID) can be defined as.

$$\text{ID} = (V, L, P) \tag{1}$$

where $V$ denotes the set of nodes, $L$ denotes the set of links, and $P$ represents the numerical model. The composition of the node set $V$ can be expressed as (2).

$$V = V_D \cup V_R \cup V_W \tag{2}$$

where $V_D$ denotes the decision node set, $V_R$ represents the set of chance nodes, $V_W$ denotes the value node to be optimized. This study uses the uppercase letters to represent the variables and lowercase letters for the value of a variable.

A simple example of influence diagrams is illustrated in Fig. 1. Fig. 1 describes the causal relationships of operations management and the performance indicators, where $V_D = \{T\}$, $V_R = \{M, N, O, P\}$, and $V_W = \{Q\}$. The meaning and states of the nodes are summarized as below.

- $M$ (Product Management): 1: good, 0: poor.
- $N$ (B.O.M. Accuracy): 1: high, 0: low.
- $O$ (Manufacturing Capacity): 1: high, 0: low.
- $P$ (Schedule Adherence): 1: high, 0: low.

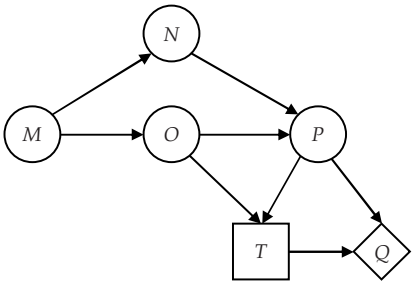



**Fig. 1.** An example of influence diagrams

– $T$ (Capacity Expansion): "take", "not take" (the action).
– $Q$ (Gain from Order Fulfilment).

In the example, the states of $N$ and $O$ are conditioned on the state of $M$. The states of $N$ and $O$ will influence the manifestation of $P$. Finally, the outcome of $P$ and the decision on $T$ will determine the value of $Q$, where the $O$ and $P$ provide the information prior to decision making. Usually, the causal relationships and decision rules in influence diagrams are expressed with probability distributions and a value table.

However, when imprecise knowledge from large-scaled data sets is involved in the reasoning systems, how to reason from the approximate information efficiently becomes a core issue to evaluate influence diagrams. Hence, this study proposes an alternative approach for modeling the causal relationships in influence diagrams, rough set theory.

## 3 Rough Sets

In this section we describe the basis and notions related to rough set theory [6].

**Information System.** Rough set theory starts with information represented by a table called an information system [6]. An information system is a 4-tuple $S = (U, A, V_a, f_a)$, where:

(i) $U$ is the universe, a nonempty finite set of *objects*.
(ii) $A = \{a_1, a_2, \ldots, a_m\}$ is a nonempty finite set of *attributes* $C \cup D$, where $C$ and $D$ is a finite set of *condition* and *decision* attributes, respectively.
(iii) $V_a$ is a *domain* of the *attribute* $a$, each attribute $a : U \rightarrow V_a$ for $a \in A$.
(iv) $f_a : U \times A \rightarrow V_a$ is the total decision function called the *information function* such that $f(x, a) \in V_a$ for $\forall a \in A, \forall x \in U$.

**Indiscernibility Relation.** Let $S = (U, A, V_a, f_a)$ be an information system, $B \subseteq A$ and $X \subseteq U$. With any subset of attributes $B \subseteq A$, a binary indiscernibility relation, is called *B-indiscernibility relation*, which is defined by:

$$IND(B) = \{(x, y) \in U \times U : a(x) = a(y), \forall a \in B\} \tag{3}$$

For any subset $X \subseteq U$, the *lower* and *upper approximation* can be expressed as (4) and (5), respectively:

$$\underline{B(x)} = \bigcup_{x \in U} \{B(x) : B(x) \subseteq X\} \tag{4}$$

$$\overline{B(x)} = \bigcup_{x \in U} \{B(x) : B(x) \cap X \neq \emptyset\} \tag{5}$$

That is, the elements of $\underline{B(x)}$ are all the elementary objects certainly belonging to $X$. The elements of $\overline{B(x)}$ are at least one object belonging to $X$. With the lower and upper approximation of a set $X \subseteq U$, the universe can be divided into three regions, the *boundary region* $BND(x)$, the *positive region* $POS(x)$, and the *negative region* $NEG(x)$:

$$BND(x) = \overline{B(x)} - \underline{B(x)} \tag{6}$$

$$POS(x) = \underline{B(x)} \tag{7}$$

$$NEG(x) = U - \overline{B(x)} \tag{8}$$

If the boundary region of $X$ is an empty set, $BND(x) = \emptyset$, then $X$ is a crisp set with respect to $B$; otherwise, if $BND(x) \neq \emptyset$, then $X$ is a rough (approximate) set with respect to $B$.

**Decision Rules.** Let $|C|$ denote the set of all objects from $U$ that have the meaning of $C$ in $S$ where $|*|$ indicates the *cardinality* of a (finite) set.

If a decision rule in $S$ is $C \rightarrow D$, meaning IF $C$ THEN $D$, where $C = \{c_1, c_2, \ldots, c_n\}$ is the *condition* attribute, and $D = \{d_1, d_2, \ldots, d_m\}$ is the *decision* attribute of the decision rule, respectively, then the *support* of the decision rule $C \rightarrow D$ in $S$ will be $supp(C, D) = card(C \cap D)$.

For any $C$ in $S$ its *probability* is defined by $\sigma(C) = p(|C|)$. With the decision rule $C \rightarrow D$ in $S$ the *conditional probability* is $\sigma(D \mid C) = p(|D| \mid |C|)$.

The *strength* of the decision rule $C \rightarrow D$ in $S$, denoted by $\sigma(C, D)$, is defined as (9):

$$\sigma(C, D) = \frac{supp(C, D)}{card(U)} \tag{9}$$

The *certainty factor* of the decision rule $C \rightarrow D$ in $S$, denoted by $cer(C, D)$, is defined as (10):

$$cer(C, D) = \sigma(D \mid C) = \frac{supp(C, D)}{card(C)} = \frac{\sigma(C, D)}{\sigma(C)} \tag{10}$$

The certainty factor is interpreted as a conditional probability that $y$ belongs to $D$ given $y$ belongs to $C$. It is easy to see that $cer(C, D) \in [0, 1]$. If $cer(C, D) = 1$, then the given decision rule is a *deterministic* or *certain* decision rule in $S$. Otherwise, $0 < cer(C, D) < 1$, the given decision rule is a *non-deterministic* or *uncertain* decision rule in $S$.

The *coverage factor* of the decision rule $C \rightarrow D$ in $S$, denoted by $cov(C, D)$, is defined as (11):

$$cov(C, D) = \sigma(C \mid D) = \frac{supp(C, D)}{card(D)} = \frac{\sigma(C, D)}{\sigma(D)} \tag{11}$$

**Probabilistic Properties.** Let $C \rightarrow D$ be the decision rule in $S$, then the following properties (12)–(17) are valid [7,8,13]:

$$\sum_{C' \in C(x)} cer(C', D) = 1 \tag{12}$$

$$\sum_{D' \in D(x)} cov(C, D') = 1 \tag{13}$$

$$\sigma(C) = \sum_{D' \in D(x)} cov(C, D') \cdot \sigma(D') = \sum_{D' \in D(x)} \sigma(C, D') \tag{14}$$

$$\sigma(D) = \sum_{C' \in C(x)} cer(C', D) \cdot \sigma(C') = \sum_{C' \in C(x)} \sigma(C', D) \tag{15}$$

$$cer(C, D) = \frac{\sigma(C, D)}{\sigma(C)} = \frac{\sigma(C, D)}{\sum_{D' \in D(x)} \sigma(C, D')} = \frac{cov(C, D) \cdot \sigma(D)}{\sigma(C)} \tag{16}$$

$$cov(C, D) = \frac{\sigma(C, D)}{\sigma(D)} = \frac{\sigma(C, D)}{\sum_{C' \in C(x)} \sigma(C', D)} = \frac{cer(C, D) \cdot \sigma(C)}{\sigma(D)} \tag{17}$$

Note that (14) and (15) are refer to total probability theorem, (16) and (17) are refer to Bayes' theorem.

## 4 Influence Diagrams with Rough Sets

Most literatures on influence diagrams [1,3,10,11] used to describe the dependency and its associated numerical model with probability theory. However, when impreciseness and large data volume involved in the domain, the decision makers may need more flexible uncertainty measures for analysis. Rough set theory can be an alternative measure in such a problem.

Given the influence diagrams structure and the original data set, rough set theory provides a basis for extracting the knowledge and expressing the dependency among nodes in the influence diagrams. In order to represent the ontology, we define that rough set-based influence diagram is a directed acyclic graph RSID $= (U, A, f)$, where:

(i) $U$ is the universe, a nonempty finite set of *objects*.
(ii) $A \equiv C \cup D \equiv V_D \cup V_R \cup V_U$.

(ii) $f$ is the *flow function* representing the strength, certainty factor, and coverage factor of the decision rule. With every branch of $(x, y)$ there is a directed arc from node $x$ to $y$, then the strength of branch $(x, y)$ is $\sigma(x, y)$. The certainty and coverage of branch $(x, y)$ are defined as $cer(x, y)$ and $cov(x, y)$. That is, $f \equiv L$.

In the following, we show the knowledge modeling and decision making in influence diagrams with rough sets.

**Example.** Consider the influence diagram in Fig. 1 where $A = \{M, N, O, P, Q, T\}$, $C = \{M, N, O, P, T\}$, and $D = \{N, O, P, Q, T\}$. An information system of the diagram is listed in Table 1 and 2, respectively. According to (9), (10), and (12), the certainty factors can be obtained as in Table 3.

Based on the information from Table 2 and Table 3, the objective of this problem is to maximize the expected utilities as (18).

$$\begin{aligned} &\max\ EV(Q = q(p, t)) \\ &= \sum_{m,n,o,p} q(p, t)\sigma(p|n, o)\sigma(n|m)\sigma(o|m)\sigma(m) \end{aligned} \tag{18}$$

where $EV(*)$ stands for the expected value of "*". Note that the uppercase and lowercase letter represents the variable and the value of a variable, respectively.

The expected values of $Q$ based on $T =$ "take" and $T =$ "not take" are computed as follows, respectively.

$$\begin{aligned} &EV(Q = q(p, t)) \\ &= \sum_{m,n,o,p} q(p, \text{“take”})\sigma(p|n, o)\sigma(n|m)\sigma(o|m)\sigma(m) = 110.63 \\ &EV(Q = q(p, t)) \\ &= \sum_{m,n,o,p} q(p, \text{“not take”})\sigma(p|n, o)\sigma(n|m)\sigma(o|m)\sigma(m) = 87.46 \end{aligned}$$

Hence, the optimal decision to maximize the utilities is $\max EV(Q = q(p, t)) = \max\{110.63, 87.46\} = 110.63$, where $T =$ "take" (take the action).

## 5 Conclusions

This study proposes an alternative numerical framework for influence diagrams, rough sets. Considering the imprecise knowledge from large-scaled data set, this study formulates the causal relationships and the decision rules among the nodes (attributes) with rough sets from information systems. The proposed knowledge model provides a comprehensive way for knowledge representation and decision support from large-scaled data sets. For future studies, there are some potential themes: (1) integrated analysis with rough sets in various graphical decision model, including Bayesian networks, decision trees, influence diagrams, and so

**Table 1.** Information system of Fig. 1

| $U$ | $M$ | $N$ | $O$ | $P$ | *support* | *strength* |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 500 | 0.24 |
| 2 | 1 | 1 | 1 | 0 | 250 | 0.12 |
| 3 | 1 | 1 | 0 | 1 | 190 | 0.09 |
| 4 | 1 | 1 | 0 | 0 | 80 | 0.04 |
| 5 | 1 | 0 | 1 | 1 | 100 | 0.05 |
| 6 | 1 | 0 | 1 | 0 | 60 | 0.03 |
| 7 | 1 | 0 | 0 | 1 | 60 | 0.03 |
| 8 | 1 | 0 | 0 | 0 | 50 | 0.02 |
| 9 | 0 | 1 | 1 | 1 | 100 | 0.05 |
| 10 | 0 | 1 | 1 | 0 | 120 | 0.06 |
| 11 | 0 | 1 | 0 | 1 | 50 | 0.02 |
| 12 | 0 | 1 | 0 | 0 | 80 | 0.04 |
| 13 | 0 | 0 | 1 | 1 | 50 | 0.02 |
| 14 | 0 | 0 | 1 | 0 | 80 | 0.04 |
| 15 | 0 | 0 | 0 | 1 | 50 | 0.02 |
| 16 | 0 | 0 | 0 | 0 | 280 | 0.13 |

**Table 2.** Value table of $Q$ in Fig. 1

| |
|---|
| $Q = q(P = 1, T =$ "take"$) = 120$ |
| $Q = q(P = 0, T =$ "take"$) = 100$ |
| $Q = q(P = 1, T =$ "not take"$) = 50$ |
| $Q = q(P = 0, T =$ "not take"$) = 130$ |

**Table 3.** Certainty factors of Fig. 1

| |
|---|
| $\sigma(M = 1) = 0.62$ |
| $cer(M = 1, N = 1) = \sigma(N = 1 \mid M = 1) = 0.79$ |
| $cer(M = 0, N = 1) = \sigma(N = 1 \mid M = 0) = 0.45$ |
| $cer(M = 1, O = 1) = \sigma(O = 1 \mid M = 1) = 0.71$ |
| $cer(M = 0, O = 1) = \sigma(O = 1 \mid M = 0) = 0.45$ |
| $cer(N = 1, O = 1, P = 1) = \sigma(P = 1 \mid N = 1, O = 1) = 0.62$ |
| $cer(N = 0, O = 1, P = 1) = \sigma(P = 1 \mid N = 0, O = 1) = 0.50$ |
| $cer(N = 1, O = 0, P = 1) = \sigma(P = 1 \mid N = 1, O = 0) = 0.58$ |
| $cer(N = 0, O = 0, P = 1) = \sigma(P = 1 \mid N = 0, O = 0) = 0.25$ |

on; (2) hybrid decision analysis with fuzzy sets and rough sets in graphical decision models; (3) potential applications of intelligent decision support with rough sets, such as biomedicine, supply chain management, business strategic analysis, and so on.

## Acknowledgement

The authors thank the anonymous referees for their fruitful comments on this paper. This study is supported by Taiwan Merit Scholarships, University-based Program Regulation, No. 94B514 (C.H. Huang) and Taiwan Merit Scholarships, No. TMS-094-2-B-009 (H.Y. Kao).

## References

1. Howard, R.A., Matheson, J.E.: Influence Diagrams. In: Howard, R.A., Matheson, J.E. (eds.): The Principles and Applications of Decision Analysis, Vol. 2. Strategic Decisions Group, Menlo Park (1981) 719–762.
2. Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer-Verlag, Berlin Heidelberg New York (2001).
3. Nease, R.F., Owens, D.K.: Use of Influence Diagrams to Structure Medical Decisions. Medical Decision Making. **17**(3) (1997) 263–275.
4. Oliver, R.M., Smith, J.Q.: Influence Diagrams, Belief Nets and Decision Analysis. John Wiley & Sons (1990).
5. Pawlak, Z.: Rough Sets. Informational Journal of Information and Computer Sciences. **11**(5) (1982) 341–356.
6. Pawlak, Z.: Rough Sets-Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991).
7. Pawlak, Z.: Rough Sets, Decision Algorithms and Bayes' Theorem. European Journal of Operational Research. **136**(1) (2002) 181–189.
8. Pawlak, Z.: Probability, Truth and Flow Graph. Electronic Notes in Theoretical Computer Science. **82**(4) (2003).
9. Pawlak, Z., Wong, S.K.M., Ziarko, W.: Rough Sets: Probabilistic versus Deterministic Approach. International Journal of Man-Machine Studies. **29**(1) (1988) 81–95.
10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc. (1988).
11. Shacher, R.D.: Evaluating Influence Diagrams. Operations Research. **34** (1986) 871–882.
12. Tatman, J.A., Shacher, R.D.: Dynamic Programming and Influence Diagrams. IEEE Transactions on Systems, Man and Cybernetics. **20**(2) (1990) 365–379.
13. Yao, Y.Y.: Probabilistic Approaches to Rough Sets. Expert Systems. **20**(5) (2003) 287-297.

# Object Class Recognition Using SNoW with a Part Vocabulary

Ming Wen, Lu Wang, Lei Wang, Qing Zhuo, and Wenyuan Wang

Department of Automation, Tsinghua University, Beijing, 100084, P.R. China
{wenm03,l-wang02,wlei04}@mails.tsinghua.edu.cn

**Abstract.** In this paper we present a novel method for object class recognition. A vocabulary of object parts is automatically constructed from sample images of the object class by AdaBoost. Images are then represented using parts from this vocabulary. Based on this representation, the Sparse Network of Winnows (SNoW) learning architecture is employed to learn to recognize instances of the object class. Experimental results show that the method achieves high recognition accuracy on different data sets, and is highly robust to partial occlusion and background clutter.

**Keywords:** Object class recognition, part-based representation, SIFT, part vocabulary, AdaBoost, SNoW.

## 1 Introduction

Object class recognition is one of the most important and challenging research topics in machine vision. Different from the recognition of specific objects, object class recognition must deal with the large intra-class variance that exists in most visual object categories. The key to solving this problem lies in finding an appropriate intermediate representation. Recently, part-based representation, where an object is modeled by a set of representative parts, has gained more and more attention. Such a representation emerges when an interest point detector is applied to an image, and then local feature descriptors are extracted from the patches around interest points highlighted by the detector [1]. This model can naturally cope with the large intra-class variance, and is also consistent with the principles of biological vision to a great extent.

Much work has been done on the basis of part-based model. Fergus et al. [2] used a generative probabilistic model to represent objects as random constellations of parts. The parameters of the model are learned using an EM algorithm. This model has been tested with great success using the Caltech database, which has since become a benchmark for other methods of object class recognition. Crandall et al. [3] presented a class of statistical models that are explicitly parameterized according to the degree of spatial structure they can represent. These models provide a way of relating different spatial priors that have been used for recognizing generic classes of objects, including joint Gaussian models and tree-structured models. Opelt et al. [4] proposed a model of object class recognition that combines four types of local features within the framework of Boosting.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 526–533, 2007.

Our approach was partially motivated by Agarwal and Roth's work [5]. In order to detect cars in images, they built a part vocabulary from a set of representative images of cars by hierarchical clustering. Images were then represented using parts from this vocabulary. However, there are some problems in their method. First, they used all parts extracted from the representative images to build the vocabulary. Some of these parts that belong to the background are useless, even harmful to the classification. There should be some schemes supporting automatic part selection. Second, the thresholds that measure the similarity between the image parts and vocabulary parts were determined by experience, and set to the same value. It's difficult to choose a single appropriate threshold for all vocabulary parts. We propose here some substantial improvement to the vocabulary construction procedure. AdaBoost is used to select those most discriminative parts for the vocabulary, and determine the appropriate thresholds. A binary feature vector is then formed for each image to indicate which of the vocabulary parts are present in an image. Due to the sparseness property of these feature vectors, we train our classifier using the Sparse Network of Winnows (SNoW) learning architecture [6,7]. The rest of this paper is organized as follows. Section 2 describes the details of our approach. Section 3 presents our experimental setup and results. Section 4 brings this paper to a conclusion.

## 2 Approach

### 2.1 Interest Point Detection

We use the Scale Invariant Feature Transform (SIFT) [8] to detect points of interest or keypoints in images. SIFT is not only an interest point detector, but also a local feature descriptor. The features extracted by SIFT are invariant to image scaling and rotation, and partially invariant to affine distortion, change in 3D viewpoint and change in illumination. Mikolajczyk and Schmid [9] compared the performance of several local feature descriptors, including SIFT, steerable filters, differential invariants and moment invariants, and concluded that the SIFT descriptor performs the best according to several evaluation criteria in [9]. Following are the major stages of computation used to generate the SIFT features of an image: (1) Select potential interest points by searching scale-space extrema in the difference-of-Gaussian (DoG) function convolved with the image. (2)Filter out the points with low contrast or poorly localized along an edge. (3) Assign a consistent orientation to each keypoint based on local image gradient directions. (4) Compute a descriptor for the local image region around each keypoint. The operations in the first three stages assign a location, scale and orientation to each keypoint. These parameters impose a 2D coordinate system on the local image region around each keypoint. Therefore, the local feature descriptor computed relative to the system is invariant to these parameters.

### 2.2 Vocabulary Construction by AdaBoost

After interest point detection, an image is represented as an unordered set of image parts. Given such a representation, object class recognition is a problem

of classification using sets of unordered features as input, which is a rather nonstandard learning problem, and rarely considered in the literature. One of the solutions to this problem is to build a part vocabulary, and convert unordered features into ordered feature vectors in the vocabulary space, for which many powerful learning algorithms have been developed. Our work was intended to proceed along this line. First all parts extracted from the training images that contain relevant objects are put into a common pool of candidate parts. Then AdaBoost is used to select from this pool those most discriminative parts for the vocabulary.

The AdaBoost algorithm, introduced by Freund and Schapire [10], is a classical ensemble learning algorithm that produces a highly accurate hypothesis by combining many weak hypotheses. Given a training set $(I_j, l_j)$ for $j = 1, ..., J$, where $I_j$ is the $j$'th example and $l_j$ is the corresponding label, we would like to learn a function $H : I \mapsto \hat{l}$ which predicts the label of example $I$. AdaBoost maintains a set of weights $\omega_j$ over the training examples, and calls a given weak learning algorithm repeatedly in a series of rounds $n = 1, ..., N$. Initially, all weights are set equally, but in each round, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training set. The weak learner's job is to find a weak hypothesis $h_n$ which has some discriminative power relative to these weights, i.e.

$$\sum_{j:h(I_j)=l_j} \omega_j > \sum_{j:h(I_j)\neq l_j} \omega_j, \tag{1}$$

such that more examples are correctly classified than misclassified relative to the weights. The process of putting weights and constructing a weak hypothesis is iterated for $N$ rounds, and the weak hypotheses $h_n$ of each round are combined into the final hypothesis $H$.

Let $v_1, ..., v_L$ be the SIFT features corresponding to the parts in the candidate pool and $d(\cdot, \cdot)$ be the Euclidean distance metric. Given the part-based representations of training images $(R(I_j), l_j), j = 1, ..., J, R(I_j) = \{f_{j,k} : k = 1, ..., K_j\}$, where $f_{j,k}$ is the $k$'th SIFT feature of image $I_j$, $l_j = +1$ if $I_j$ contains a relevant object and $l_j = -1$ if $I_j$ contains no relevant object, we design our weak learner as suggested by Opelt et al. [4]. Details for the weak learner are as follows:

1. **Calculating the minimal distance matrix:** For all features $v_i$ in the candidate pool and all images $I_j$, calculate the minimal distance between $v_i$ and features in $I_j$,
$$d_{i,j} = \min_{1\leq k\leq K_j} d(v_i, f_{j,k}). \tag{2}$$
2. **Sorting:** For each $i$, let $p_i(1), ..., p_i(J)$ be a permutation such that
$$d_{i,p_i(1)} \leq \cdots \leq d_{i,p_i(J)}. \tag{3}$$
3. **Selecting the most discriminative feature :** For all features $v_i$ in the candidate pool, calculate over all images $I_j$

$$score_i = \max_s \sum_{q=1}^{s} \omega_{p_i(q)} l_{p_i(q)}, \tag{4}$$

and select the feature $v_m$ where $score_m$ is maximum.

4. **Selecting threshold:** With the position $s$ where $score_m$ reached a maximum sum, the threshold $\theta_m$ is set to

$$\theta_m = \frac{d_{m,p_m(s)} + d_{m,p_m(s+1)}}{2}. \tag{5}$$

The weak learner finds the most discriminative feature $v_m$ relative to the current weights and determines its corresponding threshold $\theta_m$. The feature and threshold produce a simple weak hypothesis which indicates whether an image contains a feature that is sufficiently similar to $v_m$. This weak hypothesis is used to classify the training images, and the weights are updated according to the classification result. $N$-round iterations generate $N$ weak hypotheses. We don't combine these weak classifiers into a strong classifier, but use those selected features to build a vocabulary of object parts. There is still a problem in the process. The update of the weights in each round performs some sort of adaptive decorrelation of the weak hypotheses: if an image was incorrectly classified in round $n$, then its weight is increased and more emphasis is put on this image in the next round, yielding quite different hypotheses $h_n$ and $h_{n+1}$. However, this process cannot guarantee these hypotheses are all different, i.e., there may be some redundant features in the vocabulary. We have to filter out these redundant features. After selection, we get the final vocabulary , $(v_1, \theta_1), ..., (v_M, \theta_M)$.

### 2.3 Image Representation

Having constructed the part vocabulary above, images are now represented using this vocabulary. This is done by determining which of the vocabulary parts are present (active) in an image, and representing the image as a binary feature vector based on these detected parts. Each part of the image is compared to the vocabulary parts using the Euclidean distance metric. If a sufficiently similar vocabulary part is found, the corresponding element of the feature vector is set to 1, otherwise 0. Let $R(I) = \{f_k : k = 1, ..., K\}$ be the part-based representation of an image $I$, and $X = [x_1, ..., x_M]^T$ be its binary feature vector. The rule can be formulated as follows:

$$x_i = \begin{cases} 1, \ if \ \exists f_k, \ d(v_i, f_k) \leq \theta_i \\ 0, \ otherwise \end{cases} \tag{6}$$

### 2.4 Learning Classifier Using SNoW

Given a set of training images labeled as positive (Object) or negative (Non-object), each image is re-represented as a binary feature vector as described above. Note that there are many zeros in any feature vector, since only some of the vocabulary parts are actually present in any single image. Taking advantage

of this sparseness property, we train our classifier using the SNoW learning architecture that is especially well-suited for such sparse feature representations.

SNoW learns a sparse network of linear functions over the feature space using a feature-efficient learning algorithm. In its basic architecture, a two layer network is maintained. The input layer is the feature layer, and the output layer consists of target nodes, each of which corresponds to a class label. Target nodes are linked to input features via weighted edges which are allocated dynamically. SNoW expects each example to be represented as a list of indices of active features (possibly associated with a real valued strength). A feature $i$ is allocated and linked to target node $t$ if and only if $i$ is present in an example labeled $t$. Let $A_t = \{i_1, ..., i_n\}$ be the set of features that are active in an example $e$ and are linked to target node $t$, $\omega_{t,i}$ be the weight on the edge connecting the $i$'th feature to target node $t$, $s_i$ be the real valued strength associated with feature $i$ (For our task $s_i = 1$), and $\Omega_t(e)$ be $t$'s activation given $e$. Then we have the following equation:

$$\Omega_t(e) = \sum_{i \in A_t} \omega_{t,i} s_i. \tag{7}$$

We say that $t$ predicts positive if and only if $\Omega_t(e) \geq \theta_t$, where $\theta_t$ is $t$' threshold. Let $T$ be the set of all targets defined in the architecture. Example $e$ will be labeled with

$$t^*(e) = \arg\max_{t \in T} \sigma(\theta_t, \Omega_t(e)), \tag{8}$$

where $\sigma(\theta, \Omega_t(e))$ is a sigmoid function whose transition from an output close to 0 to an output close to 1 centers around $\theta$.

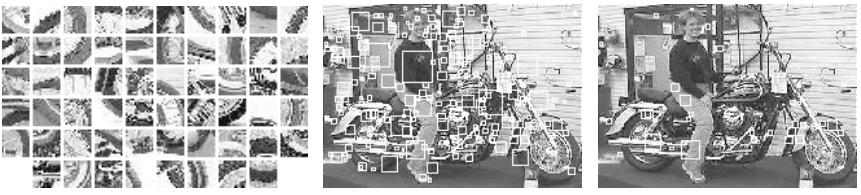
Several learning rules may be used within SNoW. One of them is the classical Winnow algorithm. Winnow has two update parameters at target node $t$: a promotion parameter $\alpha_t > 1$ and a demotion parameter $0 < \beta_t < 1$. These parameters are used to update the set of weights $\omega_{t,i}$ only when a mistake in prediction is made. Let $P_t$ be the true prediction of target node $t$. Given an example $e$, the update rule can be formulated as follows:

$$\forall i \in A_t,\ \omega_{t,i} = \begin{cases} \omega_{t,i} \alpha_t^{s_i},\ if\ \Omega_t(e) < \theta_t, P_t = +1 \\ \omega_{t,i} \beta_t^{s_i},\ if\ \Omega_t(e) \geq \theta_t, P_t = -1 \end{cases} \tag{9}$$

## 3 Experiments

We evaluated our method on the database used by Fergus et al. [2]. The database consists of 4 sets of object classes: Motorbikes, Airplanes, Faces and Cars (side view). The first three were obtained from the Caltech database [1], and the last one from the UIUC database [2]. Negative examples were also from the above two databases. The recognition was based on deciding presence or absence of a relevant object.

[1] http://www.robots.ox.ac.uk/~vgg/data
[2] http://l2r.cs.uiuc.edu/~cogcomp/Data/Car



(a) Part vocabulary (b) Keypoint detection (c) Image representation

**Fig. 1.** Image representation using parts from the vocabulary

A limited amount of preprocessing was performed on the data sets. First, we used Homomorphic Filtering [11] for illumination normalization. Second, images were rescaled to have a uniform horizontal axis length (200 pixels). This is mainly because the SIFT detector generates large numbers of parts that densely cover the image over the full range of scales and locations. A typical image of size 500×500 pixels will give rise to about 2000 parts (although this number depends on both image content and choices for various parameters) [8]. The quantity of parts is very important for object recognition. But it also increases the computational complexity. Therefore, we made a compromise on image sizes.

In each of our experiments, the training set contained 100 positive and 100 negative images. The tests were carried out on 200 new images, half belonging to the learned object class and half not. First, we constructed a vocabulary of object parts from the positive images. Fig. 1(a) shows a part vocabulary of Motorbikes. This was obtained by cropping from images those regions where the most discriminative features are extracted. There are 58 different parts in the vocabulary, and it is a result of 100-round iterations . Vocabulary construction is the most time-consuming stage in the whole process. The main computational burden is the calculation of the distances between $v_i$ and $f_{jk}$. Given these distances which can be calculated prior to AdaBoost, the remaining calculations are relatively inexpensive. Second, we represented images using parts from the vocabulary. Fig. 1(b) is an example image with keypoints shown as squares. The sizes of the squares correspond to the scales of these keypoints. This image contains a motorbike that is partially occluded by a person, and its background is cluttered. The SIFT detector totally found 298 keypoints in the image. Many of them are on the region of background or person. Fig. 1(c) shows the image with only vocabulary parts left. Most irrelevant parts have been removed. This result illustrates our method can give a more compact representation to an image. Finally, we used the SNoW learning architecture [3] to train our classifier.

Fig. 2 shows how the number of AdaBoost iterations affects the number of parts in the vocabulary, and Fig. 3 shows the recognition rate as a function of the number of AdaBoost iterations. These curves were obtained by varying the iteration number from 20 to 400 with an increment of 20. From the two figures we can see the number of vocabulary parts increases with the growth of the

[3] Software for SNoW is freely available at http://l2r.cs.uiuc.edu/~cogcomp/

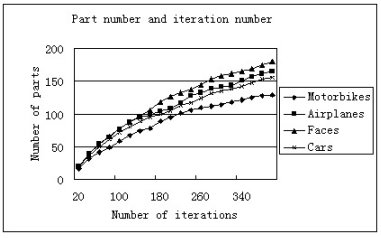



**Fig. 2.** Number of vocabulary parts as a function of the number of AdaBoost iterations

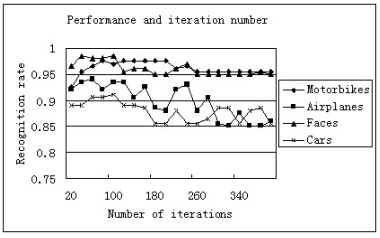



**Fig. 3.** Recognition rate as a function of the number of AdaBoost iterations

**Table 1.** Comparison of our method with [2,4]

| Data Set | Our method | Fergus et al. [2] | Opelt et al. [4] |
|---|---|---|---|
| Motorbikes | 97.5% | 92.5% | 92.2% |
| Airplanes | 94.0% | 90.2% | 88.9% |
| Faces | 98.5% | 96.4% | 93.5% |
| Cars (Side) | 91.0% | 88.5% | 83.0% |

iteration number (more and more slowly), but the increase in the number of vocabulary parts does not necessarily raise the recognition rate.

For comparison and performance evaluation, Table 1 presents the recognition rates of the various methods under consideration. Our results were obtained by selecting the number of AdaBoost iterations as follows: 160 (Motorbikes), 60 (Airplanes), 100 (Faces) and 100 (Cars). We used a validation data set of 100 images to set the iteration number. Note that the results of [2] were obtained using scale-normalized images, i.e., each object image was manually rescaled so the objects will be of the same size. Our method can naturally cope with the scale variation. Amongst the data sets, Motorbikes and Airplanes include significant scale variation. The experimental results demonstrate our method performs very well on these data sets.

## 4 Conclusions and Future Work

In this paper we have presented an approach for part-based object class recognition. Given part-based representation, object class recognition is a problem of classification using unordered features, which is a rather nonstandard learning problem. We use AdaBoost to select some most discriminative parts from a pool of candidate parts, and construct a vocabulary based on these selected parts. Images are then re-represented using this vocabulary. This is done by determining which of the vocabulary parts are present in an image , and representing the image as a binary feature vector based on these detected parts. Finally, SNoW is employed to train the classifier. Experimental results show that our method

works successfully on different data sets, and is highly robust to partial occlusion and background clutter.

Our work can be extended in several directions. First, the computational costs of the current approach are relatively high, especially in the stage of vocabulary construction. We are considering to reduce the number of candidate parts by clustering methods before selecting vocabulary parts using AdaBoost. Second, we will take into account spatial relations among the parts that can be defined in terms of the distance and direction between each pair of parts. We may incorporate some elements into the binary feature vectors of images to indicate whether or not specified relations occur in images.

## References

1. Bar-Hillel, A., Hertz, T., Weinshall, D.: Object class recognition by boosting a part-based model. In: Proc. of CVPR. Volume 1. (2005) 702– 709
2. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Proc. of CVPR. Volume 2. (2003) 264–271
3. Crandall, D., Felzenszwalb, P., Huttenlocher, D.: Spatial priors for part-based recognition using statistical models. In: Proc. of CVPR. Volume 1. (1998) 10–17
4. Opelt, A., Fussenegger, M., Pinz, A., Auer, P.: Weak hypotheses and boosting for generic object detection and recognition. In: Proc. of ECCV. Volume 2. (2004) 71–84
5. Agarval, S., Roth, D.: Learning a sparse representation for object detection. In: Proc. of ECCV. Volume 4. (2002) 113–127
6. Carlson, A.J., Cumby, C.M., Rosen, J.L., Roth, D.: The snow learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department (1999)
7. Roth, D.: Learning to resolve natural language ambiguities: A unified approach. In: Proc. of National Conference on Artificial Intelligence. (1998) 806–813
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
9. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: Proc. of CVPR. Volume 2. (2003) 257–263
10. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55** (1997) 119–139
11. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley (2001)

# Coverage in Biomimetic Pattern Recognition*

Wenming Cao[1,2] and Guoliang Zhao[2]

[1] Intelligent Information Processing Key Laboratory, Shenzhen University, Shenzhen 518060, China
[2] Institute of Semiconductors of Chinese Academy of Science, Beijing 100083, China
{wmcao,glzhao}@semi.ac.cn

**Abstract.** Coverage is a kind of method to cover points of same class samples in feature space, which is based on Biomimetic Pattern Recognition. The mathematical description of coverage is given and the discriminant boundary of coverage is shown. Coverage is tested in face recognition on ORL database. Both the COVERAGE and SVM networks are used for covering. The results show that COVERAGE act better than SVM in generalization, especially for small sample set, which are consonant with the result of the applications of BPR.

## 1 Introduction

The problem of finding a minimum covering sphere of smallest radius (equivalently, smallest volume) which contains a given set of n points in 2D space was firstly proposed by Sylvester[4] in 1857. In a high-dimensional space, convex hull was often used as a tool to simplify the issue. From a geometrical view, Hopp and Reeve[5] handled the problem via successively reducing the volume of the hyper-sphere that contains the set of points step by step with its expected computing time $O(nd^{2.3})$.

Coverage algorithm is a kind of method to cover points of same class samples in feature space, which is based on Biomimetic Pattern Recognition (BPR)[Fig. 1]. Biomimetic pattern recognition was first proposed by Academician Wang Shoujue[1] in 2002. In this theory, pattern recognition is based on "cognition" instead of "division". In another word, BPR emphasizes on "cognition of all sample classes one by one" rather than "classification of many kinds of samples". The basic idea of BPR is to finding an optimal covering of the same class in the high dimensional sample space, rather than to model a set of points in a high-dimensional space by statistical learning as the traditional pattern recognition methods do. The principle of homology-continuity is the foundation of BPR and computational information geometry, which is based on high-dimensional geometry and descriptive geometry, is used as analysis tools to deal with the practical issues.

Priority Ordered Neural Networks[2] (PONN) and Sequential Learning Ahead Masking (SLAM) model for pattern recognition[3] have many advantages in

* Supported by The National Natural Science Foundation of China NO.60576055.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 534–541, 2007.

learning and knowledge renewing. PONN's learning speed is much faster than that of the multilayered feed forward neural networks with BP algorithms. PONN also has the ability to keep previously stored knowledge, when the net is renewing with new additional training samples. In this paper, PONN is used for dealing with inseparable sample sets and as the strategy of recognition of multi-class problems.

In this paper, an algorithm of using coverage same class samples is provided. The algorithm is based on the theory of BPR and PONN. And discuss its nature by ways of experimenting with ORL face database; verify its advantages compared with other means.

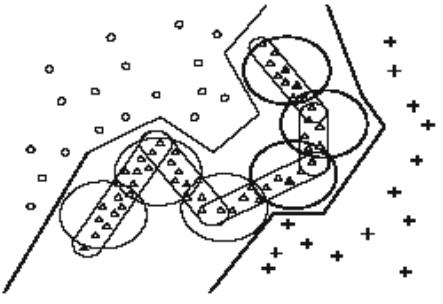


**Fig. 1.** Hyper Sausage Neuron chains based Biomimetic Pattern Recognition in comparison with BP and RBF networks

## 2 Definitions and Basic Properties

We assume that the sample nodes are given as a set of n sample points S distributed inside a n-dimensional domain. Let B be the set of sample points that define the domain boundary. For simplicity, we assume that the convex hull of the set of sample S is contained inside the domain. We also assume that every sample node has the same maximum distance. We always assume that the sample node is connected. We first give some geometry notations that will be used in the remainder of this section to mathematically formulate the problems considered. Let $||x - y||$ denote the Euclidean distance of two sample points x and y.

**Definition 1.** The distance of a point x to a set of sample points V, denoted by *dist(x,V)*, is the smallest distance of x to all sample points of V. In other words

$$dist(x, V) = \max_{y \in V} ||x - y|| \tag{1}$$

Notice that the point set V may be infinite. For example, V could be all points lying on a segment uv. We use *dist(x,V)* to denote the smallest distance from x to all points on the segment uv.

**Corollary 1.** If V is segment uv, then Actively function of Hyper Sausage Neuron is

$$f(x) = \begin{cases} 1 & dist(x,V) < \varepsilon \\ 0 & otherwise \end{cases}$$

where $\varepsilon$ is positive real.

**Corollary 2.** If V is triangle uvw, then Actively function of multi weight vector neurons is

$$f(x) = \begin{cases} 1 & dist(x,V) < \varepsilon \\ 0 & otherwise \end{cases}$$

where $\varepsilon$ is positive real. Given two point sets U and V , the breach distance *dist(x,V)* is defined as $\min_{x \in V}$ *dist(x,V)*. In other words, $dist(x,V) = \min_{x \in V, y \in V} ||x - y||$. Usually, the breach distance is called just distance in the literature.

**Definition 2.** The coverage-distance of a point set U by another point set V, denoted by *cover(U,V)*, is the maximum distance of every point $x \in U$ to V . That is,

$$cover(U,V) = \max_{x \in U} dist(x,V)$$

Notice that, the breach distance *dist(U,V)* is symmetric, i.e. *dist(U,V)* = *dist(V,U)*, while the coverage distance *dist(U,V)* is not symmetric. Here, both point sets U and V can be infinite. For example, U can be a path connecting two points s and t and V all sample nodes. Given a path $\Pi(s,t)$ inside connecting s and t, the coverage-distance $\max_{x \in \Pi(s,t)}$ *dist(x,S)* of the path $\Pi(s,t)$. specifies how well the path is protected by the samples, while, on the reverse side, the breach distance specifies how far the path is from all samples. Thus, for samples set networks, the coverage problem has two folds: the best coverage and the worst coverage, which are defined as follows:

**Definition 3.** A path $\Pi(s,t)$ that achieves the minimum coverage-distance $cover(\Pi(s,t), S)$ is called a best-coverage-path. The minimum coverage-distance $cover(\Pi(s,t), S)x \in \Pi(s,t)$ of all paths connecting s and t is called the best-coverage-distance or the support-distance.

Thus, given a set of sample S, a starting point $s \in S$ and $s \in R^n$, and an ending point $t \in S$ and $t \in R^n$, we find a path $\Pi(s,t)$ inside to connect s and t such that the coverage distance $cover(\Pi(s,t), S) = \max_{x \in \Pi(s,t)} dist(x,S)$ is minimized. In other words, we try to find a path connecting s and t such that every point x of the path is covered by some sample nodes with small distance.

**Definition 4.** A path $\Pi(s,t)$ that achieves the maximum breach distance $dist(\Pi(s,t), S)$ is called a worst-coverage-path. The maximum breach-distance $dist(\Pi(s,t), S)$ of all paths that connecting s and t is called the worst-coverage-distance or the breach-distance.

**The Principle of Homology-Continuity(PHC).** In feature space $R^d$, suppose that set A is a point set including all samples in class A. If $x, y \in A$ and $\varepsilon > 0$ are given, there must exist set B.

$$B = \{x_1 = x, x_2, ..., x_{n-1}, x_n = y | (x_m, x_{m+1}) < \varepsilon, \forall m \in [1, n-1], m \in N)\} \subset A \tag{2}$$

**The Prioritz Ordered Neural Networks(PONN).** In general, the priority ordered units in a PONN could be neurons or parts of nets or different modules as shown in [Fig. 2]. The inside of the dotted rectangle is a PONN with n modules, and output of the module with lower number of footnote means with higher priority. All the inputs of each module (sub network or subsystem) are xRd. However, the structures and priorities of different modules are generally different. The general mathematical description of a PONN is as

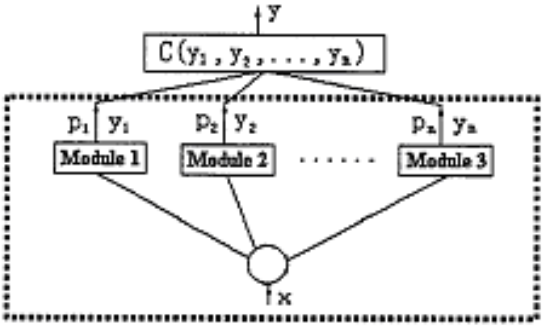



**Fig. 2.** This is the general mathematical ordered architecture of neural networks. The priority and the output of module k are denoted as $p_k$ and $y_k$ respectively. The outside of the dotted rectangle is symbolic presentation for mathematical description of the ordered priorities of modules.

follows: The mathematical description of each module depending on its structure, $C(y_1, y_2...y_n)$ is the decision function of the final output for the whole neural network. Moreover,

$$y = C(y_1, y_2...y_n) = y_s,$$

$$s = \min i | p_i = \max p_j | Q(y_j) = 1,$$

Where Q is described as the conditional map $Q : \bigcup_{j=1}^{\infty} R^{N_j} \to \{0, 1\}$.

## 3 Coverage Algorithm

The algorithm first induces a local neighborhood structure on the data, and then uses this local structure to find a group of minimum covering spheres in different lower dimensional spaces. The result of the COVERAGE algorithm is a geometrical representation of a class of samples. The algorithm takes input as

the sample points in the high-dimensional input space X, and outputs a group of lower-dimensional subspace' basis of the sample points. The only free parameter (k) appears in following steps.

- The training steps of COVERAGE algorithm of single class problem are as follows:
  – Step:
  –– 1 Construct neighborhood graph by k-nearest neighbors.
  –– 2 Compute the minimum covering subspace of each group of points in the same k-nearest neighbors. Then get the basis of the sample points and the subspace.

To cover a set of sample points in feature space has a little difference comparing with COVERAGE (P). In the view of Principle of Homology-Continuity [6] (PHC) and consider the small disturbances, all points near a sample point ought to be considered as samples of the same class. So the problem turns to finding the closed subspace of smallest distance which contains a given set of n closed subspace which distance is $\varepsilon$ in sample space. Fischer and Gärtner [7] proved that the problem of finding the smallest enclosing subspace of subspace is computationally equivalent to the problem of finding the minimum-norm point in the convex hull of a set of subspace.

- The steps of recognition of single class problem are as follows:
  – Steps:
  –– 1 Transform the signals to a point P in the vector feature space.
  –– 2 Compute the distance *dist(P, V)* from the current sample point P to every minimum covering $V_i$ and the distance *dist(P, V)* from the current sample point P to the hyper plane $S_i$ spanned by the basis of the k sample points (the algorithm calculating the distance of a point and an infinite subspace can be found in [11]). Denote the small disturbance as $\varepsilon$. If $dist(P, S_i) \leq \varepsilon$, and $dist(P, S_i) \leq \rho_i$, then proved the sample point belongs to this class.

We use covering sphere in this paper.

- The training steps of COVERAGE algorithm of multi classes' problem are as follows:
  – Steps:
  –– 1 The training steps of each class are same to table 1
  –– 2 If the distance of two centers $c_{1i}, c_{2j}$ of different classes is too near, i.e. $||c_{1i}, c_{2j}|| < \max(\rho_{1i}, \rho_{2j})$, PONN algorithm could be used to generate a priority order in order to deal with the inseparable situation, where $c_{1i}, c_{2j}$ is radius minimum covering sphere.
- The steps of recognition of multi class problem are as follows:
  – Steps:
  –– 1 The recognition steps of each class are similar to table 2
  –– 2 With the help of PONN, there is no conflict in the multi class recognition. The current sample point P should belong to class A, if $l_A = \min(l_t), t = A, B, C, ...,$ t denote the class. Where $l_t = \frac{dist(P,S_i)}{\rho_{ti}}$ with the limitation that $dist(P, S_i) \leq \delta_t$ as [Fig. 3] and [Fig. 4].

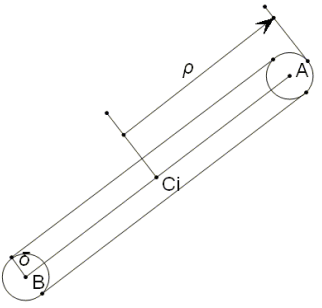



**Fig. 3.** If k=2, there are two points as the basis of a minimum covering sphere, and the actual dimensions of this sphere is only 1. Then hyper sphere only has one degree of freedom. Considering little disturbances, its shape just like a hyper-sausage.

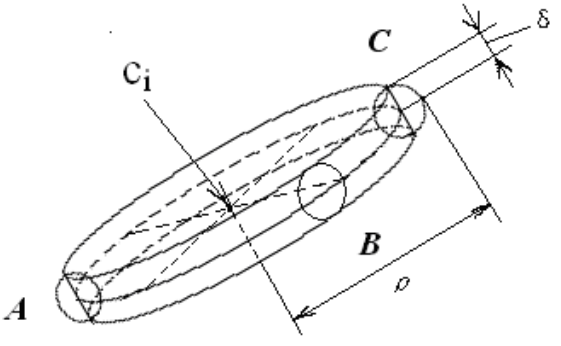



**Fig. 4.** If k=3, the minimum covering sphere is actually in a 2 dimensional subspace. The hyper sphere has two degrees of freedom, Its shape looks like a piece of cake in one of the 3 dimensional sub-spaces.

## 4 Applications of Coverage Algorithm

In this section, a face recognition experiment is designed to evaluate the performance of the proposed algorithm. Support vector machines, as one kind of popularity methods during recent time, was selected as a comparison. With the same training sets and features extraction, the ratio of correct recognition is compared. The ORL database is one of the most popular used databases recently. It contains a set of faces taken between April 1992 and April 1994 at the Olivetti Research Laboratory in Cambridge, UK. The size of each photo is 92*112, black background. The face expression and some other details are all dissimilarity, e.g. smiling or expressionless, eyes open or closed, with glasses or without glasses, and different postures. All the photos only subtract a lighting surface, which is gained by linearity estimation. And their average values and variances are tuned to 168 and 50. The experiments use each person's former 5 photos as training data set. Principal Components Analysis technique[10] is

utilized to extract features. 100 eigenvectors are selected from all the 200s. The ratio of the sum of the selected eigenvalues to the sum of all the eigenvalues is 91.72%. All the 400 photos in the database are employed as the testing set. The support vector machines as the comparison method, use the radial based kernel, which has the form of : $exp(-\gamma|X(:,i) - X(:,j)|^2)$. The SVM tool box is OSU_SVM 3.00. And the test result is as follows:

**Table 1.** Experiments Result

| Method | Ratio of correct |
|---|---|
| Coverage | 81.5 |
| SVM(Gamma=1) | 78.75 |
| SVM(Gamma=2) | 74 |

From the above result, it can be seen that Coverage algorithm has better performance than SVMs.

## 5 Conclusion and Future Work

With the result of experiments, Coverage algorithm emerges fine potential power in learning abilities. From above, we conclude that:

1. Coverage Algorithm emphasizes analyzing the distribution of a certain class samples in feature space firstly.
2. The prior information on the distribution of the sample set can improve the generalization ability greatly.
3. Coverage Algorithm has obtained better results than traditional pattern recognition methods, such as SVM in many applications.
4. Coverage Algorithm has more flexibility to fit multidimensional subspace manifold embedding. The maximum subspace dimension is determined by k.

There is still much room to improve the efficiency of the proposed algorithm. The minimum volumes of subspace covering methods can be used with more accurate approach algorithm based on some kinds of monotype.

## References

1. Wang Shoujue: Biomimetic Pattern Recognition (in Chinese). ACTA Electronica Sinica, Vol. 30, No. 10. (2002) 1417-1420
2. Wang Shoujue: Priority Ordered Neural Networks with Better Similarity to Human Knowledge Representation. Chinese Journal of Electronics, Vol. 8, No. 1. (1999) 1-4
3. Wang Shoujue, et al: The Sequential Learning Ahead Masking(SLAM) model neural networks for pattern classification. Proceedings of JCIS'98, Vol. IV. RTP, N.C. USA (1998) 199-202

4. J. J. Sylvester: On Poncelot's approximate linear valuation of surd forms. Philos. Mag. Ser. 4 (20) (1860) 203-222
5. T. H. Hopp, C. P. Reeve: An algorithm for computing the minimum covering sphere in any dimension. Technical Report NISTIR 5831, National Institute of Standards and Technology (1996)
6. Wang Shoujue, Zhao Xingtao: Biomimetic Pattern Recognition Theory and Its Applications. Chinese Journal of Electronics, Vol. 13, No. 3. (2004) 56-60
7. Kaspar Fischer, Bernd Gärtner: The Smallest Enclosing Ball of Balls: Combinatorial Structure and Algorithms. SoCG'03, San Diego, California, USA (2003)
8. E. Welzl: Smallest enclosing disks (balls and ellipsoids). In h. Maurer, editor, New Results and New Trends in computer Science, Vol. 555 of Lecture Notes Comput. Sci. Springer-Verlag, Berlin Heidelberg New York (1991) 359-370
9. V. Chvtal: Linear programming. W. H. Freeman, New York, NY (1983)
10. R. O. Duda, P. E. Hart, D. G. Stork: Pattern Classification. New York: John Wiley Sons (2001)
11. Wang Shoujue, Zhao Guliang: An Algorithm of Analysis Tools On Points Distribution In High Dimension Space: The Distance of A Point And An Infinite Sub-space. Proceedings of ICNNB05, Vol. 3. (2005) 1503-1506

# A Texture-Based Algorithm for Vehicle Area Segmentation Using the Support Vector Machine Method

Ku-Jin Kim[1], Sun-Mi Park[1], and Nakhoon Baek[2,⋆]

[1] Dept. of Computer Engineering, Kyungpook National Univ., Daegu 702-701, Korea
kujinkim@yahoo.com, disvogue@graphics.knu.ac.kr
http://graphics.knu.ac.kr
[2] School of EECS, Kyungpook National Univ., Daegu 702-701, Korea
oceancru@gmail.com

**Abstract.** The vehicle area segmentation is important for the various applications including ITS (Intelligent Transportation System). We present a novel approach for segmenting a vehicle area from still images of vehicles on the asphalt paved road captured from outdoor CCD cameras. Our algorithm classifies the partitioned grid areas in the input vehicle image into road or vehicle classes. Texture features are used for representing each class, and we use SVM (Support Vector Machine) method for the classification. Our preprocessing process partitions given sample images into a set of grids, and classifies each grid area into two classes: i) road class, and ii) vehicle (non- road) class. We use GLCM technique to extract the feature values for each class, and sample classes are trained by using the SVM. The SVM constructed in preprocessing step is applied for each given input image to decide whether the grid in the image belongs to the road area or not. After marking the grids as road or vehicle classes, we find the optimal rectangular grid area containing the vehicle. The optimal area is found by using a dynamic programming technique. Our method efficiently achieves high reliability against noises, shadows, illumination changes, and camera tremors. We experimented on various vehicle image set, where the images in each set are captured in different road environment. For the largest set, by using 50 sample images, where each image with $1280 \times 960$ resolution or $13 \times 12$ grid areas, our algorithm shows 94.31% of successful vehicle segmentation from 211 images with various kinds of shadows and illumination changes.

**Keywords:** Vehicle area segmentation, Texture-based, Support vector machine.

## 1 Introduction

In this paper, we present a method of segmenting the vehicle area from the asphalt paved road images. Our vehicle segmentation algorithm assumes that a

⋆ Corresponding author.

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 542–549, 2007.

single input image is given. The given image is captured from the stationary outdoor CCD camera which is mounted on a fixed pole in the air appropriately high above the road. The segmented vehicle area can be used for the further various applications. As an example, the license plate recognition and vehicle classification, which are used in various applications including automatic toll fee collection systems, traffic monitoring systems, and Intelligent Transportation Systems(ITS), use the vehicle segmentation as one of their fundamental operations.

When the input images have a reference frame of background, that is, the background image without a vehicle is additionally given, the background subtraction method is one of the most widely used method for the vehicle segmentation. It compares a background and a vehicle image in a pixel-by-pixel manner, to report a set of altered pixels as the vehicle area. Although it is intuitive and straightforward, it is sensitive to the illumination changes, camera tremors, shadows and other noises. For the cases where the vehicle color is similar to the background color or where shadows of the target vehicle itself or other vehicles exist, it may fail to find the vehicle area. It also has drawbacks that background images and threshold values should be dynamically updated for deriving correct results[1,2,3,4].

As one of extensions to the background subtraction methods, Lam et al.[5] focused on the observation that the road, vehicle, the reflection from the vehicle surface, and the cast shadow of the vehicle have different texture properties. They computed the texture features to differentiate the road, vehicle, shadow, and reflection, and then constructed a texture likelihood map, a luminance likelihood map, and a chrominance likelihood map. By combining those maps, they construct a region mask for the vehicle area.

For the vehicle segmentation, having background reference frame is helpful. However, to get more accurate vehicle segmentation, the background part in the vehicle image and the background reference frame should have similar illumination conditions; thus, it is necessary to capture the background image in a short time before capturing the input vehicle image. Usually, to generate a background image for each input vehicle image, additional cost is necessary. Moreover, in some cases, we are not able to get the background image at all. Therefore there are needs for segmenting a vehicle from the given single vehicle image without a background reference frame.

Given a single vehicle image, the difficulties of differentiating unnecessary visual information such as lanes, shadows, and other noises from the vehicle, such as the reflection from the shiny exterior, are well known. For the vehicle area segmentation, there is a nice survey paper written by Sun et al[6]. They roughly classified segmentation approaches into three categories: i) knowledge-based approaches, ii) stereo-based approaches, and iii) motion-based approaches. While the knowledge-based approaches require a single vehicle image, other approaches require two or more images or a video sequence. Though they focused on systems where the camera is mounted on the vehicle, the knowledge-based approaches are applicable on our problem. Knowledge-based approaches use a priori knowledge

to detect vehicle locations. There are representative approaches uses the information on vehicles, such as symmetry feature, color, shadow, geometric features such as vertical/horizontal edges or corner points, vehicle lights, and texture. Compared to the others, there are only a few researches based on the textural feature.

Kalinke et al.[7] proposed a vehicle segmentation method based on entropy, which contains the information of the intensity value distribution for a region. Entropy is introduced as a measure of expected information for the quantity of attention for given region[8]. Kalinke et al. found the region of interest having high entropy, and assumed that region as a vehicle area. The measure of entropy efficiently estimates the quantity of structure or texture of the region, but it is known as rather inaccurate compared to the co-occurrence based methods.

In this paper, we present a vehicle segmentation method, aiming to an integrated vehicle recognition system with the capability of license plate recognition, vehicle classification, and so on. Our method classifies the local regions of input vehicle images based on the texture feature represented by GLCM(Gray-Level Co-occurrence Matrix)[9,10]. The SVM(Support Vector Machine) method[11,12] is used for the classification. Given a vehicle image, our method partitions it into a set of grids. Each grid is classified as a road or vehicle class. After the classification, we found the optimal rectangular area containing the vehicle based on the dynamic programming approach. Our algorithm currently can be applied for segmenting the area of one vehicle in the image, but it can be extended to segment two or more vehicle areas by slightly changing the global optimization part which is implemented by the dynamic programming approach.

We experimented on various vehicle image sets, where the images in each set are captured in different road environments. For the largest set, by using 50 sample images, where each image with $1280 \times 960$ resolution or $13 \times 12$ grid areas, our algorithm shows 94.31% of successful vehicle segmentation from 211 images with various kinds of shadows and illumination changes. Other small size sets show the success rate of up to 100%.

This paper is organized as follows. In section 2, details of our vehicle segmentation method will be presented. Section 3 contains the experimental results from the prototype implementation. Conclusions and remarks on future work will be followed in Section 4.

## 2 Vehicle Area Segmentation Method

In this section, we are focusing on the method of segmenting the vehicle area from the road images. More precisely, we present a method of extracting the moving vehicle area from a set of images captured from a pre-specified location, through removing background road areas. During these formulations, to distinguish the background road areas more systematically, we first divide the image into a set of grid areas, and introduce a feature vector for each independent grid area.

Our method consists of two stages: the preprocessing stage starts from defining feature vectors of grid areas, and an SVM is trained with these feature vectors,

to finally decide whether a given grid area belongs to the background road class or not. In the main stage, we choose the grid areas belonging to an optimal vehicle area, based on a dynamic programming technique.

### 2.1 Feature Vector Construction

An input image is partitioned into a set of axis-aligned rectangular areas with the size of width $W$ and height $H$ as follows:

$$I_{ij} = \{p_{xy} | i \cdot W \leq x < (i+1)W, j \cdot H \leq y < (j+1)H\}, \quad (1)$$

where $p_{xy}$ is a pixel at the position $(x, y)$. To analyze the internal texture information in a grid area, we convert the input image into a grayscale one. And then, the image is quantized into $D$ grayscale levels to filter unnecessary noises out. In the next section, we use the parameter values of $W = 100$, $H = 80$ and $D = 32$, for our experiments. Since we first select a specific set of parameters and then apply SVM training stage, we can also choose another set of parameters for another set of images.

For a grid area of $W \times H$ pixels, we need to extract the abstraction of internal texture information, and we have plenty of previous works for this purpose. We use GLCM(gray level co-occurrence matrix) method, which is one of widely-used ones in the field of texture analysis.

In the GLCM method, to abstractly express neighborhood information, we build up the co-occurrence matrix, which is a two-dimensional square matrix whose element corresponds to a transition from one pixel to its neighbor pixel. Since we use $D$ quantized gray levels, our GLCM becomes a $D \times D$ square matrix. When a pixel with the quantized gray level $d$ has a neighbor pixel with the gray level of $d'$, we can interpret it as a transition from $d$ to $d'$ and increase the corresponding GLCM element by one.

Considering the symmetry conditions, a pixel may have four neighbors: its south-west($SW$), south($S$), south-east($SE$) and east($E$) pixels. To apply the GLCM method, we can use any combination of these neighborhood relations. In the case of artifacts including vehicles, there would be relatively many horizontal and/or vertical edges and they may act as noises in the texture analysis. As we can see in the next section, our experiments also show that accumulating the co-occurrence of $S$ and $E$ neighbors decreases the final success rate. Thus, we use the $SW$ and $SE$ neighbors in our experiments.

In the case of $SE$ neighboring relations, the matrix element $GLCM[d][d']$ equals to the number of transitions from a pixel $p_{xy}$ with the quantized gray level $d$ to its $SE$ neighboring pixel $p_{(x+1)(y+1)}$ with the quantized gray level $d'$, where $p_{xy}, p_{(x+1)(y+1)} \in I_{ij}$. The $SW$ neighboring relations are also handled in a similar way.

### 2.2 Decision by Support Vector Machine

Since the SVM method has difficulties to directly handle matrices, we re-arrange the resulting GLCM matrices in a row-major order, to get $32 \times 32 = 1,024$

dimensional vectors, with assigning $D = 32$. To apply the SVM method in a traditional way, we first train the SVM with a set of images. These sample images are grayscale quantized into $D = 32$ levels, and partitioned into a set of grid areas whose size is $W \times H = 100 \times 80$. After deciding each grid area whether it belongs to the background area, which includes asphalt-paved loads, painted guide lanes, shadows and so on, we put all these information into the SVM, to train it up. The trained SVM will answer yes or no to the question of whether the given grid area belongs to the background area or not, for the input $D \times D = 1,024$ dimensional vectors resulting from the GLCM method.

### 2.3 Global Optimization

For an input image, we have a set of grid areas $I_{ij}$'s. Through calculating GLCM matrices for each grid area, we can decide whether it belongs to the background area or not. Since all these procedures locally focus on the inter-grid information, there would be noises and/or wrong decisions from a global point of view. Thus, we need to remove the noises and find a globally optimal solution. Additionally, we still have no way to decide the final vehicle area, among the scattered non-background grid areas.

In this paper, we use a dynamic programming approach to find these globally optimal solutions for the vehicle area segmentation. We assigned the weight values of $W_{\text{back}}$ and $W_{\text{fore}}$ for background grid areas and non-background (or foreground) grid areas, respectively. For a grid area $I_{ij}$, let its weight be $w_{ij}$, according to the decision of SVM machine. Now, the total weight of rectangular $m \times n$ grid areas whose top-left grid area is $I_{pq}$ can be expressed recursively as follows:

$$w_{pqmn} = \begin{cases} 0, & \text{if } m = 0 \text{ or } n = 0 \\ w_{pq}, & \text{if } m = 1 \text{ and } n = 1 \\ w_{pq} + w_{(p+1)q(m-1)1} + w_{p(q+1)1(n-1)} & \\ \quad + w_{(p+1)(q+1)(m-1)(n-1)}, & \text{if } m > 1 \text{ or } n > 1. \end{cases} \tag{2}$$

Now, we evaluate all the possible $w_{pqnm}$'s and report the $m \times n$ rectangular region:

$$\{I_{ij} | p \le i < p + m, q \le j < q + n\}. \tag{3}$$

with the maximum weight. We have results for various $W_{\text{back}}$ and $W_{\text{fore}}$ values and the final result will be shown in the next section.

## 3 Experimental Results

We perform the experiments on three different vehicle images sets, where those sets were generated in three different places. For a specific set, depending on its size, we choose some training images. The number of training images and grids, the number of test images, and the rate of successful vehicle segmentation are shown in Table 1. For each set, training and test images are disjoint.

**Table 1.** Statistics on our experiments

| image set | no. of training images | no. of test images | no. of successful segmentation(%) |
|---|---|---|---|
| Set $A$ | 50 | 211 | 199 (94.31%) |
| Set $B$ | 20 | 37 | 37 (100.00%) |
| Set $C$ | 17 | 17 | 17 (100.00%) |

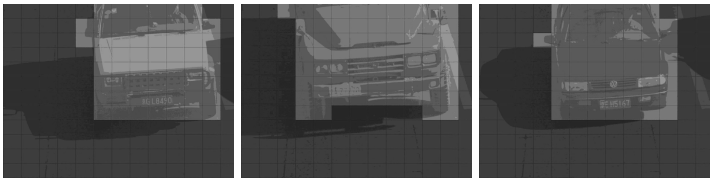
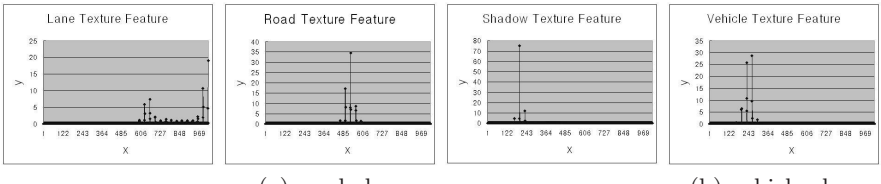


**Fig. 1.** Training images and grids for Set $A$

For the image set $A$, the training images with grids are shown in Fig. 1. We show the grid areas which are trained as the background road class in dark gray areas. The grid areas in light gray colors are trained as the non-background (or vehicle) class. Notice that the grid area containing the boundary of the vehicle and a part of road is trained as the vehicle class. The road class grid is containing the shadows, lanes, and the road surfaces. To decide the proper neighbors for generating GLCM matrix, we tried several combinations. We found that using the co-occurrences of $S$ and/or $E$ neighbors leads the lower rate of successful vehicle segmentation than the case of using only $SE$ and/or $SW$ neighbors. For the set $A$, we used $SW$ and $SE$ neighbors to generate GLCM matrix. Fig. 2 presents the GLCM elements in row-major order as graphs. Fig. 2(a) shows the examples of road surface, lane, and shadow textural features. Fig. 2(b) shows one example of vehicle textural feature.




(a) road class (b) vehicle class

**Fig. 2.** Texture features represented in graphs

In Fig. 3, we present some test images in the set $A$, and their segmentation results. In the resulting images, light and dark gray areas represent that the corresponding grid is classified as a vehicle class and a road class, respectively. To check the validity of our method, our test images include various noised ones: some images have very dark areas around the vehicle boundaries mainly due to shades, and actually they are hard to decide as the vehicle area even with

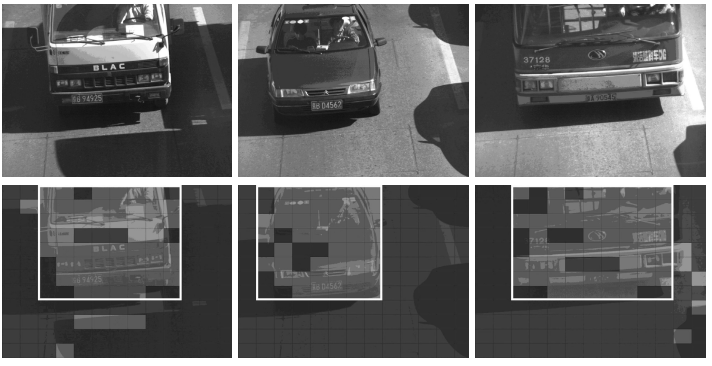


**Fig. 3.** Test result of Set $A$: test images(first row), and their corresponding segmentation results(second row)

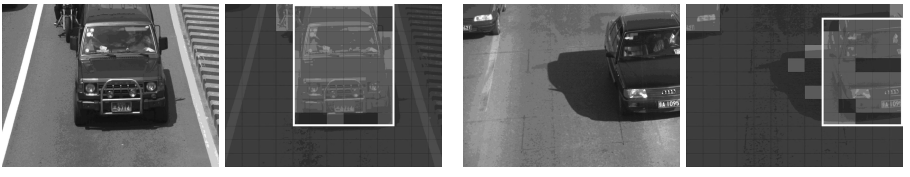


**Fig. 4.** Test images and segmentation results from Sets $B$ and $C$

human eyes. Thus, we treat the difference of one row or one column from grids including the exact boundary of the vehicle as the successful cases. Additionally, we originally aim to use the vehicle segmentation result as the input of the next stage license plate identification and/or vehicle classification, and they usually endure one row or one column differences. Though there are some wrong classifications in the example, the global optimization process segments the vehicle area successfully. Fig. 4 shows the segmentation results for test images in the sets $B$ and $C$, where only $SW$ neighbors are used to construct the GLCM.

## 4 Conclusion and Future Work

In this paper, we presented a vehicle area segmentation method from the outdoor road images. We classified the partitioned grid areas into two classes as a background road class and a vehicle (non-background) class. The vehicle images are partitioned into a set of grids, and the grids which contain the road surface, lanes, or the cast shadow from the vehicles are classified into a road class. The grids contain vehicle parts are classified into a vehicle class. For the classification, we use the SVM method based on GLCM textures of each grid as feature values. We experimented on a several sets of vehicle images, where each set was composed by the captured images in different places with different illumination and road surface conditions. For different sets, different samples were used for

training the classes. Depending on the data set, experiments show the rate of successful vehicle segmentation from 94.31% to 100%.

Currently, the proposed method is restricted to the input images with sunny or cloudy weather conditions. If the weather condition changes to rainy or snowy, we may need different feature values. As a future work, we are planning to refine the classes to include wider types of regions in the vehicle image by using the multi class SVM.

## Acknowledgment

Prof. Kim was supported by the Korea Research Foundation Grant funded by Korean Government (MOEHRD) (R04-2004-000-10099-0).

## References

1. Michalopoulos, P.G.: Vehicle detection video through image processing: The autoscope system. IEEE Trans. Vehicular Technol. **40**(1) (1991) 21–29
2. Fathy, M., Siyal, M.Y.: An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis. Pattern Recognition Letters **16**(12) (1995) 1321–1330
3. Fathy, M., Siyal, M.: A window-based image processing technique for quantitative and qualitative analysis of road traffic parameters. IEEE Trans. Vehicular Technol. **47**(4) (1998) 1342–1349
4. Gupte, S., Masoud, O., Martin, R.F.K., Papanikolopoulos, N.P.: Detection and classification of vehicles. IEEE Trans. Intell. Transport. Syst. **3**(1) (2002) 37–47
5. Lam, W., Pang, C., Yung, N.: Highly accurate texture-based vehicle segmentation method. Optical Engineering **43**(3) (2004) 591–603
6. Sun, Z., Bebis, G., Miller, R.: On-road vehicle detection: A review. IEEE Trans. Pattern Anal. Mach. Intell. **28**(5) (2006) 694–711
7. Kalinke, T., Tzomakas, C., von Seelen, W.: A texturebased object detection and an adaptive model-based classification. In: Procs. IEEE Intell. Vehicles Symp. '98. (1998) 341–346
8. Shannon, C.: A mathematical theory of communication. Bell Syst. Tech. J. **27** (1948) 379–423, 623–656
9. Haralick, R.M., Shanmupm, K., Dinstein, I.: Textural features for image classification. IEEE Trans. Syst., Man, and Cybern. **SMCZ-3**(6) (1973) 610–621
10. Chantler, M.J.: The effect of variation in illuminant direction on texture classification. PhD thesis, Dept. Computing and Electrical Eng., Heriot-Watt Univ. (1994)
11. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other Kernel-Based Learning Methods. Cambridge Univ. Press (2000)
12. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. J. Mach. Learning Res. **2** (2001) 265–292

# The Study of Some Important Theoretical Problems for Rough Relational Database

Qiusheng An

School of Science, Xi'an Jiaotong University, 710049, Xi'an, P.R. China
aaqqss@sina.com
Shanxi Normal University, 041004, Linfen, P.R. China

**Abstract.** In this paper, some important theoretical problems about rough relational database(RRDM) are studied. Firstly, the relationship between rough relational database and non-deterministic information systems is analyzed, secondly, the decomposition operator is introduced based on rough relational operator and its basic properties are discussed. In addition, the definability of rough relational database and the rough description of attribute values of rough relational database are investigated respectively. The redundant factor of rough functional dependency is proposed, as well as rough functional dependency and its inference rules are mainly studied .

**Keywords:** RRDM, rough data querying, rough functional dependency, rough normal forms

## 1 Introduction

The rough relational database model was introduced by Theresa Beaubouef. In RRDM, Theresa Beaubouef defined some rough relational operator, studied the information-theoretic measures of uncertainty measures of rough sets of uncertainty for rough sets and rough relational database, gave the definitions of rough functional dependencies, rough data querying and investigated the normal forms of RRDM [1-6].

Non-deterministic information systems (NIS) and incomplete information systems have been proposed for handling information incompleteness[7].

In this paper, the relationship between RRDM and NIS is firstly analyzed, and the decomposition operator is introduced. Moreover, the definability of rough relational database and the rough description of attribute values of RRDM are investigated respectively. Finally, the redundant factor of rough functional dependencies is proposed, rough functional dependency and its inference rules are mainly studied.

## 2 The Relationship Between RRDM and NIS

To all appearances, RRDM and NIS have same expression form, while there are some differences, which focus on the following issues:

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 550–556, 2007.

1) Data redundancy. Like classical relational database, the reduplicate tuples are unallowed in RRDM according to rough normal form, whereas the reduplicate tuples are allowed in NIS.

2) The relationships among the elements of an attribute value. Like information system is the generalization of relational database, a NIS is the generalization of RRDM. In NIS, the relationship between the elements of an attribute value is "or" in general, and in RRDM, the relationship between the elements of an attribute value are "or" or "and".

3) Their research areas are different. NIS focuses on the definability of a set in NIS, the consistency of an object, data dependency in NIS, rules in NIS, reduction of attributes in NIS; whereas RRDM concentrates on rough relational operation, rough data querying, rough functional dependency and rough normal forms.

## 3 Rough Relational Operator and Decomposition Operator

As we known, Theresa Beaubouef defined some rough relational operators. Decomposition operator is introduced in this paper and some of its properties are studied.

**Definition 1.** ([2,8]) *An interpretation* $\alpha = (a_1, a_2 \ldots, a_m)$ *of a rough tuple* $t_i = (d_{i1}, d_{i2}, \ldots, d_{im})$ *is any value assignment such that* $a_j \in d_{ij}$ *for all* $1 \leq j \leq m, a_j$ *is called a sub-interpretation of* $d_{ij}$ .

**Definition 2.** ([8]) *Let* $r_1$ *and* $r_2$ *be rough relations, and* $r_1$ *has attribute set* $(A_1, A_2, \ldots, A_m)$*, its attribute domain is* $(D_1, D_2, \ldots, D_m), r_1(A_{ij})$ *is one of attribute value of* $r_1$*, rough relation* $r_2$ *has same attributes and attributes domain with* $r_1$*, and its attribute value is denoted by* $r_2(A_{ij})$*, if* $r_2(A_{ij}) \subseteq r_1(A_{ij})$ *for all* $i, j$*, then we call rough* $r_2$ *a decomposition of* $r_1$*, denote* $r_2 = \Gamma(r_1)$*, we use* $\Gamma$ *stands for the decomposition operator.*

**Proposition 1.** *Let* $(R_1, R_2, \ldots, R_n)$ *be the set of tuples of rough relation* $r$*,* $\alpha_1, \alpha_2 \ldots, \alpha_n$ *be the interpretations of* $r$*, then the rough relation* $s$ *composed by* $(\alpha_1, \alpha_2 \ldots, \alpha_n)$ *must be a decomposition of* $r$*.*

**Theorem 1.** *If rough relation* $s$ *is the unique decomposition of rough relation* $r$*, then* $\underline{R}r = s, \overline{R}r = r$ *hold.*

*Proof.* Let rough relation s be the unique decomposition of rough relation $r$, and $r$ is composed by $(t_1, t_2, \ldots, t_n), s$ is composed by $(\alpha_1, \alpha_2 \ldots, \alpha_n)$, $k_{ij}$ be arbitrary attribute value of arbitrary tuple $\alpha_i$ of $s$, $v_{ij}$ be arbitrary attribute value of arbitrary tuple $t_i$ of $r$. Because s is the unique decomposition of r, so $k_{ij} \subseteq v_{ij}$ holds, and $\alpha_i \subseteq t_i, \alpha_i \subseteq \underline{R}t_i$ hold, in addition, we only have $\alpha_i \subseteq \underline{R}t_i$ for $r(j \neq i)$, so $\underline{R}t_i = \alpha_i$ holds. Similarly, we have $\alpha_j \subseteq \underline{R}t_j, \underline{R}t_j = \alpha_j$, moreover $\underline{R}r = \{\underline{R}t_1, \underline{R}t_2, \ldots, \underline{R}t_n\} = \{\alpha_1, \alpha_2 \ldots, \alpha_n\} = s$ based on rough set theory, and $t_i \cap t_i = t_i \neq \emptyset$ for arbitrary tuple $t_i$. So $t_i \subseteq \overline{R}t_i$, and only $\overline{R}t_i = t_i, \overline{R}r =$

$\{\overline{R}t_1, \overline{R}t_2, \ldots, \overline{R}t_n\} = \{t_1, t_2, \ldots, t_n\} = r$ hold; in conclusion, $\underline{R}r = s, \overline{R}r = r$ hold.

**Theorem 2.** *If rough relation s is the arbitrary decomposition of rough relation r, then* $\overline{R}r = r$ *must be hold.*

**Proposition 2.** *Let rough relation s be the arbitrary decomposition of rough relation* $r, k_{ij}$ *be arbitrary attribute value of arbitrary tuple* $\alpha_i$ *of s,* $v_{ij}$ *be arbitrary attribute value of arbitrary tuple* $t_i$ *of r, if* $\exists k_{ij} \subset v_{ij}$ *for all* $\alpha_i$*, then* $\overline{R}s = \emptyset$ *must be hold.*

**Proposition 3.** *Rough relation s is a decomposition of rough relation r if and only if* $k_{ij} \subseteq v_{ij}$ *(where* $k_{ij}$ *be arbitrary attribute value of arbitrary tuple* $\alpha_i$ *of s,* $v_{ij}$ *be arbitrary attribute value of arbitrary tuple* $t_i$ *of r).*

## 4 The Definability of RRDM and Rough Description of Attribute Values

### 4.1 The Definability of RRDM and Data Querying

**Definition 3.** *For a RRDM S, let X be the result set of rough data querying. When X can be expressed by these tuples of S, X is accurately definable in S,* $\overline{R}X = \underline{R}X = \{r_i | r_i \in S \wedge r_i \in X, 1 \leq i \leq |U|\}$*, when X can be expressed by these tuples of S, and can't be expressed accurately, X is rough definable in S,* $\overline{X} = \{r_i | \exists i(r_i \in S) \wedge |r_i(a)| \geq 1 \wedge r_i(a_i) \cap C \neq \emptyset, 1 \leq i \leq |U|, C \in X\}, \underline{R}X = \{r_i | r_i \in S \wedge |r_j(a_j)| = 1 \wedge r_i \in X, 1 \leq i \leq |U|, 1 \leq j \leq |A|\}$*, where* $r_i$ *denotes any tuples of S, and* $|r_i(a_j)|$ *denotes the number of sub-interpretation , C is an attribute value of X.*

Here we divide the rough querying into two classes: certain data querying and possible data querying. Certain data querying is to search these objects fully matching the querying conditions, possible data querying finds these records satisfied all possible matching with querying conditions.

**Theorem 3.** *The result of certain data querying is the minimal set that satisfies querying conditions,* $X = \overline{R}X = \underline{R}X = \{r_i | r_i \in S \wedge r_i \in X \wedge r_i(a) = C, 1 \leq i \leq |U|\}$ *where* $r_i(a)$ *is one of attribute values, and C is the attribute value that user want to query.*

**Theorem 4.** *The result of possible data querying is the maximal set that satisfies querying conditions, and we denote its result as follows:*

$$\overline{R}X = \{r_i | \exists i(r_i \in S) \wedge |r_i(a)| \geq 1 \wedge \Gamma_j(r_i(a_i)) = C, 1 \leq i \leq |U|, 1 \leq j \leq K\}$$

*where X is the result of rough relation operations, K is the maximal number of sub-interpretation for an attribute value, and* $r_i(a)$ *is an attribute value based on attribute* $a, \Gamma_j(r_i(a_i))$ *is a sub-interpretation of* $r_i(a)$*, and C is the attribute value that user want to query.*

### 4.2 The Rough Representation of Attribute Value of RRDM

According to the point of view of T.Y.Lin [9], each attribute value equal to an equivalence granule and each attribute equal to an equivalence relation, this theory can be extend to RRDM for studying the representation of attribute value.

**Proposition 4.** *Given a rough relational database R, let* $t[A_j]$ *and* $\{x_1, x_2, \ldots, x_n\}$ *be* $R'$*s arbitrary attribute value and a set of objects respectively. If* $t[A_j] \in x_i \wedge |t[A_j]| = 1$*, then* $x_i \subseteq \underline{R}t[A_j]$*; if* $t[A_j] \in x_i \wedge |t[A_j]| \geq 1$*, then* $x_i \subseteq \overline{R}t[A_j]$*, where* $1 \leq i \leq n, 1 \leq j \leq m, ||$ *is the cardinal number of sub-interpretation.*

As can be seen from Table 1, according to proposition 4, the following results for attribute "COUNTRY" are obtained:

$$\underline{R}_{US} = \{x_1, x_2, x_3, x_4, x_5\}, \overline{R}_{US} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$$

**Proposition 5.** *The certain data querying results to attribute value* $t[A_i]$ *is its lower approximation* $\underline{R}t[A_i]$*, the possible data querying results to attribute value* $t[A_i]$ *is its upper approximation* $\overline{R}t[A_i]$*.*

**Table 1.** Subregions

| *OBJ* | *ID* | *COUNTRY* | *FEATURE* |
|---|---|---|---|
| $x_1$ | U123 | US | MARSH, LAKE |
| $x_2$ | U124 | US | MARSH |
| $x_3$ | U125 | USA | MARSH, PASTURE, RIVER |
| $x_4$ | U126 | US | FOREST, RIVER |
| $x_5$ | U147 | US | SAND, ROAD, URBAN |
| $x_6$ | U157 | US, MEXICO | SAND, ROAD |
| $x_7$ | M007 | MEXICO | SAND, ROAD |
| $x_8$ | M008 | MEXICO | BEACH |
| $x_9$ | M009 | MEXICO | SAND |
| $x_{10}$ | CO39 | BELIZE | JUNGLE |
| $x_{11}$ | CO40 | BELIZE, INT | JUNGLE, COAST, SEA |

## 5 Rough Functional Dependency and Its Inference Rules

### 5.1 Rough Functional Dependency and Redundant Factor

In [2], Beaubouef Theresa gave the definitions of tuples redundant, tuples roughly-redundant and rough functional dependency, in this paper we introduce the redundant factor and analyze its some properties.

**Definition 4.** ([2]) *A rough functional dependency* $X \longrightarrow_R Y$*, for a relation schema R exists if for all instances* $T(R)$*,*

*(1) For any two tuples* $t, t' \in \underline{R}T$*,* $redundant(t(X), t'(X)) \rightarrow redundant(t(Y), t'(Y))$*, and*

*(2) For any two tuples* $s, s' \in \overline{R}T, roughly - redundant(s(X), s'(X)) \rightarrow roughly - redundant(s(Y), s'(Y))$*.*

**Definition 5.** ([11])*(redundant factor) In definition 4, we call the similarity measure between* $t(X)$ *and* $t'(X)$ *antecedent lower redundant factor, denote* $\alpha$ *, we call the similarity measure between* $t(Y)$ *and* $t'(Y)$ *as consequent lower redundant factor, denote* $\beta$ *, then the definition 4 (1) can denote* $X_\alpha \to Y_\beta$*.Similarly, we call the similarity measure between s(X) and s (X) antecedent upper redundant factor, denoted , the similarity measure between* $s(Y)$ *and* $s'(Y)$ *call consequent upper redundant factor, denote* $\beta'$ *, so the definition 4 (2) can denote* $X_{\alpha'} \longrightarrow Y_{\beta'}$ *where*

$$\alpha = \frac{card(t(X) \cap t'(X))}{card(t(X) \cup t'(X))},\ \beta = \frac{card(t(Y) \cap t'(Y))}{card(t(Y) \cup t'(Y))}$$

*where* $\alpha, \beta \in [0, 1]$*,* $card()$ *denote the cardinal number of the set, the definitions of* $\alpha', \beta'$ *are similar to* $\alpha, \beta$*.*

In this paper, we use $X \longrightarrow_R Y$ stand for rough functional dependency, or denote $X_\alpha \to Y_\beta$ and $X_{\alpha'} \longrightarrow Y_{\beta'}$.

### 5.2 Rough Functional Dependency and Armstrong Axiom

**Proposition 6.** *For a RRDM, if its arbitrary tuples* $t = (d_{x1}, d_{x2}, \ldots, d_{xm})$ *and* $t' = (d_{y1}, d_{y2}, \ldots, d_{ym})$ *are tuple-redundant, then they must be roughly-redundant. If* $t, t'$ *is roughly-redundant, it's unnecessary to be tuple-redundant.*

**Proposition 7.** *In rough functional dependency* $X_\alpha \to Y_\beta$ *and* $X_{\alpha'} \to Y_{\beta'}$*, if* $\alpha = 1$*, then* $\beta = 1$*,when* $\alpha' = 1, \beta'$ *is unnecessarily equal to 1.*

**Proposition 8.** *Dissimilar antecedent attributes values don't influence the dependency.*

**Theorem 5.** *Classical functional dependency satisfies rough functional dependency* $X_\alpha \to Y_\beta$*,* $X_{\alpha'} \longrightarrow_R Y_{\beta'}$*.*

Let $U$ be the set of attributes, $F$ be a group of rough functional dependency, $< U, F >$ be the rough relational schema, we obtain following inference rules on rough functional dependency (about Armstrong axiom).

**RFD1:** Reflexive rule: if $Y \subseteq X \subseteq U$ holds, then $X \longrightarrow_R Y$ holds in $F$.

**RFD2:** Transitivity rule: if $X \longrightarrow_R Y, Y \longrightarrow_R Z$ holds in $F$, then $X \longrightarrow_R Z$ holds in $F$.

**RFD3:** Augmentation rule: if $X \longrightarrow_R Y$ holds in $F$, and $Z \subseteq U$, then $XZ \longrightarrow_R YZ$ holds in $F$.

**Theorem 6.** *The inference rules RFD1, RFD2, RFD3 are sound.*

*Proof.* RFD1: Let $t, t'$ be arbitrary tuples of RRDM, $t, t' \in \underline{R}T$, now $Y \subseteq X \subseteq U$, so $X \cap Y = Y$, if exists $redundant(t(X), t'(X))$, then according to the definition of tuple-redundant, $t(X) = t'(X)$ holds. Because attribute $Y$ is subset of $X, Y$ satisfies $t(Y) = t'(Y)$, and $redundant(t(Y), t'(Y))$ holds, so

$redundant(t(X), t'(X)) \rightarrow redundant(t(Y), t'(Y))$ holds; let $t, t' \in \overline{R}T$, if exists $roughly - redundant(t(X), t'(X))$, then according to the definition of roughly-redundant, $t(X) \cap t'(X) \neq \emptyset$ holds, let $k \in X$, then $t(k) \cap t'(k) \neq \emptyset$ , because attribute Y is a subset of X, so $t(Y) \cap t'(Y) \neq \emptyset$ holds, and then roughly - redundant$(t(Y), t'(Y))$ holds, namely $roughly - redundant(t(X), t'(X)) \rightarrow roughly - redundant(t(Y), t'(Y))$ holds, as a result, $X \longrightarrow_R Y$ is included in $F$, RFD1 holds.

RFD2: If rough functional dependency $X \longrightarrow_R Y, Y \longrightarrow_R Z$ holds, for arbitrary tuples $t, t' \in \overline{R}T$, $redundant(t(X), t'(X)) \rightarrow redundant(t(Y), t'(Y))$, $redundant(t(Y), t'(Y)) \rightarrow redundant(t(Z), t'(Z))$ holds, namely $t(X) = t'(X)$ holds, implies $t(Z) = t'(Z)$ holds, i.e, $redundant(t(X), t'(X)) \rightarrow redundant$ $(t(Z), t'(Z))$ holds—(1); similarly, for arbitrary tuples $t, t' \in \overline{R}T$, $redundant$ $(t(X), t'(X)) \rightarrow redundant(t(Z), t'(Z))$ holds—(2); from (1),(2) we obtain $X \longrightarrow_R Z$ is included in $F$, RFD2 holds.

RFD3: Let $X, Y, Z$ are the attributes sets on the RRDM, and $t, t'$ are arbitrary tuples with the RRDM, $t, t' \in \underline{R}T$, if $X \longrightarrow_R Y$ holds, $redundant(t(X), t'(X)) \rightarrow redundant(t(Y), t'(Y))$ holds, namely $t(X) = t'(X) \rightarrow t(Y) = t'(Y)$ holds; if $t(XZ) = t'(XZ)$ holds, then $t(X) = t'(X), t(Z) = t'(Z)$ holds, so to $t, t'$ , we have $t(YZ) = t'(YZ)$ holds, $redundant(t(XZ), t'(XZ)) \rightarrow redundant(t(YZ)$, $t'(YZ))$ is included by $F$; similarly, for $t, t' \in \overline{R}T$, $redundant(t(XZ), t'(XZ)) \rightarrow redundant(t(YZ), t'(YZ))$, $roughly - redundant(t(XZ), t(XZ)) \rightarrow roughly - redundant(t(YZ), t(YZ))$ holds, so $XZ \longrightarrow_R YZ$ is included in $F$, RFD3 holds.

For RFD, we obtain following additional inference rules:
**RFD4:** Union rule: $X \longrightarrow_R Y, X \longrightarrow_R Z \models X \longrightarrow_R YZ$.
**RFD5:** Decomposition rule: $X \longrightarrow_R Y, Z \subseteq Y \models X \longrightarrow_R Z$.
**RFD4:** Pseudo transitivity rule: $X \longrightarrow_R Y, WY \longrightarrow_R Z \models XW \longrightarrow_R Z$.

**Theorem 7.** *The inference rules RFD4, RFD5, RFD6 are sound.*

The proof is similar to rules RFD1-RFD3.

## Acknowledgements

This work was Supported by Postdoctoral Science Foundation (No.2005038603) Natural Science Foundation of Shanxi Province (No.2006011038) and the National Natural Science Foundation of China (No. 60573074).

## References

1. Beaubouef Theresa, Frederick, E. Petry, Bill, P. Buckles: Extension of the relational database and its algebra with rough set techniques. Computational Intelligence, Vol. 11, 2 (1995) 233-245
2. Beaubouef Theresa:Uncertainty processing in a relational database model via a rough set representation. University Microfilms International.A Bell & Howell Information Company. Doctor dissertation, (1994) 67-76

3. Beaubouef Theresa, Petry, F. E., Arora, G.: Information-theoretic measures of uncertainty measures of rough sets of uncertainty for rough sets and rough relational databases. Information Sciences, 109 (1-4), (1998) 185-195
4. Beaubouef Theresa, Petry, F.: Rough Querying of Crisp Data in Relational Databases. Proc. Third Int. Workshop on Rough Sets and Soft Computing (RSSC'94), San Jose, November (1994) 368-375
5. Beaubouef Theresa, Petry, F.: Rough Functional Dependencies. 2004 Multiconferences: Int.Conf. on Information and Knowledge Engineering (IKE'04). Las Vegas, June 21-24, (2004) 175-179
6. Beaubouef Theresa, Frederick, E. Petry, Roy Ladner: Normalization in a Rough Relational Database. In: Elzak et al. Eds.: RSFDGrC 2005, LNAI 3641, ? Springer-Verlag Berlin Heidelberg , (2005) 275 - 282
7. Hiroshi Sakai, Akimichi Okuma: Basic Algorithms and Tools for Rough Nondeterministic Information Analysis. In: J.F. Peters et al. (Eds.): Transactions on Rough Sets I. LNCS 3100, Springer-Verlag Berlin Heidelberg, (2004) 209-231
8. Qiusheng An, Jiucheng Xu, Junyi Shen, Guoyin Wang: Rough Relational Database Model and its Relational Operations. Computer Science (China), Vol. 29, 7, (2002) 72-74
9. Lin, T. Y., Eric Louie: Data Mining Using Granular Computing: Fast Algorithms for Finding Association Rules. In: Data Mining, Rough Sets and Granular Computing, Lin, T. Y., Yao, Y. Y. and L. A. Zadeh, Eds. Physics, Heidelberg (2003) 22-42
10. Qiusheng An, Guoyin Wang, Junyi Shen, Jiucheng Xu: Querying data from RRDM based on Rough Sets Theory. The 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'2003). LNAI 2639, Chongqing, China, (2003) 342-345
11. Qiusheng An, Junyi Shen, Guoyin Wang, Jiucheng Xu: Rough Functional Dependency and Its Inference Rules. Mini-Micro Systems, Vol. 25, 4, (2004) 638-641

# Interval Rough Mereology for Approximating Hierarchical Knowledge

Pavel Klinov and Lawrence J. Mazlack

Applied Computational Intelligence Laboratory
University of Cincinnati
Cincinnati, OH 45221-0030

**Abstract.** The paper proposes an approach based on Rough Mereology to approximating hierarchical relationships between imprecise concepts in knowledge representation systems. The approach employs Interval Analysis to capture the imprecision caused by the granularity of knowledge. Interval rough inclusion functions are defined. It is demonstrated that they can be effectively used to compute the IS-A relationships by measuring the inclusion of one approximated concept into another. It is shown that the functions are superior to the previously suggested in literature.

## 1 Introduction

Representing and reasoning about knowledge is critical in Artificial Intelligence. There is a distinction between factual and ontological knowledge and the methods for their representation. Factual knowledge represents set of facts about individual objects that are known or believed whereas ontological (a.k.a. background) knowledge represents concepts and relationships that are assumed to exist in a domain. Ontological knowledge is often represented as a hierarchy of concepts because splitting things of the real world into categories and subcategories is a natural way of human thinking. One example of conceptual hierarchies in AI is *ontologies* that are widely used in such areas as Natural Language Processing, Semantic Web, etc.

Representation of both types of knowledge (factual and ontological) becomes difficult when the knowledge is imprecise. This paper investigates the case of granulated knowledge where not all objects are fully distinguishable. In this case, knowledge cannot be represented precisely but can be approximated with respect to granularity of the domain. Approximation of factual knowledge has been extensively researched and often employs Rough Set Theory [6] for dealing with indiscernibility of objects. Similar approaches have been applied on ontological knowledge, in particular, on hierarchical conceptual structures - ontologies [3]. The shortcoming of that approach is the insufficient attention is paid to approximating hierarchical relationships between concepts. To address it, Rough Mereology [9] complemented by Interval Analysis [5] will be used.

The principal contribution of this paper is to provide rough mereological approach to approximating hierarchical ("IS-A") relationships between imprecise

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 557–564, 2007.

concepts. In particular, definitions of rough inclusion functions suitable for computing degrees of membership and subsumption will be provided. It will also be demonstrated how the interval based definitions are superior to previously suggested in the literature [1]. The remainder of the paper is organized as follows: Section 2 and 3 give brief introductions into Rough Mereology and Interval Analysis. Section 4 presents the approach to approximating hierarchical knowledge and Section 5 concludes the paper.

## 2 Rough Mereology

Rough Mereology is a recently proposed paradigm for approximate reasoning based on Rough Set Theory. It was developed for "specifying and analyzing of a complex structure where the inference engine must take into account the uncertain character of knowledge about objects and complex structures" [9].

The primitive notion of Rough Mereology is a function called *rough inclusion* that defines the extent to which one possibly complex object is a part of another. Rough inclusion functions give a formal semantics to how complex structures are constructed from smaller parts. Rough inclusion functions can be used in knowledge representation [1]. For example, the following problems that often occur in ontology engineering may be approached by means of rough inclusion functions:

- Whether the object, say, x belongs to the concept A?
- Whether the concept A "IS-A" concept B (whether A *is subsumed by* B)?

The following rough inclusion functions will be defined:

- *Degree of membership* returns the degree to which the object x belongs to the concept A.
- *Degree of subsumption* returns the degree to which concept A is a part of concept B.

To apply Rough Mereology methods, membership and subsumption functions should obey certain properties that are mandatory for all rough inclusions [4]:

**R0:** $\mu : 2^{|U|} \times 2^{|U|} \rightarrow \Omega; \forall X, Y \in U; \epsilon \leq \mu(X, Y) \leq \gamma$.
Partial order relation must be defined on the co-domain. Co-domain is bounded where $\epsilon, \gamma \in \Omega$ are the least and greatest elements respectively.
**R1:** $\mu(X, Y) = \gamma, \forall X, Y \subseteq U$
**R2:** $\mu(X, Y) = \gamma \Rightarrow \mu(Z, X) \leq \mu(Z, Y), \forall X, Y, Z \subseteq U$ (monotonicity)
**R3:** $\mu(X, Y) = \mu(Y, X) \Rightarrow \mu(Z, X) = \mu(Z, Y), \forall X, Y, Z \subseteq U$
**R4:** There exists $\mu - null$ object N such that: $\mu(N, X) = \gamma, \forall X \subseteq U$

## 3 Interval Analysis and Computations

Using single real numbers between 0 and 1 is not always satisfactory for representing imprecision and other certainty domains have been proposed. One such

domain consists of pairs of real numbers. For example, Dempster-Shafer Theory uses pairs of belief and plausibility, Theory of Possibility operates with possibility and necessity, interval and intuitionistic fuzzy sets deal with imprecise membership, etc. The reason of extending certainty domain is that just one real number often fails to express the imprecision that exists in assessing the uncertainty itself (a.k.a. "meta-uncertainty"). For example, in the case of rough sets, it is not possible to compute rough membership values for union and intersections of sets [7]. Hence single values are inadequate and it is natural to use intervals to represent the range that the actual uncertainty must fall into.

Important relations and arithmetic operations on intervals are defined in the area of Interval Analysis and Computations. It was introduced by Moore as a way of handling inevitable imprecision in scientific computations [5]. In particular, the following will be used below for dealing with imprecision intervals in knowledge representation:

- Partial order relation $\leq$. It is needed to compare certainty values and define properties of interval functions (such as monotonicity). For the purpose of handling imprecision the following is appropriate [2]:
  Let $X = [\underline{x}, \overline{x}], Y = [\underline{y}, \overline{y}]$ be real-valued intervals
  $X \leq Y \Leftrightarrow (\forall x \in X, \exists y \in Y : x \leq y)$ and $(\forall y \in Y, \exists x \in X : x \leq y)$
  This definition allows simple and efficient computation:
  $X \leq Y \Leftrightarrow \underline{x} \leq \underline{y}$ and $\overline{x} \leq \overline{y}$
- Interval $X = [\underline{x}, \overline{x}]$ is said to be *degenerate* iff: $\underline{x} = \overline{x}$
- Arithmetic operations. Let $\bullet \in \{+, -, \times\}$ be the set of allowed arithmetic operations. Then:

$$[\underline{x}, \overline{x}] \bullet [\underline{y}, \overline{y}] = [min(\underline{x} \bullet \underline{y}, \underline{x} \bullet \overline{y}, \overline{x} \bullet \underline{y}, \overline{x} \bullet \overline{y}), max(\underline{x} \bullet \underline{y}, \underline{x} \bullet \overline{y}, \overline{x} \bullet \underline{y}, \overline{x} \bullet \overline{y})] \quad (1)$$

## 4 Interval Rough Mereology

The goal of this section is to show how Rough Mereology can support formal approximations of conceptual hierarchies. For that purpose rough inclusion functions to compute degrees of membership and subsumption for approximated concepts will be defined. It will also be shown that the developed functions return plausible results when previously proposed definitions are inadequate.

### 4.1 Degree of Membership

First, Rough Mereology can be used to approximate the membership of individual objects to imprecise concepts. Rough Set Theory provides this approximation through rough membership function [7]:

$$\mu_A(x) = \frac{[x]_R \cap A}{[x]_R} \quad (2)$$

where R is a crisp equivalence relation, and $[x]_R$ is the equivalence class of $x$ (sometimes called granule). $\mu_A(x)$ is provably a rough inclusion function.

The problem with the rough membership function in hierarchical knowledge representation and reasoning, is the computation of membership degrees to intersections and unions of concepts. It is desirable to have explicit formulas for $\mu_{A\cap B}$ and $\mu_{A\cup B}$ using $\mu_A$ and $\mu_B$. However, Pawlak showed that it is not possible in general to compute *exact* values of rough membership functions for union and intersection knowing only the values of functions for each set individually [7]. To formally capture this type of imprecision, it is possible to view rough membership values as conditional probabilities. Then the following relationships must hold [10]:

**RM0:** $\mu_{A^c}(x) = 1 - \mu_A(x)$.
**RM1:** $\mu_{A\cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_{A\cap B}(x)$
**RM2:** $max(0, \mu_A(x) + \mu_B(x) - 1) \leq \mu_{A\cap B}(x) \leq min(\mu_A(x), \mu_B(x))$
**RM3:** $max(\mu_A(x), \mu_B(x)) \leq \mu_{A\cup B}(x) \leq min(1, \mu_A(x) + \mu_B(x))$

Although exact values for $\mu_{A\cap B}$ and $\mu_{A\cup B}$ and are unknown, they must fall into intervals that can be precisely computed basing on $\mu_A$ and $\mu_B$. This property suggests that instead of using real numbers to represent membership degrees, it is possible to use intervals and treat exact values as a special case. This is the kind of imprecision for which Interval Analysis has been originally proposed, and it is reasonable to use its theoretical machinery.

If a set of all closed intervals that are connected subsets of [0,1], is denoted $\nu$, interval membership function $\mu : 2^{|U|} \times 2^{|U|} \rightarrow \nu$ can be defined as an extension of standard rough membership function [7]:

$$\mu_A(x) = \mu([x]_R, A) = \left[\frac{[x]_R \cap A}{[x]_R}, \frac{[x]_R \cap A}{[x]_R}\right] \tag{3}$$

Then approximate degrees of memberships to union or intersection of concepts can be computed according to RM2 and RM3:

$$\mu_{A\cap B}(x) = [max(0, \mu_A(x) + \mu_B(x) - 1), min(\mu_A(x), \mu_B(x))] \tag{4}$$

$$\mu_{A\cup B}(x) = [max(\mu_A(x), \mu_B(x)), min(1, \mu_A(x) + \mu_B(x))] \tag{5}$$

Then property RM1 can be verified using the interval calculus. This demonstrates that the proposed interval extension of rough membership as a representation of imprecision has the mandatory properties for rough membership functions. It is also straightforward to verify properties R0-R4. This is one of the two basic functions to be used in approximate knowledge representation.

### 4.2 Degree of Subsumption

Second common problem in hierarchical knowledge representation is determining whether or not a concept is more specific/general than another concept. The problem is often reduced to measuring the inclusion of one possibly imprecise set into another. In the case of crisp or fuzzy sets, the computation is often simple - a set is either a subset of another or it is not. This paper is concerned

with the *degree* to which a set is a subset of another - i.e. an inclusion function is required.

The most trivial function is the conventional one that is based on cardinalities of sets. It trivially obeys the rough inclusion properties R0-R4:

$$\mu(X,Y) = \frac{|X \cap Y|}{|Y|} \tag{6}$$

However, in the case of approximated concepts, the inclusion function should be adapted to work with more general rough and rough fuzzy sets. Two such rough inclusion functions have been recently proposed for rough sets [1]:

$$\rho(X,Y) = \frac{|B(X) \cap B(Y)|}{|B(X) \cup B(Y)|} \tag{7}$$

$$\vartheta(X,Y) = \frac{1}{2} \times \left( \frac{|\underline{X} \cap \underline{Y}|}{|\underline{X} \cup \underline{Y}|} + \frac{|\overline{X} \cap \overline{Y}|}{|\overline{X} \cup \overline{Y}|} \right) \tag{8}$$

where B(X) and B(Y) are boundary regions of X and Y respectively.

Unfortunately, these functions have some undesirable properties. Consider the case shown in the Fig. 1 when rough set X is strictly inside rough set Y (cells represent equivalence classes of the universe).

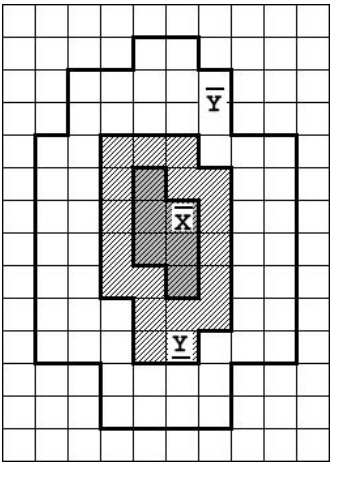



**Fig. 1.** Set X is strictly inside rough set Y

Here, upper approximation of X is a subset of lower approximation of Y. In this case function 7 returns counterintuitive zero because boundary regions do not overlap. Function 8 gives plausible result, namely 1, but suffers from less obvious but still important shortcoming. Consider Fig. 2a and 2b:

In both situations function 8 outputs $\frac{1}{2}$ despite that the situations are quite different. The difference is in the imprecision represented by boundary regions of

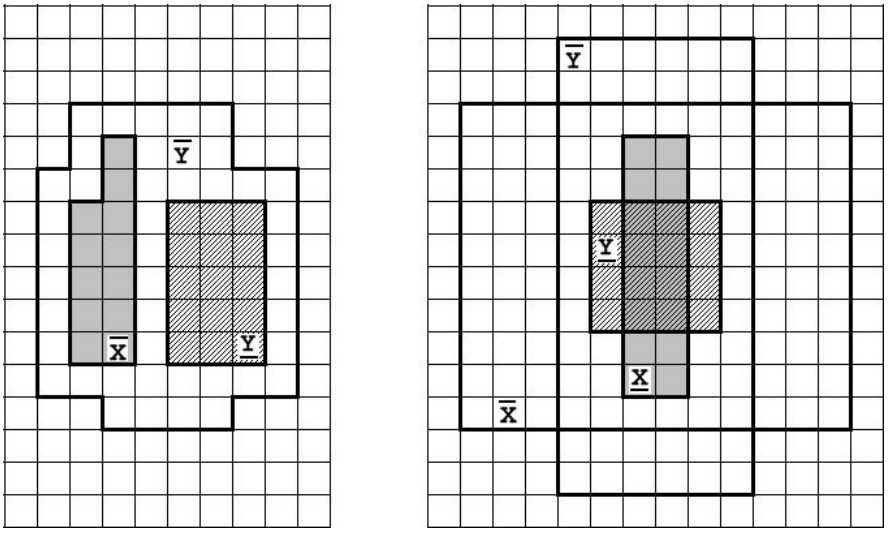



**Fig. 2.** a) Set X is strictly inside rough set Y. b) Sets X and Y have overlapping lower and upper approximations.

the sets. Recall that boundary region is comprised of those objects that cannot be certainly classified either as belonging to the set or to its complement. It is intuitively obvious that in the case shown on Fig. 2a set X can be both - strictly outside and strictly inside Y, depending on the exact shape of the boundary region of Y. Indeed, if Y collapses to its lower approximation, X is always outside of it. Conversely, if Y expands to its upper approximation, X is always inside. Therefore degree of inclusion X into Y may fluctuate between 0 and 1. The situation on Fig. 2b is different. Here, X can never be strictly outside of Y because their lower approximations overlap. It is certainly known that there are objects that are both in X and Y and the share of those objects cannot drop below certain percentage. However, degree of inclusion of X into Y may or may not rise up to one depending on whether boundary region of X is or is not fully contained in Y.

The problem of functions 7 and 8 is their inability to express that the degree of imprecision *certainly* falls in an interval and may *possibly* have any value in it. An intelligent agent computing those functions cannot distinguish between situations on Fig. 2a and 2b (and many others similar). This is a significant loss of information that may hurt the reasoning quality in an imprecise environment.

Analogously to the membership function, imprecision cannot be characterized precisely, but can be bounded by a pair of real values: $\eta : 2^{|U|} \times 2^{|U|} \rightarrow \nu$. Computing the inclusion of one rough set into another involves computing inclusion degree for crisp sets. This is because any rough set is represented as a pair of its approximations that are crisp sets. Therefore rough inclusion function will be defined with respect to some crisp inclusion function - $\eta_c$. Then the function $\eta$ must obey the following properties:

$$\forall X, Y \subseteq U, \eta(X,Y) = [\alpha, \beta] \Rightarrow \epsilon \leq \alpha \leq min_{\underline{X} \subseteq A \subseteq \overline{X}; \underline{Y} \subseteq B \subseteq \overline{Y}}(\eta_c(A,B)) \quad (9)$$

$$\forall X, Y \subseteq U, \eta(X,Y) = [\alpha, \beta] \Rightarrow max_{\underline{X} \subseteq A \subseteq \overline{X}; \underline{Y} \subseteq B \subseteq \overline{Y}}(\eta_c(A,B)) \leq \beta \leq \gamma \quad (10)$$

To understand the motivation behind properties 9 and 10, consider a rough set X as a collection of all sets enclosed between its lower and upper approximations. If any imprecision were eliminated, it would be possible to say which set in the collection actually represents X. But as long as imprecision exists, it is assumed that X can be any set in the collection. Then it is possible to give the following semantics to $[\alpha, \beta]$ when formulating properties 9 and 10:

- For any X and Y, degree of inclusion of X into Y cannot drop below $\alpha$
- For any X and Y, degree of inclusion of X into Y cannot exceed $\beta$

Now being equipped with properties 9 and 10, it is possible to propose a subsumption function $\eta$ to compute the degree of inclusion of one rough set into another. Although $\eta$ can be computed directly from 9 and 10, it incurs a significant overhead. It requires $O(2^{max(|B(X)|,|B(Y)|)})$ calls of $\eta_c$ i.e. it is exponential in cardinality of boundary regions of X and Y.

However, it is possible to compute only the following:

$$[\alpha, \beta] = [min(V_1, V_2, V_3, V_4), max(V_1, V_2, V_3, V_4)] \quad (11)$$

where: $V_1 = \eta_c(\underline{X}, \underline{Y}), V_2 = \eta_c(\underline{X}, \overline{Y}), V_3 = \eta_c(\overline{X}, \underline{Y}), V_4 = \eta_c(\overline{X}, \overline{Y})$. It can be verified that the function 11 is free of the previously described shortcomings.

Note how similar this definition is to the formula 1 that defines arithmetic operations on intervals. This is not an accident. Rough sets can be viewed as interval structures. Analogously to how real numbers fall into imprecision intervals, crisp sets fall into intervals induced by approximation spaces. The function 11 can be verified to obey properties R0-R4:

**R0:** Holds, $\eta$ is a well-defined function and $\nu$ is bounded by [0,0] and [1,1].

**R1:** For the traditional definition of rough subset [8] - $X \subseteq Y \Leftrightarrow \underline{X} \subseteq \underline{Y}$ and $\overline{X} \subseteq \overline{Y}$ the property does not hold. The reason is that this definition of rough subset is inconsistent with the properties 9 and 10. Instead, the *certain* inclusion relation defined as: $X \prec Y \Leftrightarrow \overline{X} \subseteq \underline{Y}$ and interpreted as: "all objects that are *possibly* in X are *certainly* in Y", can be used. Then, the property holds.

**R2:** Assume $\eta(X,Y) = \gamma$. For some $Z \subseteq U$ compute $\eta(Z,X) = [min(V_i'), max(V_i')]$ and $\eta(Z,Y) = [min(V_i''), max(V_i'')]$, $i = 1,2,3,4$. As long as $\eta_c$ obeys R2, $V_i' \leq V_i''$, so $\eta(Z,X) \leq \eta(Z,Y)$

**R3:** $\eta(X,Y) = \eta(Y,X)$ implies that $\underline{X} = \overline{X} = \underline{Y} = \overline{Y}$. Therefore, $\forall Z \subseteq U, \eta(Z,X) = \eta(Z,Y)$

**R4:** $N = \langle \emptyset, \emptyset \rangle$ may serve as a null-object.

Therefore the subsumption function 11 is a rough inclusion and may be used together with membership function 3 for approximating hierarchical knowledge.

## 5 Conclusion

The paper proposes a rough mereological approach to approximating concepts and hierarchical relationships for knowledge representation. The principal contribution is the interval based rough inclusion functions that can be used for approximate representation and reasoning with imprecise conceptual hierarchies. The functions operate with intervals and can formally capture the imprecision caused by computing rough membership values for union and intersection of rough sets.

The approach can be effectively used in knowledge representation systems that employ Rough Set Theory to handle the granularity of knowledge. Hierarchical relationships between concepts can be approximated using the subsumption function that computes the degree of inclusion of one rough concept into another. This is useful, for example, in ontology engineering where concept hierarchy occupies the central place. Formal ontologies for imprecise domains can be approximated using the proposed mereological approach and be used by intelligent agents for approximate reasoning in the Semantic Web.

The approach can be generalized to use rough and fuzzy methods to handle different types of imprecision in concepts and relationships. In this case, rough inclusion functions will be used to compute degrees of membership and subsumption for rough fuzzy sets [10]. Intervals will represent boundaries of fuzzy membership values computed with respect to approximations spaces.

## References

1. Cao, C., Sui, Y., Zhang, Z.: Rough Mereology in Knowledge Representation. In: Proc. of Int. Conf. On Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (2003), 329-333
2. Chiriaev, D., Walster, G.W.: Interval Arithmetic Specifications. Man. J3/97-199 for ANSI X3J3 (1997)
3. Doherty, P., Grabowski M., Lukaszewicz, W., Szalas, A.: Towards a Framework for Approximate Ontologies. Fundamenta Informaticae, 57(2-4) (2003), 147-165
4. Inuiguchi, M., Polkowski, L.: Meticulous Rough Inclusion and Its Relations to Fuzzy Inclusion. In: Proc of FUZZ-IEEE (2001), 1535-1538
5. Moore, R.E.: Interval Analysis, Prentice-Hall, Englewood Cliffs, NJ (1966)
6. Pawlak, Z.: Rough Sets, Int. J. of Comp. and Inf. Sci., 11 (1982), 341-356
7. Pawlak, Z., Skowron, A.: Rough Membership Functions. In: Yaeger, R.R., Fedrizzi, M., Kacprzyk, J. (eds.): Advances in The Dempster Shafer Theory of Evidence. John Wiley & Sons (1994), 251-271
8. Pawlak, Z.: Rough Sets, Rough Relations and Rough Functions, Fundamenta Informaticae. 27(2-3) (1996), 103-108
9. Polkowski, L., Skowron, A.: Rough Mereology: A New Paradigm for Approximate Reasoning. Int. J. Appr. Reason. 15 (1996), 333-365
10. Yao, Y.Y.: Semantics of Fuzzy Sets in Rough Set Theory. In: Peters, J.P, Skowron, A., Dubois, D., Grzymala-Busse J.W., Inuiguchi, M, Polkowski L. (eds.): Transactions on Rough Sets II. Lecture Notes in Computer Science, Vol. 3135. Springer (2004)

# Description Logic Framework for Access Control and Security in Object-Oriented Systems

Jung Hwa Chae and Nematollaah Shiri

Concordia University
Dept. of Computer Science & Software Engineering
Montreal, Quebec, Canada
{chae,shiri}@cse.concordia.ca

**Abstract.** Integrating the RBAC model in object-oriented systems is a natural way to describe authorization policies. We extend the RBAC model for access control in object-oriented systems in the context of the Access Control List. In this paper, we discuss access control issues categorizing three cases: subject to object, inter-objects, and intra object. It may be desirable in some applications to have a fine-grained access control at the level of the individual attributes or the methods of an object. We also demonstrate how access control decisions are made using $\mathcal{ALCQ}$ language, a family member of description logics.

**Keywords:** Role-based access control, object-oriented system, access control List, description logic.

## 1 Introduction

All access control problems seek to answer the fundamental question: "Is subject $S$ allowed to access type $A$ on object $O$?". A well known access control principle known as role-based access control (RBAC), is utilized to organize subjects into access control groups, based on their roles in an organization [1]. This simplifies the task to grant and revoke authorizations to entire groups of subjects at a time.

In this paper, we integrate the RBAC model within an object-oriented paradigm (ORBAC) to describe authorization policies. We argue that authorization mechanisms are needed to restrict access to objects, within a domain, based on defined access control policies. Since RBAC is not originally designed for object-oriented systems, we extend it by adding mechanisms for access control among objects.

Roles and objects automatically obtain permission from the class in which they are instantiated. They can also inherit permission from superclasses. Although inheritance can reduce the complexity of the permission assignment, under special cases it is difficult to introduce various permission restrictions. However, on occasion it is desirable to restrict some users from the given access. In addition, we may need to assign different permissions for each attribute or method within an object, rather than applying the assigned permission to all elements within that object. Therefore, in object-oriented systems, it may be

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 565–573, 2007.

desirable, in some applications, to have fine-grained access control at the level of the individual methods or attributes of an object.

Every object can be further broken down into smaller access units, i.e. the object's attributes or methods. Attributes and methods come equipped with their own *Access Control List* (ACL). To maintain the secure system, ACLs [2,3] are created to limit to accesses between the methods and attributes of objects. In this paper, access control policies are thereby represented by invocation sets of methods, ACLs of object attributes, ACLs of method variables, and ACLs of values returned by methods.

Several advancements in authorization specification and enforcement have been carried out with reference to specific applications and data models. Authorization models proposed for object-oriented systems [4,5,6] exploit the encapsulation concept, meaning the fact that access to objects is always carried out through methods. Each derived function, i.e. method can be specified as supporting static or dynamic authorizations [4]. A similar feature is also proposed in [6], where each method is associated with a principal, and accesses requested during a method execution are checked against the authorization of the method's principal. McCollum et al. [7] propose a dissemination control system that maintains access control over one's data by attaching an access control list that imposes access restrictions to the data object. The access control list propagates, through subject and object labels, to all objects into which its content may flow. In [8] each object has two protection attributes: the current access and the potential access. The model proposed in [3] controls information flows in object-oriented systems. It utilizes ACLs of objects to compute ACLs of executions, and then obtains a secure information flow condition. In spite of research results aforementioned, there is little work that links both object-oriented and DL methodologies. Research has been published describing roles and permissions, however, few include the role-object relationship.

We formally define the properties and relationships that should hold in the access control specifications using description logics (DLs) [9]. A formal description of access control policies is necessary in order to check if security requirements are satisfied or not. We use the DL language $\mathcal{ALCQ}$ [10] to define and reason about authorizations and privileges. In practice, we present an example of reasoning on an access control via RACER [11].

The paper is organized as follows. In Section 2, we describe how to integrate the RBAC model within an object-oriented paradigm. In Section 3, we follow this by showing how DL can be used to model ORBAC. In Section 4, we illustrate an example how an access control could be used in a university domain and how a reasoner is used to evaluate user's access requests. We conclude with a summary of the contributions of this paper along with some future work.

## 2 ORBAC Model

The access control problem has been contained within the framework of subjects, objects, and authorizations. Within this subject-object paradigm of access

control, an object refers to any entity that holds data. When we discuss access control in object-oriented systems, we must map this general notion of objects to class objects in the object-oriented sense. We distinguish between three different categorizations for structural and behavioral access control issues in terms of where the access takes place.

- Subject to object: This concerns access control issues if a subject requests to access an object. We are concerned with how a subject establishes an initial authorized point of contact with an object;
- Inter-objects: This level of access is concerned with how an object can access another object. With an object-oriented paradigm, access control is concerned with issues such as the visibility and propagation of authorizations from object to object;
- Intra object: This deals with access control within the internal structure and behavior of individual objects. This issue is thus irrelevant to other objects in the system.

The RBAC is a rather straightforward approach for subject to object access control. Using roles for access control generalizes the assignment of permissions. A subject playing a role possesses the permissions of the role. A major advantage of the RBAC is that permissions are bound to roles instead of users.

We extend the RBAC model to illustrate how a subject is granted authorization to the given objects. For object to object access, it is required that the access control mechanisms be flexible enough to support varying granularity of access units. Access policies are based on inherited authorizations derived along the class structure hierarchy. Users automatically obtain permissions from the class they are instantiated and inheritance from superclasses. However, it is occasionally desirable to restrict some users given access. Therefore, it is necessary to add several mechanisms to handle such situations.

The need for access control can also appear within an object. We consider how we control visibilities and interactions of elements within objects. If we wish to control the visibilities of method $m$, then we should restrict client methods that can invoke $m$. Every method $m$ can be associated with a set of methods that are allowed to invoke $m$. This set contains the names of methods and classes to which $m$ is made visible. In this paper, the association of methods is defined via a method access list of the form: $\{O_1.m_1, O_2.m_2\}$ which allows method $O_1.m_1$ to invoke method $O_2.m_2$. Therefore, method access lists, in ORBAC, limit method invocation.

In the case of a class, $m$ is visible to all methods in the class. Further investigation is necessary for the inheritance hierarchy. If a method $m_0$ is visible to class $C_1$, then $C_1$ is part of the invoker set of $m_0$. A locally defined method $m_1$ in $C_1$ can thus invoke $m_0$. Now we will consider the class $C_2$ which is a subclass of $C_1$. Class $C_2$ inherits $m_1$ in the class hierarchy and has a locally defined method $m_2$. Through inheritance, class $C_2$ is automatically placed in the invoker set for $m_0$. Under some circumstances, we may not want $C_2$ to be placed in the invoker set, therefore, we need to override the inheritance or add some restrictions to deny $C_2$ from those privileges.

It must be recognized that in object-oriented systems, access control and integrity mechanisms are closely linked. This is because methods modify the states of objects, i.e. attributes or variables, and we often enforce access control on method invocations. Integrity is concerned with the improper modification of data. It may be desirable in some applications to have fine-grained access control at the level of the individual attributes or methods of an object. In other words, attributes and methods should be protected independently. It may be desirable to allow only certain methods in the object to access a local attribute. An obvious way to accomplish this would be for every attribute to maintain a list of methods that are allowed access to it.

Within an object, we may want to restrict the visibility of local methods to each other. Again an obvious way to accomplish this would be to provide a list for every method. We use the ACL of the access control mechanism in order to establish meaningful interconnections and visibilities across objects. In the ACLs, each object is associated with a list indicating the actions that each subject can exercise on the object. Attributes and methods come equipped with their own ACL. There are ACLs for object attributes, method variables, and for method return values. Each one is defined as following:

$$\text{attACL}_i = (\text{attName}, \text{className}, \text{RACL}_{\text{attName}}, \text{WACL}_{\text{attName}})$$
$$\text{mdVarACL}_i = (\text{varName}, \text{mdName}, \text{className}, \text{RACL}_{\text{varName}}, \text{WACL}_{\text{varName}})$$
$$\text{mdRetACL}_i = (\text{mdName}, \text{className}, \text{RACL}_{\text{mdName}}, \text{WACL}_{\text{mdName}})$$

An ACL in ORBAC is composed of a Read ACL (RACL) and a Write ACL (WACL). An attribute's ACL is composed of the attribute's name, the class containing the attribute, the methods that are allowed to read the attribute (RACL), and the methods that are allowed to write to that attribute (WACL). Similarly, a method variable's ACL and a method's return value ACL are defined. In the list, mdName indicates methods that contain the variable varName. The definitions of varName, className, $\text{RACL}_{\text{varName}}$, $\text{WACL}_{\text{varName}}$ are respectively similar to those of attName, className, $\text{RACL}_{\text{attName}}$, $\text{WACL}_{\text{attName}}$.

RACL and WACL lists are used to grant more refined accesses between elements. As a result, this limits the role's permission on elements of the class to which it is given access. By default, we assume that elements within a class (as well as inter-objects) do not have rights to access each other. These permissions can only be granted through ACLs.

## 3 Representation of the ORBAC Model in ALCQ

We conceptualize the ORBAC model and use a DL to represent the characteristics of the ORBAC. Given an ORBAC model, we first define a DL knowledge base $\mathcal{K}$. We use atomic concepts and roles to express an access control policy.

An access control policy is composed of classes, a set of methods authorized to interact with other methods, ACLs of class attributes, ACLs of method variables, and ACLs of method return values. It is represented in Table 1.

**Table 1.** Access control policy roles and description

| Roles | Description |
|---|---|
| ∃hasClass.Class | Classes |
| ∃hasClass.∃hasMd.∃canInvoke.∃grant.Md | Method access list |
| ∃hasClass.∃hasAtt.∃hasAttRACL.Md | Read ACL of mehtods that can read an attribute |
| ∃hasClass.∃hasAtt.∃hasAttWACL.Md | Write ACL of mehtods that can write an attribute |
| ∃hasClass.∃hasMd.∃hasMdVar.- ∃hasMdVarRACL.Md | Read ACL of methods that can read arguments of a method |
| ∃hasClass.∃hasMd.∃hasMdVar.- ∃hasMdVarWACL.Md | Write ACL of methods that can write arguments of a method |
| ∃hasClass.∃hasMd.∃hasMdRetRACL.Md | Read ACL of methods that can read return value of a method |
| ∃hasClass.∃hasMd.∃hasMdRetWACL.Md | Write ACL of methods that can write return value of a method |

A class consists of attributes and methods, which can be local or inherited, as defined below:

$$\text{Class} \sqsubseteq \exists \text{hasAtt.Att} \sqcap \exists \text{hasMd.Md}$$

The concept description ∃hasAtt.Att indicates a set of classes that have attributes. Similarly, concept ∃hasMd.Md is interpreted as classes that have methods. A method is composed of an argument list and a return value. This is represented by the following concept:

$$\text{Method} \sqsubseteq \exists \text{hasMdVar.MdVar} \sqcap \exists \text{hasMdRetVal.MdRetVal}$$

Concept ∃hasMdVar.MdVar gives us methods that have the set of arguments associated with it. Concept ∃hasMdRetVal.MdRetVal indicates methods that have return values. A set of arguments are required for other methods that invoke it. The method return value may also be used by other methods that invoke it. In this case, authorization is given to read the return value of that method.

Since classes can inherit behavior from one another, the access control policy access lists include permissions inherited from superclasses. Therefore, concept ∃hasClass.- ∃hasAtt.∃hasAttRACL.Md includes methods defined in a particular class as well as the methods which that class inherits which are authorized to read that attribute. Method access lists are defined as a set of methods which

can invoke another method. The invoker list of methods is also used to confine the invocation accesses between methods. Each method has an associated access list:

$$\exists \text{hasClass}.\exists \text{hasMd}.\exists \text{canInvoke}.\exists \text{grant}.\text{Md}$$

This indicates all the methods that are currently granted to invoke method Md.

Now we will describe each ACL. There may be a situation where a child class wants to override an inherited attribute's ACL. This can be done at the local level and can include the inherited ACL as part of the new set. If class $C_0$ inherits an attribute *att* from class $C_1$, then this attribute's ACL can be redefined as:

$$\text{attACL}_{C_0} = \text{attACL}_{C_1} \cup (\text{att}, C_0, \text{RACLatt}, \text{WACLatt})$$

where $\text{attACL}_{C_1}$ is the inherited attribute's ACL from class $C_1$, and (att, $C_0$, RACLatt, WACLatt) is the locally defined ACL. An RACL is defined as follows:

$$\exists \text{hasClass}.\exists \text{hasAtt}.\exists \text{hasAttRACL}.\text{Md}$$

This is described as the list of methods that can read a specific attribute of a class. Similarly, WACL is defined as:

$$\exists \text{hasClass}.\exists \text{hasAtt}.\exists \text{hasAttWACL}.\text{Md}$$

This defines the set of methods that can write to a specific attribute in a class. In the same way, method variables have their own associated RACL and WACL lists as shown in the following descriptions:

$$\exists \text{hasClass}.\exists \text{hasMd}.\exists \text{hasMdVar}.\exists \text{hasMdVarRACL}.\text{Md}$$
$$\exists \text{hasClass}.\exists \text{hasMd}.\exists \text{hasMdVar}.\exists \text{hasMdVarWACL}.\text{Md}$$

Method return values also have their own RACL and WACL lists below:

$$\exists \text{hasClass}.\exists \text{hasMd}.\exists \text{hasMdRetRACL}.\text{Md}$$
$$\exists \text{hasClass}.\exists \text{hasMd}.\exists \text{hasMdRetWACL}.\text{Md}$$

## 4 A Case Study

In this section, we illustrate a practical example and demonstrate how to accomplish reasoning tasks via RACER. There are four roles: *Administrator*, *StudentAdvisor*, *FacultyAdministrator*, and *ForeignStudentAdvisor*. Suppose that there are four classes: *Person*, *Student*, *ForeignStudent*, and *Teacher*. In our example, we assume the following accesses for the above role and class hierarchies in our example. *Teacher* can read and write *Student*'s grade, but can only read *Student*'s name, ID, course history, and GPA. *Teacher* can only read *ForeignStudent*'s attributes. *Student* and *ForeignStudent* can only read *Teacher*'s course information.

The following role inclusion axioms describe the inheritance relations among roles.

$$\begin{aligned}
&\mathit{Administrator} \sqsubseteq \mathit{StudentAdvisor},\\
&\mathit{StudentAdvisor} \sqsubseteq \mathit{ForeignStudentAdvisor},\\
&\mathit{Administrator} \sqsubseteq \mathit{ForeignStudentAdvisor},\\
&\mathit{Administrator} \sqsubseteq \mathit{FacultyAdministrator}.
\end{aligned}$$

For each of the roles above, we define permission assignment axioms in (1). We define authorization axioms in (2). The inheritance relations among classes are defined by class inclusion axioms in (3). In (4), some of the elements of each class are defined. For inherited attributes, we refer to child attributes by prefixing the attribute and method. The prefix "s" is used for *Student* class, "fs" for *ForeignStudent*, and "t" for *Teacher*. We show examples as we define the RACL (5) and WACL (6) for some attributes. Method variables are defined in (7). Method return variables are listed in (8) and their RACL lists are described in (9).

$$\begin{aligned}
&\mathit{Administrator} \sqsubseteq \exists\text{perform}.\exists\text{execute}.\mathit{Person},\\
&\mathit{StudentAdvisor} \sqsubseteq \exists\text{perform}.\exists\text{execute}.\mathit{Student},\\
&\mathit{ForeignStudentAdvisor} \sqsubseteq \exists\text{perform}.\exists\text{execute}.\mathit{ForeignStudent},\\
&\mathit{FacultyAdministrator} \sqsubseteq \exists\text{perform}.\exists\text{execute}.\mathit{Teacher}.
\end{aligned} \tag{1}$$

$$\begin{aligned}
&\exists\text{assign}.\exists\text{perform}.\exists\text{execute}.\mathit{Person} \sqsubseteq \exists\text{authorize}.\exists\text{execute}.\mathit{Person},\\
&\exists\text{assign}.\exists\text{perform}.\exists\text{execute}.\mathit{Student} \sqsubseteq \exists\text{authorize}.\exists\text{execute}.\mathit{Student},\\
&\exists\text{assign}.\exists\text{perform}.\exists\text{execute}.\mathit{Teacher} \sqsubseteq \exists\text{authorize}.\exists\text{execute}.\mathit{Teacher},\\
&\exists\text{assign}.\exists\text{perform}.\exists\text{execute}.\mathit{ForeignStudent} \sqsubseteq \exists\text{authorize}.\text{-}\\
&\qquad\qquad\qquad\qquad\qquad\qquad \exists\text{execute}.\mathit{ForeignStudent}.
\end{aligned} \tag{2}$$

$$\mathit{ForeignStudent} \sqsubseteq \mathit{Student},\ \mathit{Student} \sqsubseteq \mathit{Person},\ \mathit{Teacher} \sqsubseteq \mathit{Person}. \tag{3}$$

$$\begin{aligned}
&\mathit{Person} \sqsubseteq \exists\text{hasAtt.name},\ \mathit{Person} \sqsubseteq \exists\text{hasAtt.ID},\\
&\mathit{Student} \sqsubseteq \exists\text{hasMd.setGrade},\ \mathit{Student} \sqsubseteq \exists\text{hasMd.getCourseInfo},\\
&\mathit{Teacher} \sqsubseteq \exists\text{hasMd.getCourseInfo},\ \mathit{Teacher} \sqsubseteq \exists\text{hasMd.setCourseInfo}.
\end{aligned} \tag{4}$$

$$\text{sName} \sqsubseteq \exists\text{hasAttRACL.sGetName},\ \text{fsID} \sqsubseteq \exists\text{hasAttRACL.fsGetID}, \tag{5}$$

$$\text{sGrade} \sqsubseteq \exists\text{hasAttWACL.sGetGrade},\ \text{sGPA} \sqsubseteq \exists\text{hasAttWACL.sGetGPA}, \tag{6}$$

$$\text{sSetGrade} \sqsubseteq \exists\text{hasMdVar.sGrade},\ \text{fsSetGrade} \sqsubseteq \exists\text{hasMdVar.fsGrade}, \tag{7}$$

$$\text{sGetName} \sqsubseteq \exists\text{hasMdRet.sName},\ \text{fsGetGrade} \sqsubseteq \exists\text{hasMdRet.fsGrade}, \tag{8}$$

$$\begin{aligned}
&\text{sGrade} \sqsubseteq \exists\text{hasMdRetRACL.tSetStudentGrade},\\
&\text{sGPA} \sqsubseteq \exists\text{hasMdRetRACL.tGetStudentGPA}.
\end{aligned} \tag{9}$$

In the following, we show some representation of our DL facts and rules in RACER. Afterwards, we show some queries to represent how we can check cohesion and consistency of our access control policy.

The ABox is used to verify access permissions when requests are made. For example, if we have class objects *Teacher* T1, *Student* S1, and *ForeignStudent* FS1, we can perform authorization checks when requests are made from one object to another. Suppose T1 wants to obtain a student's GPA. The local method which will get this information is T1.getStudentGPA. This local method needs to invoke the getGPA method in S1. The reasoner will first check if the local method can invoke getGPA. This is true due to the assertion: assign(T1.getStudentGPA, S1.getGPA). The following query asks which objects contain a method that is grants access to the method sGetName:

(`retrieve`(?o)(?o (`some hasMd` (`some canInvoke` (`some grant sGetName`)))) )

The return value is "`(?O T1)`" indicating the *Teacher* object T1. Next, S1. getGPA has a return value sGPA from the statement sGetGPA $\sqsubseteq \exists$hasMdRet. sGPA. After the method is invoked, the reasoner verifies sGPA is checked if it can be read by the invoking method. This is true from the statement sGPA $\sqsubseteq$ $\exists$hasMdRetRACL.tGetStudentGPA, so sGPA is passed back to tGetStudentGPA.

Note that even though class *ForeignStudent* inherits these RACL from the *Student* class, class *Teacher* is not allowed to get the GPA of FS1. This is because *ForeignStudent* creates a new RACL definition for this method, thereby overriding the inherited RACL and method assignments. The following query returns all objects that contain the method "tGetStudentName" and this method belongs to an RACL of some return value:

```
(retrieve (?o) (?o (some hasMd (some canInvoke
                    (some grant (some hasMdRet
                    (some hasMdRetRACL tGetStudentName))))) ))
```

Not all ACLs need to be overridden. In these cases, the lists are inherited from the parent and the invoker method will have the access permission on the child's attributes or methods. ACLs are required to add a finer level of security. Otherwise, the object having access to another object will have access to all of its attributes and methods. For example, *Teacher* is allowed to obtain a *ForeignStudent*'s Name, but is not allowed get the GPA or Grade.

## 5 Conclusion and Future Work

In this paper, we demonstrated how to express the RBAC concept in object-oriented systems using a logical framework called DL. The goal was based on the fine-grained access control at the level of individual attributes or methods of an object using access control list. We defined the access control policy, which consists of classes, methods invocation lists, RACLs and WACLs of class

attributes, method variables, and method return values. In addition, we illustrated how this access control policy may be used in a DL framework to make authorization decisions between the role hierarchy and the object hierarchy. The formalization of ORBAC in a logical approach makes it feasible to reason about a specified policy and verifies its correctness. Future study is required in order to incorporate role delegation and conflict resolution.

**Acknowledgments.** The research of Jung Hwa Chae has been supported by Institute for Information Technology Advancement (IITA) & Ministry of Information and Communication (MIC), Republic of Korea. The research of Nematollaah Shiri has been supported by Natural Science and Engineering Council (NSERC) of Canada.

## References

1. Sandhu, R., Coyne, E., Feinstein, H., Youman, C.: Role-based access control models. IEEE Computer **29** (1996) 38 – 47
2. Chou, S.: Embedding role-based access control model in object-oriented systems to protect privacy. Journal of Systems and Software **71** (2004) 143 – 61
3. Samarati, P., Bertino, E., Ciampichetti, A., Jajodia, S.: Information flow control in object-oriented systems. IEEE Transactions on Knowledge and Data Engineering **9** (1997) 524 – 538
4. Ahad, R., Davis, J., Gower, S., Lyngbaek, P., Marynowski, A., Onuegbe, E.: Supporting access control in an object-oriented database language. Advances in Database Technology - EDBT '92. 3rd International Conference on Extending Database Technology Proceedings (1992) 184 – 200
5. Fernandez, E., Gudes, E., S., H.: A model for evaluation and administration of security in object-oriented databases. IEEE Transactions on Knowledge and Data Engineering **6** (1994) 275 – 292
6. Richardson, J., Schwarz, P., Cabrera, L.: Cacl: Efficient fine-grained protection for objects. SIGPLAN Notices (ACM Special Interest Group on Programming Languages) **27** (1992) 263 – 275
7. McCollum, C., Messing, J., Notargiacomo, L.: Beyond the pale of mac and dac–defining new forms of access control. Proceedings of the Symposium on Security and Privacy (1990) 190 – 200
8. Stoughton, A.: Access flow: a protection model which integrates access control and information flow. Proceedings of the 1981 Symposium on Security and Privacy (1981) 9 – 18
9. Baader, F., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge university Press, Cambridge, United Kingdom (2003)
10. Calvanese, D., De Giacomo, G., Lenzerini, M.: Description logics: foundations for class-based knowledge representation. Proceedings 17th Annual IEEE Symposium on Logic in Computer Science (2002) 359 – 370
11. Haarslev, V., Moller, R.: Racer system description. Automated Reasoning. First International Joint Conference, IJCAR 2001. Proceedings (Lecture Notes in Artificial Intelligence Vol.2083) (2001) 701 – 705

# Rough Neural Networks for Complex Concepts

Dominik Ślęzak[1] and Marcin Szczuka[2]

[1] Infobright Inc.
218 Adelaide St. W, Toronto, ON, M5H 1W8 Canada
[2] Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
slezak@infobright.com, szczuka@mimuw.edu.pl

**Abstract.** Rough neural networks aim at hierarchical construction of compound concepts. Although the structure of such concepts is assumed to be more complicated than numbers in case of standard feedforward neural networks, some mechanisms can be generalized to achieve efficient propagation and learning. One of possible generalizations, called the normalizing neural networks, enables to propagate vectors instead of single signals. Neurons take form of multidimensional functions, which model cross-dependencies among importance of particular vector components. In this way, we are able to represent some types of compound concepts using relatively simple neural network structure. As an illustration, we consider the case study related to the task of magnetic resonance images' segmentation. We put a special emphasis on how the nature of objects and attributes in a given decision system influences the network's architecture. We also compare our model to other rough-neural approaches.

**Keywords:** Rough Neurons, Multi-Dimensional Neurons, Complex Concepts.

## 1 Introduction

Rough neural networks and rough neural computing have been studied in literature in many aspects. First, rough set methodology was considered together with feedforward neural networks within the framework of KDD – knowledge discovery in databases. Algorithms for reduction of attributes and simplification of rules were considered by means of optimizing the inputs to a neural network. Vice versa, neural networks were also applied to learn from data the definitions of attributes, then treated with the rough set algorithms. Finally, rough sets and neural networks were considered within hybrid approaches together with other appropriately defined methods, e.g., fuzzy logic and domain-specific analytical techniques. The reader is referred to the following papers as examples of the above-mentioned strategies [9,14].

The second direction of research in this area relates to generalization of neurons (their transition functions and connections) by means of the theory of rough sets. The initial concept of rough neuron in [7] – a neuron consisting of sub-neurons responsible for the lower and upper approximations – was modified in many ways, e.g., in [2] where lower and upper approximations are replaced by lower approximation and boundary, as a model of signal analysis, based on the concept of partitioning the signal into the predictable and random parts. In [13], the model of a multi-dimensional neuron operating

A. An et al. (Eds.): RSFDGrC 2007, LNAI 4482, pp. 574–582, 2007.

with vectors of signals has been developed. Its relevance to the main stream of the rough set methodology may be explained by papers [8,11], where dealing with vectors of memberships to various decision classes is presented as one of the main aspects of rough sets, confronted to the other methods concentrating on single decision classes in the classification/prediction process. In the next sections, we will see that comparison of rough neurons and multi-dimensional neurons becomes quite intriguing, especially when we further modify rough neurons to let them play with lower approximations and complements of upper approximations (negative regions).

Generalizations of neurons have been followed by generalizations of the entire network structures and corresponding learning mechanisms. New models of neural networks have been studied more and more often as hierarchical structures of complex concepts (granules), which finally resulted in the methodology of *rough-neural computing* (RNC) [9]. We focus on RNC's features analogous to those most commonly attributed to the *classical* neural computing:

- Construction of systems performing complex tasks using simple rough neurons and their straightforward generalizations transforming parameters of concepts
- Hierarchical structure that represents gradual formation of more complex granules (concepts) modeling complex phenomena or structures, or projection onto simpler granules (concepts) modeling aggregation of information, conflict resolution etc.
- Flexibility and robustness originating in highly adjustable structure of possibly generalized rough neurons, their connections, and intermediate transformations enabling to vary the structures of granules (concepts) throughout the network
- Ability to learn from examples a desired setting of the network weights, just like in case of standard neural network models, in particular ability to adapt the mechanism of backpropagation for networks involving complex granules and neurons

The paper is organized as follows: In Section 2, we introduce basic notions of rough sets and possible types of rough neurons. In Section 3, we illustrate how to switch from basic rough neurons in information systems to multi-dimensional neurons modeling vectors of decision memberships in decision systems. In Section 4, we show examples of applications of multi-dimensional neurons to deal with complex concepts and decision processes related to the task of magnetic resonance images' segmentation (cf. [15]). Section 5 concludes the paper.

## 2 Rough Sets and Rough Neurons

Classification systems often attempt to find possibly direct input-output mappings, which may be not learnable. The target concepts may be by nature complex, consisting of simpler sub-concepts. The desired solutions should then have an internal structure. The solutions' components, as well as the way we combine them, should be able to take complex forms too. In our research, we addressed neural network models representing such complex concepts, e.g. by propagating rough membership distributions, weighted sets of rough-set-based decision rules, etc. [12]. The origins of transmitting complex, rough-set-related information throughout neural networks are, however, back to [7], where *rough* neurons correspond to the lower and upper rough set approximations, transformed and transmitted as a kind of two-dimensional signals.

Figure 1 shows some approaches to handling rough set-related information in complex neurons. Let us assume that a set of examples $U$ is given, and every $u \in U$ is represented by a vector of attribute values $a(u)$, where $a : U \rightarrow V_a$, $a \in A$. A tuple $\mathbb{A} = (U, A)$ is referred to as an information system [10]. Assume that $U$ actually consists of positive and negative examples of some target concept (decision). We represent positive examples by $X \subseteq U$. We *approximate* $X$ by indiscernibility classes of examples with the same values, i.e. $[u]_A = \{x \in U : \forall_{a \in A} a(x) = a(u)\}$. We consider the following lower and upper approximations of $X$:

$$Low(X) = \{u \in U : [u]_A \subseteq X\} \quad \text{and} \quad Upp(X) = \{u \in U : [u]_A \cap X \neq \emptyset\}$$

as well as the following positive, negative, and boundary regions for $X$:

$$Pos(X) = Low(X), Neg(X) = U \setminus Upp(X), Bnd(X) = Upp(X) \setminus Low(X)$$

$Pos(X)$ gathers all cases that certainly satisfy the concept represented by $X$, $Neg(X)$ – the cases certainly not satisfying the concept, and $Bnd(X)$ – the undecided ones. $Upp(X) = Pos(X) \cup Bnd(X)$ gathers the cases that possibly satisfy the concept.

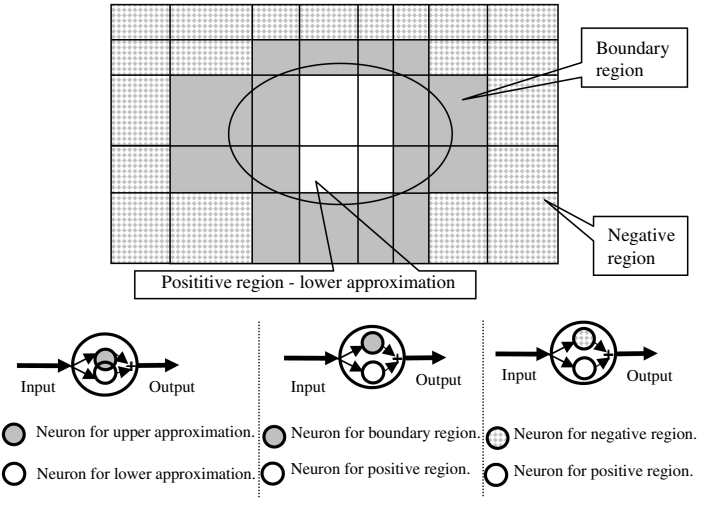



**Fig. 1.** Three rough neurons, transmitting information about: $Low(X)$ and $Upp(X)$ [7], $Pos(X)$ and $Bnd(X)$ [2], as well as $Pos(X)$ and $Neg(X)$ (see next section)

All above regions may take different forms for different sets of attributes and universes. The usage of different regions while, e.g., classifying new cases requires their tuning and combining based on, e.g., neural network architecture. As illustrated by Figure 1, rough neurons may transmit the signals of the form, e.g., $(Low, Upp)$ [7], $(Pos, Bnd)$ [2], or $(Pos, Neg)$ further analyzed in next section. Signals can be interpreted, e.g., as pairs of memberships of analyzed cases to particular regions. The neuron may receive such degrees basing on different information (sub-)systems. Then it combines, compares, and processes them. This way, the networks of rough neurons are able

to learn how to synthesize information from different (rough) classifiers, which is the topic of permanent interest in the area of machine learning [3].

Ability to compare the signal components *inside* a rough neuron means that its transition function is two-dimensional. Relative increase of membership into $Pos$ in comparison to, e.g., $Bnd$ should result in increase of the $Pos$-output component on the cost of decrease of the $Bnd$-output component. One can imagine such situation when the underlying data starts to provide more precise information about the positive examples of a given concept. A similar play could be expected between $Pos$ and $Neg$. It is, however, less intuitive in case of $Low$ and $Upp$ because increase of membership into $Low$ implies that membership into $Upp$ potentially increases too. This is why it was claimed in [2] that replacement of $(Low, Upp)$ by a pair of disjoint, mutually counteractive regions $(Pos, Bnd)$ may improve the model's performance.

## 3 Decision Systems and Normalizing Neural Networks

In the classification tasks we search for a method that labels each given example with one out of potentially long list of possible decision classes/values. Then, within the rough set framework, we should not restrict to the sets of positive and negative examples of a given concept $X \subseteq U$, but rather consider the sets of positive examples of many mutually disjoint decision classes. Information systems $\mathbb{A} = (U, A)$ are replaced by decision systems $\mathbb{A} = (U, A, d)$, where additional decision attribute $d : U \to \{1, ..., r\}$ determines decision classes/concepts $X_k = \{u \in U : d(u) = k\}$, $k = 1, ..., r$. From the perspective of rough neurons, it is then reasonable to replace two dimensions – for $(Low, Upp)$, $(Pos, Bnd)$, or $(Pos, Neg)$ – with $r$ dimensions, one for each decision class. More precisely, this is a straightforward extension of the $(Pos, Neg)$-model. – The concept $X \subseteq U$ can be interpreted in terms of two decision classes: $X_1 = X$ and $X_2 = U \setminus X$. Further, $Pos(X) = Low(X_1)$ and $Neg(X) = Pos(U \setminus X) = Low(X_2)$. Hence, $(Pos, Neg)$ is a special case of multidimensional model $(Low_1, ..., Low_r)$, where $Low_k$ reflects positive examples of decision class $X_k$, for $k = 1, ..., r$, $r \geq 2$.

In [13], we considered multidimensional neural network models in terms of the vectors of memberships of the classified objects into particular decision classes. We call them *Normalizing Neural Networks* (NNN) given that $r$-dimensional signals are *normalized* to the elements of $\triangle_{r-1}$, i.e. vectors of non-negative numbers summing up to 1, while their processing through non-linear multidimensional transition functions $\phi : \mathbb{R}^r \to \triangle_{r-1}$. The origin and meaning of vectors may obviously differ. Given more examples in the next section, here we outline basic foundations.

Figure 2 presents NNN with one hidden layer. Vectors $x_i \in \mathbb{R}^r$ correspond to the outcomes of some (sub-)classifiers. Each $j$-th neuron in the hidden layer takes as an input the vector $s_j \in \mathbb{R}^r$ and provides as output $y_j = \phi(s_j)$. The input to the output neuron is denoted by $t \in \mathbb{R}^r$ and its output – by $h = \phi(t)$. Vectors $s_1, ..., s_m, t$ are the weighted sums of outcomes of previous layers, i.e.: $t = \sum_{j=1}^{m} w_j y_j$ and $s_j = \sum_{i=0}^{n} v_{ij} x_i$, for the real-valued weights $v_{ij}, w_j \in \mathbb{R}$, $i = 0, ..., n$, $j = 1, ..., m$.

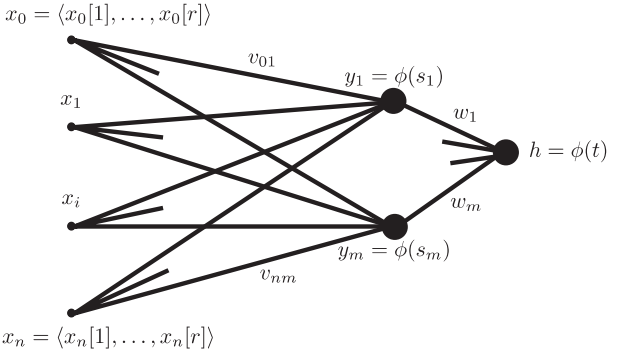



**Fig. 2.** Normalizing neural network: exemplary three-layered structure

Inside neurons, NNN should generalize the classical case, where we use monotone, mostly sigmoidal transition functions. In [13], we use the following $\phi_\alpha : \mathbb{R}^r \to \triangle_{r-1}$, where parameter $\alpha > 0$ determines the steepness of transition:

$$\phi_\alpha(s) = \left\langle \frac{e^{\alpha s[1]}}{\sum_{l=1}^{r} e^{\alpha s[l]}}, \ldots, \frac{e^{\alpha s[k]}}{\sum_{l=1}^{r} e^{\alpha s[l]}}, \ldots, \frac{e^{\alpha s[r]}}{\sum_{l=1}^{r} e^{\alpha s[l]}} \right\rangle$$

$\phi_\alpha$ can be compared to the Gibbs' *softmax* method [4]. Whenever $s[k] > s[l]$, there is also $\phi_\alpha(s)[k] > \phi_\alpha(s)[l]$. Further, increase of $s[k]$ results in increase of $\phi_\alpha(s)[k]$ on the cost of all other $\phi_\alpha(s)[l]$, $l \neq k$. It is also easily differentiable, which enables to adapt backpropagation procedure [6]. Let us denote by $d = \langle d[1], \ldots, d[r] \rangle$ the distribution, that we would like to obtain. Consider the the normalized Euclidean distance $E = \frac{1}{2}\sum_{k=1}^{r}(h[k] - d[k])^2$ [11] as the error function. Vector $h = \langle h[1], \ldots, h[r] \rangle$ is the output of NNN, as shown in Figure 2. We use negative gradient of $E$, treated as a function of the weight vectors, to tune the network weights:

$$\frac{\partial E}{\partial w_j} = \left\langle h - d \,\middle|\, \frac{\partial h}{\partial w_j} \right\rangle \quad \text{where} \quad \left[\frac{\partial h}{\partial w_j}\right]^T = D\phi_\alpha(t)\,[y_j]^T$$
$$\frac{\partial E}{\partial v_{ij}} = \left\langle h - d \,\middle|\, \frac{\partial h}{\partial v_{ij}} \right\rangle \quad \text{where} \quad \left[\frac{\partial h}{\partial v_{ij}}\right]^T = D\phi_\alpha(t) w_j D\phi_\alpha(s_j)\,[x_i]^T$$

and where $D\phi_\alpha$ is the derivative matrix of $\phi_\alpha$, given by formula $D\phi_\alpha(s) =$

$$\alpha \cdot \begin{bmatrix} \phi_\alpha(s)[1](1-\phi_\alpha(s)[1]) & \ldots & -\phi_\alpha(s)[1]\phi_\alpha(s)[k] & \ldots & -\phi_\alpha(s)[1]\phi_\alpha(s)[r] \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ -\phi_\alpha(s)[k]\phi_\alpha(s)[1] & \ldots & \phi_\alpha(s)[k](1-\phi_\alpha(s)[k]) & \ldots & -\phi_\alpha(s)[k]\phi_\alpha(s)[r] \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ -\phi_\alpha(s)[r]\phi_\alpha(s)[1] & \ldots & -\phi_\alpha(s)[r]\phi_\alpha(s)[k] & \ldots & \phi_\alpha(s)[r](1-\phi_\alpha(s)[r]) \end{bmatrix}$$

The above mechanism was implemented and proved to work efficiently in [12,13].

## 4 Handling Complex Concepts – Case Studies

As we mentioned in Section 1, one of the features of many rough set-based classifiers is that they do not tend to particular decisions immediately, but rather operate with complete vectors of decision memberships. From this perspective, the neural network model presented in the previous section fits rough set methodology perfectly. It can be further applied to more compound hierarchical classification schemes, as reported in [12]. In this paper, however, we would like to focus on observation that analysis of such more complex structures, like vectors instead of single values, are often justified by specification of the decision problem itself.

The major case study in this section relates to segmentation of Magnetic Resonance Images (MRI). It entails labeling pixels with tissue types. In case of the brain images, we have usually five of such types: background, bone, white matter, gray matter, and cerebral spinal fluid [5]. Segmentation may be done by an expert who visually inspects a series of MRI scans. In a clinical setting, however, the tools analyzing MRI's in an automated manner are very valuable. Further, MRI's can be performed at various levels of accuracy, involving noise and thickness of horizontal slices. The scans can be generated per every 1mm, 3mm, etc. across the brain's volume. This is why we should rather talk about *voxels* than pixels in MRI images. For thicker slices, voxels may actually overlap with multiple tissue types, which is called as Partial Volume Effect (PVE) [15].

Figure 3 illustrates how one can deal with PVE by applying a two-stage process. First, we identify voxels affected by PVE. We build decision table $\mathbb{A} = (U, A \cup \{d\})$ where objects $u \in U$ correspond to voxels and attributes $a \in A$ – to the voxels' features extracted from a given image, usually available in three *modalities* [5]. Although the attributes' values label particular voxels, they are calculated based on information about the entire images. They can be obtained from, e.g., *magnitude frequency histograms* and self-organizing networks applied to cluster the images' regions (cf. [15]). Decision $d \notin A$ determines the "PVE" and "NOE" (no PVE) classes of voxels. Further, depending on the results of the first stage, we classify voxels into the particular tissue type classes (NOE case), or we *estimate* (*predict*) their overlaps with particular tissue types (PVE case). Such membership-based approach is still valuable for further phases of the MRI analysis, where the results of segmentation are transformed into higher-level attributes describing proportions/distributions of tissue types in particular slices.

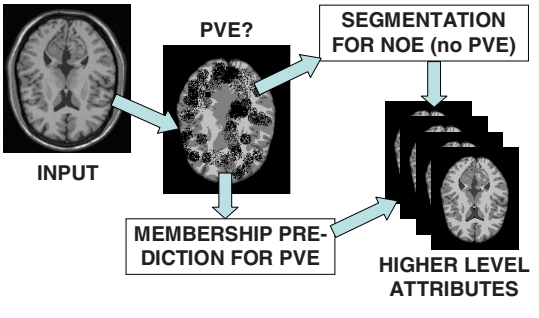



**Fig. 3.** The scheme for two-stage MRI analysis, involving "PVE" and "no PVE" (NOE) voxels

The PVE-case involves decisions in form of distributions $d = \langle d[1], \ldots, d[r] \rangle$, where $r$ equals to the number of tissue types. Further, given the classifier's output $h$, its error can be measured by $E = \frac{1}{2} \sum_{k=1}^{r} (h[k] - d[k])^2$, like in Section 3. Similarly stated decision problems occur in many applications. As another example, let us mention about the post-surgery survival analysis [1], where decision rules identify the groups of patients with specific tendencies (distributions) of tumor reoccurrences. The rule-based method introduced in [1] is a symbolic counterpart of neural network-based approach studied in this paper. In both cases, the objects and attributes are quite standard but decisions take form of compound distributions. These two techniques may be actually combined, as the rules' results may feed the input layer of NNN.

Let us continue with the MRI analysis and note that PVE-NOE classification can be redefined for a different universe of objects. So far, voxels corresponded to the objects in $U$. In [15], $U$ actually consisted of voxels taken from various images, with the training and testing samples based on disjoint sets of slices. Now, we specify each single training/testing case – and, therefore, an element of $U$ – as the whole slice, and the decision value – as the ordering of voxels from those most to those least affected by PVE. Such an approach enables the network to compare to each other the *atomic* concepts (voxels) while processing the complex ones (whole images).

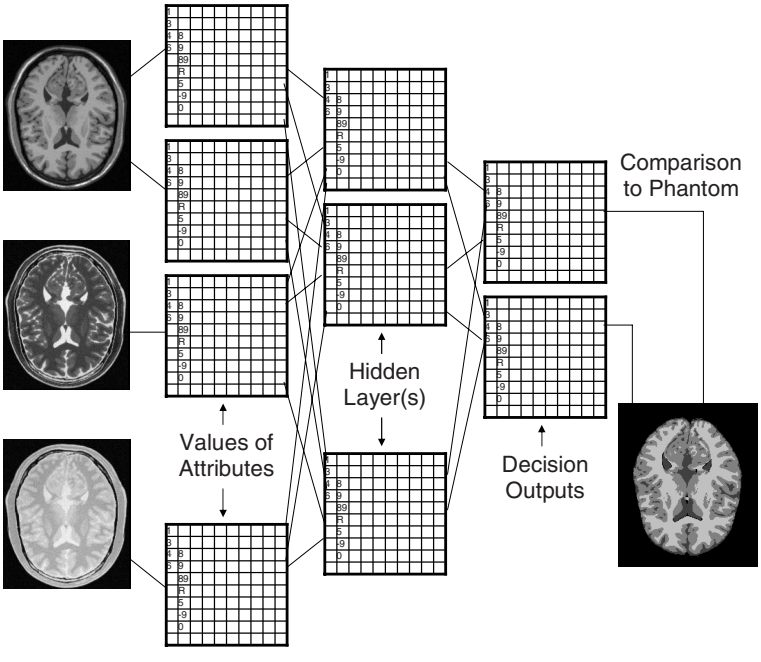



**Fig. 4.** NNN processing the vectors of voxels' values for the whole MRI slices. The input layer corresponds to the attributes calculated from three MRI modalities [5]. The output neurons represent the normalized vector of weights of voxels' memberships into decision concepts (like "PVE", "white matter", etc.). The network can be learnt using backpropagation described in Section 3, where the target signal is taken from *Phantom* – an image segmented by the expert.

While comparing Figures 3 and 4, one can see that the network's architectures are the same. The difference lays in the signals' dimensions – the number of tissue types versus the number of voxels in MRI slices. Further, voxels are labeled by the same values, though now the attribute values are multi-dimensional themselves, as the objects correspond to the whole voxels' collections. It is also interesting to analyze the role of function $\phi_\alpha : \mathbb{R}^r \rightarrow \triangle_{r-1}$, now with $r$ equal to the number of voxels. We can see that voxels *compete* to each other while being processed through multi-dimensional neurons, with ability to learn the competition's laws within a backpropagation-like framework. This example shows an important advantage of normalizing neural networks with respect to more standard models, when applied to complex decision problems.

## 5 Conclusions

We discussed rough neural networks as handling complex concepts being learnt from data. As the basic mechanism, we adapted normalizing neural networks [12,13] equipped with neurons processing multi-dimensional signals. We showed how such neurons generalize and complement the previously investigated types of rough neurons [2,7]. We studied real-life examples requiring operating with complex concepts and appropriately defined learning models. As a case study, we considered the tasks related to segmentation of magnetic resonance images [5,15]. We also discussed how the proposed methodology fits into a more general framework of rough-neural computing [9].

**Acknowledgements.** The work of the second co-author is partly supported by grant 3T11C00226 from the Polish Ministry of Scientific Research and Higher Education.

## References

1. Bazan, J.G., Skowron, A., Ślęzak, D., Wróblewski, J.: Searching for the complex decision reducts: The case study of the survival analysis. In: Proc. of ISMIS'2003, LNAI, **2871**, Springer (2003) pp. 160–168.
2. Chandana, S., Mayorga, R.V.: Rough Approximation based Neuro-Fuzzy Inference System. In: Proc. of IEEE HIS'2005, Rio (2005) pp. 518–521.
3. Dietterich, T.: Machine learning research: four current directions. AI Magazine **18/4** (1997) pp. 97–136.
4. Gibbs, J.W.: Elementary Princinples in Statistical Mechanisms. Dover, NY (1960).
5. Kaus, M., Warfield, S.K., Nabavi, A., Black, P.M., Jolesz, F.A., Kikinis, R.: Automated Segmentation of MRI of Brain Tumors. Radiology **218** (2001) pp. 586–591.
6. le Cun, Y.: A theoretical framework for backpropagation. In: Neural Networks – concepts and theory. IEEE Computer Society Press (1992).
7. Lingras, P.: Rough Neural Networks. In: Proc. of IPMU'1996, Granada (1996) pp. 1445–1450.
8. Nguyen, H.S.: Approximate Boolean Reasoning: Foundations and Applications in Data Mining. Transactions on Rough Sets V, LNCS, **4100**, Springer (2006) pp. 334–506.
9. Pal, S.K., Polkowski, L., Skowron, A. (eds.): Rough-Neural Computing. Cognitive Technologies Series, Springer (2004).
10. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer (1992).

11. Ślęzak, D.: Various approaches to reasoning with frequency-based decision reducts: a survey. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.): Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems. Physica Verlag (2000) pp. 235–285.
12. Ślęzak, D., Szczuka, M., Wróblewski, J.: Feedforward Concept Networks. In: B. Dunin-Kęplicz, A. Jankowski, A. Skowron, M. Szczuka (eds.), Monitoring, Security, and Rescue Techniques in Multiagent Systems. Advances in Soft Computing, Springer (2005) pp. 281–292.
13. Ślęzak, D., Wróblewski, J., Szczuka, M.: Neural Network Architecture for Synthesis of the Probabilistic Rule Based Classifiers. ENTCS, **82/4**, Elsevier (2003).
14. Szczuka, M.: Rough Sets and Artificial Neural Networks. In: Polkowski, L., Skowron, A. (eds.), Rough Sets in Knowledge Discovery I & II, Physica Verlag (1998).
15. Widz, S., Revett, K., Ślęzak, D.: A Rough Set-Based Magnetic Resonance Imaging Partial Volume Detection System. In: Proc. PReMI'2005, Calcutta, Springer (2005) pp. 756–761.

# Author Index