Davide Ciucci Masahiro Inuiguchi
Yiyu Yao Dominik Ślęzak
Guoyin Wang (Eds.)

LNAI 8170

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

14th International Conference, RSFDGrC 2013
Halifax, NS, Canada, October 2013
Proceedings

Springer

Lecture Notes in Artificial Intelligence 8170

Subseries of Lecture Notes in Computer Science

Davide Ciucci Masahiro Inuiguchi
Yiyu Yao Dominik Ślęzak
Guoyin Wang (Eds.)

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

14th International Conference, RSFDGrC 2013
Halifax, NS, Canada, October 11-14, 2013
Proceedings

Springer

Volume Editors

Davide Ciucci
University of Milano-Bicocca, Italy
E-mail: ciucci@disco.unimib.it

Masahiro Inuiguchi
Osaka University, Japan
E-mail: inuiguti@sys.es.osaka-u.ac.jp

Yiyu Yao
University of Regina, SK, Canada
E-mail: yyao@cs.uregina.ca

Dominik Ślęzak
University of Warsaw and Infobright Inc., Poland
E-mail: slezak@mimuw.edu.pl

Guoyin Wang
Chongqing Institute of Green and Intelligent Technology, CAS, Chongqing, China
E-mail: wanggy@ieee.org

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-41217-2 e-ISBN 978-3-642-41218-9
DOI 10.1007/978-3-642-41218-9
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013948999

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume comprises papers accepted for presentation at the 14th Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC) International conference, which, along with the 8th International conference on Rough Sets and Knowledge Technology (RSKT) conference, was held as a major part of Joint Rough Set Symposium (JRS) during October 11–14, 2013 in Halifax, Canada. JRS was organized for the first time in 2007 in Toronto, Canada, and was re-established in Chengdu, China 2012, as the major event assembling different rough-set-related conferences and workshops. In addition to RSFDGrC and RSKT, JRS 2013 also hosted the 4th Rough Set Theory Workshop (RST) and the Rough Set Applications Workshop (RSA), both held on October 10, 2013. RSFDGrC is a series of scientific events spanning the last 15 years. It investigates primarily rough sets in connection with the other disciplines outlined in its title, with respect to both foundations and applications.

JRS 2013 received 106 submissions which were carefully reviewed by two or more Program Committee (PC) members or additional reviewers. After the rigorous process finally 44 regular papers (acceptance rate 41.5 %) and 25 short papers were accepted for presentation at the symposium and publication in two volumes of the JRS proceedings.

This volume contains original research papers submitted to the conference RSFDGrC 2013 and lecture notes of keynote speakers: Andrzej Skowron, Bo Zhang, Vijay Raghavan, Boris Mirkin, and Jian Pei. We would like to thank all the authors, both those whose papers were accepted and those whose papers did not appear in the proceedings, for their best efforts – it is their work that gives meaning to the conference.

It is a pleasure to thank all those people who helped this volume to come into being and JRS 2013 to be a successful and exciting event. It would not be possible to hold the symposium without the committees and the sponsors. We deeply appreciate the work of the PC members and additional reviewers (Yasunori Endo, Faeze Eshragh, Wenxin Yang) who assured the high standards of accepted papers. We hope that the resulting proceedings are evidence of the high-quality and exciting RSFDGrC 2013 program. This program also included four special sessions: Fuzzy and Rough Hybridization (Chris Cornelis, Richard Jensen, Neil Mac Parthalain, Wei-Zhi Wu), Covering-Based Rough Sets and Their Applications (William Zhu, Fan Min), Soft Clustering (Pawan Lingras, Manish Joshi), Granular Computing Theory Research and Applications (Yanping Zhang, Ling Zhang, Shu Zhao, Xuqing Tang, Deyu Li, Qinghua Zhang).

We would like to express our gratitude to the special session chairs and both RST and RSA workshops' chairs (JingTao Yao, Ahmad Taher Azar, Stan Matwin) for their great work.

We deeply acknowledge the conscientious help of all the JRS chairs (Pawan Lingras, Yuhua Qian, Chris Cornelis, Sushmita Mitra, Hai Wang, Andrzej Janusz) whose valuable suggestions and various pieces of advice made the process of proceedings preparation and conference organization much easier to cope with.

We also gratefully thank our sponsors: David Gauthier, Vice President - Academic and Research, Saint Mary's University, Halifax, for sponsoring the reception; Kevin Vessey, Associate Vice President - Research, Saint Mary's University, Halifax, for sponsoring the data mining competition; Steven Smith, Dean of Science, Saint Mary's University, Halifax, for sponsoring the conference facilities; Danny Silver, Director, Jodrey School of Computer Science, Acadia University, Wolfville, for sponsoring the second day of the conference in the beautiful Annapolis valley and Acadia University; Stan Matwin, Canada Research Chair and Director, Institute for Big Data Analytics, Dalhousie University, Halifax, for sponsoring RST and RSA workshops at Dalhousie University; finally Infobright Inc. corporation for being the industry sponsor of the entire event.

Our immense gratitude goes once again to Pawan Lingras for taking charge of JRS 2013 organization and his invaluable help and support throughout the whole preparation of the symposium.

We are very thankful to Alfred Hofmann and the excellent LNCS team at Springer for their help and co-operation. We would also like to acknowledge the use of EasyChair, a great conference management system.

Finally, let us express our hope that the reader will find all the papers in the proceedings interesting and stimulating.

October 2013

Davide Ciucci
Masahiro Inuiguchi
Yiyu Yao
Dominik Ślęzak
Guoyin Wang

Organization

General Chairs

Pawan Lingras
Yiyu Yao

Steering Committee Chairs

Dominik Ślęzak
Guoyin Wang

Joint Program Chairs

Davide Ciucci
Yuhua Qian

Program Co-chairs for RSFDGrC 2013

Chris Cornelis
Sushmita Mitra

Program Co-chairs for RSKT 2013

Masahiro Inuiguchi
Piotr Wasilewski

Program Co-chairs for RSA 2013

Stan Matwin
Ahmad Taher Azar

Program Co-chairs for RST 2013

Marcin Wolski
JingTao Ya

Data Mining Competition Chairs

Hai Wang
Andrzej Janusz

Joint Program Committee

Arun Agarwal
Adel M. Alimi
Simon Andrews
Piotr Artiemjew
S. Asharaf
Sanghamitra Bandyopadhyay
Mohua Banerjee
Andrzej Bargiela
Alan Barton
Jan Bazan
Theresa Beaubouef
Rafael Bello
Rabi Nanda Bhaumik
Jurek Błaszczyński
Nizar Bouguila
Yongzhi Cao
Salem Chakhar
Mihir K. Chakraborty
Chien-Chung Chan
Chiao-Chen Chang
Santanu Chaudhury
Degang Chen
Mu-Chen Chen
Mu-Yen Chen
Igor Chikalov
Zoltán Csajbók
Jianhua Dai
Bijan Davvaz
Martine Decock
Dayong Deng
Thierry Denoeux
Jitender Deogun
Lipika Dey
Fernando Diaz
Maria Do Carmo
Ivo Düntsch
Zied Elouedi
Francisco Fernandez
Wojciech Froelich
G. Ganesan
Yang Gao
Guenther Gediga
Neveen Ghali
Anna Gomolińska
Salvatore Greco
Jerzy Grzymała-Busse
Jianchao Han
Aboul Ella Hassanien
Jun He
Christopher Henry
Francisco Herrera
Chris Hinde
Shoji Hirano
Władysław Homenda
Feng Hu
Qinghua Hu
Shahid Hussain
Dmitry Ignatov
Hannah Inbarani
Ryszard Janicki
Andrzej Jankowski
Richard Jensen
Xiuyi Jia
Manish Joshi
Jouni Järvinen
Janusz Kacprzyk
Byeong Ho Kang
C. Maria Keet
Md. Aquil Khan
Yoo-Sung Kim
Michiro Kondo
Beata Konikowska
Jacek Koronacki
Witold Kosiński
Bożena Kostek
Adam Krasuski
Vladik Kreinovich
Rudolf Kruse
Marzena Kryszkiewicz
Yasuo Kudo
Yoshifumi Kusunoki
Sergei Kuznetsov
Tianrui Li
Jiye Liang
Churn-Jung Liau
Diego Liberati

Antoni Ligęza
T. Y. Lin
Kathy Liszka
Dun Liu
Guilong Liu
Qing Liu
Dickson Lukose
Neil Mac Parthaláin
Pradipta Maji
A. Mani
Victor Marek
Barbara Marszał-Paszek
Tshilidzi Marwala
Benedetto Matarazzo
Nikolaos Matsatsinis
Jesús Medina-Moreno
Ernestina Menasalvas
Jusheng Mi
Duoqian Miao
Alicja Mieszkowicz-Rolka
Tamás Mihálydeák
Fan Min
Pabitra Mitra
Sadaaki Miyamoto
Mikhail Moshkov
Tetsuya Murai
Kazumi Nakamatsu
Michinori Nakata
Amedeo Napoli
Kanlaya Naruedomkul
Hung Son Nguyen
Linh Anh Nguyen
Vilem Novak
Mariusz Nowostawski
Hannu Nurmi
Hala Own
Nizar Banu
Piero Pagliani
Krzysztof Pancerz
Piotr Paszek
Alberto Guillen Perales
Georg Peters
James F. Peters
Frederick Petry
Jonas Poelmans
Lech Polkowski
Henri Prade
Keyun Qin
Mohamed Quafafou
Anna Maria Radzikowska
Vijay V. Raghavan
Sheela Ramanna
Zbigniew Raś
Kenneth Revett
Leszek Rolka
Leszek Rutkowski
Henryk Rybiński
Wojciech Rząsa
Hiroshi Sakai
Abdel-Badeeh Salem
Miguel Ángel Sanz-Bobi
Gerald Schaefer
Noor Setiawan
Siti Mariyam Shamsuddin
Marek Sikora
Arul Siromoney
Andrzej Skowron
Vaclav Snasel
John G. Stell
Jarosław Stepaniuk
Zbigniew Suraj
Piotr Synak
Andrzej Szałas
Marcin Szczuka
Tomasz Szmuc
Marcin Szpyrka
Roman Słowiński
Domenico Talia
Shusaku Tsumoto
Gwo-Hshiung Tzeng
Nam Van Huynh
Changzhong Wang
Junhong Wang
Xin Wang
Junzo Watada
Ling Wei
Arkadiusz Wojna
Karl Erich Wolff
Michał Woźniak
Wei-Zhi Wu

Ronald Yager
Yan Yang
Yingjie Yang
Yong Yang
Yubin Yang
Nadezhda G. Yarushkina
Dongyi Ye
Hong Yu
Sławomir Zadrożny
Yan-Ping Zhang
Shu Zhao
William Zhu
Wojciech Ziarko
Beata Zielosko

Table of Contents

Invited Keynote Lectures

Inconsistency, Incompleteness, Non-Determinism

Fuzzy and Rough Hybridization

Granular Computing and Covering-Based Rough Sets

Soft Clustering

Image and Medical Data Analysis

30 Years of Rough Sets and Future Perspectives

Andrzej Skowron[1], Andrzej Jankowski[2], and Roman Swiniarski[3,4,⋆]

[1] Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl
[2] Institute of Computer Science, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
a.jankowski@ii.pw.edu.pl
[3] Department of Computer Science, San Diego State University
5500 Campanile Drive San Diego, CA 92182, USA
[4] Institute of Computer Science Polish Academy of Sciences
Jana Kazimierza 5, 01-248 Warsaw, Poland
rswiniarski@mail.sdsu.edu

Abstract. In the development of rough set theory and applications, one can distinguish three main stages. While the first period was based on the assumption that objects are perceived by means of partial information represented by attributes, in the second period it was assumed that information about the approximated concepts is partial too. Approximation spaces and searching strategies for relevant approximation spaces were recognized as the basic tools for rough sets. Important achievements both in theory and applications were obtained. Nowadays, a new period for rough sets is emerging.

Keywords: rough sets, granular computing, (approximate) Boolean reasoning, interactions, adaptive judgment.

1 Introduction

The rough set approach was proposed by Professor Zdzisław Pawlak in 1982 [11,12] as a tool for dealing with imperfect knowledge, in particular with vague

⋆ The authors would like to express sincere appreciation and gratitude to Professor Dominik Ślęzak for his comments and corrections which helped to improve the paper. This work was supported by the Polish National Science Centre grants 2011/01/B/ST6/03867, 2011/01/D/ST6/06981, 2012/05/B/ST6/03215, the Foundation for Polish Science within the Homing Plus program, Edition 3/2011, co-financed from the European Union Regional Development Fund, and the Polish National Centre for Research and Development grants SP/I/1/77065/10 ("Interdisciplinary System for Interactive Scientific and Scientific-Technical Information") and O ROB/0010/03/001 ("Modern engineering tools for decision support for commanders of the State Fire Service of Poland during Fire & Rescue operations in the buildings").

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 1–10, 2013.

concepts. Rough set theory has attracted attention of many researchers and practitioners all over the world, who have contributed essentially to its development and applications.

The developed methods based on rough set theory alone or in combination with other approaches found applications in many areas including: acoustics, bioinformatics, business and finance, chemistry, computer engineering (*e.g.*, data compression, digital image processing, digital signal processing, parallel and distributed computer systems, sensor fusion, fractal engineering), decision analysis and systems, economics, electrical engineering (*e.g.*, control, signal analysis, power systems), environmental studies, digital image processing, informatics, medicine, molecular biology, musicology, neurology, robotics, social science, software engineering, spatial visualization, Web engineering, and Web mining.

The rough set approach is of fundamental importance in artificial intelligence and cognitive sciences, especially in machine learning, data mining and knowledge discovery, pattern recognition, decision support systems, expert systems, intelligent systems, multiagent systems, (complex) adaptive systems, autonomous systems, cognitive systems, conflict analysis, risk management systems.

Rough sets have established relationships with many other approaches such as fuzzy set theory, granular computing, evidence theory, formal concept analysis, (approximate) Boolean reasoning, multicriteria decision analysis, statistical methods, decision theory, matroids. Despite the overlap with many other theories rough set theory may be considered as an independent discipline in its own right. There are reports on many hybrid methods obtained by combining rough sets with other approaches such as soft computing (fuzzy sets, neural networks, genetic algorithms), statistics, natural computing, mereology, principal component analysis, singular value decomposition or support vector machines.

The main advantage of rough set theory in data analysis is that it does not need any preliminary or additional information about data like probability distributions in statistics, basic probability assignments in evidence theory, a grade of membership or the value of possibility in fuzzy set theory.

One can observe the following about the rough set approach: (i) introduction of efficient algorithms for finding hidden patterns in data, (ii) determination of optimal sets of data (data reduction), evaluation of the significance of data, (iii) generation of sets of decision rules from data, (iv) easy-to-understand formulation, (v) straightforward interpretation of obtained results, (vi) suitability of many of its algorithms for parallel processing.

It is worthwhile to mention that rough sets play a crucial role in the development of granular computing (GC) [15]. The extension to interactive granular computing (IGR) requires generalization of the basic concepts such as complex granules (including both physical and abstract parts [8]), information (decision) systems as well as methods of inducing hierarchical structures of information (decision) systems. The current research projects are aiming at developing

foundations of IGC based on the rough set approach in combination with other soft computing approaches, in particular with fuzzy sets. The approach is called interactive rough granular computing (IRGC). In IRGC computations are based on interactions of complex granules. IRGC can be treated as the basis for (see, *e.g.*,[17] and references in this book): (i) Wisdom Technology, in particular for approximate reasoning (called adaptive judgment) about properties of interactive computations, (ii) context inducing and discovery of structural objects, (iii) reasoning about changes, (iv) process mining (this research was inspired by Professor Pawlak in 1992), (v) perception based computing, (vi) risk management in computational systems [16,8].

Due to the space limitation we restrict in this paper the references on rough sets to two basic papers by Professor Zdzisław Pawlak [11,12], some survey papers [13] and books [17,5,10] including long lists of references to papers on rough sets. The basic ideas of rough set theory and its extensions as well as many interesting applications can be found in a number of books, issues of the Transactions on Rough Sets, special issues of other journals, numerous proceedings of international conferences, and tutorials (see, *e.g.*, [13,17,5]). The reader is referred to the cited books and papers, references in them as well as to web pages `www.roughsets.org`, `rsds.univ.rzeszow.pl`.

In this paper we present comments on some research directions in rough sets over the last 30 years and we also outline future perspectives of rough sets.

2 From Partitions to Coverings

The rough set philosophy is founded on the assumption that with every object of the universe of discourse we associate some information (data, knowledge). Objects characterized by the same information are indiscernible (similar) in view of the available information about them. The *indiscernibility relation* generated in this way is the mathematical basis of rough set theory. This understanding of indiscernibility is related to the idea of Gottfried Wilhelm Leibniz that objects are indiscernible if and only if all available functionals take on them identical values (Leibniz's Law of Indiscernibility: The Identity of Indiscernibles) [9]. However, in the rough set approach indiscernibility is defined relative to a given set of functionals (attributes).

Any set of all indiscernible (similar) objects is called an elementary set, and forms a basic granule (atom) of knowledge about the universe. Any union of some elementary sets is referred to as *crisp* (precise) set. If a set is not crisp then it is called *rough* (imprecise, vague). Note that due to the computational complexity of searching for relevant crisp sets for the considered problem, the searching is usually restricted to a feasible subfamily of the family of all possible unions of elementary sets.

Consequently, each rough set has *borderline cases*, *i.e.*, objects which cannot be classified with certainty as members of either the set or its complement. Obviously crisp sets have no borderline elements at all. This means that borderline cases cannot be properly classified by employing available knowledge.

Thus, the assumption that objects can be "seen" only through the information available about them leads to the view that knowledge has granular structure. Due to the granularity of knowledge, some objects of interest cannot be discerned and appear as the same (or similar). As a consequence, vague concepts in contrast to precise concepts, cannot be characterized in terms of information about their elements. Therefore, in the proposed approach, we assume that any vague concept is replaced by a pair of precise concepts – called the lower and the upper approximation of the vague concept. The lower approximation consists of all objects which definitely belong to the concept and the upper approximation contains all objects which possibly belong to the concept. The difference between the upper and the lower approximation constitutes the boundary region of the vague concept. Approximations are two basic operations in rough set theory.

Hence, rough set theory expresses vagueness not by means of membership, but by employing a 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 means that our knowledge about the set is not sufficient to define the set precisely.

In the literature one can find more details on different aspects of rough set approximations of vague concepts.

The original approach by Professor Pawlak was based on indiscernibility defined by equivalence relations. Any such indiscernibility relation defines a partition of the universe of objects. Over the years many generalizations of this approach were introduced many of which are based on coverings rather than partitions. In particular one can consider similarity (tolerance) based rough set approach, binary relation based rough sets, neighborhood and covering rough sets, dominance based rough set approach, hybridization of rough sets and fuzzy sets, and many others.

One should note that dealing with coverings requires solving several new algorithmic problems such as selection of family of definable sets or resolving problems with selection of relevant definition of approximation of sets among many possible ones. One should also note that for a given problem (*e.g.*, classification problem) one should discover the relevant covering for the target classification task. In the literature there are numerous papers dedicated to theoretical aspects of the covering rough set approach. However, still much more work should be done on rather hard algorithmic issues for the relevant covering discovery.

Another issue to be solved is related to inclusion measures. Parameters of such measures are tuned to induce of the high quality approximations. Usually, this is done on the basis of the minimum description length principle. In particular, approximation spaces with rough inclusion measures have been investigated. This approach was further extended to rough mereological approach. More general cases of approximation spaces with rough inclusion were also discussed in the literature including approximation spaces in GC. Finally, it is worthwhile to mention the approach for ontology approximation used in hierarchical learning of complex vague concepts [17].

3 Rough Sets and Induction

Rough sets are strongly related to inductive reasoning (*e.g.*, in rough set based methods for inducing classifiers or clusters). In this section, we present an illustrative example of the rough set approach to induction of concept approximations. The approach can be generalized to the rough set approach to inductive extensions of approximation spaces.

Let us consider the problem of approximation of concepts over a universe U^∞ (concepts that are subsets of U^∞). We assume that the concepts are perceived only through some subsets of U^∞, called samples. This is a typical situation in the machine learning, pattern recognition, or data mining approaches [6].

We assume that there is given an information system $\mathcal{A} = (U, A)$ and let us assume that for some $C \subseteq U^\infty$ there is given the set $\Pi_U(C) = C \cap U$. In this way we obtain a decision system $\mathcal{A}_d = (U, A, d)$, where $d(x) = 1$ if $x \in \Pi_U(C)$ and $d(x) = 0$, otherwise.

We would like to illustrate how from the decision function d may be induced a decision function μ_C defined over U^∞ with values in the interval $[0, 1]$ which can be treated as an approximation of the characteristic function of C.

Let us assume that $RULES(\mathcal{A}_d)$ is a set of decision rules induced by some rule generation method from $\mathcal{A}_d$. For any object $x \in U^\infty$, let $MatchRules(\mathcal{A}_d, x)$ be the set of rules from this set supported by x.

Now, the rough membership function $\mu_C : U^\infty \to [0, 1]$ approximating the characteristic function of C can be defined as follows

1. Let $R_k(x)$, for $x \in U^\infty$ be the set of all decision rules from $MatchRules(\mathcal{A}_d, x)$ with right hand side $d = k$, where $d = 1$ denotes that the rule r is voting for C and $d = 0$ – that the rule r is voting against C, respectively.
2. We define real values $w_k(x)$, where $w_1(x)$ is called the weight "for" and $w_0(x)$ the weight "against" membership of the object x in C, respectively, by $w_k(x) = \sum_{r \in R_k(x)} strength(r)$, where $strength(r)$ is a normalized function depending on $length$, $support$, $confidence$ of the decision rule r and on some global information about the decision system $\mathcal{A}_d$ such as the size of the decision system or the class distribution.
3. Finally, one can define the value of $\mu_C(x)$ in the following way: $\mu_C(x)$ is undefined if $\max(w_1(x), w_0(x)) < \omega$; $\mu_C(x) = 0$ if $w_0(x) - w_1(x) \geq \theta$ and $w_0(x) > \omega$; $\mu_C(x) = 1$ if $w_1(x) - w_0(x) \geq \theta$ and $w_1(x) > \omega$ and $\mu_C(x) = \frac{\theta + (w_1(x) - w_0(x))}{2\theta}$, otherwise, where ω, θ are parameters set by user.

For computing of the value $\mu_C(x)$ for $x \in U^\infty$ the user should select a strategy resolving conflicting votes "for" and "against" membership of x in C. The degree of these conflicts are represented by values $w_1(x)$ and $w_0(x)$, respectively. Note that for some cases of x due to the small differences between these values the selected strategy may not produce the definite answer and these cases will create the boundary region.

We can now define the lower approximation, the upper approximation and the boundary region of the concept C relative to the induced rough membership function μ_C as follows

$$\begin{aligned} &LOW(C, \mu_C) = \{x \in U^\infty : \ \mu_C(x) = 1\}, \\ &UPP(C, \mu_C) = \{x \in U^\infty : \ \mu_C(x) > 0 \text{ or } \mu_C(x) \text{ is undefined}\}, \\ &BND(C, \mu_C) = UPP(C, \mu_C) \setminus LOW(C, \mu_C). \end{aligned} \tag{1}$$

The whole procedure can be generalized for the case of approximation of more complex information granules than concepts.

4 Boolean Reasoning and Scalability

Solutions for many algorithmic problems related to rough sets were proposed using the (approximate) Boolean reasoning approach [2,3,1,4,17]. Some progress was also made in developing methods scalable for large data sets. In this section we present comments on some applications of Boolean reasoning approach for solving different problems related to rough sets.

The discernibility relations are closely related to indiscernibility and belong to the most important relations considered in rough set theory. Tools for discovering and classifying patterns are based on *reasoning schemes* rooted in various paradigms. Such patterns can be extracted from data by means of methods based, *e.g.*, on Boolean reasoning and discernibility.

The ability to discern between perceived objects is important for constructing many entities like reducts, decision rules or decision algorithms. In the standard approach the discernibility relation $DIS(B) \subseteq U \times U$ is defined by $x \ DIS(B) \ y$ if and only if $non(x \ IND(B) \ y)$, *i.e.*, $B(x) \cap B(y) = \varnothing$, where $B(x)$, $B(y)$ are neighborhoods of x and y, respectively. However, this is not the case for generalized approximation spaces.

The idea of Boolean reasoning is based on construction for a given problem P of 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. Let us mention that to solve real-life problems it is necessary to deal with Boolean functions with large sizes.

A successful methodology based on the discernibility of objects and Boolean reasoning has been developed for computing of many important ingredients for applications. These applications include generation of 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 (see, *e.g.*, [13,17]).

Most of the problems related to generation of the above mentioned entities are NP-complete or NP-hard. However, it was possible to develop efficient heuristics returning suboptimal solutions of the problems. The results of experiments on many data sets are very promising. They show very good quality of 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 efficient from the point of view of time necessary for computing of the solution. Many of these methods are based on discernibility matrices. However, it is possible to compute the necessary information about these matrices without their explicit construction (*i.e.*, by sorting or hashing original data).

The considered methodology makes it possible to construct heuristics having a very important *approximation property* which can be formulated as follows: expressions, called *approximate implicants*, generated by heuristics that are *close* to prime implicants define approximate solutions for the problem.

Mining large data sets is one of the biggest challenges in KDD. In many practical applications, there is a need of data mining algorithms running on terminals of possibly distributed database systems where the only access to data is enabled by SQL queries or NoSQL operations.

Let us consider two illustrative examples of problems for large data sets: (i) searching for short reducts, (ii) searching for best partitions defined by cuts on continuous attributes. In both cases the traditional implementations of rough sets and Boolean reasoning based methods are characterized by the high computational cost. The critical factor for time complexity of algorithms solving the discussed problems is the number of data access operations. Fortunately some efficient modifications of the original algorithms were proposed by relying on concurrent retrieval of higher level statistics which are sufficient for the heuristic search of reducts and partitions (see, *e.g.*, [13,17]). The rough set approach was also applied in development of other scalable big data processing techniques (*e.g.*, Infobright `http://www.infobright.com/`).

5 Rough Sets and Logic

Rough set theory has contributed to some extent to various kinds of deductive reasoning. Particularly, various kinds of logics based on the rough set approach have been investigated, rough set methodology contributed essentially to modal logics, many-valued logics (especially different types of 3-valued logics), intuitionistic logics, paraconsistent logics and others (see, *e.g.*, references in book [17] and in articles [13]).

There are numerous issues related to approximate reasoning under uncertainty including inductive reasoning, abduction, analogy based reasoning and common sense reasoning.

We would like to stress that still much more work should be done to develop approximate reasoning methods about complex vague concepts for making progress in development of intelligent systems. This idea was very well expressed by Professor Leslie Valiant (the 2011 winner of the ACM Turing Award, the highest distinction in computer science, "for his fundamental contributions to the development of computational learning theory and to the broader theory of computer science") (`http://people.seas.harvard.edu/~valiant/researchinterests.htm`):

> *A fundamental question for artificial intelligence is to characterize the computational building blocks that are necessary for cognition. A specific challenge is to build on the success of machine learning so as to cover broader issues in intelligence. [...] This requires, in particular a reconciliation between two contradictory characteristics – the apparent logical nature of reasoning and the statistical nature of learning.*

It is worthwhile to present two more views. The first one by Professor Lotfi A. Zadeh, the founder of fuzzy sets and the computing with words paradigm (see [18] and also `http://www.cs.berkeley.edu/~zadeh/presentations.html`):

> *Manipulation of perceptions plays a key role in human recognition, decision and execution processes. As a methodology, computing with words provides a foundation for a computational theory of perceptions - a theory which may have an important bearing on how humans make- and machines might make - perception-based rational decisions in an environment of imprecision, uncertainty and partial truth. [...] computing with words, or CW for short, is a methodology in which the objects of computation are words and propositions drawn from a natural language.*

and another view by Judea Pearl (the 2011 winner of the ACM Turing Award, "for fundamental contributions to artificial intelligence through the development of a calculus for probabilistic and causal reasoning") [14]:

> *Traditional statistics is strong in devising ways of describing data and inferring distributional parameters from sample. Causal inference requires two additional ingredients: a science-friendly language for articulating causal knowledge, and a mathematical machinery for processing that knowledge, combining it with data and drawing new causal conclusions about a phenomenon.*

The question arises about the logic relevant for the above mentioned tasks. First let us observe that the satisfiability relations in the IRGC framework can be treated as tools for constructing new information granules. If fact, for a given satisfiability relation, the semantics of formulas relative to this relation is defined. In this way the candidates for new relevant information granules are obtained. We would like to emphasize a very important feature. The relevant satisfiability relation for the considered problems is not given but it should be induced (discovered) on the basis of a partial information encoded in information (decision) systems. For real-life problems, it is often necessary to discover a hierarchy of satisfiability relations before we obtain the relevant target level. Information granules constructed at different levels of this hierarchy finally lead to relevant ones for approximation of complex vague concepts related to complex information granules expressed using natural language (see Figure 1). The reasoning making it possible to derive relevant information granules for solutions of the target tasks is called adaptive judgment. Deduction and induction as well as abduction or analogy based reasoning are involved in adaptive judgment. Among

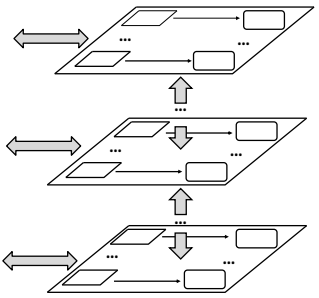


Fig. 1. Interactive hierarchical structures (gray arrows show interactions between hierarchical levels and the environment, arrows at hierarchical levels point from information (decision) systems representing partial specifications of satisfiability relations to induced from them theories consisting of rule sets)

the tasks for adaptive judgment are the following ones supporting reasoning toward: searching for relevant approximation spaces, discovery of new features, selection of relevant features, rule induction, discovery of inclusion measures, strategies for conflict resolution, adaptation of measures based on the minimum description length principle, reasoning about changes, perception (action and sensory) attributes selection, adaptation of quality measures over computations relative to agents, adaptation of object structures, discovery of relevant context, strategies for knowledge representation and interaction with knowledge bases, ontology acquisition and approximation, learning in dialogue of inclusion measures between information granules from different languages (*e.g.*, the formal language of the system and the user natural language), strategies for adaptation of existing models, strategies for development and evolution of communication language among agents in distributed environments, strategies for risk management in distributed computational systems.

The discussed concepts such as interactive computation and adaptive judgment are among the basic ingredient elements in the Wisdom Technology (WisTech) [7,8]. Let us mention here the WisTech meta-equation: *wisdom = interactions + adaptive judgment + knowledge*. In particular, extension of the rough set approach on interactive computations is one of the current challenges.

6 Conclusions

In the paper, we have discussed some issues related to the development of rough sets over 30 years together with challenges for the rough set approach, especially in the environment where computations are progressing due to interactions on physical and abstract (information) granules, and where they can be controlled by performing actions activated on the basis of satisfiability to a degree of complex vague concepts modeled by approximations. Interactive computations and issues related to them are discussed in the book [8], currently under preparation.

References

1. Blake, A.: Canonical expressions in Boolean algebra. Dissertation, Dept. of Mathematics, University of Chicago. University of Chicago Libraries (1937)
2. Boole, G.: The Mathematical Analysis of Logic (1847); (reprinted by Philosophical Library). Philosophical Library (1948)
3. Boole, G.: An Investigation of the Laws of Thought (1854); (reprinted by Dover Books). Dover Books (1954)
4. Brown, F.: Boolean Reasoning. Kluwer Academic Publishers, Dordrecht (1990)
5. Chikalov, I., Lozin, V., Lozina, I., Moshkov, M., Nguyen, H.S., Skowron, A., Zielosko, B.: Three Approaches to Data Analysis. Test Theory, Rough Sets and Logical Analysis of Data. Springer (2012)
6. Cios, K., Pedrycz, W., Swiniarski, R.W., Kurgan, L.A.: Data mining: A knowledge discovery approach. Springer Science & Business Media, LLC, New York (2007)
7. Jankowski, A., Skowron, A.: A WisTech paradigm for intelligent systems. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.) Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 94–132. Springer, Heidelberg (2007)
8. Jankowski, A., Skowron, A.: Practical Issues of Complex Systems Engineering: Wisdom Technology Approach. Springer, Heidelberg (2014) (in preparation)
9. Leibniz, G.W.: Discourse on metaphysics. In: Leibniz, G.W. (ed.) Philosophical Essays, pp. 35–68 (1686)
10. Pal, S.K., Polkowski, L., Skowron, A. (eds.): Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies. Springer, Heidelberg (2004)
11. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
12. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory, Knowledge Engineering and Problem Solving, vol. 9. Kluwer Academic Publishers, Dordrecht (1991)
13. Pawlak, Z., Skowron, A.: Rudiments of rough sets; Rough sets: Some extensions; Rough sets and Boolean reasoning. Information Sciences 177(1), 3–27, 28–40, 41–73 (2007)
14. Pearl, J.: Causal inference in statistics: An overview. Statistics Surveys 3, 96–146 (2009)
15. Pedrycz, W., Skowron, S., Kreinovich, V. (eds.): Handbook of Granular Computing. John Wiley & Sons, Hoboken (2008)
16. Skowron, A., Jankowski, A., Wasilewski, P.: Risk management and interactive computational systems. Journal of Advanced Mathematics and Applications 1, 61–73 (2012)
17. Skowron, A., Suraj, Z. (eds.): Rough Sets and Intelligent Systems. Professor Zdzislaw Pawlak in Memoriam. Series Intelligent Systems Reference Library. Springer (2013)
18. Zadeh, L.A.: From computing with numbers to computing with words – from manipulation of measurements to manipulation of perceptions. IEEE Transactions on Circuits and Systems 45, 105–119 (1999)

Multi-granular Computing in Web Age

Bo Zhang[1,2,3] and Ling Zhang[4]

[1] College of Information Science & Technology
[2] Department of Computer Science & Technology
Tsinghua University, Beijing, China 100084
[3] State Key Lab of Intelligent Technology & Systems
Tsinghua National Lab for Information Science & Technology
[4] Department of Computer Science, Anhui University, Anhui, China

Abstract. In web age, the traditional information processing faces a new challenge. Due to the change of man-machine interaction modes, computers have to know the intention or interest of users. So computer information processing has to use the human brain processing principle for reference. One of its key principles is the multi-granular computing. In the talk, we will discuss the problem both from artificial intelligence and traditional information processing viewpoints. And we show that the new trend of information processing is to combine these two methods.

Keywords: Granular computing, structure mining, structured prediction, data driven, knowledge driven, deep learning.

1 Introduction

In web age, man-machine interaction mode had shifted. When a user interacts with a single computer, after a program and data have been input to the machine, it simply processes the data based on the program while needs not to know the user's intention or what the program means. When a user interacts with a web, the situation has changed. The web, a set of computers, has to know the user's intention or interest in order to provide a high quality service. Therefore, in web information processing such as information retrieval, recommendation systems, and data mining, besides computers deal with the form of information it's also needed to concern with the meaning of information.

Meaning independent underlies the traditional information processing theory. In traditional information processing the grain-size of processing units is quite small such as bag of words in text processing, colors, textures or line segments in image processing. There exists a big gap between meaning and these processing units, namely, the semantic gap. The semantic gap blocks the ability of computers to deal with the meaning of information. In order to provide a high quality service, the semantic gap should be narrowed down.

In human cognitive processing such as visual information processing, there does not have the semantic gap. How human beings to overcome the difficulty, the main

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 11–14, 2013.

strategies they adopted are multi-granular computing and the combination of data-driven and knowledge-driven methodologies. Over the last decade, a number of physiological studies in brain have established several basic facts about the cortical mechanisms of visual perception. One of the main characteristics is the hierarchical architecture, for example, from the primary visual cortex (V1) to the inferotemporal cortex (IT), there is an increase in the size of the receptive fields [1]. Namely, the visual information processing in human brain is in a hierarchical way (multi-granular computing), i.e., from the fine level (primary cortex with small receptive fields) to the coarse level (inferotemporal cortex with large receptive fields) or vice versa. In recent article [2], Yao showed that "hierarchy of information granules supports an important aspect of perception of phenomena…" As we discussed in [3], the multi-granular computing strategy is also available to human deliberative behaviors such as problem solving, planning, scheduling, etc. Hobbs [4] point out "One of the basic characteristics in human problem solving is the ability to conceptualize the world at different granularities and translate from one abstraction level to the others easily, i.e. deal with them hierarchically". It seems that the multi-granular computing is aimed at narrowing down the semantic gap and improving efficiency since coarse grain-size patterns (information) have more semantically meaningful.

1.1 Structure Mining

In order to carry out multi-granular computing, it's first needed to mine the hierarchical structures with different grain-sizes behind a huge amount of data. These structures include the contextual structures of texts, the spatial structures of images, temporal structures of speech, temporal-spatial structures of video, etc. In artificial intelligence (AI), the structures are obtained by prior knowledge generally. For example, the contextual structure of a text can be obtained by syntactic, lexical, and semantic knowledge. The disadvantage of AI methods is that it's hard to deal with the uncertainty of structures. In traditional information processing, it can be regarded as a data mining problem and can be solved by probabilistic methods. Several existed data mining methods are available in principle. But due to the difficulty of high-level features mining, only low-level features are mined generally. For example, Olshausen [5] uses a sparse coding network to mine the simple features, line segments with different orientations, from natural images that are similar to the features extracted in human's primary visual cortex V1. Deep learning is the famous method presented recently. In deep learning by using multilayer neural networks, more complex patterns can be learned effectively [6][7]. For example, Le [8] based on a 9 layers sparse deep auto-encoder, object level features such as human face, cat and human body can be learned by using unsupervised deep learning. But in order to have the results, 1.000 machines with 16,000 cores are used and it takes about 3 days to train. In [9], we using a 2 layers network, the more complex patterns similar to the patterns in human visual area V2 can be learned by a hierarchical K-means algorithm. Recently researchers pay close attention to deep learning, it's expected that deep learning may find a way to mining the multi-granular hierarchical structures behind the data. But structure mining is still a hard problem that people still need to put in lots of efforts to solve it.

1.2 Multi-level Inference

After we have had different gran-size structures (worlds), the key is to process or reason the information over different grain-size worlds. In our previous works [3], we have discussed the problem from AI viewpoint and present several multi-levels reasoning methods based on the quotient space theory. We show that in multi-granular reasoning, the homomorphism principle should be guaranteed, i.e., the results inferred from a coarse grain-size world are still available to fine grain-size worlds in a certain extent. On the other hand, the information synthesis of different grain-size worlds is also needed. The synthetic methods are also discussed in [3]. Since the elements of coarse grain-size worlds have complex structured, from the traditional information processing viewpoint, the inference over hierarchical structured data then becomes a structured prediction learning problem. Its aim is to learn a function that maps a structured input to a structured output. Therefore, other than general machine learning, the processing units of structured prediction are structures (graphs) rather than simple points (vectors). So structured prediction learning can be used to handle coarse grain size worlds. There are many well-known structured prediction learning algorithms [10]-[12] which are the expansions of general machine learning algorithms. We present a new structured prediction method called maximal entropy discrimination Markov network [13] that is the expansion of a general learning algorithm, maximal entropy discrimination learning. Using these methods, information at different grain-size worlds can be inferred.

1.3 Data-Driven and Knowledge-Driven

In AI, problems are usually solved by using symbolic reasoning and domain knowledge while the domain knowledge is represented by symbols. It's called a knowledge driven method that imitates human problem solving behaviors. The processing unit, i.e., knowledge, in knowledge driven methods is the coarsest one. It does not have semantic gap but is domain dependent and has a poor generalization capacity. Contrary, as mentioned before, in traditional information processing, the processing unit is finer. So traditional method is data driven and has a big semantic gap. The combination of these two methods means to combine the information processing at both coarse and fine levels so that the semantic gap is narrowed down and at the same time the generalization capacity is remained in a certain extent.

This is a recent trend in information processing. For example, Judea Pearl, the winner of 2011 ACM Turing award, one of his main contributions is the introduction of probability to AI. Tenenbaum, et al [14] point out the future trend of information processing is statistical inference over abstract structured declarative knowledge representation. Now, the key is how to introduce knowledge into traditional methods. Here, the knowledge includes prior knowledge, domain knowledge, the knowledge behind data, etc. Recently, there are several methods to deal with the issue. For example, in the regularized Bayesian inference [15]-[18], based on the optimization theory, the posterior regularization are added to the traditional Bayesian inference. Therefore, not only prior distribution but also posterior constraints can be considered.

And the posterior regularization may be obtained from domain knowledge or/and problem attributes so that more knowledge can join the processing process. Certainly, the research on the combination of data-driven and knowledge driven is still in the early stage. There are a lot of issues to be resolved.

References

1. Serre, T., Oliva, A., Poggio, T.: A feed-forward architecture accounts for rapid categorization. Proc. of the National Academy of Sciences (PNAS) 104(15), 6424–6429 (2007)
2. Yao, J., Vasilakos, A.V., Pedryez, W.: Granular computing: perspectives and challenges. will appears in IEEE Trans. on Cybernetics (2012)
3. Zhang, B., Zhang, L.: Theory and Application of Problem Solving. North-Holland Elsevier Science Publishers B.V. (1992)
4. Hobbs, J.R.: Granularity. In: Proc. of IJCAI, Los, Angeles, USA, pp. 432–435 (1985)
5. Olshausen, B.: Emergence of simple-sell receptive properties by learning a sparse code for natural image. Nature (1996)
6. Benjio, Y., Lamblin, P., Popovici, D., Larochelle: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems (NIPS 2006), vol. 19, pp. 153–160. MIT Press (2007)
7. Hinton, G.E., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Communication 18, 1527–1554 (2006)
8. Le, Q.V., et al.: Building high level feature using large scale unsupervised learning. In: Proc. 29th ICML, Edinburgh, Scotland, UK (2012)
9. Hu, X., Qi, P., Zhang, B.: Hierarchical K-means algorithm for modeling visual area V2 neurons. In: 19th International Conference on Neural Information Processing, Doha, Qatar, November 12-15 (2012)
10. Baum, L.E., Petrie, T.: Statistical inference for probabilistic functions of finite state Markov chains. The Annals of Mathematical Statistics 37(6), 1554–1563 (1966)
11. Lafferty, J., et al.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proc. of International Conference on Machine Learning, ICML (2001)
12. Taskar, B., et al.: Max-margin Markov networks. In: Advances in Neural Information Processing Systems (INIPS) (2003)
13. Zhu, J., Xing, E., Zhang, B.: Laplace maximum margin Markov networks. In: Proc. of International Conference on Machine Learning (ICML) (2008)
14. Tenenbaum, J.E., et al.: How to grow a mind. Science 331(6022), 1279–1285 (2011)
15. Xu, M., Zhu, J., Zhang, B.: Bayesian non-parameter max-margin matrix factorization for collaborative prediction. In: Advances in Neural Information Processing Systems, NIPS (2012)
16. Zhu, J., Chen, N., Perkins, H., Zhang, B.: Gibbs max-margin supervised topic models with fast ampling algorithms. In: ICML (2013)
17. Xu, M., Zhu, J., Zhang, B.: Fast max-margin matrix factorization with data augmentation. In: ICML (2013)
18. Zhu, J., Zheng, X., Zhou, L., Zhang, B.: Scalable inference in max-margin topic models. In: SIGKDD (2013)

Representations for Large-Scale Sequence Data Mining: A Tale of Two Vector Space Models

Vijay V. Raghavan[1], Ryan G. Benton[1], Tom Johnsten[2], and Ying Xie[3]

[1] Center for Advanced Computer Studies,
University of Louisiana at Lafayette, Louisiana, USA
{vijay,rbenton}@cacs.louisiana.edu
[2] School of Computing, University of South Alabama, Alabama, USA
tjohnsten@southalabama.edu
[3] Department of Computer Science, Kennesaw State University, Georgia, USA
yxei2@kennesaw.edu

Abstract. Analyzing and classifying sequence data based on structural similarities and differences is a mathematical problem of escalating relevance. Indeed, a primary challenge in designing machine learning algorithms to analyzing sequence data is the extraction and representation of significant features. This paper introduces a generalized sequence feature extraction model, referred to as the Generalized Multi-Layered Vector Spaces (GMLVS) model. Unlike most models that represent sequence data based on subsequences frequency, the GMLVS model represents a given sequence as a collection of features, where each individual feature captures the spatial relationships between two subsequences and can be mapped into a feature vector. The utility of this approach is demonstrated via two special cases of the GMLVS model, namely, Lossless Decomposition (LD) and the Multi-Layered Vector Spaces (MLVS). Experimental evaluation show the GMLVS inspired models generated feature vectors that, combined with basic machine learning techniques, are able to achieve high classification performance.

Keywords: Sequence Data, Classification, Feature Representation.

1 Introduction

Analyzing and classifying sequence data based on structural similarities and differences, no matter how subtle, is a mathematical problem of escalating relevance and surging importance in many different disciplines, particularly those in biology and information sciences. Characterizing patterns of all topologies at various levels of sophistication is a colossal problem lurking in the backdrop. One of the primary challenges in designing machine learning algorithms for the purpose of analyzing sequence data is the extraction and representation of significant features.

Most feature extraction methods are designed to represent sequence data based on the frequency of subsequences. For example, computational methods designed to analyze protein sequences typically represent a sequence as a set of features

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 15–25, 2013.

corresponding to the frequency of subsequences of amino acids. It is easy to realize that such a simplistic approach fails to capture the complex relationships – be it temporal, spatial, local or global – in collections of sequence data. In response, we propose a generalized sequence feature extraction model, referred to as the Generalized Multi-Layered Vector Spaces (GMLVS) model, along with two special cases of the model referred to as the Lossless Decomposition (LD) model [1] and the Multi-Layered Vector Spaces (MLVS) model [2]. The GMLVS model represents a given sequence as a collection of features in which each individual feature can be mapped to a corresponding feature vector. The GMLVS model has the flexibility to generate diverse types of feature vectors. However, the size of the set of all possible features that can be generated is huge. This fact led to the development of the LD and MLVS models, which are able to generate different types of feature vectors using a well-defined subset of features represented through the GMLVS model. We believe the resulting feature vectors have the potential of penetrating into the micro structures embedded in sequences to provide an infrastructure for various forms of analysis at the local level, while concurrently addressing global patterns over those sequences.

The rest of this paper is organized as follows. Section 2 proposes the Generalized Multi-Layered Vector Spaces Model (GMLVS) for representing sequence data. Section 3 formally defines the Lossless Decomposition (LD) model and describes its application to the problem of pair-wise sequence alignment. Section 4 formally defines the Multi-Layered Vector Spaces (MLVS) model for representing sequence data and describes its application to the classification of biological sequences. Finally, Section 5 provides a discussion and summary of the work.

2 Generalized Multi-Layered Vector Spaces (GMLVS)

The proposed GMLVS model has several significant properties that collectively have the potential to discover interesting and novel patterns from sequence data. These properties include the ability to 1) discover both local and global patterns embedded in a sequence, 2) discover patterns defined in terms of the alphabet defined over a target collection of sequences, 3) reconstruct a sequence from its model representation, and 4) facilitate both descriptive and predictive data mining tasks. We now formally present the Generalized Multi-Layered Vector Spaces model for representing sequence data.

2.1 Model Formulations

A sequence S of finite length |S| defined over a finite alphabet β is viewed as a collection of generated subsequences, β_t^*, of length t where t = 1,..., |S|-1. Let β^* denote the set of all possible subsequences.

$$\bigcup_{t=1}^{|S|-1} \beta_t^* \tag{1}$$

The set of all possible pairs of subsequences (i, j), where i and j are elements of β^* is $\beta^* \times \beta^*$. Hence, the number of possible subsequences for a given t is equal to$|\beta|^t$

and the number of possible pairs of subsequences (i, j) for all t ($1 \leq t \leq |S|$) is equal to $(\beta^*)^2 * (k + 1))$. A *feature* is defined as a pair of subsequences $f = (i, j)$, where i and $j \in \beta_t^*$, along with a specified step value m where $0 \leq m \leq k$. The parameter m stands for the number of spaces between the elements of a given feature. If m=1, then f represents a consecutive subsequence and if $m > 1$ then f is a subsequence with a gap, where the gap is filled by an arbitrary sequence of $(m - 1)$ symbols (i.e. don't care). In the latter case, subsequences i and j are called, respectively, as leading and trailing subsequence. When $m = 0$, the leading subsequence is an element from β_t^* and the trailing element is a null symbol, which takes no space (i.e. size of trailing subsequence is zero). The upper bound for parameter k is (|R|-1), where R is the maximum admissible value of m. For instance, R is equal to |S| - 1, if the feature space is represented by all pairs of symbols (i, j), where i and $j \in \beta_1^*$. It should be noted that in order for a feature $f=(i, j)$ to be valid, the sum of the length of subsequences i and j plus the value of m must be less than |S|. As a result, the number of possible features is less than or equal to the number of possible subsequences. Allowing multiple spaces between the elements of a feature generates a multitude of m-step pairs (families) $P_0, P_1, P_2, ..., P_i, \ldots, P_k$, creating a multi-layered *k-clustering* C_k made up of sets $P_{ml(i,j)}$ where m=0,1,2,...,k. In general, the size of a cluster C_k is $|\beta|^t * (k + 1)$, where t is equal to the sum of the length of the subsequences i and j. Using this notation, a sequence S can be represented by a set of features, which, in turn can be converted into a set of feature vectors. A feature is mapped into a corresponding feature vector only if it appears at least once in one of the sequences in a given collection of sequences. This fact can significantly reduce the size of the feature space. Assume S is <g, c, t, g, g, g, c, t, c, a, g, c, t, a, a, t, g, a, g, c>, t=1, and m=1. The feature (g,c), where g is the leading symbol and c is the trailing symbol, is present in the locations {1, 6, 11, 19}; this can be represented as a vector <1,6,11,10>. The resulting vector can be used to compare different sequences, or utilized to generate new representations. How this is done will be shown in the next section, which will present two specialized versions of the GMLVS model. The first model is the Lossless Decomposition Model, which corresponds to m=0 and t≥1. The second model is the Multi-Layered Vector Spaces model which corresponds to the case where m≥1 and t=2.

3 Lossless Decomposition Model

The Lossless Decomposition (LD) model creates a set of feature vectors **G** from a set of extracted features of the form $f = (i, NULL)$, where $i \in \beta_t^*$ in which $m = 0$ such that $\mathbf{G} = \{<f_p> \mid f_p$ is the starting position of the p^{th} instance of feature f in S$\}$. The resulting feature vectors **G** represent a lossless decomposition since S can be reconstructed directly from **G**. The maximum number of LD feature vectors that can be generated from a sequence S is

$$\sum_{t=1}^{|S|-1} |\beta|^t \tag{2}$$

Example-1: Given the alphabet β = {a,c,g,t}, with |β| = 4 and the sequence defined over β S=[g, c, t, g, g, g, c, t, c, a, g, c, t, a, a, t, g, a, g, c]. The following GMLVS

extracted features *a*, *gc*, and *gct* have corresponding LD generated feature vectors <10,14,15,18>, <1,6,11,19>, and <1,6,11>, respectively.

3.1 Pairwise Sequence Similarity

Measuring the degree of similarity between two sequences is an important task in several different domains. The LD model has been designed, in part, to facilitate the pairwise similarity measurement of sequences. By decomposing two sequences into a set of LD feature vectors, we are able to calculate the pairwise similarity of the sequences using parallel processes without sacrificing accuracy. For illustration purpose, we assume the feature vectors are based on GMLVS extracted features corresponding to the set β_2^* (m=0). In other words, we assume the generated feature vectors represent all possible consecutive subsequences of length two. Formally, given two sequences S1 and S2, the corresponding sets of feature vectors **G1** and **G2** are defined as follows:

G1 = {$<f_p>$ | f_p is the starting position of the p^{th} instance of feature f in S1}
G2 = {$<f_p>$ | f_p is the starting position of the p^{th} instance of feature f in S2}

Let a feature vector v1 ∈ **G1** be represented as $f_1, f_2, \ldots, f_i, \ldots, f_m$ where f_i is the i^{th} starting position of feature v1 in S1. Likewise, let a feature vector v2 ∈ **G2** be represented as $g_1, g_2, \ldots, g_i, \ldots, g_n$ where g_i is the i^{th} starting position of feature v2 in S2. We now define the distance between v1 and v2, which is denoted as *dist*(v1 , v2), to be a minimal cumulative distance calculated based on an optimal warping path between the feature vectors. The optimal warping path can be computed by the dynamic programming process, where the minimal cumulative distance $Y(f_i, g_j)$ is recursively defined as:

$$Y(f_i, g_j) = d(f_i, g_j) + min(Y(f_i-1, g_j-1), Y(f_i-1, g_j), Y(f_i, g_j-1)) \quad (3)$$

For example, assume the feature vector v1 is <0, 5, 9, 121, 130>, and the feature vector v2 is <4, 11, 100>. Then, by dynamic programming, the optimal alignment of these two vectors is illustrated in Figure 1. Then the distance between v1 and v2 can be calculated according to the optimal alignment as (4−0)+(5−4)+(11−9)+(121−100)+(130−100) = 58.

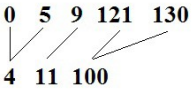


Fig. 1. Alignment between position sequences of two granules

Given the fact a feature vector represented in terms of the LD model is much shorter than the original sequence, the alignment between vectors by dynamic

programming should be much more efficient than the alignment between the original sequences. The calculation of similarity between two sequences by pairwise alignment can be distributed across individual feature vectors. For this purpose, we define the distance between the sequence S1 and the sequence S2 as the aggregation of the distances between corresponding feature vectors. Let $v1_f$ and $v2_f$ represent the feature vector corresponding to feature $f = (i, NULL)$, where $i \in \beta_t^*$, in sequences S1 and S2, respectively:

$$\text{dist(S1, S2)} = \sum_{f=(i,NULL) \in \beta_t^*} dist(v1_f, v2_f) \quad (4)$$

From this definition, the calculation of the distance between two sequences can be distributed to $|\beta|^t$ calculations of distances between $|\beta|^t$ feature vectors.

3.2 Experimental Investigation

We studied the performance of the proposed LD generated feature vectors in classifying 53 SCOP protein families. The data set of the 53 SCOP protein families can be downloaded from [11]. Each of the SCOP families contains a training data set and a testing data set as described in [3]. We simply used 1-nearest neighbor (1NN) approach to predict if a test sequence belongs to the given family or not. More specifically, for each test sequence, we evaluate its similarity with each training sequence, and then use the class label of the most similar training sequence as the label for this test sequence. The accuracy rate of the prediction for each family is reported.

We used the following approaches to evaluate similarity between two protein sequences: 1) the Needleman-Wunsch algorithm (NW) [4] ; 2) the Smith-Waterman algorithm (SW) [5]; 3) the proposed granular approach based on single amino acids (Single), and 4) the proposed granular approach based on pairs of amino acids (Pair). For NW and SW, we set the match reward to be 10 and mismatch penalty to be -8. No external scoring matrix is used for this preliminary experimental study. The classification results are summarized in Table-1.

As can be seen in Table-1, the proposed granular approach based on single amino acids reaches the same level of accuracy rate as the Needleman-Wunsch algorithm and the Smith-Waterman algorithm. In other words, the proposed granular approach is able to distribute the calculation of pairwise similarity to 20 parallel processes without sacrificing accuracy. The accuracy rate of the proposed granular approach based on pairs of amino acids is approximately 6% worse than the other three methods; however the calculation of similarity of two protein sequences under this setting can be distributed to 400 parallel processes, each of which deals with much smaller data. Therefore, this approach may be suitable for online analysis of very large scale protein sequence database, where the tradeoff between efficiency and accuracy is necessary.

Table 1. Preliminary experimental results

Protein		Accuracy	Rate %		Protein		Accuracy	Rate %	
Family	NW	SW	Single	Pair	Family	NW	SW	Single	Pair
7.3.5.2	99.4872	99.4872	99.3162	98.1766	3.32.1.11	95.7746	98.3568	98.3568	97.6526
2.56.1.2	99.3996	99.072	99.2358	99.5633	3.32.1.13	95.7478	97.5073	97.9472	97.8006
3.1.8.1	98.2691	98.5838	98.8198	99.2919	7.3.6.1	99.6599	99.4331	99.093	98.6395
1.27.1.1	99.5172	99.2414	99.5172	99.5172	7.3.6.2	98.98	98.8623	98.5092	98.9015
1.27.1.2	99.5346	99.4312	99.5863	99.4829	7.3.6.4	98.9796	98.9796	98.7755	98.3673
3.42.1.1	99.0135	98.834	98.3857	98.7443	2.38.4.1	99.1909	99.0291	98.0583	98.3819
1.45.1.2	99.1031	99.1031	99.4021	98.2063	2.1.1.1	96.9973	96.2693	96.5423	96.3603
1.4.1.1	98.8597	98.7605	98.3639	98.1656	2.1.1.2	97.0513	97.0513	97.1795	96.282
2.9.1.2	98.2697	98.4224	98.6768	99.2875	3.32.1.1	97.0052	98.0469	98.0469	97.6563
1.4.1.2	98.8588	98.7161	98.2882	98.2882	2.38.4.3	99.0441	98.6029	98.5294	98.3824
2.9.1.3	99.0014	98.5735	98.2882	99.2867	2.1.1.3	96.8198	96.4664	96.4664	96.4664
1.4.1.3	98.8593	98.4791	99.2395	98.4791	2.1.1.4	96.6667	96.6667	97.4359	96.837
2.44.1.2	94.4224	92.2905	94.4968	82.4988	2.38.4.5	99.1015	98.832	98.4726	98.6523
2.9.1.4	98.6458	98.4319	99.0021	99.1447	2.1.1.5	96.8652	96.3427	95.9247	96.8652
3.42.1.5	99.0345	98.6207	98.4828	98.8276	7.39.1.2	99.3794	99.2908	98.9361	99.0248
3.2.1.2	98.4768	97.3343	97.639	98.4006	2.52.1.2	99.4531	99.6094	99.4531	99.5313
3.42.1.8	99.1023	98.9228	97.666	99.1023	7.39.1.3	99.3351	99.2465	99.0691	99.1135
3.2.1.3	97.835	97.1583	98.2409	98.1055	1.36.1.2	99.1726	99.1726	98.818	98.9362
3.2.1.4	97.561	96.6899	97.561	98.0836	3.32.1.8	96.2687	98.1876	98.1876	97.1215
3.2.1.5	98.6063	96.8641	97.7352	98.2578	1.36.1.5	99.1728	98.791	98.1864	98.6637
3.2.1.6	97.0732	96.5854	97.0732	97.561	7.41.5.1	99.5556	99.5062	99.308	99.4074
2.28.1.1	97.6683	97.215	97.215	98.1865	7.41.5.2	99.5556	99.4074	99.0617	99.5556
3.3.1.2	98.5712	98.7619	98.5714	98	1.41.1.2	98.8728	99.1948	98.5507	98.0676
3.2.1.7	97.561	97.8049	97.561	98.5366	2.5.1.1	99.4483	99.1976	99.2477	99.2477
2.28.1.3	98.3373	98.3373	98.5748	98.0998	2.5.1.3	99.3933	99.2278	99.3932	99.1338
3.3.1.5	97.8342	97.8342	98.7922	99.0837	1.41.1.5	98.9954	98.609	98.3771	98.493
7.3.10.1	98.5592	97.5454	96.2913	95.7044	Average of All	98.37638302	98.2450566	98.26318491	98.06838

4 Multi-Layered Vector Spaces Model

The Multi-Layered Vector Spaces Model (MLVS) creates a set of feature vectors **G** based on GMLVS features of the form (i,j), where i and $j \in \beta_1^*$. The total number of feature vectors that can be generated from an alphabet β is $|\beta|^2$. In this specialized case, a sequence S is viewed to have a multi-layered structure made up of a set of m-step ordered pairs (features) (i,j), where i and $j \in \beta_1^*$, denoted by $P_{ml(i,j)}$, where $1 \leq m \leq k$. Ordered pairs made up of consecutive elements of the sequence are said to form the family of 1-step (one-step) pairs, $P_{1l(i,j)}$. The concept of a multi-layered k-clustering C_k, as defined in the context of the GMLVS model, also applies to the MLVS model. Thus, the MLVS model views a sequence S as the as the union of all ordered pairs (i,j), where i and $j \in \beta_1^*$ at k distinct layers. The following example demonstrates how the said structures are built.

Example-2: Given the alphabet β = {a,c,g,t}, with |β|=4, $|\beta|^2$ =16, and the sequence S = [g, c, t, g, g, g, c, t, c, a, g, c, t, a, a, t, g, a, g, c]. The following are sample m-step pairs (β_1^*): 1-step ordered pairs for (g,c) are located at step locations [1,2], [6,7], [11,12], and [19,20]; 1-step ordered pairs for (g,g) are located at step locations [4,5], and [5,6]; 2-step ordered pairs for (g,t) are located at step locations [1,3], [6,8], and [11,13]; 4-step ordered pairs for (c,g) are located at step locations [2,6], and [7,11].

4.1 Feature Vector Creation

For a selected value of *m* and a given GMLVS extracted feature $f = (i,j)$ $(i, j \in \beta_1^*)$, the sequence of anchor positions is taken as forming the scalar components of an n-dimensional feature vector $\mathbf{V_{m|(i,j)}}$ associated with the feature (i,j). The union of such vectors for all features (for a given *m*) forms a vector cluster $\check{\mathbf{Z}}_\mathbf{m}$ at step size *m*, providing a single-step representation for the sequence.

$$\check{Z}_m = \bigcup_{(i,j)} V_{m|(i,j)} \tag{5}$$

The union of vector clusters $\check{\mathbf{Z}}_\mathbf{m}$ provides a multi-layered feature vector space $\check{\mathbf{Z}}_\mathbf{k}$, one layer for each value of *m*, for the original sequence.

$$\check{Z}_k = \bigcup_m \bigcup_{(i,j)} V_{m|(i,j)} \tag{6}$$

Feature vectors for each *m*-step feature can be structured in at least two different ways. One approach is to simply record the step (spatial index) locations of anchor positions as Boolean values (1, 0). This approach is suitable for collections of equal length sequences. An alternative approach is to partition a sequence into *n* equal segments and record the number of anchor positions that fall into each segment. The number of segments *n* will determine the dimension of the vectors thus formed. The size of *n* can be adjusted to meet restrictions or expectations on resolution and accuracy. This approach has the advantage of mapping sequences of unequal length into fixed length feature vectors. For a given *m*, the construction scheme for $\mathbf{V_{m|(i,j)}}$ can be implemented in two different ways: a vector can be constructed for each feature in the sequence to generate a vector cluster over the whole sequence, or feature vectors in the cluster are concatenated into a single vector to represent the entire sequence. The steps involved in the feature mapping process are illustrated in Fig. 2. As is the case with LD feature vectors, MLVS feature vectors can also be analyzed in a distributed manner. In particular, MLVS feature vectors can be processed in parallel based on either specific sets of ordered pairs (i, j) and / or range of step sizes (m).

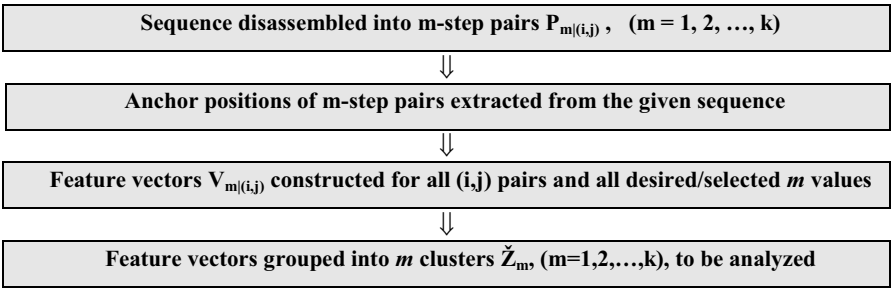


Fig. 2. Proposed feature mapping process

Example-2: Using the same alphabet and sequence as used in the previous examples, the following are sample feature vectors for a select group of m-step MLVS features:

Anchor positions of 1-step feature (g,c) are located at step (index) locations [1,6,11,19]; vector $\mathbf{V}_{1|(gc)}$, is represented by the Boolean feature vector <1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,0> if step locations for the anchors are used directly as vector components. If we instead partition the sequence into 4 equal segments (n = 4), the vector $\mathbf{V}_{1|(gc)}$, is represented by the 4D feature vector <1,1,1,1> with vector components representing the number of anchor elements in each segment; anchor positions of the 1-step feature (g,g) are located at step (index) locations [4,5]; vector $\mathbf{V}_{1|(gg)}$ is represented by the Boolean feature vector <0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0> or by the 4D vector <2,0,0,0>; anchor positions of 2-step feature (g,t) are located at step (index) locations [1,6,11]; vector $\mathbf{V}_{2|(gt)}$ is represented by the Boolean vector <1,0,0,0,0,1 ,0,0,0,0,1,0,0,0,0,0,0, 0,0,0> or by the 4D vector <1,1,1,0>.

4.2 Experimental Investigation

Experiments were conducted to determine the potential usefulness of the MLVS generated feature vectors in classifying biological sequences. Specific objectives included: investigating the accuracy of classifiers constructed from various n-dimensional feature vectors $\mathbf{V}_{ml(i,j)}$; and, the accuracy of ensemble classifiers constructed from individual vector clusters $\check{\mathbf{Z}}_m$. The results obtained from these classifiers were compared with results obtained from the (k,m)-mismatch kernel method [6,7].

The biological sequences utilized in the experiments corresponded to the classification of the 3PGK-DNA sequences, Eukaryota vs. Euglenozoa [8]. There were a total of forty-three instances belonging to the class Eukaryota and forty-four instances belonging to the class Euglenozoa. The alphabet β consisted of the elements {a,c,g,t}. Each instance was mapped into the following vector clusters $\check{\mathbf{Z}}_1$, $\check{\mathbf{Z}}_2$, $\check{\mathbf{Z}}_3$, and $\check{\mathbf{Z}}_{10}$. For the experiments, we set n=100; that is, we segmented each $\mathbf{V}_{ml(i,j)}$ into 100 equal segments. In addition, we arbitrarily selected the step sizes m=1,2,3, and 10. We utilized the decision tree classifier C4.5 [9] as implemented in the Weka data mining application [10]. The performance of the decision trees was evaluated using the hold-out method in which the feature vectors, $\mathbf{V}_{ml(i,j)}$, for a given GMLVS feature f = (i, j) (i, j$\in$ β_1^*), were randomly divided into five pairs of training and test sets. The reported performance is the average accuracy over five runs.

The results of the experiments are shown in Tables 2 and 3. Table-2 shows the accuracy of the decision trees constructed from the feature vectors for each ordered pair feature. For instance, the decision tree constructed from the feature vectors corresponding to the ordered pair (a,a) has an estimated predicted accuracy of 75%, 82%, 75%, and 69% with respect to step sizes 1, 2, 3, and10, respectively. The results show for the selected step sizes, the decision trees are performing better than random guessing but not at a desired level. A significant improvement in performance is obtained from the use of ensemble (multiple) classifiers constructed from decision trees belonging to a single vector cluster. Table-3 shows the accuracy values obtained by combining multiple decision trees at step sizes 1, 2, 3, and 10. The grouping of classifiers into ensembles was based on the accuracy of individual decision trees constructed from single ordered pairs. Specifically, for a given step size, the decision

trees were selected based on accuracy and the *r* most accurate decision trees were combined to form an ensemble of size *r*. The decision trees of a given ensemble were combined using *un-weighted* majority voting. Several of the constructed ensemble classifiers shown in Table-3 have a high degree of accuracy, and in particular the ensemble classifier consisting of fifteen decision trees at step size m=1 (15:96) has a 96% level of accuracy.

Table 2. Decision tree accuracy values for selected feature vectors

$V_{m\|(i,j)}$	m=1	m=2	m=3	m=10
(a,a)	75	82	75	69
(a,c)	77	63	69	64
(a,g)	75	89	83	75
(a,t)	77	78	82	71
(c,a)	69	68	71	67
(c,c)	76	75	82	87
(c,g)	64	78	74	67
(c,t)	76	68	77	67
(g,a)	70	78	74	72
(g,c)	75	67	82	82
(g,g)	70	66	84	85
(g,t)	76	64	69	76
(t,a)	66	68	72	67
(t,c)	87	75	74	70
(t,g)	72	70	67	67
(t,t)	76	61	75	72
Average	74	72	76	72

Table 3. Ensemble decision tree accuracy values for selected vector clusters

m	# Classifiers : Accuracy (%)
1	3:90; 5:93; 7:93; 9:94; 11:92; 13:92; **15:96**
2	3:90; 5:87; 7:87; 9:90; 11:87; 13:83; 15:79
3	3:87; 5:92; 7:91; 9:92; 11:93; 13:94; 15:94
10	3:92; 5:90; 7:92; 9:92; 11:90; 13:89; 15:87

Table 4. (k,m)-mismatchmethod accuracy values

K	m = 0 (%)	m=1 (%)
4	90	89
5	93	88
6	93	90
7	91	93
8	91	93
9	90	91
10	86	90

To evaluate the results recorded in Tables-2 and -3, we repeated the experiments using the (k,m)-mismatch kernel method. Specifically, the five pairs of training and test sets were evaluated using the (k,m)-mismatch method as implemented by the authors of [6,7]. Table-4 shows the classification accuracy results, averaged over the five runs, for contiguous subsequences of length k = 4, 5, …, 10 and zero or one mismatches (m). The maximum achieved accuracy was 93%, which is less than the 96% accuracy value obtained through the use of the proposed multi-layer vector space model. In addition, the comprehensibility of a decision tree classifier is, in general, much greater as compared to SVM classifiers (i.e. (k,m)-mismatch method). This difference is significant if one wishes to obtain a deep characterization of a collection of biological sequences.

5 Discussion and Summary

It is anticipated that the transparent quality, simplicity and therefore the interpretation of the feature extraction models discussed in this paper will shed light into the inner workings of the system being studied. The Generalized Multi-Layered Vector Spaces (GMLVS) model allows an investigator to map a collection of sequences into a very large space of feature vectors for the purpose of analyzing and classifying data. The generated feature vectors can be logically partitioned along multiple dimensions based on sets of specific GMLVS features (i, j) (i and $j \in \beta_t^*$) and/or specific step values m ($0 \leq m \leq k$). We believe a large feature vector space whose vectors can be partitioned into semantically related groups will provide a user-friendly mathematical habitat in which an investigator can discover the intrinsic elements of the system being studied such as the plausibility of interactions among micro patterns and causal connections embedded in a sequence. More generally, an investigator has the opportunity to discover relationships among various groups of feature vectors and to discover characteristics of the feature space as the step values (m) are increased to their limit.

We have also developed two related sequential data models, referred to as the Lossless Decomposition (LD) model and the Multi-Layered Vector Spaces (MLVS) model. These two models are able to generate different types of feature vectors using a well-defined subset of features represented through the GMLVS model. Preliminary experimental results reported on in this paper indicate both the LD and MLVS models have the capability to identify important relationships within individual sequences.

In the future, we plan to explore the utility of GMLVS (and specialized cases) in a variety of ways. One area of study is to explore the applicability of the GMLVS for signal peptide prediction; that is, to identify sections of amino acids that used to direct nascent, or newly formed, proteins to their correct locations. Moreover, we believe the GLMVS format (or a derivative) could be used to create human-interpretable rules; this is something currently lacking in of the current signal peptide detection techniques. Second, we also wish to explore the applicability of the MLVS model, combined with association mining, to detect potential mutations and frequent co-occurrences of mutations within cancer cells. Third, we are interested in exploring

how to incorporate external scoring matrices into the LD model, along with developing adaptive search methods to exploit the LD representation. Finally, the MLVS and LD methods represent only two special cases of the GMLVS model; developing complementary special case models may yield additional advantages and insights.

References

1. Xie, Y., Fisher, J., Raghavan, V.V., Johnsten, T., Akkoc, C.: Granular approach for protein sequence analysis. In: Yao, J., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 414–421. Springer, Heidelberg (2012)
2. Akkoç, C., Johnsten, T., Benton, R.: Multi-layered vector spaces for classifying and analyzing biological sequences. In: BICoB, pp. 160–166 (2011)
3. Liao, L., Noble, S.: Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. Journal of Computational Biology, 857–868 (2003)
4. Needleman, B., Wunsch, D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 443–453 (1970)
5. Smith, F., Waterman, S.: Identification of common molecular subsequences. Journal of Molecular Biology, 195–197 (1981)
6. Leslie, C., Eskin, E., Noble, W.S.: The spectrum kernel: A string kernel for SVM protein classification. In: Pacific Symposium on Biocomputing, pp. 564–575 (2002)
7. Leslie, C., Eskin, E., Weston, J., Noble, W.S.: Mismatch string kernels for SVM protein classification. In: Neural Information Processing Systems, pp. 1441–1448 (2003)
8. Sonego, P., Pacurar, M., Dhir, S., Kertesz-Farkas, A., Kocsor, A., Gaspari, Z., Leunissen, J., Pongor, S.: A protein classification benchmark collection for machine learning, D232-D236 (2007)
9. Quinlan, J.: C4.5: Programs for machine learning. Morgan Kaufmann (1993)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDDD Explorations, 10–18 (2009)
11. Supplementary data (from paper [3]), `http://noble.gs.washington.edu/proj/svm-pairwise/`

Individual Approximate Clusters: Methods, Properties, Applications

Boris Mirkin

Division of Applied Mathematics
Higher School of Economics, Moscow, Russian Federation
and
Department of Computer Science and Information Systems,
University of London, Birkbeck, United Kingdom
bmirkin@hse.ru

Abstract. A least-squares data approximation approach to finding individual clusters is advocated. A simple local optimization algorithm leads to suboptimal clusters satisfying some natural tightness criteria. Three versions of an iterative extraction approach are considered, leading to a portrayal of the cluster structure of the data. Of these, probably most promising is what is referred to as the incjunctive clustering approach. Applications are considered to the analysis of semantics, to integrating different knowledge aspects and consensus clustering.

1 Individual Clusters in Graph Theory and Clustering

In spite of the ubiquitous use of partitions and hierarchies as the only two cluster structures of interest (see, for example, [8]), individual clusters are prominent in the analysis of similarity data from the start. Intuitively, cluster is a set of highly similar entities that are dissimilar from entities outside of the cluster.

Currently, the most popular format for similarity data is of square matrix $A = (a_{ij})$ of pair-wise indices a_{ij} expressing similarity between entities $i, j \in I$. The greater the value of a_{ij}, the greater the similarity between i and j. Some examples of similarity data are (i) individual judgements of similarity expressed using a fixed range, (2) correlation coefficients between variables or time series, (3) graphs represented by 1/0-similarity matrices, (4) weighted graphs, or networks, (5) probabilities of common ancestry, especially in proteomics, (6) affinnity data obtained by transformation of distances using a Gaussian or another kernel function. Consider an example of a data set of this type.

Eurovision Song Contest Scoring
Table 1 presents the average scores given by each country to her 10 top choices at the Eurovision song contests (up to and including year 2011). I compiled this using public data at http://www.escstats.com/ (visited 28/2/2013). Each row of the table corresponds to one out of selected nineteen European countries, and assigns a non-zero score to those of the other eighteen that have been among the 10 best choices. The cluster structure of the table should quantify to what extent the gossip of the effects of cultural and ethnical links on voting is justified, because the quality of songs and performances may be considered random from year to year, so that in the ideal case when no cultural preferences are involved at evaluations, the similarity matrix should be of a random structure too.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 26–37, 2013.

Table 1. Eurovision scoring: Each row contains the average score given by the row country to the column country in Eurovision song contests (multiplied by 10)

Country	Az	Be	Bu	Es	Fr	Ge	Gr	Is	It	Ne	Pol	Por	Ro	Ru	Se	Sp	Sw	Ukr	UK
1 Azerbajan	0	0	0	0	0	0	61	48	0	0	0	0	50	65	0	0	0	90	0
2 Belgium	38	0	0	0	0	39	40	0	0	47	0	0	0	0	0	34	0	0	42
3 Bulgaria	67	0	0	0	0	0	93	0	0	0	0	0	0	48	60	0	0	44	0
4 Estonia	41	0	0	0	0	0	0	0	43	0	0	0	0	88	0	0	0	43	0
5 France	0	37	43	0	0	0	0	56	47	0	0	54	0	0	80	0	0	0	41
6 Germany	0	0	0	0	34	0	37	35	0	0	55	0	0	0	70	0	0	0	42
7 Greece	54	0	80	0	41	0	0	0	0	0	0	0	40	0	80	44	0	38	0
8 Israel	50	0	0	0	0	0	0	0	0	43	0	0	66	74	50	0	0	62	43
9 Italy	0	0	100	0	54	0	0	0	0	0	0	0	120	0	0	0	0	65	52
10 Netherlands	39	46	0	0	0	38	0	45	0	0	0	0	0	0	70	0	0	0	0
11 Poland	84	43	0	39	0	0	0	0	90	0	0	0	0	0	0	0	0	82	0
12 Portugal	0	35	0	0	0	45	0	41	81	0	0	0	52	0	57	42	0	74	43
13 Romania	52	0	0	0	0	0	82	0	60	0	0	0	0	49	80	0	0	35	0
14 Russia	99	0	0	0	0	0	37	36	0	0	0	0	0	0	80	0	0	77	0
15 Serbia	0	0	53	0	0	0	73	0	0	0	0	0	0	44	0	0	0	44	0
16 Spain	0	0	78	0	0	51	45	0	74	0	0	43	79	0	47	0	0	46	0
17 Switzerland	0	0	0	0	44	0	0	42	47	0	0	0	0	0	106	41	0	0	41
18 Ukraine	111	0	0	0	0	0	0	0	0	0	60	0	0	98	90	0	0	0	0
19 UK	0	0	0	36	0	39	38	0	0	0	0	0	0	0	37	0	0	0	0

There are several individual cluster related graph-theoretic concepts: (a) *connected component* (a maximal subset of nodes in which there is a path connecting each pair of nodes), (b) *bicomponent* (a maximal subset of nodes in which each pair of nodes belongs to a cycle), and (c) *clique* (a maximal subset of nodes in which each pair of nodes is connected by an edge). Even more relevant is a more recent concept of (d) the *maximum density subgraph* [5]. The density $g(S)$ of a subgraph $S \subset I$ is the ratio of the number of edges in S to the number of elements $|S|$. For an edge weighted graph with weights specified by the matrix $A = (a_{ij})$, the density of a subgraph on $S \subseteq I$ $g(S)$ is defined by the *Rayleigh quotient* $s^T As/s^T s$, where $s = (s_i)$ is the characteristic vector of S, viz. $s_i = 1$ if $i \in S$ and $s_i = 0$ otherwise. The maximum value of the Raleigh quotient of a symmetric matrix over any real vector s is equal to the maximum eigenvalue and is attained at an eigenvector corresponding to this eigenvalue. This gives rise to the so-called (e) *spectral clustering*.

Cluster-specific individual cluster concepts include those of B-cluster [7] and Apresian's cluster [1].

2 Approximation Models for Summary and Semi-average Criteria

2.1 Least-Squares Approximation

The idea is to find such a subset $S \subseteq I$ that its binary matrix $s = (s_{ij})$ approximates a given symmetric similarity matrix A as close as possible. To take into account the difference in the unit of measurement of the similarity as well as for its zero point, matrix

s should be also supplied with (adjustable) scale shift and rescaling coefficients, say λ and μ. That would mean that the approaximation is sought in the set of all binary $\lambda + \mu$ / μ matrices $\lambda s + \mu$ with $\lambda > 0$. Unfortunately, such an approximation, at least when follows the least squares approach, would have little value as a tool for producing a cluster, because the optimal values for λ and μ would not separate the optimal S from the rest [10,11]. This is why this author uses only one parameter λ, change of the unit of measurement, in formulating approximation problems in clustering. The issue of adjustment of similarity zero point, in such a setting, is moved out of the modeling stage to the data pre-processing stage. This amounts to substraction of a similarity shift value from all the similarity values before doing data analysis. Choice of the similarity shift value may affect the clustering results, which the user can take advantage of to differently contrast within- and between- cluster similarities. In the remainder, it is assumed that a similarity shift value has been subtracted from all the similarity entries. Another assumption, for the sake of simplicity, is that the diagonal entries a_{ii} are all zero (after the pre-processing step). From now on, S is represented by a vector $s = (s_i)$ such that $s_i = 1$ if $i \in S$ and $s_i = 0$, otherwise. Our approximation model is

$$a_{ij} = \lambda s_i s_j + e_{ij} \tag{1}$$

where a_{ij} are the preprocessed similarity values, $s = (s_i)$ is the unknown cluster belongingness vector and λ, the rescaling value, also referred to as the cluster intensity value. To fit the model (1), only the least squares criterion $L^2 = \sum_{i,j \in I} e_{ij}^2$ is considered here.

Pre-specified Intensity. We first consider the case in which the intensity λ of the cluster to be found is pre-specified. Since $s_i^2 = s_i$ for any 0/1 variable s_i, the least squares criterion can be expressed as

$$L^2(S,\lambda) = \sum_{i,j \in I} (a_{ij} - \lambda s_i s_j)^2 = \sum_{i,j \in I} a_{ij}^2 - 2\lambda \sum_{i,j \in I} (a_{ij} - \lambda/2) s_i s_j \tag{2}$$

Since $\sum_{i,j} a_{ij}^2$ is constant, for $\lambda > 0$, minimizing (2) is equivalent to maximizing the summary within-cluster similarity after subtracting the threshold value $\pi = \lambda/2$, i.e.,

$$f(S,\pi) = \sum_{i,j \in I} (a_{ij} - \pi) s_i s_j = \sum_{i,j \in S} (a_{ij} - \pi). \tag{3}$$

This is the so-called summary similarity criterion which satisfies the following properties:

Statement 1. *A cluster S optimizes criterion (3) over similarity matrix A if and only if S optimizes it over symmetric similarity matrix $A + A'$.*

Statement 2. *The optimal cluster size according to criterion (3) can only decrease when π grows.*

One more property of the criterion is that it leads to provably tight clusters. Let us refer to cluster S as suboptimal if, for any entity i, the value of criterion (3) can only decrease if i changes its state in respect to S. Entity i changes its state in respect to S if it is added to S, in the case that $i \notin S$, or removed from S if $i \in S$.

Statement 3. *If S is a suboptimal cluster, then the average similarity $a(i,S)$ of i with other entities in S is greater than π if $i \in S$, or less than π if $i \notin S$.*

An algorithm for producing a suboptimal cluster S starting from any entity i by adding/removing a single entity can be drawn using property:

$$\Delta(S,k) = f(S \pm k) - f(S) = -2z_k \sum_{i \in S} a_{ik}, \tag{4}$$

under the assumption that the diagonal similarities a_{ij} are not considered and z_k in (4) corresponds to S, that is, taken before the change of sign.

Optimal Intensity. When λ in (2) is not fixed but can be adjusted to further minimize the criterion, it is easy to prove that the optimal λ is

$$\lambda = a(S) = s^T A s / [s^T s]^2, \tag{5}$$

where $a(S)$ is the average within cluster S similarity.

By putting this equation in the least-squares criterion (2), one can prove:

$$L^2(S) = (A,A) - [s^T A s / s^T s]^2, \tag{6}$$

which implies that the optimal cluster S is a maximizer of

$$g^2(S) = [s^T A s / s^T s]^2 = a^2(S)|S|^2 \tag{7}$$

According to (7), the maximum of $g^2(S)$ may correspond to either positive or negative value of $a(S)$. The focus here is on maximizing (7) only for positive $a(S)$. This is equivalent to maximizing its square root, that is the Rayleigh quotient,

$$g(S) = s^T A s / s^T s = a(S)|S| \tag{8}$$

This criterion is a form of the so-called semi-average clustering criterion which has a number of properties similar to those of the summary similarity criterion. In particular a cluster tightness property is:

Statement 4. *If S is a suboptimal cluster, then the average similarity $a(i,S)$ of i with other entities in S is greater than $a(S)/2i$ if $i \in S$, or less than $a(S)/2$ if $i \notin S$.*

An algorithm for producing a suboptimal cluster S starting from any $i \in I$ can be drawn by selecting such an entity i whose adding to S if $i \notin S$ or removal from S if $i \in S$ makes the greatest increment of criterion (8).

2.2 Partitional, Additive and Incjunctive Clusters: Iterative Extraction

The approximation model can be extended to a set of (not necessarily disjoint) similarity clusters $S_1, S_2, ..., S_K$:

$$a_{ij} = \biguplus_{k=1}^{K} \lambda_k s_i^k s_j^k + e_{ij}, \quad \text{for } i,j \in I, \tag{9}$$

where $s^k = (s_i^k)$ and λ_k are k-th cluster belongingness vector and the intensity. The symbol $\biguplus$ denotes an operation of integration of the binary values together with their intensities. We consider three versions of the operation: (a) additive clusters: $\biguplus$ is just summation; (b) partitional clusters: $\biguplus$ denotes the fact that clusters are disjunct, no overlapping; (c) incjunctive clusters: $\biguplus$ is maximum over $k = 1, 2, ..., K$, that is, operation of inclusive disjunction.

The goal is to minimize the residuals e_{ij} with respect to the unknown relations R^k and intensities λ_k.

Additive cluster model was introduced, in the English language literature, by Shepard and Arabie in [18], and independently, and even earlier, in a more general form embracing other cluster structures as well, by the author in mid-seventies in Russian ([10], see references in [11]). Incjunctive clusters have not been considered in the literature, to our knowledge.

We maintain that cluster structures frequently are similar to that of the Solar system so that clusters hidden in data much differ with respect to their "contributions". We proposed an iterative extraction method [10] to find clusters one by one (see also [11,13]). Depending on the setting, that is, meaning of $\biguplus$ in (9), one may use the following options:

i **Additive clusters.** The iterative extraction works as this:
   (a) Initialization. Given a preprocessed similarity matrix A, compute the data scatter $T = (A,A)$. Put $k = 0$.
   (b) General step. Add 1 to k. Find cluster S (locally) maximizing criterion $g(S)$ in (8). Output that as S_k, the intensity of this cluster, the within-cluster average $a(S)$ as λ_k, and its contribution to the data scatter, $w_k = a(S)^2|S|^2$.
   (c) Test. Check a stopping condition. If it does hold, assign $K = k$ and halt. Otherwise, compute the residual similarity matrix as $A - \lambda_k s_k s_k^T$ and go back to General step with the residual matrix as A.

   The stopping condition can be either reaching a prespecified number of clusters or contribution of the individual cluster has become too small or the total contribution of the so far found clusters has become too large. The individual cluster contributiona are additive in this process. Moreover, the residual matrix in this process tends to 0 when k increases [10,11].

ii **Partitional clusters.** This method works almost like the iterative extraction at the additive clustering model, except that here no residual matrix is considered, but rather the found clusters are removed from the set of entities.

iii **Incjunctive clusters.** Make a loop over $i \in I$. Run the semi-average criterion suboptimal algorithm at $S = \{i\}$ for each i. Remove those of the found clusters that overlap with others too much. This can be done by applying the same algorithm to the cluster-to-cluster similarity matrix; entries in this matrix are defined as proportional to the overlap values. The individual cluster over this matrix contains those clusters that overlap too much - only one of them shoild be left.

For an example, let us apply each of these three strategies to the Eurovision matrix, preliminarily made symmetric with zeroed diagonal entries.

a Additive clusters one by one: With the condition to stop when the contribution of an individual cluster becomes less than 1.5% of the total data scatter, the algorithm

Table 2. Additive clusters found at the Eurovision song contest dataset

n.	Cluster	Intensity	Contribution, %
1	Azerbajan, Bulgaria, Greece, Russia, Serbia, Ukraine	70.0	21.43
2	Azerbajan, Israel, Romania, Russia, Ukraine	49.5	7.13
3	Bulgaria, Greece, Italy, Romania, Spain	46.8	6.38
4	Azerbajan, Poland, Ukraine	66.8	3.90
5	Italy, Portugal, Romania	53.0	2.46
6	Greece, Romania, Serbia	43.7	1.67

found, in addition to the universal cluster I with the intensity equal to the similarity average, six more clusters (see Table 2). We can see that, say, pair Azerbajan and Ukraine belong to three of the clusters and contribute, therefore, the summary intensity value 70.0+49.5+66.8=186.3 as the "model" similarity between them (the summary similarity between them in Table 1 is 201).

b Partitional clusters one by one. Here the algorithm is run on the entities remaining unclustered after the previous step (see Table 3).

Table 3. Partitional clusters found one-by-one at the Eurovision song contest dataset

n.	Cluster	Intensity	Contribution, %
1	Azerbajan, Bulgaria, Greece, Russia, Serbia, Ukraine	70.0	21.43
2	Italy, Portugal, Romania, Spain	56.1	5.50
3	Belgium, Netherlands	57.3	0.96
4	Germany, UK	45.3	0.60
5	France, Israel, Switzerland	11.6	0.12
6	Estonia, Poland	3.3	0.00

There are only two meaningful clusters, East European and Latin South European, in Table 3; the other four contribute too little. The first of the clusters is just a replica of that in the additive clustering computation. Yet the second cluster combines clusters 3 and 5 cleaned from the Balkans in the additive clusters results Table 2.

c Incjunctive clusters from every entity. The semi-average algorithm has been applied starting from $S = \{i\}$ for every $i \in I$. Most of the final clusters coincide with each other, so that there are very few different clusters (see Table 4).

According to the data recovery model, these clusters lead to a recovered similarity matrix as follows: first of all, the subtracted average value, 35.72, should be put at every entry. Then the two entries of Belgium/Netherlands link are to be increased by the intensity of cluster 2, 57.3. Similarly, the intensities of clusters 1 and 4 are to be added for any pair of entities within each. Then entries for pairs from cluster 3 are to be changed for 35.7+110.6=146.3.

This is an example at which the local nature of the algorithm is of an advantage rather than a drawback. Clusters in Table 4 reflect cultural interrelations rather than anything else.

Table 4. All four different incjunctive clusters found at the Eurovision song contest dataset starting from every entity

Cluster	Intensity	Contribution, %
1. Azerbajan, Bulgaria, Greece, Russia, Serbia, Ukraine	70.0	21.43
2. Belgium, Netherlands	57.3	0.96
3. Bulgaria, Greece, Serbia	110.6	10.7
4. Italy, Portugal, Romania, Spain	56.1	5.50

3 Applications

3.1 Semantics of Domain-Specific Nouns

The idea that semantics of domain-specific nouns lies in their relation to specific situations, functions, etc., a few decades back was not that obvious in cognitive sciences as it is now. In the absence of Internet, the researchers used the so-called sorting experiments to shed light on semantics of domain specific nouns [16,4]. In a sorting experiment, a set of domain-specific words is specified and written down, each on a small card; a respondent is asked then to partition cards into any number of groups according to their perceived similarity among the nouns. Then. a similarity matrix between the words can be drawn so that the similarity score between two words is defined as the number of respondents who put them together in the same cluster. A cognitive scientist may think that behind the similarity matrix can be some "additive" elementary meanings. In the analysis of similarities between 72 kitchenware terms, the iterative one-by-one extraction with the semi-average similarity suboptimal algorithm found that the clusters related to the usage only: (i) a cooking process, such as frying or boiling; (ii) a common consumption use, such as drinking or eating, and (iii) a common situation such as a banket [4]. In contrast to expectations, none of the clusters reflected logical or structural similarities between the kitchenware items.

3.2 Determining Similarity Threshold by Combining Knowledge

In [14] partitional clusters of protein families in herpes viruses are found. The similarity between them is derived from alignments of protein amino acid sequences and similarity neigbourhoods. At different similarity shifts, different numbers of clusters can be obtained, from 99 non-singleton clusters (of 740 entities) at the zero similarity shift to only 29 non-singleton clusters at the shift equal to 0.97 [14]. To choose a proper value of the shift, external information can be used – of functional activities of the proteins under consideration in [14]. Although function of most proteins under consideration was unknown, the set of pairs of functionally annotated proteins can be used to shed light onto potentially admissible values of the similarity shift. In each pair, the proteins can be synonymous (sharing the same function) or not. Because of a high simplicity of virus genomes, the synonymous proteins should belong in the same aggregate protein family, whereas proteins of different functions should belong in different protein families. The similarity shift value should be taken as that between the sets of similarity values for synonymous and nonsynonymous proteins. Then, after subtraction of

this value, similarities between not synonymous HPFs get negative while those between synonymous HPFs remain positive. In [14] no non-synonymous pair has a greater *mbc* similarity than 0.66, which should imply that the shift value 0.67 confers specificity for the production of aggregate protein families. Unfortunately, the situation is less clear cut for synonymous proteins: although the similarities between them indeed are somewhat higher, 24% pairs is less than 0.67. To choose a similarity shift that minimizes the error in assigning negative and positive similarity values, one needs to compare the distribution of similarity values in the set of synonymous pairs with that in the set of non-synonymous pairs and derive the intersection point similarity value (see details in [14]).

3.3 Consensus Clustering

Consensus clustering is an activity of summarizing a set of clusterings into a single clustering. This has become popular recently because after applying different clustering algorithms, or the same algorithm at different parameter settings, on a data set, one gets a number of different solutions. Consensus clustering seeks a unified cluster structure behind the solutions found (see, for example, [21,13]). Here some results of applying an approach from Mirkin and Muchnik [15] in the current setting will be reported (see also [13]).

Consider a partition $S = \{S_1, ..., S_K\}$ on I and corresponding binary membership $N \times K$ matrix $Z = (z_{ik})$ where $z_{ik} = 1$ if $i \in S_k$ and $z_{ik} = 0$, otherwise ($i = 1, ..., N, k = 1, ..., K$). Obviously, $Z^T Z$ is a diagonal $K \times K$ matrix in which (k,k)-th entry is equal to the cardinality of S_k, $N_k = |S_k|$. On the other hand, $ZZ^T = (s_{ij})$ is a binary $N \times N$ matrix in which $s_{ij} = 1$ if i and j belong to the same class of S, and $s_{ij} = 0$, otherwise. Therefore, $(Z^T Z)^{-1}$ is a diagonal matrix of the reciprocals $1/N_k$ and $P_Z = Z(Z^T Z)^{-1} Z^T = (p_{ij})$ is an $N \times N$ matrix in which $p_{ij} = 1/N_k$ if both i and j belong to the same class S_k, and $p_{ij} = 0$, otherwise. Matrix P_Z represents the operation of orthogonal projection of any N-dimensional vector x onto the linear subspace $L(Z)$ spanning the columns of matrix Z.

A set of partitions R^u, $u = 1, 2, ..., U$, along with the corresponding binary membership $N \times L_t$ matrices X^u, found with various clustering procedures, can be thought of as proxies for a hidden partition S, along with its binary membership matrix Z. Each of the partitions can be considered as related to the hidden partition S by equations

$$x^u_{il} = \sum_{k=1}^{K} c^u_{kl} z_{ik} + e^u_{ik} \tag{10}$$

where coefficients c^u_{kl} and matrix z_{ik} are to be chosen to minimize the residuals e^u_{ik}.

By accepting the sum of squared errors $E^2 = \sum_{i,k,u} (e^u_{ik})^2$ as the criterion to minimize, one immediately arrives at the optimal coefficients being orthogonal projections of the columns of matrices X^u onto the linear subspace spanning the hidden matrix Z. More precisely, at a given Z, the optimal $K \times L_u$ matrices $C^u = (c^u_{kl})$ are determined by equations $C^u = Z(Z^T Z)^{-1} X^u$. By substituting these in equations (10), the square error criterion can be reformulated as:

$$E^2 = \sum_{u=1}^{U} ||X^u - P_Z X^u||^2 \tag{11}$$

where $||.||^2$ denotes the sum of squares of the matrix elements. It is not difficult to show that the criterion can be reformulated in terms of the so-called consensus similarity matrix. To this end, let us form $N \times L$ matrix $X = (X^1 X^2 ... X^U)$ where $L = \sum_{u=1}^{U} L_u$. The columns of this matrix correspond to clusters R_l that are present in partitions $R^1, ..., R^U$. Then the least squares criterion can be expressed as $E^2 = ||X - P_Z X||^2$, or equivalently, as $E^2 = Tr((X - P_Z X)(X - P_Z X)^T)$ where Tr denotes the trace of $N \times N$ matrix, that is, the sum of its diagonal elements, and ^{T}, the transpose. By opening the parentheses in the latter expression, one can derive that $E^2 = Tr(XX^T - P_Z XX^T)$. Let us denote $A = XX^T$ and take a look at (i, j)-th element of this matrix $a_{ij} = \sum_l x_{il} x_{jl}$ where summation goes over all clusters R_l of all partitions $R^1, R^2, ..., R^U$. Obviously, a_{ij} equals the number of those partitions $R^1, R^2, ..., R^U$ at which i and j are in the same class. This matrix is referred to in the literature as the consensus matrix. The latter expression can be reformulated thus as

$$E^2 = NU - \sum_{k=1}^{K} \sum_{i,j \in S_k} a_{ij}/N_k.$$

This leads us to the following statement.

Statement 5. *A partition $S = \{S_1, ..., S_K\}$ is an ensemble consensus clustering if and only if it maximizes criterion*

$$g(S) = \sum_{k=1}^{K} \sum_{i,j \in S_k} a_{ij}/N_k \tag{12}$$

where $A = (a_{ij})$ is the consensus matrix.

Criterion (12) is but the sum of semi-average criteria for clusters S_1, ..., S_K. Therefore, the iterative extraction algorithm in its partitional clusters format is applicable here. We compared the performances of this algorithm and a number of up-to-date algorithms of consensus clustering (see Table 5) [19].

Table 5. Consensus clustering methods involved in the experiments

n.	Method	Author(s)	Reference
1	Bayes	Wang et al.	[23]
2	Vote	Dimitriadi et al.	[3]
3	CVote	Ayad, Kamel	[2]
4	Borda	Sevillano et al.	[17]
5	Fusion	Guenoche	[6]
6	CSPA	Strehl, Ghosh	[21]
7	MCLA	Strehl, Ghosh	[21]

These algoritms have been compared with two versions of the iterative extraction partitional clusters method above differing by the condition whether the option of zeroing all the diagonal entries of the similarity matrix has been utilized or not (Lsc1 and Lsc2). Three types of datasets have been used: (a) datasets from the Irvine Data Repository, (b) generated synthetic datasets, and (c) specially drawn artificial 2D shapes. Here we present only results of applying the algorithms to the Wisconsin Diagnostic Breast Cancer (WDBC) dataset from UCI Data Repository (569 entities, 30 features, two classes) (see Figure 1). The results are more or less similar to each other, although the superiority of our algorithms is expressed more clearly on the other datasets [19].

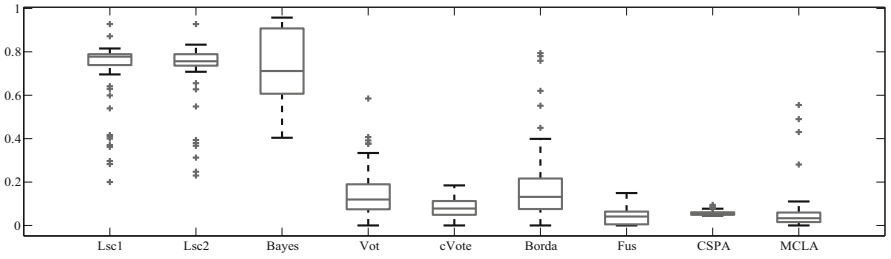


Fig. 1. Comparison of the accuracy of consensus clustering algorithms at WDBC dataset

4 Conclusion

The paper describes least squares approximation approaches for finding individual similarity clusters which can be useful in several perspectives - summary similarity criterion, semi-average criterion, spectral clustering criterion and approximation criterion. The clustering criterion involves, in different forms, the concept of similarity threshold, or similarity shift - a value subtracted from all the similarity matrix entries. The threshold can be used for bridging different aspects of the phenomenon under study together. This is demonstrated in section 3.2, in which the final choice of clustering involves the protein function and gene arrangement in the genomic circle, in addition to the original similarity derived from protein sequences.

The criterion leads to nice properties of the clusters: they are quite tight over average similarities of individual entities with them. Also, unlike methods for finding global optima, the one starting from an entity leads to recovery of the local cluster structure of the data, probably a single most important innovation proposed in this paper.

Acknowledgements. This work was partially supported by the International Laboratory of Decision Choice and Analysis at NRU HSE (headed by F. Aleskerov) and the Laboratory of Algorithms and Technologies for Network Analysis NRU HSE Nizhny Novgorod by means of RF goverment grant ag. 11.G34.31.0057 (headed by V. Kalyagin).

References

1. Apresian, Y.D.: An algorithm for finding clusters by a distance matrix. Computer. Translation and Applied Linguistics 9, 72–79 (1966) (in Russian)
2. Ayad, H., Kamel, M.: On voting-based consensus of cluster ensembles. Pattern Recognition, 1943–1953 (2010)
3. Dimitriadou, E., Weingessel, A., Hornik, K.: A Combination Scheme for Fuzzy Clustering. Journal of Pattern Recognition and Artificial Intelligence, 332–338 (2002)
4. Frumkina, R., Mirkin, B.: Semantics of domain-specific nouns: a psycho-linguistic approach. Notices of Russian Academy of Science: Language and Literature 45(1), 12–22 (1986) (in Russian)
5. Gallo, G., Grigoriadis, M.D., Tarjan, R.E.: A fast parametric maximum flow algorithm and applications. SIAM Journal on Computing 18, 30–55 (1989)
6. Guenoche, A.: Consensus of partitions: a constructive approach. Adv. Data Analysis and Classification 5, 215–229 (2011)
7. Holzinger, K.J., Harman, H.H.: Factor Analysis. University of Chicago Press, Chicago (1941)
8. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1988)
9. Kawaji, H., Takenaka, Y., Matsuda, H.: Graph-based clustering for finding distant relationships in a large set of protein sequences. Bioinformatics 20(2), 243–252 (2004)
10. Mirkin, B.: Analysis of Categorical Features, p. 166. Finansy i Statistika Publishers, Moscow (1976) (in Russian)
11. Mirkin, B.: Additive clustering and qualitative factor analysis methods for similarity matrices. Journal of Classification 4, 7–3; Erratum 6, 271–272 (1989)
12. Mirkin, B.: A sequential fitting procedure for linear data analysis models. Journal of Classification 7, 167–195 (1990)
13. Mirkin, B.: Clustering: A Data Recovery Approach, 2nd edn. Chapman and Hall, Boca Raton (2012)
14. Mirkin, B.G., Camargo, R., Fenner, T., Loizou, G., Kellam, P.: Similarity clustering of proteins using substantive knowledge and reconstruction of evolutionary gene histories in herpesvirus. Theoretical Chemistry Accounts: Theory, Computation, and Modeling 125(3-6), 569–581 (2010)
15. Mirkin, B., Muchnik, I.: Geometric interpretation of clustering criteria. In: Mirkin, B. (ed.) Methods for Analysis of Multidimensional Economics Data, pp. 3–11. Nauka Publishers (Siberian Branch), Novosibirsk (1981) (in Russian)
16. Rosenberg, S., Kim, M.P.: The method of sorting as a data-gathering procedure in multivariate research. Multivariate Behavioral Research 10, 489–502 (1975)
17. Sevillano Dominguez, X., Socoro Carrie, J.C., Alias Pujol, F.: Fuzzy clusterers combination by positional voting for robust document clustering. Procesamiento Del Lenguaje Natural 43, 245–253 (2009)
18. Shepard, R.N., Arabie, P.: Additive clustering: representation of similarities as combinations of overlapping properties. Psychological Review 86, 87–123 (1979)
19. Shestakov, A., Mirkin, B.: Least squares consensus clustering applied to k-means results (2013) (in progress)
20. Smid, M., Dorssers, L.C.J., Jenster, G.: Venn Mapping: clustering of heterologous microarray data based on the number of co-occurring differentially expressed genes. Bioinformatics 19(16), 2065–2071 (2003)

21. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal on Machine Learning Research, 583–617 (2002)
22. Swift, S., Tucker, A., Vinciotti, V., Martin, N., Orengo, C., Liu, X., Kellam, P.: Consensus clustering and functional interpretation of gene expression data. Genome Biology 5, R94 (2004)
23. Wang, H., Shan, H., Banerjee, A.: Bayesian cluster ensembles. In: Proceedings of the Ninth SIAM International Conference on Data Mining, pp. 211–222 (2009)

Some New Progress in Analyzing and Mining Uncertain and Probabilistic Data for Big Data Analytics*

Jian Pei

Simon Fraser University, BA, Canada
jpei@cs.sfu.ca
www.cs.sfu.ca/~jpei

Abstract. Uncertainty is ubiquitous in big data. Consequently, analyzing and mining uncertain and probabilistic data is important in big data analytics. In this short article, we review some recent progress in mining uncertain and probabilistic data in the hope that the problems, progress, and challenges can inspire interdisciplinary dialogues and lead to new research opportunities.

1 Introduction

According to Wikipedia, big data refers to "*data sets with sizes beyond the ability of commonly-used software tools to capture, curate, manage, and process the data within a tolerable elapsed time*"[1]. Big data posts many grand challenges for data analytics, often summarized using four V's: volume, variety, veracity, and velocity.

Uncertainty is ubiquitous in big data. Consequently, analyzing and mining uncertain and probabilistic data is important in big data analytics. For example, noise is almost unavoidable in massive data. To reduce data volume, sampling methods and statistical models are often employed to access and summarize large data sets and generate working data sets for analysis, which introduce imprecision and uncertainty. Moreover, many data analysis techniques trade off uncertainty for reduced representation cost. When integrating data from many sources, such as resolving conflicts and inconsistency and removing duplicates and redundancy, uncertainty often slides in due to noise and errors in data. Uncertainty is inherent in analysis results derived from inaccurate or uncertain data. To process streaming data, which arrives fast and may often allow only one scan, analytics results have to be approximate and thus uncertainty is inevitable.

Recently, significant progress has been achieved in mining uncertain and probabilistic data in the databases and data mining communities. Different from the

* Jian Pei's research is supported in part by an NSERC Discovery Grant and a BCFRST NRAS Endowment Research Team Program project. All opinions, findings, conclusions and recommendations in this paper are those of the author and do not necessarily reflect the views of the funding agencies.

[1] http://en.wikipedia.org/wiki/Big_data, as of July 1, 2013.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 38–45, 2013.

previous work focusing on uncertainty in inferences, this new line of work explicitly models uncertainty in data, and how the uncertainty in data affects query and data mining results. In this paper, we review some of such research results, part of them being generated in my group or by my collaborators and me. The purpose is to bring such problems, progress, and challenges to the attention of a broader audience and possible applications in the hope that interdisciplinary dialogues can be inspired and new research opportunities can be identified and pursued.

We note that a systematic survey of data processing and data mining techniques on uncertain data is far beyond the capacity of this paper. There are some recent excellent surveys on the topic, such as [3,7]. We realize that, due to the limit of space and the wide range of the related results, the coverage of this paper is biased and narrow. We apologize for any possible unintentional offenses.

In the rest of the paper, we first review the possible worlds model of uncertain data in Section 2. Then, in Section 3 we discuss ranking queries and skyline queries, which are good representatives in uncertain data processing. In Section 4, we discuss clustering and outlier detection on uncertain data, some interesting problems in mining uncertain and probabilistic data. We conclude by some interesting challenges in Section 5.

2 Uncertain and Probabilistic Data and Possible Worlds

We consider multidimensional objects, that is, objects of multiple attributes. An object is described by its values on those attributes. An object is **certain** if its value on every attribute is determined, which is the case assumed in most of the traditional data analysis methods. An object is **uncertain** if there exists at least one attribute where the object's behavior can be modeled as a random variable.

For example, we can model a tennis player as an object with multiple attributes, one being the speed of serves. A play serves in a game multiple times likely in different speeds, not to mention in different games. Thus, a player's speed of serves is a random variable.

In general, a multidimensional uncertain object is a multidimensional random variable. In practice, often an uncertain object is captured by a set of instances, where each instance can be regarded as a sample of the object.

There are two frequently used methods to represent uncertain objects. In a probabilistic table, an instance of an uncertain object is represented as a tuple associated with an existence probability. Generation rules are used to describe the relations among instances. For example, an exclusive rule can specify that several instances of an object cannot co-exist. Alternatively, we can represent each uncertain object explicitly as a set of instances, where each instance can be associated with an existence probability. More generally, we can describe an uncertain object using its joint distribution or its factored marginal distributions under some independence assumptions.

An **uncertain database** is a set of uncertain objects. In the rest of the paper, we use the terms "**uncertain data**" and "**probabilistic data**" interchangeably to refer to uncertain databases.

Table 1. Possible worlds in the toy example in Table 2

Name	Speed of serves	Return of serve rate	Existence probability
Albert	120	60%	0.40
	100	75%	0.60
Bob	135	50%	0.30
	125	35%	0.50

Table 2. A toy example

Possible world id	Tuples	Existence probability
w_1	(Albert, 120, 60%), (Bob, 135, 50%)	0.12
w_2	(Albert, 120, 60%), (Bob, 125, 35%)	0.20
w_3	(Albert, 120, 60%)	0.08
w_4	(Albert, 100, 75%), (Bob, 135, 50%)	0.18
w_5	(Albert, 100, 75%), (Bob, 125, 35%)	0.30
w_6	(Albert, 100, 75%)	0.12

A possible configuration of an uncertain database is called a **possible world** [16]. To describe the semantics of an uncertain/probabilistic data set completely, we have to enumerate all possible configurations thoroughly and their existence probability.

For example, suppose we model tennis players using two attributes, speed of serves and return of serve rate. As a toy example, Table 2 shows two players, Albert and Bob, each having two instances. Table 1 shows the six possible worlds.

Given an uncertain database D, let $\mathcal{PW}(D)$ be the set of possible worlds of D. Each possible world is a database associated with an existence probability. For an analytics task Q, such as a query or a data mining task, we can apply Q on every possible world of D. Denote by $Q(w)$ $(w \in \mathcal{PW}(D))$ the result of Q on possible world w. The problem of analyzing and mining uncertain data is to summarize the result set $\{Q(w) \mid w \in \mathcal{PW}(D)\}$. For more details about the possible world model, please see [16].

3 Ranking Queries and Skyline Queries on Uncertain Data

Consider the task of ranking objects, a simple yet frequently used database query. Given a set of multidimensional objects and an objective function defined using the dimensions, we want to sort all objects according to their values in the objective function. While ranking queries on certain data have been well studied and popular in commercial products, ranking uncertain data is far from trivial.

The challenge comes from the fact that an uncertain tuple/object may have different ranks in different possible worlds. There are different ways to summarize their ranks. For example, one may use the expected rank, which is the average of

the ranks in all possible worlds weighted by the existence probabilities of those worlds. That is,

$$E(rank(x)) = \sum_{w \in \mathcal{PW}(D)} rank_w(x)p(w)$$

where $p(w)$ is the existence probability of a possible world, and $rank_w(x)$ is the rank of object x in possible world w.

One possible drawback of the method using expectation is that a tuple may not take the expected rank in any possible world. An alternative is to find for each uncertain tuple/object the most likely rank, that is, the rank of the highest probability. That is,

$$M(rank(x)) = \arg\max_{r>0} \left\{ \sum_{w \in \mathcal{PW}(D), rank_w(x)=r} p(w) \right\}$$

However, the probability of even the most likely rank may be very low on a large uncertain database. Moreover, two tuples/objects may reach their most likely ranks in different possible worlds. In other words, such ranks may not be used for comparison.

To tackle the problem, among several recently proposed models, we proposed the probabilistic threshold approach [8]. Given a probability threshold $p > 0$ and a ranking threshold $k > 0$, a **probabilistic threshold top-k query** finds all tuples that have a probability of at least p to be ranked at the top k positions in all possible worlds. That is,

$$rank(x, p) = \max \left\{ k > 0 \mid \sum_{w \in \mathcal{PW}(D), rank_w(x) \geq k} p(w) \geq p \right\}$$

Most recently, Li *et al.* [13] developed a unified model.

The models of ranking queries on uncertain data demonstrate the challenges in summarizing different query answers in different possible worlds and some interesting ideas.

It is computationally prohibitive to enumerate all possible worlds and compute a query in each world. We observed that Poisson binomial recurrence can be used in computing many types of probabilistic threshold based queries on uncertain data [8]. Later, Bernecker *et al.* [4] defined probabilistic threshold frequent patterns on uncertain transactions, where each item takes a probability to appear in a transaction. Their algorithm uses Poisson binomial recurrence, too. Using Poisson binomial recurrence and some other techniques, we try to avoid enumerating all possible worlds and computing queries on each possible world in analyzing and mining uncertain data.

Ranking queries on uncertain data can be extended to address needs in various applications. We discussed a few interesting extensions [7], including top-k typicality queries, online ranking query answering, continuous ranking queries on uncertain data streams, ranking queries on probabilistic linkages, and probabilistic path queries on road networks.

Skyline queries are another type of useful analytic queries related to ranking queries. An object x is said to dominate another object y if x is not worse than y in every dimension, and there exists at least one dimension where x is better than y. Given a set D of objects, $x \in D$ is a skyline object if there does not exist any other object $y \in D$ such that y dominates x. A skyline query finds all skyline objects in a given set.

We developed a bounding-pruning-refining framework for probabilistic threshold based skyline queries on uncertain data [15]. Given a probability threshold $p > 0$, we compute all objects that take a probability of at least p to be in the skyline. The problem formulation and the general framework inspire many more recent methods. The bounding-pruning-refining framework is a general heuristic method for analyzing and mining uncertain data.

4 Clustering Analysis and Outlier Detection on Uncertain Data

Clustering analysis, also known as unsupervised learning, partitions a set of objects into groups such that objects falling into a group are similar to each other, and objects falling into different groups are dissimilar. Uncertainty is considered in clustering, such as fuzzy clustering [6], where the assignment of an object to a cluster may be probabilistic. However, how to cluster uncertain objects was not systematically investigated until very recently.

The need of clustering analysis on uncertain data comes from a few applications. For example, marketing surveys may collect customers' opinion about products, such as hotels, on multiple attributes, such as room quality, location convenience, and service quality. One hotel may receive multiple reviews and thus can be naturally modeled as an uncertain object of multiple instances, where each review is captured by an instance. An often useful analytic task is to cluster uncertain objects, such as clustering hotels according to their reviews.

Some studies [11,12,14,10] extend traditional clustering methods, such as K-means, density-based clustering, and hierarchical clustering, to uncertain data. Cormode and McGregor [5] provided theoretical analysis on extending partitioning methods, such as K-means and K-medoids, to uncertain data.

Most of the clustering methods for uncertain data extended from traditional approaches only explore the geometric properties of data objects and focus on instances of uncertain objects. One important issue in uncertain data clustering is that the distribution of an uncertain object is an inherent feature, which should be considered in clustering. For example, Figure 1 shows the instances of two uncertain objects. The two objects have very similar mean values, but their distributions of instances are very different.

To address the new challenge in uncertain data mining that distributions are inherent features for uncertain data, we advocated clustering uncertain objects according to their distributions [9]. Concretely, we can estimate the distribution of an uncertain object using the instances of the object. Then, we can measure the similarity between the distributions of two objects, for example, using

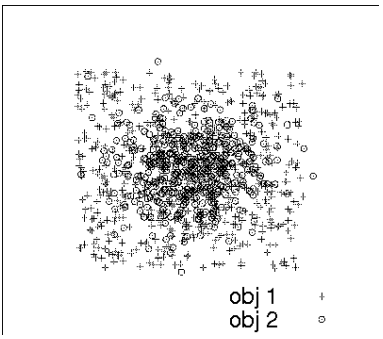


Fig. 1. Two objects of different distributions but similar means. The artwork is adopted from [9].

Kullback-Leibler divergence. However, computing the distribution similarity between every two uncertain objects in a large data set is very costly. To tackle the cost, we developed a fast Gauss transform method.

Outlier detection is another frequently used data mining task, which finds objects that are significantly different from the majority. Outlier detection on certain data has been well studied [1].

Aggarwal and Yu [2] proposed the notion of (δ, η)-outlier. An uncertain object O is a (δ, η)-outlier if the probability of O lying in a region in some subspace with density at least η is less than δ. To mine (δ, η)-outliers, one can first enumerate all non-empty subspaces in a bottom-up, breadth-first manner. For each subspace, one can use sampling and micro-clusters to estimate the density distribution, and check whether there is an (δ, η)-outlier. In this method, only outlier objects are detected, but not outlier instances.

To detect both outlier objects and outlier instances, a straightforward way to extend existing outlier detection methods to handle uncertain objects works in two steps. First, for each uncertain object, we can detect and removed outlier instances. After this step, we can represent each object using an aggregate, such as mean or median, of all instances of the object. Second, we can detect and remove outlier objects. However, such a straightforward extension suffers from a critical drawback. As we just discussed, distribution is an inherent feature of uncertain objects. Using aggregates, such as mean and median, may not be able to represent an object well.

In many applications, an uncertain object is associated with some inherent properties described by a set of conditioning attributes, and a set of instances described by a set of dependent attributes. For example, to collect environment surveillance data, the meteorological measures at a location, such as temperature, pressure, and humidity, may be modeled as a multidimensional random variable, that is, an uncertain object. Multiple co-located monitors may provide readings to estimate the random variable. At the same time, the values of those meteorological measures at a location depend on some conditioning attributes, such as latitude and longitude.

To provide a comprehensive solution to detect outliers on uncertain data at both object level and instance level [9], we observe that objects with similar properties, such as latitude and longitude, tend to have similar instance distributions. Consequently, we can learn the normal instances of each object by taking into account the instances of objects with similar properties. Technically, we learn the conditional distribution of dependent attributes given the conditioning attributes, and measure the normality, which is the opposite of outlyingness, of an instance by its conditional probability. At the object level, we detect outlier objects most of whose instances are outliers.

5 Summary and Challenges

Uncertain data and probabilistic data are ubiquitous in big data and big data analytics. Therefore, effective and efficient techniques are of high demand in practice. We reviewed some of the recent progress in analyzing and mining uncertain data. Particularly, we illustrated the challenges and some ideas about summarizing results in possible worlds, reducing enumeration of all possible worlds and computing queries on each possible world, a heuristic bounding-pruning-refining framework, using distributions of instances in objects in analytics, and integrating analysis at both instance level and object level.

Although good progress has been achieved, there are still many grand challenges. Particularly, many of the state-of-the-art methods for analyzing and mining uncertain data are developed in the traditional databases and data mining community. For example, fuzzy methods and rough set methods are two important approaches to capture and analyze uncertainty. However, not many existing uncertain data analysis and mining methods, particularly those on pattern mining, consider fuzzy methods and rough set methods systematically.

Uncertainty may exist in multiple aspects in big data analytics, such as data level and analysis level. While uncertain data analysis and mining methods mainly target on uncertainty at the data level, and many existing machine learning methods, rough set methods, and fuzzy methods embrace uncertainty in inference process, it is important and interesting to develop a comprehensive framework to address uncertainty in different aspects in an integrative and consistent way. Instead of categorizing different methods according to the traditional schools, we have to adopt and develop whatever feasible and effective methods to tackle the grand challenges posted by uncertainty in big data and big data analytics.

References

1. Aggarwal, C.: An Introduction to Outlier Analysis. Springer, New York (2013)
2. Aggarwal, C.C., Yu, P.: Outlier detection with uncertain data. In: SIAM Data Mining Conference (2008)
3. Aggarwal, C.C., Yu, P.S.: A survey of uncertain data algorithms and applications. IEEE Transactions on Knowledge and Data Engineering 21(5), 609–623 (2009)

4. Bernecker, T., Kriegel, H.-P., Renz, M., Verhein, F., Zuefle, A.: Probabilistic frequent itemset mining in uncertain databases. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 119–128. ACM, New York (2009)
5. Cormode, G., McGregor, A.: Approximation algorithms for clustering uncertain data. In: Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, pp. 191–200. ACM, New York (2008)
6. Höppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition. Wiley (July 1999)
7. Hua, M., Pei, J.: Ranking Queries on Uncertain Data. Springer, USA (2011)
8. Hua, M., Pei, J., Zhang, W., Lin, X.: Ranking queries on uncertain data: A probabilistic threshold approach. In: Proc. ACM International Conference on Management of Data, SIGMOD 2008, Vancouver, Canada (June 2008)
9. Jiang, B., Pei, J.: Outlier detection on uncertain data: Objects, instances, and inferences. In: Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, ICDE 2011, pp. 422–433. IEEE Computer Society, Washington, DC (2011)
10. Kao, B., Lee, S.-D., Cheung, D., Ho, W.-S., Chan, K.F.: Clustering uncertain data using voronoi diagrams. In: Eighth IEEE International Conference on Data Mining, ICDM 2008, pp. 333–342 (2008)
11. Kriegel, H.-P., Pfeifle, M.: Density-based clustering of uncertain data. In: Proceeding of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD 2005, pp. 672–677. ACM, New York (2005)
12. Kriegel, H.-P., Pfeifle, M.: Hierarchical density-based clustering of uncertain data. In: Proceedings of the Fifth IEEE International Conference on Data Mining, ICDM 2005, pp. 689–692. IEEE Computer Society, Washington, DC (2005)
13. Li, J., Saha, B., Deshpande, A.: A unified approach to ranking in probabilistic databases. The VLDB Journal 20(2), 249–275 (2011)
14. Ngai, W.K., Kao, B., Chui, C.K., Cheng, R., Chau, M., Yip, K.Y.: Efficient clustering of uncertain data. In: Proceedings of the Sixth International Conference on Data Mining, ICDM 2006, pp. 436–445. IEEE Computer Society, Washington, DC (2006)
15. Pei, J., Jiang, B., Lin, X., Yuan, Y.: Probabilistic skylines on uncertain data. In: Proceedings of the 33rd International Conference on Very Large Data Bases, VLDB 2007, Viena, Austria (September 2007)
16. Sarma, A.D., Benjelloun, O., Halevy, A., Widom, J.: Working models for uncertain data. In: ICDE 2006: Proceedings of the 22nd International Conference on Data Engineering, p. 7. IEEE Computer Society, Washington, DC (2006)

Three Approaches to Deal with Inconsistent Decision Tables - Comparison of Decision Tree Complexity

Mohammad Azad, Igor Chikalov, and Mikhail Moshkov

Computer, Electrical and Mathematical Sciences and Engineering Division
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
{mohammad.azad,igor.chikalov,mikhail.moshkov}@kaust.edu.sa

Abstract. In inconsistent decision tables, there are groups of rows with equal values of conditional attributes and different decisions (values of the decision attribute). We study three approaches to deal with such tables. Instead of a group of equal rows, we consider one row given by values of conditional attributes and we attach to this row: (i) the set of all decisions for rows from the group (many-valued decision approach); (ii) the most common decision for rows from the group (most common decision approach); and (iii) the unique code of the set of all decisions for rows from the group (generalized decision approach). We present experimental results and compare the depth, average depth and number of nodes of decision trees constructed by a greedy algorithm in the framework of each of the three approaches.

Keywords: Decision Trees, Greedy Algorithms, Inconsistent Decision Tables, Boundary Subtables.

1 Introduction

It is not uncommon to have inconsistent decision tables where there are groups of rows (objects) with equal values of conditional attributes and different decisions (values of the decision attribute). In this paper, we consider three approaches to deal with inconsistent decision tables.

The first approach is called many-valued decisions – *MVD*. We transform an inconsistent decision table into a decision table with many-valued decisions. Instead of a group of equal rows with, probably, different decisions we consider one row given by values of conditional attributes and we attach to this row the set of all decisions for rows from the group [6]. Our aim here is to find, for a given row r, a decision from the set of decisions attached to rows equal to r.

The second approach is called the most common decision – *MCD*. We transform an inconsistent decision table into consistent decision table with one-valued decisions. Instead of a group of equal rows with, probably, different decisions, we consider one row given by values of conditional attributes and we attach to

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 46–54, 2013.

this row the most common decision for rows from the group. Our aim here is to find, for a given row r, the most common decision attached to rows equal to r.

The third approach is well known in the rough set theory [9, 10] and is called generalized decision – *GD*. In this case we transform an inconsistent decision table into the table with many-valued decisions and after that encode each set of decisions by a number (decision) such that equal sets are encoded by equal numbers and different sets – by different numbers. Our aim here is to find, for a given row r, all decisions attached to rows equal to r.

In literature, often, problems that are connected with multi-label data are considered from the point of view of classification (multi-label classification problem) [3–5, 8, 11–13]. But here our aim is to show that the proposed approach based on many-valued decisions can be useful from the point of view of knowledge representation.

In [2, 7] we studied a greedy algorithm for construction of decision trees for decision tables with many-valued decisions. This algorithm can be used also in the cases of *MCD* and *GD* approaches: we can consider decision tables with one-valued decisions as decision tables with many-valued decisions where sets of decisions attached to rows have one element.

This paper is an extension of the conference publication [2]. It is devoted to the comparison of depth, average depth and number of nodes of trees constructed by the greedy algorithm in the framework of *MVD*, *MCD* and *GD* approaches. All definitions are given for binary decision tables. However, they can be easily extended to the decision tables filled by numbers from the set $\{0, \ldots, k-1\}$, where $k \geq 3$.

We present experimental results for data sets from UCI Machine Learning Repository [1] that are converted into inconsistent decision tables by removal of some conditional attributes. These results show that the use of *MCD* and, especially, *MVD* approaches can reduce the complexity of trees in comparison with *GD* approach. It means that *MVD* and *MCD* approaches can be useful from the point of view of knowledge representation.

This paper consists of six sections. Section 2 contains main definitions. In Sect. 3, we consider decision tables which have at most t decisions in each set of decisions attached to rows. In Sect. 4, we present the greedy algorithm for construction of decision trees. Section 5 contains results of experiments and Sect. 6 – conclusions.

2 Main Definitions

A *(binary) decision table* with one-valued decisions is a rectangular table T filled by numbers from the set $\{0, 1\}$. Columns of this table are labeled with conditional attributes $f_1, \ldots, f_n$. Each row is labeled with a natural number (decision) which is interpreted as a value of the decision attribute. It is possible that T is inconsistent, i.e., contains equal rows with different decisions. The table T can contain also equal rows with equal decisions.

A *(binary) decision table with many-valued* decisions is a rectangular table filled by numbers from the set $\{0, 1\}$. Columns of this table are labeled with conditional attributes $f_1, \ldots, f_n$. Rows of the table are pairwise different, and each row is labeled with a nonempty finite set of natural numbers (set of decisions). Note that each consistent decision table with one-valued decisions can be interpreted also as a decision table with many-valued decisions. In such table, each row is labeled with a set of decisions which has one element.

The most frequent decision attached to rows from a group of rows in a decision table T with one-valued decisions is called the *most common decision* for this group of rows. If we have more than one such decision we choose the minimum one.

To work with inconsistent decision tables we consider three approaches:

- many-valued decisions – *MVD*,
- most common decision – *MCD*,
- generalized decision – *GD*.

For approach called *many-valued decisions – MVD*, we transform an inconsistent decision table T into a decision table T_{MVD} with many-valued decisions. Instead of a group of equal rows with, probably, different decisions, we consider one row from the group and we attach to this row the set of all decisions for rows from the group [6].

T^0 =

	f_1	f_2	f_3	
r_1	1	1	1	1
r_2	0	1	0	1
r_3	0	1	0	3
r_4	1	1	0	2
r_5	0	0	1	2
r_6	0	0	1	3
r_7	1	0	0	1
r_8	1	0	0	2

T^0_{MVD} =

	f_1	f_2	f_3	
r_1	1	1	1	{1}
r_2	0	1	0	{1, 3}
r_3	1	1	0	{2}
r_4	0	0	1	{2, 3}
r_5	1	0	0	{1, 2}

T^0_{GD} =

	f_1	f_2	f_3	
r_1	1	1	1	1
r_2	0	1	0	2
r_3	1	1	0	3
r_4	0	0	1	4
r_5	1	0	0	5

T^0_{MCD} =

	f_1	f_2	f_3	
r_1	1	1	1	1
r_2	0	1	0	1
r_3	1	1	0	2
r_4	0	0	1	2
r_5	1	0	0	1

Fig. 1. Transformation of inconsistent decision table T^0 into decision tables T^0_{MVD}, T^0_{GD} and T^0_{MCD}

For approach called *most common decision – MCD*, we transform inconsistent decision table T into consistent decision table T_{MCD} with one-valued decision. Instead of a group of equal rows with, probably, different decisions, we consider one row from the group and we attach to this row the most common decision for the considered group of rows.

For approach called *generalized decision – GD*, we transform inconsistent decision table T into consistent decision table T_{GD} with one-valued decisions. Instead of a group of equal rows with, probably, different decisions, we consider one row from the group and we attach to this row the set of all decisions for rows from the group. Then instead of a set of decisions we attach to each row a code of this set – a natural number such that the codes of equal sets are equal and the codes of different sets are different. We have shown in Fig. 1 the transformation of an inconsistent decision table T^0 using all the three approaches.

To unify some notions for decision tables with one-valued and many-valued decisions, we will interpret decision table with one-valued decisions as a decision table with many-valued decisions where each row is labeled with a set of decisions that has one element.

We will say that T is a *degenerate* table if either T is empty (has no rows), or the intersection of sets of decisions attached to rows of T is nonempty.

A table obtained from T by removal of some rows is called a *subtable* of T. A subtable T' of T is called *boundary* subtable if T' is not degenerate but each proper subtable of T' is degenerate. We denote by $B(T)$ the number of boundary subtables of the table T. It is clear that T is a degenerate table if and only if $B(T) = 0$. The value $B(T)$ will be interpreted as *uncertainty* of T.

$T_1 =$

	f_1	f_2	f_3	d
r_2	0	1	0	$\{1,3\}$
r_4	0	0	1	$\{2,3\}$
r_5	1	0	0	$\{1,2\}$

$T_2 =$

	f_1	f_2	f_3	d
r_1	1	1	1	$\{1\}$
r_4	0	0	1	$\{2,3\}$

$T_3 =$

	f_1	f_2	f_3	d
r_2	0	1	0	$\{1,3\}$
r_3	1	1	0	$\{2\}$

$T_4 =$

	f_1	f_2	f_3	d
r_1	1	1	1	$\{1\}$
r_3	1	1	0	$\{2\}$

Fig. 2. All boundary subtables of the decision table T^0_{MVD} (see Fig. 1)

Figure 2 presents all four boundary subtables of the decision table T^0_{MVD} depicted in Fig. 1. The number of boundary subtables of the decision table T^0_{MCD} is equal to six. The number of boundary subtables of the decision table T^0_{GD} is equal to 10. Each boundary subtable of tables T^0_{MCD} and T^0_{GD} has two rows (see Proposition 1).

Let $f_{i_1}, \ldots, f_{i_m} \in \{f_1, \ldots, f_n\}$ and $\delta_1, \ldots, \delta_m \in \{0, 1\}$. We denote by

$$T(f_{i_1}, \delta_1) \ldots (f_{i_m}, \delta_m)$$

the subtable of the table T which consists of all rows that at the intersection with columns $f_{i_1}, \ldots, f_{i_m}$ have numbers $\delta_1, \ldots, \delta_m$ respectively. For example, the subtable $T(f_1, 0)$ will contain rows from T for which the value of the attribute f_1 is equal to 0. Similarly, the subtable $T(f_1, 0)(f_2, 1)$ will have the rows from T, for which, attribute f_1 has the value 0 and f_2 has the value 1. In this way, we construct subtables by choosing attributes and their corresponding values.

A *decision tree* over T is a finite tree with root in which each terminal node is labeled with a decision (a natural number), and each nonterminal node is labeled with an attribute from the set $\{f_1, \ldots, f_n\}$. Two edges start in each nonterminal node. These edges are labeled with 0 and 1 respectively.

Let Γ be a decision tree over T and v be a node of Γ. There is one to one mapping between node v and subtable of T i.e. for each node v, we have an unique subtable of T. We denote $T(v)$ as a subtable of T that is mapped for a node v of decision tree Γ. If node v is the root of Γ then $T(v) = T$. Otherwise, let nodes and edges in the path from the root to node v be labeled

with attributes $f_{i_1}, \ldots, f_{i_m}$ and numbers $\delta_1, \ldots, \delta_m$ respectively. Then $T(v)$ is the subtable $T(f_{i_1}, \delta_1) \ldots (f_{i_m}, \delta_m)$ of the table T.

It is clear that for any row r of T there exists exactly one terminal node v in Γ such that r belongs to $T(v)$. The decision attached to v will be considered as the result of the work of Γ on the row r. We denote by $l_\Gamma(r)$ the length of the path from the root of Γ to the node v. We will say that Γ is a decision tree for the table T if, for any row r of T, the result of the work of Γ on the row r belongs to the set of decisions attached to the row r. An example of a decision tree for the table T^0_{MVD} can be found in Fig. 3.

We denote by $h(\Gamma)$ the *depth* of Γ which is the maximum length of a path from the root to a terminal node. Let $\Delta(T)$ be the set of rows of T. We denote by $h_{avg}(\Gamma)$ the average depth of Γ which is equal to $\sum_{r \in \Delta(T)} l_\Gamma(r)/|\Delta(T)|$. We denote by $N(\Gamma)$ the number of nodes in the decision tree Γ.

3 Set $Tab(t)$ of Decision Tables

We denote by $Tab(t)$, where t is a natural number, the set of decision tables with many-valued decisions such that each row in the table has at most t decisions (is labeled with a set of decisions which cardinality is at most t).

Proposition 1. *[6] Each boundary subtable of a table $T \in Tab(t)$ has at most $t + 1$ rows.*

Therefore, for tables from $Tab(t)$, there exists a polynomial time algorithm for the finding of all boundary subtables and the computation of parameter $B(T)$. For example, for any decision table T with one-valued decisions (in fact, for any table from $Tab(1)$), the equality $B(T) = P(T)$ holds, where $P(T)$ is the number of unordered pairs of rows of T with different decisions. Hence, we count number of boundary subtables by checking over all possible subtables from T. First, we start from all possible combination of 2-rows subtables of T. For each 2-rows subtable, if it is boundary subtable then we count it. However, if it is degenerate subtable, then we expand it into 3-rows subtable by adding another row which was not added before. Henceforth, in a similar way, we check this new 3-rows subtable for boundary subtable, and continue expanding if it is degenerate. We continue this process until we expand it into $(t + 1)$-rows subtables of T, and finish the algorithm by returning all boundary subtables that have been counted.

4 Greedy Algorithm U

The greedy algorithm U, for a given decision table T with many-valued decisions, constructs a decision tree $U(T)$ for T (see Algorithm 1).

Now, we present an example of the greedy algorithm work for construction of a decision tree for the decision table T^0_{MVD} depicted in Fig. 1. All boundary

Algorithm 1. Greedy algorithm U

Require: Binary decision table T with many-valued decisions and attributes $f_1, \ldots, f_n$.
Ensure: Decision tree $U(T)$ for T.
Construct the tree G consisting of a single node labeled with the table T;
while (true) **do**
if No one node of the tree G is labeled with a table **then**
Denote the tree G by $U(T)$;
else
Choose a node v in G which is labeled with a subtable T' of the table T;
if $B(T') = 0$ **then**
Instead of T' mark the node v with the most common decision for T';
else
for $i = 1, \ldots, n$, compute the value

$$Q(f_i) = \max\{B(T'(f_i, 0)), B(T'(f_i, 1))\};$$

Instead of T' mark the node v with the attribute f_{i_0}, where i_0 is the minimum i for which $Q(f_i)$ has the minimum value;
For each $\delta \in \{0, 1\}$, add to the tree G the node $v(\delta)$ and mark this node with the subtable $T'(f_{i_0}, \delta)$;
Draw an edge from v to $v(\delta)$ and mark this edge with δ;
end if
end if
end while

subtables of the decision table T^0_{MVD} are depicted in Fig. 2. The table T^0_{MVD} is not degenerate, so for $i \in \{1, 2, 3\}$, we compute the value $Q(f_i)$: $Q(f_1) = \max\{0, 1\} = 1$, $Q(f_2) = \max\{0, 2\} = 2$, $Q(f_3) = \max\{1, 1\} = 1$. We choose the attribute f_1 and assign it to the root of the constructed tree. The decision tree $U(T^0_{MVD})$ is depicted in Fig. 3.

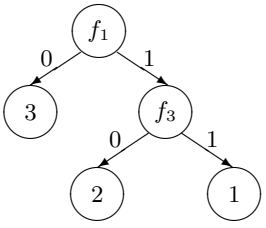


Fig. 3. Decision tree $U(T^0_{MVD})$

Table 1. Characteristics of inconsistent decision tables

Decision table T	Rows	Attr	Spectrum					
			#1	#2	#3	#4	#5	#6
balance-scale-1	125	3	45	50	30			
breast-cancer-1	193	8	169	24				
breast-cancer-5	98	4	58	40				
cars-1	432	5	258	161	13			
flags-5	171	21	159	12				
hayes-roth-data-1	39	3	22	13	4			
kr-vs-kp-5	1987	31	1564	423				
kr-vs-kp-4	2061	32	1652	409				
lymphography-5	122	13	113	9				
mushroom-5	4078	17	4048	30				
nursery-1	4320	7	2858	1460	2			
nursery-4	240	4	97	96	47			
spect-test-1	164	21	161	3				
teeth-1	22	7	12	10				
teeth-5	14	3	6	3	0	5	0	2
tic-tac-toe-4	231	5	102	129				
tic-tac-toe-3	449	6	300	149				
zoo-data-5	42	11	36	6				

5 Experimental Results

We consider a number of decision tables from UCI Machine Learning Repository [1]. In some tables there were missing values. Each such value was replaced with the most common value of the corresponding attribute. Some decision tables contain conditional attributes that take unique value for each row. Such attributes were removed. We removed from these tables some conditional attributes. As a result, we obtain inconsistent decision tables. After that we transform each such table T into tables T_{MVD}, T_{MCD} and T_{GD} as it was described in Section 2. The information about these decision tables can be found in Table 1. This table contains name of inconsistent table T in the form "name of initial table from [1]"-"number of removed conditional attributes", number of rows in T_{MVD}, T_{MCD}, T_{GD} (column "Rows"), number of attributes in T_{MVD}, T_{MCD}, T_{GD} (column "Attr"), and spectrum of the table T_{MVD} (column "Spectrum"). Spectrum of a decision table with many-valued decisions is a sequence #1, #2,..., where $\#i$, $i = 1, 2, \ldots$, is the number of rows labeled with sets of decisions with the cardinality equal to i.

Table 2 contains depth, average depth and number of nodes for decision trees $U(T_{MVD})$, $U(T_{MCD})$ and $U(T_{GD})$ constructed by the greedy algorithm U for decision tables T_{MVD}, T_{MCD} and T_{GD} derived from 18 inconsistent decision tables T (see Table 1). The obtained results show that the decision trees constructed in the frameworks of *MVD* approach are usually simpler than the decision trees constructed in the frameworks of *MCD* and the decision trees constructed in

Table 2. Depth, average depth and number of nodes for decision trees $U(T_{MVD})$, $U(T_{GD})$ and $U(T_{MCD})$

Decision table T	depth			average depth			number of nodes		
	MVD	*MCD*	*GD*	*MVD*	*MCD*	*GD*	*MVD*	*MCD*	*GD*
balance-scale-1	2	3	3	2	2.72	3	31	121	156
breast-cancer-1	6	6	7	3.72	3.731	4.119	154	159	220
breast-cancer-5	3	4	4	1.837	2.184	2.602	49	77	102
cars-1	5	5	5	2.903	3.646	4.507	90	300	462
flags-5	6	6	6	3.825	3.889	3.906	217	224	232
hayes-roth-data-1	2	3	3	1.744	1.974	2.308	17	26	39
kr-vs-kp-5	13	14	14	8.241	8.575	9.487	783	1135	1811
kr-vs-kp-4	12	14	14	8.125	8.531	9.471	785	1107	1833
lymphography-5	6	7	7	3.803	4.221	4.361	79	109	121
mushroom-5	7	8	8	2.782	2.797	2.898	249	260	265
nursery-1	7	7	7	2.825	4.446	4.946	430	1380	2130
nursery-4	2	4	4	1.333	2.783	2.417	9	133	61
spect-test-1	7	9	10	3.348	3.354	4.335	37	47	61
teeth-1	4	4	4	2.818	2.818	2.818	35	35	35
teeth-5	3	3	3	2.214	2.214	2.214	20	20	20
tic-tac-toe-4	5	5	5	2.996	4.247	4.541	76	200	257
tic-tac-toe-3	6	6	6	4.265	4.804	5.276	223	365	513
zoo-data-5	4	6	7	3.214	3.452	4.095	19	25	41
average	**5.56**	**6.33**	**6.5**	**3.44**	**3.91**	**4.3**	**183.5**	**317.94**	**464.39**

the frameworks of *MCD* approach are usually simpler than the decision trees constructed in the framework of *GD* approach.

6 Conclusions

We considered three approaches for the work with inconsistent decision tables, and compared for these approaches the complexity of decision trees constructed by the greedy algorithm. Experimental results show that the approach based on the many-valued decisions outperforms the approaches based on the generalized decisions and the most common decisions.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, `http://www.ics.uci.edu/~mlearn/`
2. Azad, M., Chikalov, I., Moshkov, M., Zielosko, B.: Greedy algorithm for construction of decision trees for tables with many-valued decisions. In: Popova-Zeugmann, L. (ed.) Proceedings of the 21st International Workshop on Concurrency, Specification and Programming, Berlin, Germany, September 26-28. CEUR-WS.org, pp. 13–24 (2012)

3. Clare, A.J., King, R.D.: Knowledge discovery in multi-label phenotype data. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 42–53. Springer, Heidelberg (2001)
4. Comité, F.D., Gilleron, R., Tommasi, M.: Learning multi-label alternating decision trees from texts and data. In: Perner, P., Rosenfeld, A. (eds.) MLDM 2003. LNCS, vol. 2734, pp. 35–49. Springer, Heidelberg (2003)
5. Mencía, E.L., Fürnkranz, J.: Pairwise learning of multilabel classifications with perceptrons. In: Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, pp. 2899–2906. IEEE (2008)
6. Moshkov, M., Zielosko, B.: Combinatorial Machine Learning – A Rough Set Approach. SCI, vol. 360. Springer, Heidelberg (2011)
7. Moshkov, M., Zielosko, B.: Construction of α-decision trees for tables with many-valued decisions. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 486–494. Springer, Heidelberg (2011)
8. Moshkov, M.J.: Greedy algorithm for decision tree construction in context of knowledge discovery problems. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 192–197. Springer, Heidelberg (2004)
9. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
10. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1992)
11. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. IJDWM 3(3), 1–13 (2007)
12. Tsoumakas, G., Katakis, I., Vlahavas, I.P.: Mining multi-label data. In: Data Mining and Knowledge Discovery Handbook, 2nd edn., pp. 667–685. Springer (2010)
13. Zhou, Z.H., Zhang, M.L., Huang, S.J., Li, Y.F.: Multi-instance multi-label learning. Artif. Intell. 176(1), 2291–2320 (2012)

Rough Set-Based Information Dilution by Non-deterministic Information

Hiroshi Sakai[1], Mao Wu[2], Naoto Yamaguchi[2], and Michinori Nakata[3]

[1] Department of Basic Sciences, Faculty of Engineering,
Kyushu Institute of Technology Tobata, Kitakyushu 804-8550, Japan
sakai@mns.kyutech.ac.jp
[2] Graduate School of Engineering, Kyushu Institute of Technology
Tobata, Kitakyushu, 804-8550, Japan
wumogaku@yahoo.co.jp, KITYN1124@gmail.com
[3] Faculty of Management and Information Science,
Josai International University
Gumyo, Togane, Chiba 283, Japan
nakatam@ieee.org

Abstract. We have investigated rough set-based concepts for a given *Non-deterministic Information System* (*NIS*). In this paper, we consider generating a *NIS* from a *Deterministic Information System* (*DIS*) intentionally. A *NIS* Φ is seen as a diluted *DIS* ϕ, and we can hide the actual values in ϕ by using Φ. We name this way of hiding *Information Dilution* by non-deterministic information. This paper considers information dilution and its application to hiding the actual values in a table.

Keywords: Rough sets, NIS-Apriori algorithm, Information dilution, Privacy preserving, Randomization, Perturbation.

1 Introduction

In our previous research, we coped with rule generation in *Non-deterministic Information Systems* (*NISs*) [7, 11–13]. In contrast to *Deterministic Information Systems* (*DISs*) [9, 14], *NISs* were proposed by Pawlak [9], Orłowska [8] and Lipski [5, 6] in order to better handle information incompleteness in data. We have proposed *certain* and *possible* rules in *NISs*, and proved an algorithm named *NIS-Apriori* is *sound* and *complete* for defined certain and possible rules. We have also implemented *NIS-Apriori* [10].

This paper considers the connection between information incompleteness and information hiding (or the randomization and the perturbation in privacy-preserving [2]). We intentionally add information incompleteness, i.e., non-deterministic values, to a *DIS* for hiding the actual values, then a *DIS* is translated to a *NIS*. For such a *NIS*, we can apply our previous framework including *NIS-Apriori*.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 55–66, 2013.

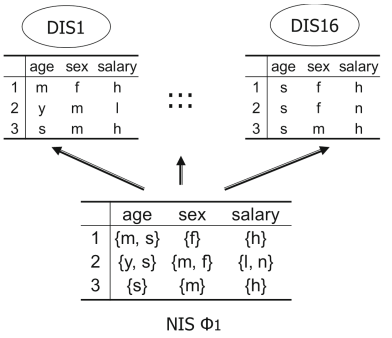


Fig. 1. NIS Φ_1 and 16 derived $DISs$. Here, $Domain_{age}$=$\{\underline{y}oung, \underline{m}iddle, \underline{s}enior\}$, $Domain_{sex}$=$\{\underline{m}ale, \underline{f}emale\}$, $Domain_{salary}$=$\{\underline{l}ow, \underline{n}ormal, \underline{h}igh\}$. The number of derived $DISs$ is finite. However, it usually increases in the exponential order with respect to the level of incompleteness of $NIS's$ values.

The paper is organized as follows: Section 2 recalls rule generation in $NISs$. Section 3 introduces the framework of information dilution, and considers properties. Section 4 considers an algorithm for dilution and its relation to reduction [9], and Section 5 concludes the paper.

2 Apriori-Based Rule Generation in NISs

We omit any formal definition. Instead, we show an example in Figure 1. We identify a DIS with a standard table. In a NIS, each attribute value is a set. If the value is a singleton, there is no incompleteness. Otherwise, we have a set of possible values. We can interpret this situation by saying that each set includes the actual value but we do not know which of them is the actual one.

A *rule* (more correctly, a candidate for a rule) is an implication τ in the form of $Condition\_part \Rightarrow Decision\_part$. In a NIS, the same τ may be generated from different tuples, so we use notation τ^x to express that τ is generated by an object x. For example in Φ_1, an implication $\tau : [age, senior] \Rightarrow [salary, high]$ occurs in objects 1 and 3. Therefore, there are τ^1 and τ^3. If τ^x is the unique implication from an object x, we say τ^x is $definite$, and otherwise we say τ^x is $indefinite$. In this example, τ^1 is indefinite and τ^3 is definite.

In a DIS, the following holds for each $y \in [x]_{CON} \cap [x]_{DEC}$ (CON: condition attributes, DEC: decision attributes).

$support(\tau^y)$=$support(\tau^x)$, $accuracy(\tau^y)$=$accuracy(\tau^x)$.

Therefore, we may identify τ^x with τ. However in a NIS, this may not hold. The property of each τ^1 and τ^3 is slightly different, namely the one is indefinite and the other is definite. If there is at least one τ^x satisfying constraint, we see this τ^x is the *evidence* for causing τ is a rule. There may be other τ^y not satisfying the constraint. We employ this strategy for rule generation in a NIS.

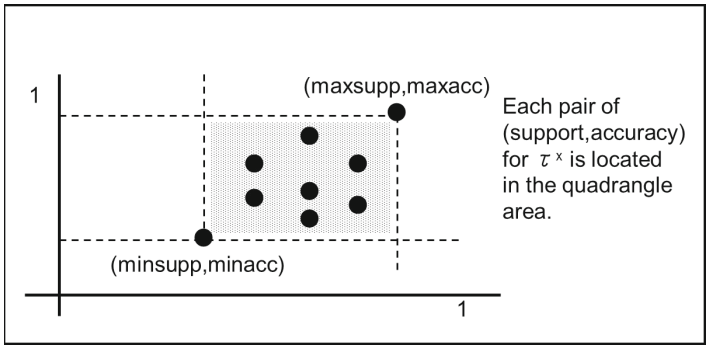


Fig. 2. A distribution of pairs (*support*,*accuracy*) for τ^x. There exists $\phi_{min} \in DD(\tau^x)$ which makes both $support(\tau^x)$ and $accuracy(\tau^x)$ the minimum. There exists $\phi_{max} \in DD(\tau^x)$ which makes both $support(\tau^x)$ and $accuracy(\tau^x)$ the maximum. We denote such quantities as *minsupp*, *minacc*, *maxsupp* and *maxacc*, respectively.

Let $DD(\Phi)$ and $DD(\tau^x)$ denote $\{\phi \mid \phi$ is a derived $DISs$ from NIS $\Phi\}$ and $\{\phi \in DD(\Phi) \mid \tau^x$ occurs in ϕ $\}$, respectively. According to rule generation (employing *support* and *accuracy*) in $DISs$ [9], rule generation with missing values [3, 4] and data mining in transaction data [1], we defined the next tasks in rule generation in $NISs$ [11].

Specification of the Rule Generation Tasks in a NIS
Let us consider the threshold values α and β $(0 < \alpha, \beta \leq 1)$.

(The Lower System: Certain rule generation task) Find each definite implication τ^x such that $support(\tau^x) \geq \alpha$ and $accuracy(\tau^x) \geq \beta$ hold in each $\phi \in DD(\tau^x)$. We say such τ is a *certain* rule and τ^x is an evidence of supporting τ in a NIS.

(The Upper System: Possible rule generation task) Find each implication τ^x such that $support(\tau^x) \geq \alpha$ and $accuracy(\tau^x) \geq \beta$ hold in some $\phi \in DD(\tau^x)$. If such τ is not certain rule, we say τ is a *possible* rule and τ^x is an evidence of supporting τ in a NIS.

Both the above tasks depend on $|DD(\tau^x)|$. In [11], we proved some simplifying results illustrated by Figure 2. We also showed how to effectively compute $support(\tau^x)$ and $accuracy(\tau^x)$ for ϕ_{min} and ϕ_{max} independently from $|DD(\tau^x)|$. Due to Figure 2, we have the following equivalent specification.

Equivalent Specification of the Rule Generation Tasks in a NIS

(The Lower System: Certain rule generation task) Find each definite τ^x such that $minsupp(\tau^x) \geq \alpha$ and $minacc(\tau^x) \geq \beta$ (see Figure 2).

(The Upper System: Possible rule generation task) Find each implication τ^x such that $maxsupp(\tau^x) \geq \alpha$ and $maxacc(\tau^x) \geq \beta$ (see Figure 2).

Example. In $NIS\ \Phi_1$, we at first generate two blocks inf and sup for each descriptor. These two blocks are the extensions from Grzymała-Busse's *blocks* [3, 4], and inf defines the minimum equivalence class. On the other hand, sup defines the maximum equivalence class. For example,

$inf([age, s])=\{3\}$, $sup([age, s])=\{1, 2, 3\}$,
$inf([salary, h])=\{1, 3\}$, $sup([salary, h])=\{1, 3\}$.

Since $sup([age, s]) \cap sup([salary, h])=\{1, 3\}$, we know there are τ^1 and τ^3 for an implication $\tau : [age, s] \Rightarrow [salary, h]$. As for τ^3, $3 \in inf([age, s]) \cap inf([salary, h])$ holds, so we know τ^3 is definite. In this case, we have the following.

$minsupp(\tau^3)=(|inf([age, s]) \cap inf([salary, h])|)/3=|\{3\}|/3=1/3$.
$minacc(\tau^3)=\frac{|inf([age,s]) \cap inf([salary,h])|}{(|inf([age,s])|+|OUT|)}=|\{3\}|/(|\{3\}|+|\{2\}|)=1/2$.
$maxsupp(\tau^3)=(|sup([age, s]) \cap sup([salary, h])|)/3=|\{1, 3\}|/3=2/3$.
$maxacc(\tau^3)=\frac{|sup([age,s]) \cap sup([salary,h])|}{(|inf([age,s])|+|IN|)}=|\{1, 3\}|/(|\{3\}|+|\{2\}|)=2/2=1.0$.
$OUT=(sup([age, s]) \setminus inf([age, s])) \setminus inf([salary, h])$,
$IN=(sup([age, s]) \setminus inf([age, s])) \cap sup([salary, h])$.

In the above calculation, we do not handle $DD(\Phi_1)$ at all. By using blocks inf and sup, it is possible to calculate four criterion values. We extended rule generation to $NISs$ and implemented a software tool with *NIS-Apriori* algorithm [11]. *NIS-Apriori* does not depend on the number of derived $DISs$, and the complexity is almost the same as the original *Apriori* algorithm [1].

3 Information Dilution

This section considers a framework of information dilution.

3.1 An Intuitive Example

We at first consider DIS_{16} and Φ_1 in Figure 1. Since a DIS is a special case of a NIS, we can apply *NIS-Apriori* to each DIS. In this case, the lower and the upper systems generate the same rules. The following is the real execution under the decision attribute *salary*, α=0.5 and β=0.6.

```
?-step1. /* Rule generation in DIS16 under α=0.5 and β=0.6 */
File Name for Read Open: dis16.pl.
SUPPORT:0.5, ACCURACY:0.6
===== Lower System ========================================
[1] MINSUPP=0.667, MINACC=0.667
[age,senior] ==> [salary,high] [1,3] /* Obtained rule */
[2] MINSUPP=0.333, MINACC=0.5
(Lower System Terminated)
===== Upper System ========================================
```

```
[1] MAXSUPP=0.667, MAXACC=0.667
           :   :   :
EXEC_TIME=0.0 (sec)
```

We obtained an implication $[age, senior] \Rightarrow [salary, high]$ from DIS_{16}. Now, we consider the 2nd person's tuple $(senior, female, normal)$. If we employ the following replacement,

$senior$ to $[young, senior]$ (semantically $young$ or $senior$),
$female$ to $[male, female]$,
$normal$ to $[low, normal]$,

the 2nd person's tuple is changed to

$([young, senior], [male, female], [low, normal])$.

This is an example of information dilution. There are 8 possible tuples and one of the tuple is actual, so in such case we say the actual tuple is *diluted* with $1/8$ degree. Similarly, DIS_{16} is diluted to Φ_1 with $1/16$ degree in Figure 1. The following is the real execution of rule generation in Φ_1.

```
?-step1. /* Rule generation in Φ1 under α=0.5 and β=0.6 */
File Name for Read Open: Phi1.pl.
SUPPORT:0.5, ACCURACY:0.6
===== Lower System ========================================
(Lower System Terminated)
===== Upper System ========================================
[1] MAXSUPP=0.667, MAXACC=1.0
[age,senior] ==> [salary,high] [1,3] /* Obtained rule */
[2] MAXSUPP=0.333, MAXACC=1.0
[3] MAXSUPP=0.333, MAXACC=1.0
(Upper System Terminated)
EXEC_TIME=0.0 (sec)
```

In this execution, we know that the results (an obtained rule) in Φ_1 is the same as the original DIS_{16}. Namely, DIS_{16} and Φ_1 are equivalent in rule generation, but some actual values are hidden in Φ_1. Even though this example depends on threshold values α=0.5 and β=0.6, these DIS_{16} and Φ_1 give an example of *information dilution with obtainable rules preserved*.

Figure 3 shows the chart of information dilution, namely the relation between a DIS, a NIS and obtained rules. In data mining, we usually do not open the original data set, namely a DIS, to save privacy-preserving. However, we may open the diluted data set, namely a NIS, because some data in a NIS are changed to disjunctive information. We may consider diluting some specified

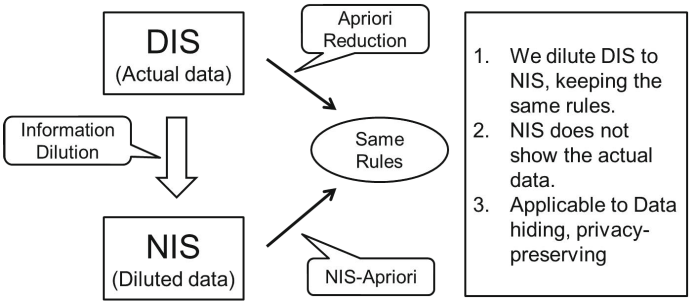


Fig. 3. Formalization of information dilution with constraint

person's data intentionally. Like this, information dilution may take the role of hiding the actual values in a table.

3.2 Some Properties and a Formalization of a Problem

Now, we confirm the following facts.

(Fact 1). A DIS ϕ is diluted to a NIS Φ.
(Fact 2). NIS-$Apriori$ is applicable to Φ.
(Fact 3). For Φ diluted from ϕ, each rule in ϕ is obtainable by the upper system in Φ.

(Fact 3) is the key background. Let us suppose an implication τ^x satisfies $support(\tau^x) \geq \alpha$ and $accuracy(\tau^x) \geq \beta$ in ϕ, and ϕ is diluted to Φ. Then, we know $\phi \in DD(\tau^x) \subseteq DD(\Phi)$. According to the specification of the upper system, τ satisfies the condition of a possible rule, namely τ is obtainable in the upper system. However, we also have a problem. For $\phi' \in DD(\Phi)$ $(\phi' \neq \phi)$, the upper system may pick up another implication η as a possible rule. Therefore, we need to know the next fact.

(Fact 4). For Φ diluted from ϕ, some rules not related to ϕ may be obtained by the upper system in Φ. We name such rules *unexpected rules*.
(Fact 5). If we dilute much more attribute values, we may have much more unexpected rules. On the other hand, if we dilute less attribute values, we will have less unexpected rules.

According to five facts, we have the problem in the following.

(Problem of Information Dilution). Dilute a DIS ϕ to a NIS Φ so as not to generate any unexpected rules.

4 A Example on an Algorithm for Information Dilution

We are now starting this work, and we are considering how to dilute a DIS to a NIS. Therefore, we employ an exemplary DIS ϕ_1 in Table 1 for considering an algorithm. For simplicity, we fix constraint such that the decision attribute is D, $maxsupp(\tau^x) = \alpha > 0$ and $maxacc(\tau^x) = \beta = 1.0$. In this example, we dilute ϕ_1 to a NIS with obtainable 7 rules preserved in Table 2. We can easily obtain them by using a software tool [10].

Table 1. An exemplary DIS ϕ_1. Here, $Domain_A$=$\{1, 2, 3\}$, $Domain_B$=$\{1, 2\}$, $Domain_C$=$\{1, 2\}$ and $Domain_D$=$\{1, 2\}$.

OB	A	B	C	D
1	3	1	1	1
2	2	1	1	1
3	1	1	1	2
4	3	1	2	1
5	3	1	1	1
6	2	2	2	2
7	1	2	1	2
8	2	2	2	2

Table 2. Seven rules in ϕ_1

	Rules	*Objects*
(Imp 1)	[A,1]==>[D,2]	[3,7]
(Imp 2)	[A,3]==>[D,1]	[1,4,5]
(Imp 3)	[B,2]==>[D,2]	[6,7,8]
(Imp 4)	[A,2]&[B,1]==>[D,1]	[2]
(Imp 5)	[A,2]&[C,1]==>[D,1]	[2]
(Imp 6)	[A,2]&[C,2]==>[D,2]	[6,8]
(Imp 7)	[B,1]&[C,2]==>[D,1]	[4]

4.1 Reduction and Dilution

Reduction seems to be applicable to information dilution, namely we apply reduction to a table, and we replace non-necessary attribute values with the set of all attribute values. However, this way is not sufficient for preserving the rules.

In ϕ_1, the degree of data dependency from $\{A, B, C\}$ to $\{D\}$ is 1.0, and 8 objects are consistent for condition attributes A, B, C and decision attribute D. In reduction, we have a tuple $(3, -, -, 1)$ from object 1, 4, 5, and a tuple $(1, -, -, 2)$ from object 3, 7, because they are still consistent. After this reduction, it seems possible to replace each $-$ symbol with all attribute values, i.e., $[1, 2]$. Like this we have a tuple $(1, [1, 2], [1, 2], 2)$ from object 3. In this tuple, we need to consider four cases $(1, 1, 1, 2)$, $(1, 1, 2, 2)$, $(1, 2, 1, 2)$ and $(1, 2, 2, 2)$. An implication $\tau^2 : [B, 1]\&[C, 1] \Rightarrow [D, 1]$ contradicts to $\eta^3 : [B, 1]\&[C, 1] \Rightarrow [D, 2]$ related to the tuple $(1, 1, 1, 2)$. However in other three cases, τ^2 does not contradict to any implication, and τ^2 becomes the unexpected rule.

4.2 Base Step Dilution: Dilution in Each Attribute

We propose a dilution process related to reduction. We start with NIS Φ_2 in Table 3, then we fix some attribute values which induce 7 rules.

Table 3. NIS Φ_2 at the beginning

OB	A	B	C	D
1	$\{1,2,3\}$	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
2	$\{1,2,3\}$	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
3	$\{1,2,3\}$	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
4	$\{1,2,3\}$	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
5	$\{1,2,3\}$	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
6	$\{1,2,3\}$	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
7	$\{1,2,3\}$	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
8	$\{1,2,3\}$	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$

(Step 1-1). In order to generate (Imp 1), (Imp 2) and (Imp 3), we fix $[A,3]$ and $[D,1]$ in object $1 \in [1,4,5]$, $[A,1]$ and $[D,2]$ in object $7 \in [3,7]$, $[B,2]$ and $[D,2]$ in object $8 \in [6,7,8]$.
(Step 1-2). In order to generate inconsistency, we fix $[A,2]$ and $[D,1]$ in object 2, $[A,2]$ and $[D,2]$ in object 6, $[B,1]$ and $[D,1]$ in object 2, $[B,1]$ and $[D,2]$ in object 3, $[C,1]$ and $[D,1]$ in object 2, $[C,1]$ and $[D,2]$ in object 3, $[C,2]$ and $[D,1]$ in object 4, $[C,2]$ and $[D,2]$ in object 6.

Table 4. NIS Φ_3 after the base step

OB	A	B	C	D
1	$\{3\}$	$\{1,2\}$	$\{1,2\}$	$\{1\}$
2	$\{2\}$	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1,2,3\}$	$\{1\}$	$\{1\}$	$\{2\}$
4	$\{1,2,3\}$	$\{1,2\}$	$\{2\}$	$\{1\}$
5	$\{1,2,3\}$	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$
6	$\{2\}$	$\{1,2\}$	$\{2\}$	$\{2\}$
7	$\{1\}$	$\{1,2\}$	$\{1,2\}$	$\{2\}$
8	$\{1,2,3\}$	$\{2\}$	$\{1,2\}$	$\{2\}$

After these two steps, we have Φ_3 in Table 4. Since three implications (Imp 1), (Imp 2) and (Imp 3) appear in each derived DIS, they are also rules in the upper system. We have the next important fact.

(Fact 6). Any implication $\tau^x : [A,1]\&Condition\_part \Rightarrow [D,2]$ in a derived DIS $\phi \in DD(\Phi_3)$ is redundant for (Imp 1). Therefore, $accuracy(\tau^x) = 1.0$ holds in this ϕ. Any implication $\eta^y : [A,1]\&Condition\_part \Rightarrow [D,1]$ in a derived DIS $\phi' \in DD(\Phi_3)$ is inconsistent, because (Imp 1) also appears in this ϕ'. Therefore, $accuracy(\eta^y) < 1.0$ holds in this ϕ'. According to the above consideration, we do not have to pay any attention to any implication with a descriptor $[A,1]$. The same holds for descriptors $[A,3]$, $[B,2]$.

4.3 Recursive Steps Dilution: Dilution in a Set of Attributes

Similarly to the base step, we fix some attribute values for (Imp 4), (Imp 5), (Imp 6) and (Imp 7).

(Step 2-1). The attribute values of (Imp 4) and (Imp 5) are fixed in Φ_3. We fix $[B, 1]$ in object 4 and $[C, 2]$ in object 6.

According to (Fact 6), we do not have to consider any implication including descriptors $[A, 1]$, $[A, 3]$ and $[B, 2]$. It is enough to consider descriptors $[A, 2]$, $[B, 1]$, $[C, 1]$ and $[C, 2]$. Then, we have 10 implications, where unexpected rules may exist.

```
(1) [A,2]&[B,1] ==> [D,1],  (2) [A,2]&[B,1] ==> [D,2],
(3) [A,2]&[C,1] ==> [D,1],  (4) [A,2]&[B,1] ==> [D,2],
(5) [A,2]&[C,2] ==> [D,1],  (6) [A,2]&[C,2] ==> [D,2],
(7) [B,1]&[C,1] ==> [D,1],  (8) [B,1]&[C,1] ==> [D,2],
(9) [B,1]&[C,2] ==> [D,1],  (10) [B,1]&[C,2] ==> [D,2].
```

(Step 2-2). Here, (1) is (Imp 4), (3) is (Imp 5). They are obtainable in object 2. (6) is (Imp 6), which is obtainable in object 6. (9) is (Imp 7), and we fix $[B, 1]$ in object 4. According to (Fact 6), any of (2), (4), (5) and (10) does not satisfy $accuracy(\tau^x)$=1.0 in any derived $DISs$. (7) in object 2 and (8) in object 3 are inconsistent in any derived $DISs$.

After (Step 2-1) and (Step 2-2), we have Φ_4 below.

Table 5. NIS Φ_4 after the 2nd step

OB	A	B	C	D
1	{3}	{1, 2}	{1, 2}	{1}
2	{2}	{1}	{1}	{1}
3	{1, 2, 3}	{1}	{1}	{2}
4	{1, 2, 3}	{1}	{2}	{1}
5	{1, 2, 3}	{1, 2}	{1, 2}	{1, 2}
6	{2}	{1, 2}	{2}	{2}
7	{1}	{1, 2}	{1, 2}	{2}
8	{1, 2, 3}	{2}	{1, 2}	{2}

In Φ_4, all 7 implications (Imp 1) to (Imp 7) are all obtainable. There is a conjunction of descriptors $[B, 1]\&[C, 1]$ which causes inconsistency, so we need to consider a conjunction of descriptors $[A, \_]\&[B, 1]\&[C, 1]$. However, such conjunction is redundant, and we do not have to consider it. The following is the real execution. If there is an implication τ^x, $maxsupp(\tau^x)$>0.1 holds. Therefore, we set α=0.1 instead of α>0.

```
?-step1. /* Rule p ⇒ q in Φ4 under α=0.1 and β=1.0 */
File Name for Read Open: Phi4.pl.
SUPPORT:0.1, ACCURACY:1.0
===== Lower System ========================================
           :   :   :
(Next Candidates are Remained) [[[1,1],[4,2]],[[1,2],[4,1]], :::
===== Upper System ========================================
[1] MAXSUPP=0.125, MAXACC=0.5
[2] MAXSUPP=0.375, MAXACC=1.0
[a,1] ==> [d,2] [3,7,8]   /* (Imp 1) in φ1 */
           :   :   :
[5] MAXSUPP=0.375, MAXACC=1.0
[a,3] ==> [d,1] [1,4,5]   /* (Imp 2) in φ1 */
           :   :   :
[10] MAXSUPP=0.5, MAXACC=1.0
[b,2] ==> [d,2] [5,6,7,8]   /* (Imp 3) in φ1 */
(Next Candidates are Remained) [[[1,2],[4,1]],[[1,2],[4,2]], :::
EXEC_TIME=0.0 (sec)

?-step2. /* Rule p1&p2 ⇒ q in Φ4 under α=0.1 and β=1.0 */
===== Lower System ========================================
           :   :   :
(Next Candidates are Remained) [[[1,2],[2,1],[4,1]],[[1,2], :::
===== Upper System ========================================
[1] MAXSUPP=0.375, MAXACC=1.0
[a,2]&[b,1] ==> [d,1] [2,4,5]   /* (Imp 4) in φ1 */
           :   :   :
[3] MAXSUPP=0.25, MAXACC=1.0
[a,2]&[c,1] ==> [d,1] [2,5]   /* (Imp 5) in φ1 */
           :   :   :
[6] MAXSUPP=0.375, MAXACC=1.0
[a,2]&[c,2] ==> [d,2] [5,6,8]   /* (Imp 6) in φ1 */
           :   :   :
[9] MAXSUPP=0.375, MAXACC=1.0
[b,1]&[c,2] ==> [d,1] [1,4,5]   /* (Imp 7) in φ1 */
           :   :   :
(Next Candidates are Remained) [[[2,1],[3,1],[4,1]], :::
EXEC_TIME=0.0 (sec)

?-step3. /* Rule p1&p2&p3 ⇒ q in Φ4 under α=0.1 and β=1.0 */
===== Lower System ========================================
[1] MINSUPP=0.125, MINACC=0.333
           :   :   :
[4] MINSUPP=0.0, MINACC=0.0
(Lower System Terminated)
```

```
===== Upper System ========================================
(Upper System Terminated)
EXEC_TIME=0.0 (sec)
```

In step 1, we obtained three implications (Imp 1), (Imp 2) and (Imp 3) in the upper system. In step 2, we obtained four implications (Imp 4) to (Imp 7) in the upper system. In step 3, we obtained no implications. In view of the above results, we have the following:

$$\{\tau \mid \tau \text{ is either a possible rule or a certain rule in } \Phi_4\}=\{\tau \mid \tau \text{ is a rule in } \phi_1\}.$$

This means that Φ_4 and ϕ_1 are equivalent in rule generation, and they are satisfying the formalization of Figure 3. Each tuple of ϕ_1 stores the actual values, therefore we should not open ϕ_1. However, it may be possible to open Φ_4, because some attribute values are diluted. Especially, the tuple of object 5 is completely diluted.

5 Concluding Remarks

We have proposed a framework of information dilution, which depends on the research on $RNIA$ (*Rough Non-deterministic Information Analysis*) and NIS-*Apriori* algorithm. This is an attempt to apply information incompleteness and $RNIA$ to the randomization and the perturbation in privacy-preserving [2].

We investigated the formal algorithm of diluting a DIS and its implementation. In Figure 1, we unexpectedly obtained that rules in DIS_{16} and Φ_1 are the same under $support \geq 0.5$ and $accuracy \geq 0.6$. In this paper, we handled the most simple case $support > 0$ and $accuracy$=1.0. The procedure proposed in this paper is a preliminary work towards more general cases.

In Φ_4 and ϕ_1, 13 attribute values are diluted for totally 32 attribute values. The ratio is about 1/3. We figure that this ratio is depending on the number of rules and total number of objects. Furthermore, (Fact 6) seems very important. If most descriptors are fixed in the base step, the number of implications are reduced in the recursive steps. Like several variations of reduction with several constraints, there may be several variations of information dilution.

Acknowledgment. The authors would be grateful for anonymous referees for their useful comments.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. of VLDB, pp. 487–499 (1994)
2. Aggarwal, C., Yu, P.: Privacy-Preserving Data Mining. In: Advances in Database Systems, vol. 34. Springer (2008)
3. Grzymała-Busse, J.: A new version of the rule induction system LERS. Fundamenta Informaticae 31, 27–39 (1997)

4. Grzymała-Busse, J., Rząsa, W.: A local version of the MLEM2 algorithm for rule induction. Fundamenta Informaticae 100, 99–116 (2010)
5. Lipski, W.: On semantic issues connected with incomplete information data base. ACM Trans. DBS 4, 269–296 (1979)
6. Lipski, W.: On databases with incomplete information. Journal of the ACM 28, 41–70 (1981)
7. Nakata, M., Sakai, H.: Twofold rough approximations under incomplete information. International Journal of General Systems 42(6), 546–571 (2013)
8. Orłowska, E., Pawlak, Z.: Representation of nondeterministic information. Theoretical Computer Science 29, 27–39 (1984)
9. Pawlak, Z.: Rough Sets. Kluwer Academic Publishers (1991)
10. RNIA software logs, http://www.mns.kyutech.ac.jp/~sakai/RNIA
11. Sakai, H., Ishibashi, R., Nakata, M.: On rules and apriori algorithm in non-deterministic information systems. Transactions on Rough Sets 9, 328–350 (2008)
12. Sakai, H., Okuma, H., Nakata, M.: Rough non-deterministic information analysis: Foundations and its perspective in machine learning. In: Smart Innovation, Systems and Technologies, ch. 9, vol. 13, pp. 215–247. Springer (2013)
13. Sakai, H., Okuma, H., Wu, M., Nakata, M.: Rough non-deterministic information analysis for uncertain information. In: The Handbook on Reasoning-Based Intelligent Systems, ch. 4, pp. 81–118. World Scientific (2013)
14. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Intelligent Decision Support. Handbook of Advances and Applications of the Rough Set Theory, pp. 331–362. Kluwer Academic Publishers (1992)

Belief Discernibility Matrix and Function for Incremental or Large Data

Salsabil Trabelsi[1], Zied Elouedi[1], and Pawan Lingras[2]

[1] Larodec, Institut Superieur de Gestion de Tunis, University of Tunis, Tunisia
[2] Saint Mary's University Halifax, Canada

Abstract. This paper proposes an incremental attribute selection method based on rough sets from partially uncertain and incremental or large decision system. The uncertainty exists only in the decision attributes (classes) and is represented by the belief function theory. The simplification of large or incremental uncertain decision table is based on computing possible reducts by the means of belief discernibility matrix and function under the belief function framework from two or more sub-decision tables.

Keywords: Uncertainty, incremental data, belief function theory, rough sets, attribute selection, discernibility matrix and function.

1 Introduction

Feature selection is an important topic in data mining, especially for high dimensional datasets to take away the unnecessary attributes [4,5,6]. The discernibility matrix and function was proposed by Skowron and Rauszer [8] to select relevant features from data using rough set theory. It provides an easy approach to compute the possible reducts and the core of a decision table. However, the original discernibility matrix and function cannot deal with uncertain decision system. In previous works [13,14], we have proposed belief discernibility matrix and function to compute the possible reducts and core from partially uncertain data. The uncertainty exists only in the decision attribute and is represented by the belief function theory. It is considered as a useful theory for representing and managing total or partial uncertain knowledge because of its relative flexibility. In addition, this theory is not competitive but complementary to the rough set theory and can be often used jointly with it [12]. In this paper, we use the Transferable Belief Model (TBM), one interpretation of belief function theory [11].

Our earlier solutions [13,14] are not suitable for large databases characterized by huge number of instances, attributes and attribute values. Besides, these solutions are not suitable for incremental data where the complete training set is not given at the beginning. This paper proposes incremental belief discernibility matrix and function to solve the dynamically changing big data by gathering the results obtained from many parts of datasets. The original discernibility matrix and function from large and incremental data relative to crisp decision tables was proposed in [2]. The proposed new feature selection method deals

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 67–76, 2013.

with both incremental and partially uncertain data. The rest of the paper is organized as follows: Section 2 provides an overview of the rough set theory. Section 3 introduces the belief function theory as understood in the Transferable Belief Model (TBM). Section 4 describes the attribute selection method based on rough sets under uncertainty to deal with large and incremental data. In Section 5, experimental results have been done to show the efficiency of the solution.

2 Rough Set Theory

In this section, we give some notions related to information systems and rough sets [3,4]. An information system is a pair $A = (\mathrm{U}, \mathrm{C})$, where U is the universe of discourse with a finite number of objects (or entities) and C is a non-empty, finite set of attributes. We also consider a special case of information systems called decision tables. A decision table is an information system of the form $DT = (\mathrm{U}, \mathrm{C} \cup \{\mathrm{d}\})$, where $d \notin C$ is a distinguished attribute called *decision*. In this paper, the notation $c_i(x_j)$ is used to represent the value of a condition attribute $c_i \in C$ for $x_j \in U$. For every set of attributes $B \subseteq C$, an equivalence relation denoted by IND_B and called the B-indiscernibility relation, is defined by

$$IND_B = U/B = \{[x_j]_B | x_j \in U\} \tag{1}$$

where

$$[x_j]_B = \{x_i | \forall c \in B\, c(x_i) = c(x_j)\} \tag{2}$$

Let $B \subseteq C$ and $X \subseteq U$. We can approximate X by constructing the $B-lower$ and $B-upper\ approximations$ of X, denoted $\underline{\mathrm{B}}(\mathrm{X})$ and $\bar{\mathrm{B}}(\mathrm{X})$, respectively, where

$$\underline{\mathrm{B}}(X) = \{x_j | [x_j]_B \subseteq X\}\ and\ \bar{\mathrm{B}}(X) = \{x_j | [x_j]_B \cap X \neq \emptyset\} \tag{3}$$

2.1 Reduct and Core

A reduct [5,6] is a minimal subset of attributes from C that preserves the positive region and the ability to perform classifications as the entire attributes set C. A subset $B \subseteq C$ is a reduct of C with respect to d, iff B is minimal and:

$$Pos_B(\{d\}) = Pos_C(\{d\}) \tag{4}$$

where $Pos_C(\{d\})$ is called a positive region of the partition $U/\{d\}$ with respect to C.

$$Pos_C(\{d\}) = \bigcup_{X \in U/\{d\}} \underline{\mathrm{C}}(X) \tag{5}$$

The core is the most important subset of attributes, it is included in every reduct.

$$Core(DT, \{d\}) = \bigcap RED(A, \{d\}) \tag{6}$$

where $RED(DT, \{d\})$ is the set of all reducts of DT relative to d.

2.2 Discernibility Matrix and Function

The discernibility matrix and function [8,9] are a way to compute reducts and cores from decision table which are defined below. Let DT be a decision table with n objects. The discernibility matrix $M(DT)$ is a symmetric n*n matrix with entries $M_{i,j}$. Each entry consists of the set of attributes upon which objects x_i and x_j differ. $For\ i, j = 1, ..., n$

$$M_{i,j} = \{c \in C | c(x_i) \neq c(x_j)\ and\ d(x_i) \neq d(x_j)\} \tag{7}$$

Thus, entry $M_{i,j}$ is the set of all attributes which discern objects x_i and x_j that do not belong to the same equivalence class $IND_{\{d\}}$. A discernibility function $f(DT)$ for a decision table DT is a boolean function of k boolean variables $c_1^*...c_k^*$ (corresponding to the attributes $c_1...c_k$) defined as follows, where $M_{i,j}^* = \{c^* | c \in M_{i,j}\}$.

$$f(DT) = \wedge\{\vee M_{i,j}^* | 1 \leq j \leq i \leq n, M_{i,j} \neq \emptyset\} \tag{8}$$

where $\wedge$ and $\vee$ are two logical operators for conjunction and disjunction. The set of all prime implicants of $f(DT)$ determines the sets of all reducts of DT.

3 Belief Function Theory

The belief function theory was proposed by Shafer [7] as a useful tool to represent uncertain knowledge. Here, we introduce only some basic notations related to the TBM [11], one interpretation of the belief function theory. Let Θ, called a frame of discernment, be a finite set of exhaustive elements to a given problem. All the subsets of Θ belong to the power set of Θ, denoted by 2^Θ. The bba (basic belief assignment) is a function representing the impact of a piece of evidence on the subsets of the frame of discernment Θ and is defined as follows:

$$m : 2^\Theta \rightarrow [0, 1]$$

$$\sum_{E \subseteq \Theta} m(E) = 1 \tag{9}$$

where *m(E)* is a basic belief mass (bbm) that shows the part of belief exactly committed to the element E. The conjunctive rule is used to combine the bba's induced from distinct pieces of evidence [10]:

$$(m_1 \bigcirc\!\!\!\!\cap\, m_2)(E) = \sum_{F,G \subseteq \Theta : F \cap G = E} m_1(F) \times m_2(G) \tag{10}$$

To make decisions from beliefs, the TBM [10] proposes using the pignistic probabilities denoted $BetP$ which are defined as :

$$BetP(\{a\}) = \sum_{F \subseteq \Theta} \frac{\mid \{a\} \cap F \mid}{\mid F \mid} \frac{m(F)}{(1 - m(\emptyset))} \text{ for all } \ a \in \Theta \tag{11}$$

4 Incremental Belief Discernibility Matrix and Function

In this section, we will first give an overview of the uncertain decision table followed by a description of the belief discernibility matrix and function [14]. Belief discernibility matrix and function can be used to get reducts and cores of uncertain decision tables, but they do not work very well with incremental or large data. Therefore, we propose an incremental method for attribute selection from partially uncertain and incremental decision system based on rough sets under the belief function framework by gathering the results obtained from two or many belief discernibility matrices and functions.

4.1 Uncertain Decision Table

Our uncertain decision table denoted UDT contains n objects x_j, characterized by a set of certain condition attributes C=$\{c_1, c_2,...,c_k\}$ and uncertain decision attribute ud. We propose to represent the uncertainty of each object by a bba m_j expressing belief on decision defined on the frame of discernment Θ=$\{ud_1, ud_2,...,ud_s\}$ representing the possible values of ud.

Example: Let us use Table 1 to describe our uncertain decision system. It contains five objects, three certain condition attributes C=$\{$a, b, c$\}$ and an uncertain decision attribute ud with possible value $\{$yes, no$\}$ representing Θ. For example, for the object x_2, belief of 0.6 is exactly committed to the decision ud_2=no, whereas belief of 0.4 is assigned to the entire frame of discernment Θ (ignorance).

Table 1. Uncertain Decision Table 1 (UDT_1)

U	a	b	c	ud
x_1	0	1	1	$m_1(\{yes\}) = 0.95 \quad m_1(\Theta) = 0.05$
x_2	1	0	2	$m_2(\{no\}) = 0.6 \quad m_2(\Theta) = 0.4$
x_3	1	0	2	$m_3(\{no\}) = 1$
x_4	1	1	1	$m_4(\{no\}) = 0.95 \quad m_4(\Theta) = 0.05$
x_5	0	0	1	$m_5(\{yes\}) = 1$

4.2 Belief Discernibility Matrix and Function

In order to compute the possible reducts from our uncertain decision table, we have previously proposed [13,14] the concepts of belief discernibility matrix $M'(UDT)$ and function $f'(UDT)$. The belief discernibility matrix was based on a distance measure to identify the similarity or dissimilarity between two bba's m_i and m_j. The threshold value is used to provide flexibility. Hence, belief discernibility matrix $M'(UDT)$ is a n*n matrix with entries $M'_{i,j}$.
For i,j=1,...,n

$$M'_{i,j} = \{c \in C | c(x_i) \neq c(x_j) \; and \, dist(m_i, m_j) \geq threshold\} \qquad (12)$$

where *dist* is a distance measure between two bba's proposed in [1] as defined below.

The belief discernibility function $f'(UDT)$ for the uncertain decision table UDT is equivalent to the certain discernibility function $f(DT)$ only it is computed from the belief discernibility matrix. The latter has the same structure as the certain discernibility matrix. The belief discernibility function is a boolean function of m boolean variables $c_1^*...c_m^*$ (corresponding to the attributes $c_1...c_m$) defined as below, where $M_{ij}'^* = \{c^* | c \in M'_{i,j}\}$

$$f'(UDT) = \wedge\{\vee M_{i,j}'^* | 1 \leq j \leq i \leq n, M'_{i,j} \neq \emptyset\} \tag{13}$$

where $\wedge$ and $\vee$ are two logical operators for conjunction and disjunction. The set of all prime implicants of $f'(UDT)$ determines the sets of all reducts of UDT.

Example: To apply our feature selection method to the uncertain decision table 1 (see Table 1), we start by computing the belief discernibility matrix (see Table 2). To obtain Table 2, we use Equation (12) with a threshold value equal to 0.1. For example, $M'_{1,5} = \emptyset$ because the two objects x_1 and x_5 have $dist(m_1, m_5) = 0.07 \leq 0.1$. The decision values of the two objects are considered similar.

Table 2. Belief discernibility matrix (M'(UDT_1))

U	x_1	x_2	x_3	x_4	x_5
x_1					
x_2	a,b,c				
x_3	a,b,c				
x_4	a	b,c			
x_5		a,c	a,c	a,b	

Next, we compute the possible reducts by computing the discernibility function. $f'(UDT) = (a \vee b \vee c) \wedge (a) \wedge (a \vee c) \wedge (b \vee c) \wedge (a \vee b) = (a \wedge b) \vee (a \wedge c)$. We find two possible reducts: {a and b} or {a and c}.

4.3 Belief Discernibility Matrix and Function for Large or Incremental Data

Since the proposed belief discernibility matrix and function [13,14] are not suitable for large and incremental data, we propose an incremental belief discernibility matrix and function computed from two uncertain decision tables. The original discernibility matrix and function from large and incremental data relative to crisp decision tables was proposed in [2]. The method has the following merits:

1. Work efficiently with big data.
2. Work very well with incremental data.
3. Disassemble decision tables into parts, and then "divide and conquer".
4. Suitable for parallel computing.

We will adapt their work on our uncertain context as follows. Let us define three uncertain decision tables $UDT_1 = (U_1, C \cup \{ud\})$, $UDT_2 = (U_1, C \cup \{ud\})$ and $UDT = (U, C \cup \{ud\})$ where $U_1 = \{x_1, x_2, ..., x_n\}$, $U_2 = \{y_1, y_2, ..., y_m\}$, C= $\{c_1, c_2, ..., c_k\}$, $U = U_1 \cup U_2$.

$M'(UDT_1)$ and $M'(UDT_2)$ are belief discernibility matrices of respectively UDT_1 and UDT_2 computed using equation (12). The $M'(UDT)$ is the belief discernibility matrix relative to UDT and can be computed as follows:

$$M'(UDT) = \begin{pmatrix} M'(UDT_1) & M'(UDT_1, UDT_2) \\ & M'(UDT_2) \end{pmatrix} \tag{14}$$

where $M'(UDT_1, UDT_2)$ is the belief discernibility matrix between two uncertain decision tables UDT_1 and UDT_2. It is a n*m matrix where each entry $M'_{i,j}$ is defined as follows:

$For\, i = 1, ..., n\, and\, j = 1, ..., m$

$$M'_{i,j} = \{c \in C | c(x_i) \neq c(y_j)\ and\, dist(m_i, m_j) \geq threshold\} \tag{15}$$

Let f'_1 and f'_2 be the belief discernibility functions of respectively UDT_1 and UDT_2 computed using equation (16). f' is the belief discernibility function of the whole UDT and is defined as follows:

$$f' = f'_1 \bigwedge f'_2 \bigwedge f'_{1,2} \tag{16}$$

where $f'_{1,2}$ is the discernibility function between UDT_1 and UDT_2 relative to the belief discernibility matrix $M'(UDT_1, UDT_2)$ and is defined as follows:

$$f'_{1,2} = \wedge\{\vee M'^{*}_{i,j} | 1 \leq j \leq i \leq n, M'_{i,j} \neq \emptyset\} \tag{17}$$

Example: To understand the notions of incremental discernibility matrix and function for the uncertain case, let us take another uncertain decision table (UDT_2) (see Table 3). The second decision table could be considered to be an incremental addition to the first.

Table 3. Uncertain Decision Table 2 (UDT_2)

U	a	b	c	ud
y_1	0	0	0	$m_1(\{yes\}) = 0.95 \quad m_1(\Theta) = 0.05$
y_2	0	1	2	$m_2(\{no\}) = 1$

We start by computing its relative belief discernibility matrix (see Table 4). Then, we compute the belief discernibility matrix between UDT_1 and UDT_2 to obtain Table 5.

Table 4. Belief discernibility matrix 2 ($M'(UDT_2)$)

U	y_1	y_2
y_1		
y_2	a,b	

Table 5. Belief discernibility matrix ($M'(UDT_1, UDT_2)$)

U	x_1	x_2	x_3	x_4	x_5
y_1		a,c	a,c	a,b,c	
y_2	c	a,b			b,c

Finally, we compute belief discernibility matrix of the whole UDT (see Table 6). f' is the belief discernibility function of the whole UDT and is equal to: $f' = (a \vee b \vee c) \wedge (a) \wedge (c) \wedge (a \vee c) \wedge (b \vee c) \wedge (a \vee b) = (a \wedge c)$. We find only one possible reduct: {a and c}.

Table 6. Belief discernibility matrix (M'(UDT))

U	x_1	x_2	x_3	x_4	x_5	y_1	y_2
x_1							
x_2	a,b,c						
x_3	a,b,c						
x_4	a	b,c					
x_5		a,c	a,c	a,b			
y_1		a,c	a,c	a,b,c			
y_2	c	a,b			b,c	a,b	

5 Experimentation

Several tests were performed on real-world databases to evaluate the proposed incremental feature selection method in comparison with non-incremental feature selection method proposed originally in [14]. The comparison is based on two evaluation criteria: the time requirement (the number of seconds needed to find the reduct) and the classification accuracy (Percent of Correct Classification (PCC)) of the generated decision rules by incorporating the two methods into a classification system called belief rough set classifier [15]. The latter is able to generate uncertain decision rules used for classification process where the feature selection is one of the important steps.

We have tested our methods on standard real-world databases obtained from the U.C.I. repository[1]. A brief description of these databases is presented in Table 7. These databases are of varying sizes (number of instances, number of attributes and number of decision values). Our incremental method can also work on very large databases by dividing them in many parts. The databases were artificially modified in order to include uncertainty in the decision attribute. We took different degrees of uncertainty based on increasing values of probabilities P used to transform the actual decision value d_i of each object x_j to a bba $m_j(\{d_i\}) = 1-P$ and $m_j(\Theta) = P$. A larger P gives a larger degree of uncertainty.

- Low degree of uncertainty: $0 < \mathrm{P} \leq 0.3$
- Middle degree of uncertainty: $0.3 < \mathrm{P} \leq 0.6$
- High degree of uncertainty: $0.6 < \mathrm{P} \leq 1$

Table 7. Description of databases

Databases	#instances	#attributes	#decision values
W. Breast Cancer	690	8	2
Balance Scale	625	4	3
C. Voting records	497	16	2
Zoo	101	17	7
Nursery	12960	8	3
Solar Flares	1389	10	2
Lung Cancer	32	56	3
Hayes-Roth	160	5	3
Car Evaluation	1728	6	4
Lymphography	148	18	4
Spect Heart	267	22	2
Tic-Tac-Toe Endgame	958	9	2

Each database is divided into ten parts. Nine parts are used as the training set, the last is used as the testing set. The procedure is repeated ten times, each time another part is chosen as the testing set. This method, called a cross-validation, permits a more reliable estimation of the evaluation criterion. In this paper, we report the average of the evaluation criteria. Each training set is divided in two parts to simulate the incremental data.

Table 8 reports the experimental results relative to the classification accuracy. From this table, we see that the proposed incremental feature selection method has the same accuracy as the non-incremental method for attribute selection. It is true for all the databases and for all degrees of uncertainty. For example, the mean PCC for Balance Scale database is equal to 83.23% with incremental and non-incremental methods. We can also conclude that when the degree of uncertainty increases there is a slight decline in accuracy.

[1] `http://www.ics.uci.edu/ mlearn/MLRepository.html`

The Table 8 also gives the experimental results relative to the second evaluation criterion, the time requirement needed to simplify the databases. Note that the time requirement is almost the same for different degrees of uncertainty. We conclude from the table that the incremental method feature selection method is faster than the non-incremental method for attribute selection. It is true for all the databases. For example, the time requirement for W. Breast Cancer database goes from 154 seconds with non-incremental method to 101 seconds with incremental method.

Table 8. The PCC and time requirement relative to non-incremental and incremental methods

Databases	Same PCC(%) for non-incremental and incremental methods				Non-incremental method Time (seconds)	Incremental method Time (seconds)
	Low	Middle	High	Mean		
W. Breast Cancer	86.87	86.58	86.18	86.54	154	101
Balance Scale	83.46	83.21	83.03	83.23	129	83
C. Voting records	98.94	98.76	98.52	98.74	110	69
Zoo	96.52	96.47	95.87	95.95	101	63
Nursery	96.68	96.21	96.07	96.32	380	199
Solar Flares	88.67	88.61	88.56	88.61	157	103
Lung Cancer	75.77	75.50	75.33	75.53	48	29
Hayes-Roth	97.96	97.15	96.75	96.95	91	67
Car Evaluation	84.46	84.17	84.01	84.21	178	112
Lymphography	83.24	83.03	82.67	82.64	102	61
Spect Heart	85.34	85.28	85.07	85.23	109	62
Tic-Tac-Toe Endgame	86.26	86.21	86.18	86.21	139	78

6 Conclusion and Future Work

In this paper, we have proposed an incremental feature selection method from large amount of data or incremental data by defining belief discernibility matrices and functions from many parts of uncertain decision tables. We handle uncertainty in decision attributes using the belief function. Experimental results show the efficiency of the method compared with non-incremental feature selection method especially for the time requirement criteria. We have also proposed a belief discernibility matrix and function between two uncertain decision tables.

As a future work, we suggest adapting the concepts of incremental belief discernibility matrix and function to select relevant features from data characterized by uncertain condition attribute values.

References

1. Bosse, E., Jousseleme, A.L., Grenier, D.: A new distance between two bodies of evidence. Information Fusion 2, 91–101 (2001)
2. Deng, D., Huang, H.: A New Discernibility Matrix and Function, Rough Sets and Knowledge Technology, pp. 114–121. Springer (2006)

3. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
4. Pawlak, Z., Zdzislaw, A.: Rough Sets: Theoretical Aspects of Reasoning About Data, 7th edn. Kluwer Academic Publishing, Dordrecht (1991) ISBN 0-7923-1472-7
5. Pawlak, Z., Rauszer, C.M.: Dependency of attributes in Information systems. Bull. Polish Acad. Sci., Math. 33, 551–559 (1985)
6. Rauszer, C.M.: Reducts in Information systems. Fundamenta Informaticae (1990)
7. Shafer, G.: A mathematical theory of evidence. Princeton University Press, Princeton (1976)
8. Skowron, A., Rauszer, C.: The Discernibility Matrices and Functions in Information Systems. In: Slowiski, R. (ed.) Intelligent Decision Support, Handbook of Applications and Advances of the Rough Set Theory, pp. 311–362. Kluwer Academic Publishers, Dordrecht (1992)
9. Skowron, A.: Rough Sets in KDD. Special Invited Speaking. In: WCC 2000 in Beijing (August 2000)
10. Smets, P., Kennes, R.: The transferable belief model. Artificial Intelligence 66, 191–236 (1994)
11. Smets, P.: The transferable belief model for quantified belief representation. In: Gabbay, D.M., Smets, P. (eds.) Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 1, pp. 207–301. Kluwer, Doordrecht (1998)
12. Skowron, A., Grzymala-Busse, J.W.: From rough set theory to evidence theory. In: Advances in the Dempster-Shafer Theory of Evidence, New York, pp. 193–236 (1994)
13. Trabelsi, S., Elouedi, Z., Lingras, P.: Heuristic for Attribute Selection Using Belief Discernibility Matrix. In: Li, T., Nguyen, H.S., Wang, G., Grzymala-Busse, J., Janicki, R., Hassanien, A.E., Yu, H. (eds.) RSKT 2012. LNCS, vol. 7414, pp. 129–138. Springer, Heidelberg (2012)
14. Trabelsi, S., Elouedi, Z., Lingras, P.: Exhaustive search with belief discernibility matrix and function. In: Zaïane, O.R., Zilles, S. (eds.) Canadian AI 2013. LNCS, vol. 7884, pp. 162–173. Springer, Heidelberg (2013)
15. Trabelsi, S., Elouedi, Z., Lingras, P.: Belief rough set classifier. In: Gao, Y., Japkowicz, N. (eds.) AI 2009. LNCS, vol. 5549, pp. 257–261. Springer, Heidelberg (2009)

An Experimental Comparison of Three Interpretations of Missing Attribute Values Using Probabilistic Approximations

Patrick G. Clark[1] and Jerzy W. Grzymała-Busse[1,2]

[1] Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA
[2] Institute of Computer Science, Polish Academy of Sciences, 01–237 Warsaw, Poland
{pclark,jerzy}@ku.edu

Abstract. This paper presents results of experiments on 24 data sets with three different interpretations of missing attribute values: lost values, attribute-concept values, and "do not care" conditions. Lost values were erased or forgotten to be inserted. Attribute-concept values are any values from the attribute domain restricted to the respective concept. "Do not care" conditions are any values from the attribute domain without any restriction. For our experiments we used concept probabilistic approximations, a generalization of standard approximations. Our main objective was to determine the best interpretation of missing attribute values, in terms of the error rate. Results of experiments indicate that the lost value interpretation of missing attribute values is the best. Our secondary objective was to test how useful proper concept probabilistic approximations (i.e., different from standard concept lower and upper approximations) are for mining data with missing attribute values. Proper concept probabilistic approximations were better than standard concept approximations for 12 data sets and worse for five data sets (out of 24).

1 Introduction

Lower and upper approximations are the most fundamental ideas of rough set theory. A probabilistic (or parameterized) approximation, associated with a probability (parameter) α, is a generalization of ordinary lower and upper approximations. If the probability α is quite small, the probabilistic approximation is reduced to an upper approximation; if it is equal to one, the probabilistic approximation becomes a lower approximation [1]. Probabilistic approximations have been studied in areas such as variable precision rough sets, Bayesian rough sets, decision-theoretic rough sets for many years. The idea was introduced in [2] and then discussed in many papers, see, e. g., [3–10].

So far, mostly theoretical properties of probabilistic approximations were discussed. Only recently probabilistic approximations, for completely specified and inconsistent data sets, were experimentally validated in [11]. For incomplete data sets probabilistic approximations were generalized and re-defined in [1]. Results

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 77–86, 2013.

of similar experiments, but restricted to only lost values and "do not care" conditions, were presented in [12].

We will distinguish three kinds of missing attribute values: lost values, attribute-concept values and "do not care" conditions. If an attribute value was originally given but now is not accessible (e.g., erased or forgotten) we will call it lost. If a data set consists of lost values, we will try to induce rules from existing, specified data. Another interpretation of a missing attribute value is based on a refusal to answer a question, e.g., some people may refuse to tell their marital status, such a value will be called a "do not care" condition. For analysis of data sets with "do not care" conditions we will replace them by all specified attribute values. Attribute-concept values are "do not care" conditions restricted to a concept to which the case with missing attribute values belongs. Any attribute-concept value may be replaced by the set of all specified attribute values restricted to a concept to which the case belongs.

For incomplete data sets there exist many definitions of approximations. Following [1], we will use so called concept approximations, generalized to concept probabilistic approximation in [1]. Concept probabilistic approximations different from standard concept lower and upper approximations are called proper.

The main objective of this paper was to determine the best interpretation of missing attribute values. Out of eight data sets, for six data sets lost values provided the smallest error rate. Our secondary objective was to study usefulness of proper concept probabilistic approximations to mining data sets with missing attribute values. Proper concept probabilistic approximations were better than standard concept approximations for 12 data sets and worse for five data sets (out of 24).

2 Data Sets

We assume that the input data sets are presented in the form of a *decision table*. An example of a decision table is shown in Table 1. Rows of the decision table represent *cases*, while columns are labeled by *variables*. The set of all cases will be denoted by U. In Table 1, $U = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Independent variables are called *attributes* and a dependent variable is called a *decision* and is denoted by d. The set of all attributes will be denoted by A. In Table 1, $A = \{$*Temperature*, *Headache*, *Cough*$\}$. The value for a case x and an attribute a will be denoted by $a(x)$.

In this paper we distinguish between three interpretations of missing attribute values: *lost values*, denoted by "?", *"do not care" conditions*, denoted by "*" and *attribute-concept values*, denoted by "−". We assume that lost values were erased or are unreadable and that for data mining we use only remaining, specified values [13, 14]. "Do not care" conditions are interpreted as uncommitted [15, 16]. Such missing attribute values will be replaced by all possible attribute values. The attribute-concept value is a special case of the "do not care" condition: it is restricted to attribute values typical for the concept to which the case belongs. For example, typical values of temperature for patients sick with flu are:

Table 1. An incomplete data set

	Attributes			Decision
Case	Temperature	Headache	Cough	Flu
1	high	yes	no	yes
2	?	yes	*	yes
3	*	?	yes	yes
4	normal	no	no	maybe
5	high	–	yes	maybe
6	*	no	yes	no
7	–	no	*	no
8	normal	no	no	no

high and very-high, for a patient the temperature value is missing, but we know that this patient is sick with flu. Using the attribute-concept interpretation, we will assume that possible temperature values are: high and very-high.

We will assume that for any case at least one attribute value is specified (i.e., is not missing) and that all decision values are specified.

For complete data sets, a *block* of a variable-attribute pair (a, v), denoted by $[(a, v)]$, is the set $\{x \in U \mid a(x) = v\}$ [17]. For incomplete data sets the definition of a block of an attribute-value pair is modified in the following way.

- If for an attribute a there exists a case x such that $a(x) = ?$, i.e., the corresponding value is lost, then the case x should not be included in any blocks $[(a, v)]$ for all values v of attribute a,
- If for an attribute a there exists a case x such that the corresponding value is a "do not care" condition, i.e., $a(x) = *$, then the case x should be included in blocks $[(a, v)]$ for all specified values v of attribute a.
- If for an attribute a there exists a case x such that the corresponding value is an attribute-concept value, i.e., $a(x) = -$, then the corresponding case x should be included in blocks $[(a, v)]$ for all specified values $v \in V(x, a)$ of attribute a, where

$$V(x, a) = \{a(y) \mid a(y) \textit{ is specified}, y \in U,\ d(y) = d(x)\}.$$

For the data set from Table 1, $V(5, Headache) = \{no\}$, $V(7, Temperature) = \{low\}$, and the blocks of attribute-value pairs are:

[(Temperature, high)] = {1, 3, 5, 6},
[(Temperature, normal)] = {3, 4, 6, 7, 8},
[(Headache, yes)] = {1, 2},
[(Headache, no)] = {4, 5, 6, 7, 8},

[(Cough, no)] = {1, 2, 4, 7, 8},
[(Cough, yes)] = {2, 3, 5, 6, 7}.

For a case $x \in U$ and $B \subseteq A$, the *characteristic set* $K_B(x)$ is defined as the intersection of the sets $K(x, a)$, for all $a \in B$, where the set $K(x, a)$ is defined in the following way:

- If $a(x)$ is specified, then $K(x, a)$ is the block $[(a, a(x)]$ of attribute a and its value $a(x)$,
- If $a(x))$ =? or $a(x) = *$ then the set $K(x, a) = U$, where U is the set of all cases.
- If $a(x) = -$, then the corresponding set $K(x, a)$ is equal to the union of all blocks of attribute-value pairs (a, v), where $v \in V(x, a)$ if $V(x, a)$ is nonempty. If $V(x, a)$ is empty, $K(x, a) = U$.

For Table 1 and $B = A$,

$K_A(1) = \{1\}$, $K_A(5) = \{5, 6\}$,
$K_A(2) = \{1, 2\}$, $K_A(6) = \{5, 6, 7\}$,
$K_A(3) = \{2, 3, 5, 6, 7\}$, $K_A(7) = \{4, 6, 7, 8\}$,
$K_A(4) = \{4, 7, 8\}$, $K_A(8) = \{4, 7, 8\}$.

Note that for incomplete data there is a few possible ways to define approximations [18, 19], we use *concept* approximations [1]. A B-*concept lower approximation* of the concept X is defined as follows:

$$\underline{B}X = \cup\{K_B(x) \mid x \in X, K_B(x) \subseteq X\}.$$

A B-*concept upper approximation* of the concept X is defined as follows:

$$\overline{B}X = \cup\{K_B(x) \mid x \in X, K_B(x) \cap X \neq \emptyset\} = \\ = \cup\{K_B(x) \mid x \in X\}.$$

Since we will use only A-concept lower and upper approximations, we will call them, for simplicity, *lower* and *upper approximations*.

For Table 1, lower and upper approximations of the concept {4, 5} are:

$\underline{A}\{4, 5\} = \emptyset$ and $\overline{A}\{4, 5\} = \{1, 2, 3, 5, 6, 7\}$.

3 Probabilistic Approximations

For incomplete data sets, a *B-concept probabilistic approximation* of the set X, denoted by $appr_\alpha(X)$, is defined by the following formula [1]

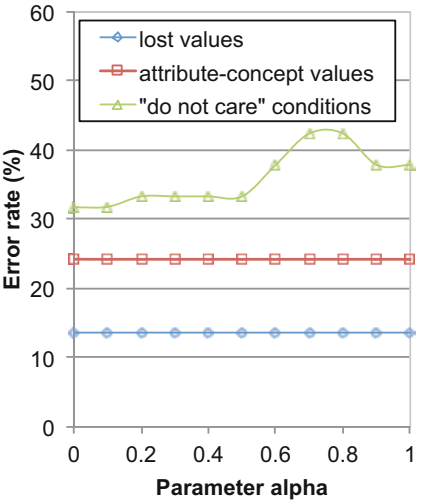


Fig. 1. Error rate for the *bankruptcy* data set

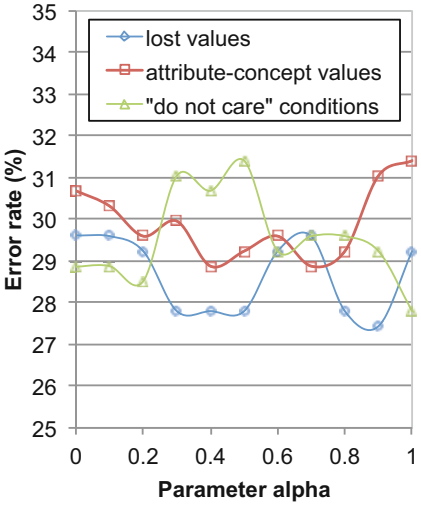


Fig. 2. Error rate for the *breast cancer* data set

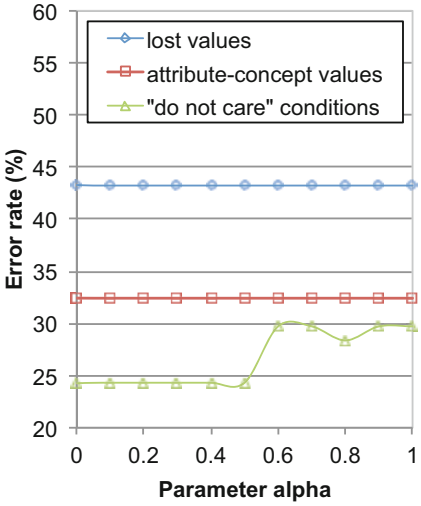


Fig. 3. Error rate for the *echocardiogram* data set

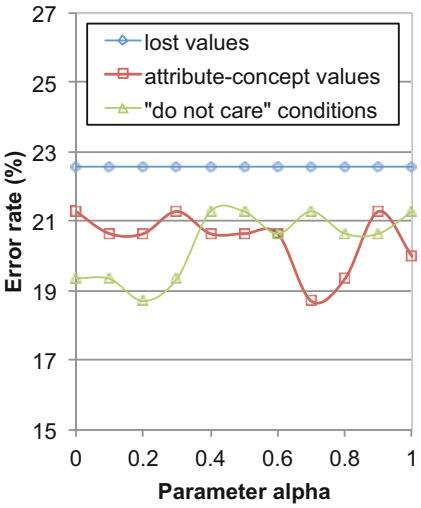


Fig. 4. Error rate for the *hepatitis* data set

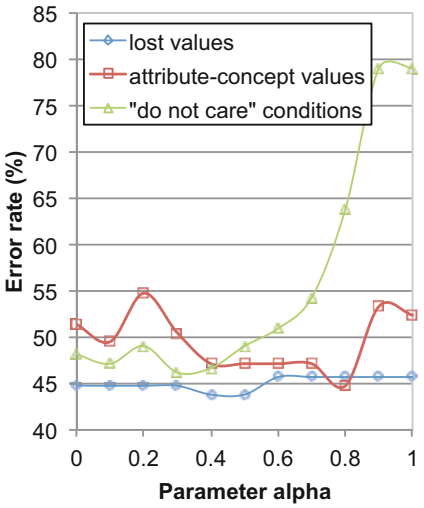


Fig. 5. Error rate for the *image segmentation* data set

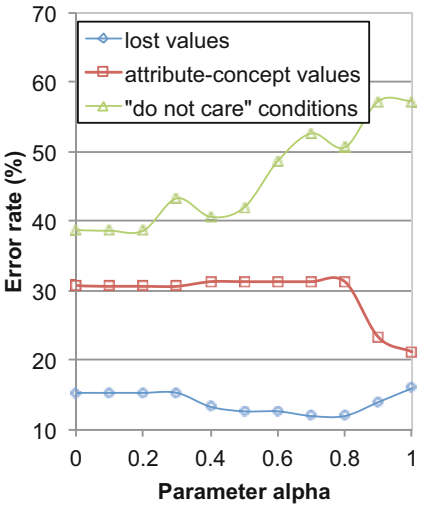


Fig. 6. Error rate for the *iris* data set

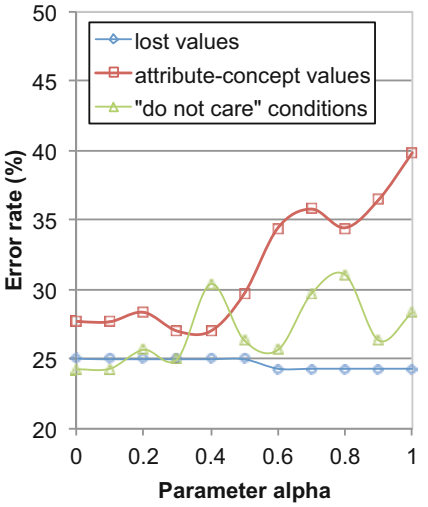


Fig. 7. Error rate for the *lymphography* data set

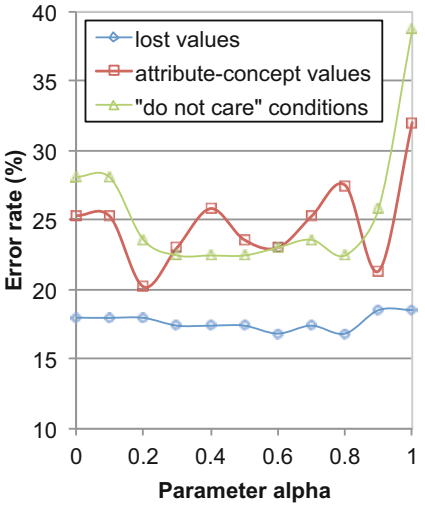


Fig. 8. Error rate for the *wine recognition* data set

Table 2. Data sets used for experiments

Data set	Number of		
	cases	attributes	concepts
Bankruptcy	66	5	2
Breast cancer	277	9	2
Echocardiogram	74	7	2
Image segmentation	210	19	7
Hepatitis	155	19	2
Iris	150	4	3
Lymphography	148	18	4
Wine recognition	178	13	3

$$\cup\{K_B(x) \mid x \in X,\ Pr(X|K_B(x)) \geq \alpha\}.$$

Since we will discuss only *A-concept* probabilistic approximations, we will call them, for simplicity, *probabilistic approximations*.

Thus, for the concept {4, 5} we may define three distinct probabilistic approximations:

$$appr_{0.333}(\{4,5\}) = \{4,5,6,7,8\}, \quad appr_{0.5}(\{4,5\}) = \{5,6\},$$
and $$appr_1(\{4,5\}) = \emptyset.$$

Note that there are only two distinct probabilistic approximations for the concept {1, 2, 3} (the standard lower and upper approximations).

4 Rule Induction with LERS

The LERS (Learning from Examples based on Rough Sets) data mining system [17, 20] starts from computing lower and upper approximations for every concept and then it induces rules using the MLEM2 (Modified Learning from Examples Module version 2) rule induction algorithm. Rules induced from lower and upper approximations are called *certain* and *possible*, respectively [21].

MLEM2 explores the search space of attribute-value pairs. Its input data set is a lower or upper approximation of a concept. In general, MLEM2 computes a local covering and then converts it into a rule set [20]. In order to induce probabilistic rules we have to modify input data sets, as described in [22].

5 Experiments

For our experiments we used eight real-life data sets that are available on the University of California at Irvine *Machine learning Repository*. These data sets

were enhanced by replacing 35% of existing attribute values by missing attribute values, separately by *lost* values, separately by attribute-concept values, and separately by *"do not care" conditions*, see Table 2. Thus, for any data set from Table 2, three data sets were used for experiments, with missing attribute values interpreted as lost values, attribute-concept values and as "do not care" conditions, respectively. Thus for our experiments 24 data sets were used. Results of our experiments for these 24 data sets may be categorized into four groups:

- For 11 data sets, there exists some $\alpha \in [0.1, 0.9]$ such that the error rate, a result of ten-fold cross validation, is smaller for the corresponding proper (i.e., different from lower and upper approximations) probabilistic approximation than for both lower and upper approximations. For example, for the *breast cancer* data set with lost values and $\alpha = 0.9$, the error rate is 27.44%, while the error rates for the lower and upper approximations are 29.24% and 29.60%, respectively. To this group belong 4 data sets with lost values, 4 data sets with attribute-concept values, and 3 data sets with "do not care" conditions.
- For 8 data sets, there exists, for any $\alpha \in [0.1, 0.9]$, the error rate for probabilistic approximations is neither larger nor smaller than the error rate for lower and upper approximations. An example is the *bankruptcy* data set with lost values. To this group belong 4 data sets with lost values, 2 data sets with attribute-concept values, and 2 data sets with "do not care" conditions.
- For 4 data sets, there exists some $\alpha \in [0.1, 0.9]$ such that the error rate is larger than the error rate for lower and upper approximations. For example, for the *bankruptcy* data set with "do not care" conditions and $\alpha = 0.7$ the error rate is 42.42%, while the error rates for the lower and upper approximations are 37.88% and 31.82%, respectively. To this group belong one data set with attribute-concept values and 3 data sets with "do not care" conditions.
- For one data set, there exists some $\alpha \in [0.1, 0.9]$ such that the error rate for the corresponding proper probabilistic approximation is smaller than for both lower and upper approximations and there exists another $\alpha \in [0.1, 0.9]$ such that the error rate for the corresponding proper probabilistic approximation is larger than for both lower and upper approximations. It is the *image segmentation* data set with attribute-concept values, where the error rates for the lower approximation is 52.38%, for the upper approximation is 51.43%, for $\alpha = 0.5$ the error rate is 47.14% and for $\alpha = 0.9$ the error rate is 53.33%.

6 Conclusions

Our objective was to compare, experimentally, three interpretations of missing attribute values using probabilistic approximations. For any original, complete data set the best overall interpretation of missing attribute values was selected among the three incomplete data sets, with lost values, attribute-concept values and "do not care" conditions. In six out of eight data sets, the smallest overall

error rate, a result of ten-fold cross validation, was associated with lost values. Note that for the *echocardiogram* data set the best interpretation was the "do not care" condition and for the *hepatitis* data set there is a tie between two interpretations: attribute-concept value and "do not care" condition.

Additionally, as follows from our experiments for each of 24 data sets separately, the best choice is again the lost value interpretation of missing attribute values since for this kind of missing attribute value the error rate for proper probabilistic approximations cannot be larger than for standard approximations.

References

1. Grzymała-Busse, J.W.: Generalized parameterized approximations. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 136–145. Springer, Heidelberg (2011)
2. Wong, S.K.M., Ziarko, W.: INFER—an adaptive decision support system based on the probabilistic approximate classification. In: Proceedings of the 6th International Workshop on Expert Systems and their Applications, pp. 713–726 (1986)
3. Grzymala-Busse, J.W., Ziarko, W.: Data mining based on rough sets. In: Wang, J. (ed.) Data Mining: Opportunities and Challenges, pp. 142–173. Idea Group Publ., Hershey (2003)
4. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences 177, 28–40 (2007)
5. Pawlak, Z., Wong, S.K.M., Ziarko, W.: Rough sets: probabilistic versus deterministic approach. International Journal of Man-Machine Studies 29, 81–95 (1988)
6. Ślęzak, D., Ziarko, W.: The investigation of the bayesian rough set model. International Journal of Approximate Reasoning 40, 81–91 (2005)
7. Yao, Y.Y.: Probabilistic rough set approximations. International Journal of Approximate Reasoning 49, 255–271 (2008)
8. Yao, Y.Y., Wong, S.K.M.: A decision theoretic framework for approximate concepts. International Journal of Man-Machine Studies 37, 793–809 (1992)
9. Ziarko, W.: Variable precision rough set model. Journal of Computer and System Sciences 46(1), 39–59 (1993)
10. Ziarko, W.: Probabilistic approach to rough sets. International Journal of Approximate Reasoning 49, 272–284 (2008)
11. Clark, P.G., Grzymala-Busse, J.W.: Experiments on probabilistic approximations. In: Proceedings of the 2011 IEEE International Conference on Granular Computing, pp. 144–149 (2011)
12. Clark, P.G., Grzymala-Busse, J.W.: Rule induction using probabilistic approximations and data with missing attribute values. In: Proceedings of the 15th IASTED International Conference on Artificial Intelligence and Soft Computing, ASC 2012, pp. 235–242 (2012)
13. Grzymala-Busse, J.W., Wang, A.Y.: Modified algorithms LEM1 and LEM2 for rule induction from data with missing attribute values. In: Proceedings of the Fifth International Workshop on Rough Sets and Soft Computing, RSSC 1997, at the Third Joint Conference on Information Sciences, JCIS 1997, pp. 69–72 (1997)
14. Stefanowski, J., Tsoukias, A.: Incomplete information tables and rough classification. Computational Intelligence 17(3), 545–566 (2001)

15. Grzymala-Busse, J.W.: On the unknown attribute values in learning from examples. In: Proceedings of the ISMIS 1991, 6th International Symposium on Methodologies for Intelligent Systems, pp. 368–377 (1991)
16. Kryszkiewicz, M.: Rules in incomplete information systems. Information Sciences 113(3-4), 271–292 (1999)
17. Grzymala-Busse, J.W.: LERS—a system for learning from examples based on rough sets. In: Slowinski, R. (ed.) Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory, pp. 3–18. Kluwer Academic Publishers, Dordrecht (1992)
18. Grzymala-Busse, J.W.: Three approaches to missing attribute values—a rough set perspective. In: Proceedings of the Workshop on Foundation of Data Mining, in conjunction with the Fourth IEEE International Conference on Data Mining, pp. 55–62 (2004)
19. Grzymala-Busse, J.W., Rzasa, W.: A local version of the MLEM2 algorithm for rule induction. Fundamenta Informaticae 100, 99–116 (2010)
20. Grzymala-Busse, J.W.: MLEM2: A new algorithm for rule induction from imperfect data. In: Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp. 243–250 (2002)
21. Grzymala-Busse, J.W.: Knowledge acquisition under uncertainty—A rough set approach. Journal of Intelligent & Robotic Systems 1, 3–16 (1988)
22. Grzymala-Busse, J.W.: Generalized probabilistic approximations. Transactions on Rough Sets 16, 1–16 (2013)

Efficient Algorithms for Attribute Reduction on Set-Valued Decision Tables*

Sinh Hoa Nguyen[1] and Thi Thu Hien Phung[2]

[1] Polish-Japanese Institute of Inf. Technology,
Koszykowa 86, 02008, Warszawa, Poland
[2] University of Economic and Technical Industries, Ha Noi, Viet Nam
hoa@pjwstk.edu.pl

Abstract. This paper concerns the problem of searching for reduct on set-valued decision systems. We present efficient algorithms for solving the the problems of *attribute reduction* and *lower and upper approximation of a set* induction for *set-valued decision tables* with predefined *tolerance relations*. Theoretical evaluation shows that the proposed algorithms outperform the known algorithms in literature.

Keywords: set-valued decision tables, tolerance relations, rough sets, attribute reduction.

1 Introduction

Rough set theory was originally developed [7] as a tool for dealing with incomplete and imperfect data. It has been successfully applied in various tasks, such as feature selection/extraction, rule synthesis and classification.

Guan et al. [3] initially introduced the set-valued information system as generalized models of the classic single-valued information systems. The generalization is based on the assumption that each pair of attribute and object is associated with a set of values instead of a single value. Recently, the set-valued information system has become a developing research area and got a lot of attention ([3],[10],[15]). In those papers, the authors proposed many interesting approaches to generalization of the standard rough sets like methods for computing the rough set concepts like lower and upper approximation, the reducts and decision rules for the case of set-valued decision tables. The generalizations are based on either tolerance-based rough sets ([4],[14],[15]) or dominance-based rough sets ([2],[15]). The proposed so far rough set methods for set-valued decision tables are not efficient for real life applications, as they are performing many matrix operations including multiplication of $n \times n$ matrices, where n is the number of objects in decision tables [15].

In this paper we present an efficient approach to computation of short reducts for set-valued decision tables. The idea is based on the contingency table and lattice traversal approach to calculate the number of occurrences of each attribute

* The paper is supported by grants 2011/01/B/ST6/03867 and 2012/05/B/ST6/03215 from the Polish National Science Centre (NCN) and No ST/SI/02/2013 from the Polish Japanese Institute of Information Technology (PJIIT).

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 87–98, 2013.

in the discernibility matrix without implicit implementation of this matrix. The proposed solution is especially effective in the case when the set-valued decision tables are large, but the number of different sets of values of each attribute is not very high. This is in fact an initial proposition, however it can be easily adopted for other rough set concepts.

The paper is organized as follows: Section 2 presents some basic concepts in a set-valued decision table. In Section 3 we present the new concepts called a contingency table and attribute lattices and discuss how to apply them to the algorithm for attribute reduction. Section 4 presents the complete scheme of the algorithm for reduct calculation. The algorithm for computing a lower and an upper approximation of a set is discussed in Section 5. In Section 6 we present some concluding remarks and propose some ideas of future research.

2 Basic Notions

Set-valued decision systems were proposed as a tool to characterize the data sets with incomplete or uncertain information [10].

Formally set-values decision table is a tuple $\mathcal{DT} = (U, A \cup \{d\})$, where U i a finite set of *objects*, A is a finite set of *set-valued attributes*, i.e the functions of form $a : U \to 2^{V_a}$ for $a \in A$, and $d \notin A$ is a distinguished attribute called *decision*. The set V_a is called the domein of attribute a, and $a(x) \subseteq V_a$ for each $a \in A$ and $x \in U$. In the case, when $|a(x)| = 1$ for any $a \in A$ and $x \in U$ we have a standard single-valued decision table.

In Table 1 we have an example of a set-valued decision system. There are ten objects and four condition attributes. Objects belong to one of two decision classes. The table is adopted from [10].

Table 1. An example of a set-valued decision table

U	Audition(**A**)	Spoken Language(**S**)	Reading(**R**)	Writing(**W**)	dec
x_1	$\{E\}$	$\{E\}$	$\{F, G\}$	$\{F, G\}$	No
x_2	$\{E, F, G\}$	$\{E, F, G\}$	$\{F, G\}$	$\{E, F, G\}$	No
x_3	$\{E, G\}$	$\{E, F\}$	$\{F, G\}$	$\{F, G\}$	No
x_4	$\{E, F\}$	$\{E, G\}$	$\{F, G\}$	$\{F\}$	No
x_5	$\{F, G\}$	$\{F, G\}$	$\{F, G\}$	$\{F\}$	No
x_6	$\{F\}$	$\{F\}$	$\{E, F\}$	$\{E, F\}$	Yes
x_7	$\{E, F, G\}$	$\{E, F, G\}$	$\{E, G\}$	$\{E, F, G\}$	Yes
x_8	$\{E, F\}$	$\{F, G\}$	$\{E, F, G\}$	$\{E, G\}$	Yes
x_9	$\{F, G\}$	$\{G\}$	$\{F, G\}$	$\{F, G\}$	Yes
x_{10}	$\{E, F\}$	$\{E, G\}$	$\{F, G\}$	$\{E, F\}$	Yes

Let $\mathcal{DT} = (U, A \cup \{d\})$ be a set-valued decision table. Any *reflexive* and *symmetric* relation $\tau \subseteq U \times U$ is called a *tolerance relation* defined on U. A tolerance relation τ_B related to a set of attributes $B \subseteq A$ can be defined by:

$$\tau_B(x, y) \Leftrightarrow \forall_{b \in B} |a(x) \cap a(y)| \neq \emptyset \tag{1}$$

For any $B \subseteq A$ we denote by $[x]_{\tau_B} = \{y \in U : (x, y) \in \tau_B\}$ the tolerance class related to object $x \in U$. We also denote by $U/\tau_B = \{[x]_{\tau_B} : x \in U\}$ the family of all tolerance classes of τ_B.

For the decision table from Table 1 the tolerance class $[x_1]_{\tau_B}$ related to $B = \{Audition, Spoken\ language\}$ is $[x_1]\tau_B = \{x_1, x_2, x_3, x_4, x_7, x_{10}\}$

Given a set-valued decision table $\mathcal{DT} = (U, A \cup \{d\})$. Each subset $X \subseteq U$ can be approximated by a pair of two definable sets $(\mathbf{L}, \mathbf{U})$, called the *lower* and *upper* approximations of X, where

$$\mathbf{L}_{\mathcal{DT}}(X) = \{x \in U : [x]_{\tau_B} \subseteq X\}; \qquad \mathbf{U}_{\mathcal{DT}}(X) = \{x \in U : [x]_{\tau_B} \cap X \neq \emptyset\}.$$

In a single-valued attribute decision system two objects are called *discernible* if their values are different in at least one attribute. Skowron and Rauszer [11] proposed to store the sets of attributes that discern pairs of objects in *discernibility matrix*. In a set-valued decision table with predefined tolerance relations $\tau(.)$, two objects x and y are discernible if there exists attribute a such that $(x, y) \notin \tau_a$. The discernibility matrix for a set-valued decision table using the tolerance relations, defined by Eq. 1, can be defined as follows:

Definition 1 (Discernibility matrix). *Let $\mathcal{DT} = (U, A \cup \{d\})$ be a set-valued decision table. The discernibility matrix $\mathcal{M}_{\mathcal{DT}} = [m_{ij}]_{i,j=1}^{n}$ of $\mathcal{DT}$ is defined by:*

$$m_{ij} = \begin{cases} \emptyset & \text{if } d(u_i) = d(u_j) \\ \{a \in A : (u_i, u_j) \notin \tau_a\} & \text{otherwise} \end{cases}$$

for any pair of objects $(u_i, u_j) \in U^2$.

An example of a discernibility matrix for a set-valued decision table based on tolerance relations is presented in Table 2.

Table 2. The discernibility matrix of the decision table shown in Table 1. The gray cells are related to the pairs of objects that have the same decision.

$\mathcal{M}_{\mathcal{IS}}$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
x_1						*A*,*S*		*S*	*A*,*S*	
x_2										
x_3						*A*			*S*	
x_4						*S*		*W*		
x_5								*W*		
x_6	*A*,*S*		*A*	*S*						
x_7										
x_8	*S*			*W*	*W*					
x_9	*A*,*S*		*S*							
x_{10}										

3 Attribute Reduction and Heuristics

Attribute reduction is an important task in many applications, e.g. feature selection, decision rule extraction or concept approximation. Intuitively, attribute reduct is a minimal set of attributes that preserves all information necessary to discern objects such as the original attribute set.

Definition 2 (Decision relative reduct). *Given a set-valued decision table* $\mathcal{DT} = (U, A \cup \{d\})$ *the decision relative reduct of* $\mathcal{DT}$ *is the minimal set of attribute* $R \subseteq A$*, which satisfying the following conditions:*

1. *for any pair* $(x, y) \in U^2$ *, if* $d(x) \neq d(y)$ *and* $(x, y) \notin \tau_A$ *then* $(x, y) \notin \tau_R$*;*
2. *no proper subset* R' *of* R *satisfies the previous condition.*

The reduct R is *optimal* if it consists of the smallest number of attributes.

Problem of finding the optimal reduct of a single-valued decision table is NP-hard [11]. Different heuristics have been investigated for this problem [5]. They differ by a searching strategy and an objective function. In this paper we concentrate on greedy forward selection algorithm. The method iteratively extends a subset of attributes by picking in each step of the algorithm the attribute that maximizes the objective function.

The critical operation in almost all heuristics is calculating the value of the objective function for a given attribute set. The operation becomes more time-consuming when we have to deal with a set-valued decision table with tolerance classes. In the next sections we will discuss an algorithm for solving this problem. We present the objective function called *discernibility function* for set-valued decision tables. We also introduce two data structures called a *contingency table* and *attribute lattices*. By using them one can speed up the time for tolerance class induction and candidate attribute set evaluation.

3.1 Discernibility Function

Usually the objective function for an attribute reduction problem is defined by using two rough set concepts *positive* and *boundary region*. Alternatively one can use the number of pairs of objects from different classes, that are discerned by a set of attributes as an evaluation measure. This measure is called *discernibility function* and it was introduced for single-valued decision tables with the indiscernibility relation [5]. In this section we discuss the properties of the discernibility function, generalized for set-valued decision tables. The algorithm for attribute reduct induction is presented with this measure. However, the idea is universal that it can be applied to another forms of objective functions too. Below we present the definitions of a *basic* and a *generalized discernibility function* for a single and a set-valued decision table, respectively.

Definition 3 (Basic discernibility measure). *Let* $\mathcal{DT} = (U, A \cup \{d\})$ *be a single-valued decision table. The discernibility measure for a set of attributes* $B \subseteq A$ *is defined by:*

$$disc(B) = |\{(x, y) \in U \times U | (d(x) \neq d(y)) \wedge \exists_{b \in B}(b(x) \neq b(y))\}|$$

Definition 4 (Generalized discernibility function). *Let $\mathcal{DT} = (U, A \cup \{d\})$ be a set-valued decision table with tolerance relations τ_a (for all $a \in A$). The mapping discern : $2^A \to R^+ \cup \{0\}$, defined by*

$$discern(B) = |\{(x, y) \in U \times U | (d(x) \neq d(y)) \wedge \exists_{b \in B}(x, y) \notin \tau_b)\}|$$

where $B \subseteq A$ is set of attributes, is called the generalized discernibility function.

Below we list some properties of the generalized function:

Property 1. *For any attribute $a \in A$, the value $discern(a)$ is equal to frequency of occurrence of attribute a in the discernibility matrix M_{DT}.*

Property 2. *Discernibility function is increasing. For any set $B \subseteq A$ and $C \subseteq A$, if $B \subseteq C$ then $discern(B) \leq discern(C)$*

Property 3. *Let $\mathcal{DT} = (U, A \cup \{d\})$ be a set-valued decision table and let $B \subseteq A$ be a set of attributes, $discern(B) = discern(A)$ iff.*

$$\forall_{(x,y) \in U^2} d(x) \neq d(y) \wedge (x, y) \notin \tau_A \Rightarrow ((x, y) \notin \tau_B)$$

3.2 Contingency Table and Tolerance-Based Contingency Table

One can observe that for any attribute $a \in A$, a frequency of occurrence of a in discernibility matrix $\mathcal{M}(\mathcal{DT})$ is computed in $DTIME(n^2)$ by scanning all cells of the matrix. This time can be improved by using the discernibility function and some additional structure called a *contingency table*. The concept was proposed in [6] and it was defined for a single-valued decision table with the indiscernibility relation. Intuitively, a contingency table is a structure, which keeps information about decision distributions of all indiscernibility classes. Using such data structure one can quickly determine a frequency of occurrence of any attribute in discernibility matrix without checking its cells. In this section at first we remind a concept of a contingency table for a single-valued decision system and then discuss a concept of a tolerance based contingency table for a set-valued decision system with a predefined tolerance relation.

Contingency Table. Let V_d be the set of decision values in decision table $\mathcal{DT} = (U, A \cup \{d\})$, and let $U/IND(B) = \{[x_1]_B, ..., [x_{n_B}]_B\}$ be partition of U defined by indiscernibility relation $IND(B)$ for $B \subseteq A$. *Contingency table* $\mathbf{CT}_B$ related to B is a two dimensional table $\mathbf{CT}_B = [CT_B[i, j]]_{i \in \{1,...,n_B\}}^{j \in \{1,...,|V_d|\}}$ where:

$$CT_B[i, j] = |\{x \in U : x \in [x_i]_B \wedge d(x) = j\}|.$$

The *local discernibility measure* related to indiscernibility class $[x_i]_B$ is defined as follows:

$$\begin{aligned} \delta_B([x_i]_B) &= |\{(x, y) \in [x_i]_B \times (U \setminus [x_i]_B) : d(x) \neq d(y)\}| \\ &= \sum_{j_1 \neq j_2, x_k \notin [x_i]_B} CT[i, j_1] \cdot CT[k, j_2] \\ &= \sum_{j_1 \neq j_2} CT[i, j_1] \cdot (|D_{j_2}| - CT[i, j_2]) \end{aligned}$$

where $|D_j|$ denotes cardinality of decision class D_j for $j = 1, ..., |V_d|$.

Table 3. The contingency tables for single attributes and values of the discern function

Audition		
Values	No	Yes
E	1	0
F	0	1
E, F	1	2
F, G	1	1
E, G	1	0
E, F, G	1	1
disc(A) = 21		

Spoken language		
	No	Yes
E	1	0
F	0	1
G	0	1
E, F	1	0
E, G	1	1
F, G	1	1
E, F, G	1	1
disc(S) = 22		

Reading		
	No	Yes
E, F	0	1
F, G	0	1
E, G	5	2
E, F, G	0	1
disc(R) = 15		

Writing		
	No	Yes
F	2	0
E, F	0	2
E, G	0	1
F, G	2	1
E, F, G	1	1
disc(W) = 22		

Hence the *basic discernibility measure* of attribute set B is defined as the number of pairs of discernible objects, i.e.

$$disc(B) = \sum_{i} \delta_B([x_i]_B) = \frac{1}{2} \sum_{i=1}^{n_B} \sum_{j_1 \neq j_2} CT[i, j_1](|D_{j_2}| - CT[i, j_2]) \qquad (2)$$

The summation is taken over the disjoint subsets induced by $IND(B)$ and over all $j_1, j_2 \in \{1, \ldots, |V_d|\}, j_1 \neq j_2$.

Table 3 presents the contingency table and the values of the discernibility function for each attribute from Table 1. We remind that the cardinality of each decision class is equal to 5. The contingency table with the indiscernibility relation is further called the *basic contingency table.*

Proposition 1. *Let $\mathcal{DT} = (U, A \cup \{d\})$ be a decision table. Let $IND(B)$ be a indiscernibility relation related to $B \subseteq A$. Let n_B denotes a number of indiscernibility classes defined by $INB(B)$. Given a contingency table $\mathbf{CT}_B$. The value discern(B) can be determined in time $O(dn_B)$, which is bounded by $O(dn)$, where $n = |U|$ and d is a number of decision classes.*

Tolerance-Based Contingency Table. For a decision table $\mathcal{DT} = (U, A \cup \{d\})$, let τ_B be a tolerance relation for $B \subseteq A$ and let $U/IND(B) = \{[x_1], ..., [x_{n_B}]\}$ be the partition of U defined by indiscernibility relation $IND(B)$. The *tolerance based contingency table* is a two-dimensional table $\mathbf{TCT}_B = [TCT_B[i, j]]_{i \in \{1,\ldots,n_B\}}^{j \in \{1,\ldots,|V_d|\}}$, which is defined as follows:

$$TCT_B[i, j] = |\{x \in [x_i]_{\tau_B} \wedge d(y) = j\}|$$

Intuitively, tolerance-based contingency table stores the decision distributions inside each tolerance class. One can observe that the tolerance classes are not disjoint in general. This may cause an error in calculation of a discernibility function if we take the same formula for a basic contingency table.

To compute the value of discernibility function we modify the concept of a local discernibility measure. For a tolerance class $[x_i]_{\tau_B}$, the local discernibility measure related to $[x_i]_{\tau_B}$ is defined by:

$$\begin{aligned}\delta_B([x_i]_{\tau_B}) &= |\{(x,y) \in [x_i]_B \times (U \setminus [x_i]_{\tau_B}) : d(x) \neq d(y)\}| \\ &= \sum_{j_1 \neq j_2, x_k \notin [x_i]_{\tau_B}} CT[i,j_1] \times TCT_B[k,j_2] \\ &= \sum_{j_1 \neq j_2} CT_B[i,j_1](|D_{j_2}| - TCT_B[i,j_2])\end{aligned}$$

The generalized discernibility measure can be calculated as follows:

$$discern(B) = \sum_i \delta_B([x_i]_{\tau_B}) = \frac{1}{2}\sum_{i=1}^{n_B} \sum_{j_1 \neq j_2} CT_B[i,j_1](|D_{j_2}| - TCT[i,j_2]) \quad (3)$$

where $B \subset A$. We denote by $\mathbf{CT}_B \otimes \mathbf{TCT}_B$ the operation in Equation 3. The summation is taken over a disjoint subsets induced by $IND(B)$ and over all $j_1, j_2 \in \{1, \ldots, |V_d|\}, j_1 \neq j_2$.

Table 4. The illustration of contingency tables and discernibility function calculation

a_1 = **Audition**					
Set values	$\mathbf{CT}_{a_1}$		$\mathbf{TCT}_{a_1}$		δ_{a_1}
	No	Yes	No	Yes	
E	1	0	4	3	$1 \cdot (5-3) + 0 \cdot (5-4) = 2$
F	0	1	3	5	$0 \cdot (5-5) + 1 \cdot (5-3) = 2$
E,F	1	2	5	5	$1 \cdot (5-5) + 2 \cdot (5-5) = 0$
F,G	1	1	4	5	$1 \cdot (5-5) + 1 \cdot (5-4) = 1$
E,G	1	0	4	4	$1 \cdot (5-4) + 0 \cdot (5-4) = 1$
E,F,G	1	1	5	5	$1 \cdot (5-5) + 1 \cdot (5-5) = 0$
$discern(a_1) = \frac{1}{2}(2+2+0+1+1+0) = 3$					

The basic and tolerance-based contingency tables related to the attribute *Audition* are shown in Table 4. In the last column we illustrate how to local discernibility measures are calculated. In the last row we have the value of discern function computed for this attribute.

Observation 1. *For a given set of attributes $B \subseteq A$ the contingency table $\mathbf{CT}_B$ can be created by simple SQL queries of the form:* `SELECT B COUNT DISTINCT GROUP BY d`.

Observation 2. *Let $\mathcal{DT} = (U, A \cup \{d\})$ be a set-value decision table. Let $\mathbf{CT}_B$ and $\mathbf{TCT}_B$ denote a basic and tolerance-based contingency table for $B \subseteq A$, respectively. Let $\{[x_1], ..., [x_{n_B}]\}$ be indiscernibility classes defined by $IND(B)$.*

The elements of $\mathbf{TCT}_B = [TCT_B[i,j]]_{i\in\{1,\ldots,n_B\}}^{j\in\{1,\ldots,|V_d|\}}$ *can be defined by adding up appropriate elements of the contingency table* $\mathbf{CT}_B$*:*

$$TCT_B[i,j] = \sum_{x_k \in [x_i]_{\tau_B}} CT[k,j]$$

Proposition 2. *The pessimistic time for generation of tolerance contingency table is* $O(n_B^2 d)$*, where* n_B *is a number of records in basic contingency table* $\mathbf{CT}_B$ *and* d *is a number of decision classes.*

The algorithm of calculating **TCT** is efficient for the decision tables with small attribute domains In the case of a large number of records in the basic contingency table, the creation of **TCT** from **CT** is time-consuming. Next section presents a speed-up technique for computing **TCT** using an additional data structure, called the *attribute value lattices*, for storing the relations between attribute values.

3.3 Lattice of Attribute Values

Formally, *lattice* is a partially ordered set (poset) in which any two elements have a supremum and an infimum. Lattice can be presented as directed graph $G = (V, E)$, where V is a set of vertices and E is a set of edges. Vertices are corresponding to the elements of the given ordered set. The directed edge $(v_i, v_j) \in E$ if the element v_i is "smaller" then v_j and between v_i and v_j there is no other element. This structure is adopted to store the values of attributes with symbolic domains.

Definition 5 (Lattice of an attribute). *Let* $\mathcal{DT} = (U, A \cup \{d\})$ *be a set-valued decision table. Let* $a \in A$ *and* V_a *be the domain of an attribute* a*. The lattice of attribute* a *is a directed graph defined as an ordered set* $Latt(a) = (V_a, r)$*, where* $r \subseteq 2^{V_a} \times 2^{V_a}$ *is a partial order defined by* $r = \{(X, Y) \in 2^{V_a} \times 2^{V_a} : X \subseteq Y\}$*.*

Let us denote the lattice of attribute a by $Latt(a)$ and the set of all lattices related to attributes from A by $Latt(A)$.

Observation 3. *Every set-valued decision table* $\mathcal{DT} = (U, A \cup \{d\})$ *can be translated to the structured decision table* $\mathcal{SDT} = (U, Latt(A), d)$*.*

The structures of attribute values are shown in Figure 1. One can observe that the tolerance classes for all nodes of $Latt(a)$ can be determined by two scanning the lattice: *bottom - up* and *top - down*. Initially a tolerance class of a node in the lowest level consists of one element, which is himself. In the bottom-up phase, every node aggregates the lists of tolerance classes from it's children. In the top-down phase the parent nodes send the tolerance class list to their children. The node in the lower level completes its tolerance class by the appropriate elements obtained from their parents.

Observation 4. *For any attribute* $a \in A$*, the time complexity of the algorithm computing tolerance classes of all nodes in lattice* $Latt(a)$ *is* $O(n_a)$*, where* n_a *is a number of nodes of the lattice* ($n_a = |U/IND(a)|$)*.*

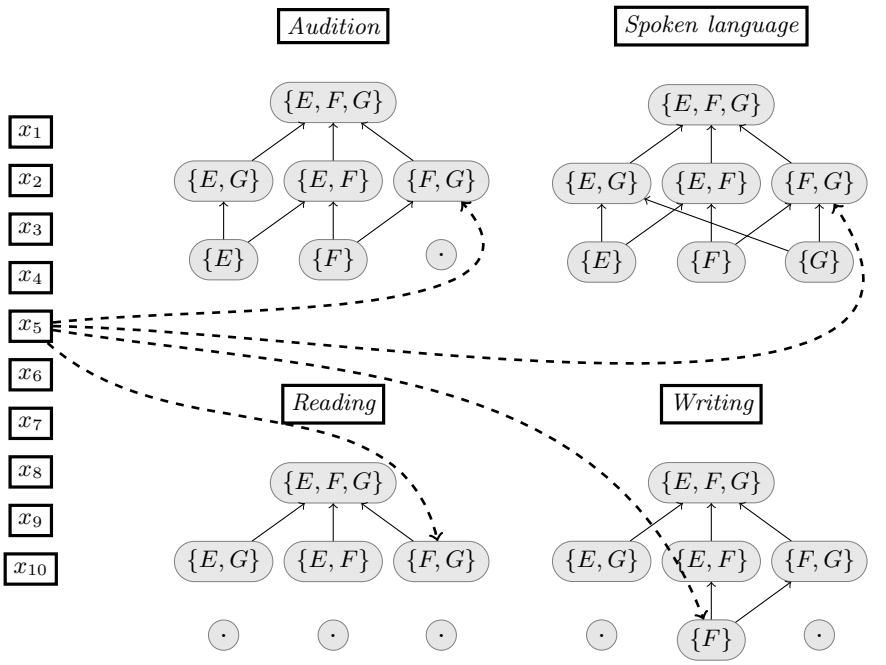


Fig. 1. The structured set-valued decision table

Observation 5. *Having a set of lattices Latt(A) with tolerance classes calculated for all nodes, the tolerance class of any record on the basic contingency table* $\mathbf{CT_B}$ *can be induced in time* $O(c)$*, where c is a cardinality of the tolerance class of the investigated record.*

Proposition 3. *Let Latt(A) be a set of lattices defined for attribute set A. Assume that for each lattice Latt(a) the tolerance classes for it's nodes are calculated. For any set of attributes* $B \subset A$*, the time complexity of the algorithm computing the tolerance contingency table* $\mathbf{TCT}_B$ *from* $\mathbf{CT}_B$ *based on Latt(A) is* $O(n_B c)$*, where* n_B *is a number of records in the basis contingency table* $\mathbf{CT}_B$ *and c is a maximal cardinality of tolerance classes of the records in* $\mathbf{CT}_B$*. The value c is bounded by* n_B*.*

4 Searching for Decision Relative Reducts

In this section we present the complete schema for decision relative reduct induction. Let us consider a set-valued decision table with a given tolerance relation. Reduct for a set-valued decision tables is a minimal set of attributes, that preserves the discernibility of the set of all attributes. The greed heuristic for attribute reduction shown in Algorithm 1.

Algorithm 1. Generalized Maximal Discernibility (**GMD**) heuristic for set-valued decision tables with tolerance relation.

Input: Set-valued decision table $D = (U, A \cup d)$.
Output: Attribute reduction R.
Generate a set of lattices $Latt(A)$;
$R \leftarrow \emptyset$;
$discern(R) \leftarrow 0$;
while $(discern(R) < discern(A))$ **do**
  $max\_discern \leftarrow 0$;
  for $(a_i \in A)$ **do**
    $B \leftarrow R \cup \{a_i\}$;
    Create $\mathbf{CT}_B$;
    Create $\mathbf{TCT}_B$ using $\mathbf{CT}_B$ and $Latt(A)$;
    Determine $discern(B) = \mathbf{CT}_B \otimes \mathbf{TCT}_B$ using Equation (3);
    if $(discern(B) > max\_dicern)$ **then**
      $max\_discern \leftarrow discern(B)$;
      $best\_attribute \leftarrow a_i$;
    end if
  end for
  $A \leftarrow A \setminus \{best\_attribute\}$;
  $R \leftarrow R \cup \{best\_attribute\}$;
end while

Proposition 4. *The time complexity of $GMD - heuristic$ is $O(k^2 * m^2 * d)$, where k is a number of attributes, m is the maximal number of distinct set-values occurring in attribute domains and d is a number of decision classes.*

5 Set Approximation Induction

Let $X \subseteq U$ be a given subset of the universe U. The goal is to find upper and lower approximation of X. The approach of using a square matrix to represent set approximations was discussed in [15]. In this paper, the authors proposed an $O(n^3)$ algorithm, where n is the number of objects in the decision table. In this section we present a new method for induction of rough approximations of sets. The time complexity can be improved by using a tolerance-based contingency table.

Given a set-valued information table $\mathcal{DT} = (U, A)$. Let τ_B be a tolerance relation related to $B \subseteq A$. Let x be an element of a set X. Let $[x]_{\tau_B}$ be a tolerance class related to x. The inclusion degree of $[x]_{\tau_B}$ in X is

$$\nu([x]_{\tau_B}, X) = \frac{|[x]_{\tau_B} \cap X|}{|[x]_{\tau_B|}}.$$

We can observe the value of inclusion function can be effectively computed by using a contingency table.

For any set of objects $X \subseteq U$ we denote by $d_X : U \to \{0, 1\}$ the characteristic function of X, i.e. $d_X(x) = \begin{cases} 1 & \text{if } x \in X \\ 0 & \text{otherwise.} \end{cases}$

Proposition 5. *Let $\mathcal{DT} = (U_{DT}, A_{DT} \cup \{d_X\})$ be a decision table with binary decision 0 and 1. Let $X \subseteq U$. Let $\mathbf{TCT}_B$ be a tolerance-based contingency table for $\mathcal{DT}$. Let $x \in X$ and τ_B be a tolerance relation. The inclusion degree of $[x]_{\tau_B}$ in X can be computed using $\mathbf{TCT}_B$ as follows:*

$$\frac{|[x]_{\tau_B} \cap X|}{|[x]_{\tau_B}|} = \frac{TCT[x, 1]}{TCT[x, 1] + TCT[x, 0]}$$

The object $x \in U$ belongs to lower approximation of X, if

$$\frac{|[x]_{\tau_B} \cap X|}{|[x]_{\tau_B}|} = 1$$

and end its belong to upper approximation of X if

$$\frac{|[x]_{\tau_B} \cap X|}{|[x]_{\tau_B}|} > 0$$

Below we present a scheme of algorithm computing upper and lower approximation of a given set of objects X.

Algorithm 2. Verify, if objects belong to lower or upper approximation

1: **Input**: Set-valued information table $\mathcal{IS} = (U, A)$, $X \subseteq U$, $B \subseteq A$, tolerance relation τ_B, $U/IND(B) = \{1, 2, ..., n_B\}$.
2: **Output**: Upper and lower approximation of X.
3: Create the decision table $\mathcal{DS} = (U, A \cup \{d_X\})$.;
4: Generate $\mathbf{CT}_B$;
5: Generate $\mathbf{TCT}_B$ from $\mathbf{CT}_B$;
6: **for** $i \in \{1, 2, ..., n_B\}$ **do**
7: Compute a inclusion degree $\nu_i = \frac{TCT[i,1]}{TCT[i,1]+TCT[i,0]}$
8: **if** $(\nu_i = 1)$ **then**
9: $LowerAppr \leftarrow \{i\}$
10: **else**
11: **if** $(\nu_i > 0)$ **then**
12: $UpperAppr \leftarrow \{i\}$
13: **end if**
14: **end if**
15: **end for**

Let us notice that the time complexity of the **Verifying algorithm** is $O(m^2)$, where m is the maximal number of distinct set-values in attribute domains.

6 Conclusions

We have presented the algorithms for solving problems of attribute reduction and lower and upper approximation of a set induction. To improve time complexity we have provided novel data structures called a *generalized contingency table* and *lattices of attribute values*. By using these structures one can reduce the time complexity of the algorithm for searching for a lower and an upper approximation of a set presented in [15] from $O(n^3)$ to $O(m^2)$, where m is the maximal number of distinct set-values in attribute domains. In next papers we will show that the proposed solution can be also modified to manage with dominance based rough sets approach to set-valued decision table.

References

1. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman & Co., New York (1979)
2. Greco, S., Matarazzo, B., Slowinski, R.: Rough approximation by dominance relation. International Journal of Intelligent Systems 17, 153–171 (2002)
3. Guan, Y.Y., Wang, H.K.: Set-valued information systems. Information Sciences 176(17), 2507–2525 (2006)
4. Kryszkiewicz, M.: Rough set approach to incomplete information systems. Information Sciences 112, 39–49 (1998)
5. Nguyen, H.S.: Approximate boolean reasoning: Foundations and applications in data mining. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 334–506. Springer, Heidelberg (2006)
6. Swieboda, W., Nguyen, H.S.: Rough Set Methods for Large and Sparse Data in EAV Format. In: Proceedings of 2012 IEEE RIVF International Conference, pp. 1–6. IEEE (2012)
7. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data, System Theory. Kluwer Academic Publishers, Dordrecht (1991)
8. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177(1), 3–27 (2007)
9. Pawlak, Z., Skowron, A.: Rough sets: some extensions. Information Sciences 177(1), 28–40 (2007)
10. Qian, Y., Dang, C., Liang, J., Tang, D.: Set-valued ordered information systems. Inf. Sci. 179(16), 2809–2832 (2009)
11. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Slowinski (ed.) Intelligent Decision Support, vol. 11, pp. 331–362. Springer, Netherlands (1992)
12. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
13. Skowron, A., Pawlak, Z., Komorowski, J., Polkowski, L.: A rough set perspective on data and knowledge. In: Kloesgen, W., Żytkow, J. (eds.) Handbook of KDD, pp. 134–149. Oxford University Press, Oxford (2002)
14. Slowinski, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Transactions on Knowledge and Data Engineering 12(2), 331–336 (2000)
15. Zhang, J., Li, T., Ruan, D., Liu, D.: Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems. International Journal of Approximate Reasoning 53(4), 620–635 (2012)

Metric Based Attribute Reduction in Incomplete Decision Tables*

Long Giang Nguyen[1] and Hung Son Nguyen[2]

[1] Institute of Information Technology, VAST, Vietnam
nlgiang@ioit.ac.vn
[2] Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
son@mimuw.edu.pl

Abstract. Metric technique has recently been applied to solve such data mining problems as classification, clustering, feature selection, decision tree construction. In this paper, we apply metric technique to solve a attribute reduction problem of incomplete decision tables in rough set theory. We generalize Liang entropy in incomplete information systems and investigate its properties. Based on the generalized Liang entropy, we establish a metric between coverings and study its properties for attribute reduction. Consequently, we propose a metric based attribute reduction method in incomplete decision tables and perform experiments on UCI data sets. The experimental results show that metric technique is an effective method for attribute reduction in incomplete decision tables.

Keywords: Rough sets, feature selection and extraction, Liang's entropy, metric based reducts.

1 Introduction

Classical rough set theory based on equivalent relation has been introduced by Pawlak [11] as one of the effective tools for rule induction, object classification in complete decision tables. Attribute reduction is one of the crucial problems in rough set theory. Recently, there have been many attribute reduction algorithms in complete decision tables based on the equivalent relation [17]. In fact, there are many cases that decision tables contain missing values for at least one conditional attribute in the value set of that attribute and these decision tables are called incomplete decision tables. To extract decision rules directly from incomplete decision tables, Kryszkiewicz [5] has extended the equivalent relation in classical rough set theory to tolerance relation and proposed tolerance rough set. Based on the tolerance relation, many uncertainty measures and attribute reduction algorithms for incomplete decision tables have been investigated [7],

* The authors are supported by grants 2011/01/B/ST6/03867 from the Polish National Science Centre (NCN), and the grant SP/I/1/77065/10 in frame of the strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information" founded by the Polish National Centre for Research and Development (NCBiR).

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 99–110, 2013.

[8], [9], [12], [13]. Huang et al [4] proposed an attribute reduction algorithm based on information quantity. Zhou et al [22], Huang et al [3] proposed attribute reduction algorithms based on tolerance matrix. The time complexity of these algorithms is $O\left(|A|^3|U|^2\right)$, where $|A|$ is the number of conditional attributes and $|U|$ is the number of objects. Zhang et al [21] improved the algorithm from [4] and the time complexity is down to $O\left(|A|^2|U|^2\right)$. Dai et al [1] presented an attribute reduction algorithm based on the coverage of an attribute set.

Metric is a distance measure between two sets [2]. In recent researches, metric technique has been applied to solve problems in data mining and rough set theory. Mantaras [16], Simovici and Jaroszewicz [18], [19] used a metric as the attribute selection criterion in the process of decision tree construction. Nguyen [10] proposed a metric based attribute reduction method in complete decision tables. Qian et al [14], [15] proposed knowledge distances between coverings in incomplete information systems and investigate its properties.

In this paper, we propose a metric based attribute reduction method in incomplete decision tables. Firstly, we generalize Liang entropy [6] in incomplete information systems and investigate its properties. Secondly, we establish a metric between coverings based on the generalized Liang entropy and study its properties in incomplete decision tables for attribute reduction. Finally, we define a reduct based on the metric, significance of attribute based on the metric and propose an attribute reduction heuristic algorithm in incomplete decision tables. The time complexity of proposed algorithm is $O\left(|A|^2|U|^2\right)$.

The structure of this paper is as follows. Section 2 presents the concept of attribute reduction in rough set theory. Section 3 presents a generalized Liang entropy in incomplete information systems and investigate its properties. Section 4 establishes a metric between coverings based on the generalized Liang entropy and study its properties. Section 5 presents a metric based attribute reduction method in incomplete decision tables. In Section 6, we perform some experiments of the proposed algorithm. The conclusions are presented in the last section.

2 Basic Notions

In this section, we introduction some basic concepts in rough set theory related to attribute reduction.

An information system [11] is a pair $\mathbb{S} = (U, A)$, where U is a non-empty, finite collection of objects and A is a non-empty, finite set, of attributes. Each $a \in A$ corresponds to the function $a : U \to V_a$, where V_a is called the value set of a. Elements of U can be interpreted as, e.g., cases, patients, observations, etc. Without loss of generality, we will assume that $U = \{u_1, ..., u_{|U|}\}$.

For a given information system $\mathbb{S} = (U, A)$, the function $\mu_{\mathbb{S}} : \mathbb{P}(A) \longrightarrow \mathbb{R}^+$, where $\mathbb{P}(A)$ is the power set of A, is called *the monotone evaluation function* if:

1. $\mu_{\mathbb{S}}(B)$ can be computed using information from B and U for any $B \subset A$;
2. $\mu_{\mathbb{S}}(.)$ is monotone, i.e., for any $B, C \subset A$, if $B \subset C$, then $\mu_{\mathbb{S}}(B) \leq \mu_{\mathbb{S}}(C)$.

In rough sets, reducts are the minimal subsets (with respect to the set inclusion) of attributes that contain a necessary portion of *information* about the objects, expressed by a *monotone evaluation function*.

Definition 1 (μ-reduct). *Any set $B \subseteq A$ is called* the reduct relative to a monotone evaluation function μ, *or briefly* μ-reduct, *if B is the smallest subset of attributes that $\mu(B) = \mu(A)$, i.e., $\mu(B') < \mu(B)$ for any proper subset $B' \subsetneq B$. We denote by $\mathcal{RED}(\mathbb{S}, \mu)$ the set of all μ-reducts, i.e.,*

$$\mathcal{RED}(\mathbb{S}, \mu) = \{R \subset A : R \text{ is } \mu\text{-reduct of } \mathbb{S}\} \tag{1}$$

The attribute $a \in A$ is called core attribute *if a presents in all reducts of A. The set of all core attributes is denoted by*

$$CORE(\mathbb{S}, \mu) = \bigcap_{\mathcal{RED}(\mathbb{S},\mu)} R \tag{2}$$

This definition is general for many existing definitions of reducts. Let us mention some well-known types of reducts used in rough set theory.

2.1 Decision Table and Decision Reducts

A decision table is a special information system $\mathbb{D} = (U, A \cup D)$, where attributes are of two types: conditional attributes (the attributes from A), and decision attributes (the attributes from D). The conditional attributes are also called *conditions*, while the decision attributes are briefly called *decisions*.

Each subset of attributes $P \subseteq A$ determines a binary indistinguishable relation $IND(P)$ as follows

$$IND(P) = \{(x, y) \in U \times U : inf_P(x) = inf_P(y)\}. \tag{3}$$

It is obvious that $IND(P)$ is an equivalence relation, as it is reflexive, symmetric and transitive, over the set U. Any element $u \in U$ the set $[u]_P = \{v \in U \,|(u, v) \in IND(P)\}$ is called the equivalent class. The relation $IND(P)$ constitutes a partition of U, which is denoted by

$$U/_P = \{[u]_P : u \in U\} \tag{4}$$

Let $\mathbb{D} = (U, A \cup D)$ be a decision table. Any set $D_i \in U/_D$ is called the decision class of $\mathbb{D}$. For any $B \subset A$, the set

$$POS_B(D) = \{u \in U : [u]_B \subseteq D_i \text{ for some } D_i \in U/_D\} \tag{5}$$

is called the *B-positive region of D*. The decision table $\mathbb{D}$ is called consistent if and only if $POS_A(D) = U$. Otherwise, $\mathbb{D}$ is called the inconsistent decision table. Any minimal subset B of A such that $POS_B(D) = POS_A(D)$ is called the *decision reduct* (or reduct based on positive region) of $\mathbb{D}$. It has been shown in [9] that $\mu_{POS}(B) = |POS_B(D)|$ is a monotone evaluation function. Thus:

Proposition 1. *The set of attributes $R \subseteq A$ is* decision reduct *if and only if it is μ-reduct with respect to the measure $\mu_{POS}(B) = |POS_B(D)|$.*

2.2 Entropy Based Methods

Let $\mathbb{D} = (U, A \cup D)$ be a decision table and $C \subset A$ is an arbitrary set of attributes. Suppose that $U/_C = \{C_1, C_2, ..., C_m\}$and $U/_D = \{D_1, D_2, ..., D_n\}$, the conditional Shannon entropy of D with respect to $C \subset A$ is defined as

$$H(D|C) = -\sum_{i=1}^{m} \frac{|C_i|}{|U|} \sum_{j=1}^{n} \frac{|C_i \cap D_j|}{|C_i|} \log_2 \frac{|C_i \cap D_j|}{|C_i|} \tag{6}$$

Proposition 2 ([19]). *Let $\mathbb{D} = (U, A \cup D)$ be a decision table. If $Q \subseteq P \subseteq A$ then $H(D|Q) \geq H(D|P)$. The equality holds when $\forall X_u, X_v \in U/_P,\ X_u \neq X_v$, if $(X_u \cup X_v) \subseteq Y_k \in U/_Q$ then $\frac{|X_u \cap D_j|}{X_u} = \frac{|X_v \cap D_j|}{X_v}$ for $\forall j \in \{1, 2, ..., n\}$.*

Thus $H(D|C)$ is monotone function with respect to set inclusion. Any μ-reduct with respect to entropy measure $\mu_{Ent}(C) = M - H(D|C)$, where M is a constant, is called *a reduct of $\mathbb{D}$ based on conditional Shannon entropy.*

Let $\mathbb{S} = (U, A)$ be a complete information system, for any $P \subseteq A$ the value

$$E(P) = \sum_{i=1}^{m} \frac{|P_i|}{|U|} \left(1 - \frac{|P_i|}{|U|}\right) \tag{7}$$

where $U/_P = \{P_1, ..., P_m\}$, is called *the Liang entropy* [6].

Let $P, Q \subseteq A$ be arbitrary sets of attributes and let $U/_P = \{P_1, ..., P_m\}$, $U/_Q = \{Q_1, ..., Q_n\}$. The *conditional Liang entropy* is defined as follows:

$$E(Q|P) = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{|Q_i \cap P_j|}{|U|} \frac{|Q_i^c - P_j^c|}{|U|} \tag{8}$$

where $Q_i^c = U - Q_i,\ P_j^c = U - P_j$ (see [6]).

It has been shown in [6] that both Liang entropy and conditional Liang entropy measures are monotone with respect to set inclusion. Thus the μ-reducts with respect to either $\mu_1(P) = E(P)$ or $\mu_2(P) = E(D|P)$ are called the Liang entropy based reducts.

3 Reducts for Incomplete Information Systems

An information system $\mathbb{S} = (U, A)$ is called *incomplete*, or IIS for short, if the value $a(u)$ is not always determined for $a \in A$ and $u \in U$. Furthermore, we will denote the missing value by * [5]. Analogically, incomplete decision table, briefly IDT, is an incomplete information system $\mathbb{D} = (U, A \cup \{d\})$ where $d \notin A$ and $* \notin V_d$. Let $\mathbb{S} = (U, A)$ be an IIS, for any $P \subseteq A$ we define a binary relation on U as follows:

$$SIM(P) = \left\{(u, v) \in U^2 : \forall a \in P, a(u) = a(v) \vee a(u) = * \vee a(v) = *\right\} \tag{9}$$

Let us notice that $SIM(P)$ is a tolerance relation (as it is reflexive and symmetric) on U and that $SIM(P) = \bigcap_{a \in P} SIM(\{a\})$. For any object $u \in U$ and

set of attributes $P \subset A$, the set $S_P(u) = \{v \in U : (u, v) \in SIM(P)\}$ is called *the tolerance class of* u, or granule of information. Let $K(P)$ denote the family of tolerance classes of all objects from U, called *the knowledge base of* P, i.e.

$$K(P) = U/_{SIM(P)} = \{S_P(u) : u \in U\} = \left\{S_P(u_1), S_P(u_2), ..., S_P\left(u_{|U|}\right)\right\}.$$

It is clear that the tolerance classes in $K(P)$ do not constitute a partition of U in general. They constitute a covering of U, i.e., $S_P(u) \neq \emptyset$ for every $u \in U$, and $\bigcup_{u \in U} S_P(u) = U$. We will denote by $COVER(U) = \{K(P) : P \subset A\}$ the set of all possible coverings on U defined by attributes from A. A partial ordered relation $(COVER(U), \preceq)$ can be defined on $COVER(U)$ as follows

1. $K(P)$ is the same as $K(Q)$, denoted by $K(P) = K(Q)$, if and only if $\forall u \in U,\ \ S_P(u) = S_Q(u)$.
2. $K(P)$ is finer than $K(Q)$, denoted by $K(P) \preceq K(Q)$, if and only if $\forall u \in U,\ \ S_P(u) \subseteq S_Q(u)$.

Let $\mathbb{S} = (U, A)$ be an IIS. The family $\omega = \{S_A(u) = \{u\} \,|u \in U\}$ is called *the discrete covering* and $\delta = \{S_A(u) = U \,|u \in U\}$ is called *the complete covering.*

Definition 2 (generalized Liang entropy). *Let* $\mathbb{S} = (U, A)$ *be an IIS and* $P \subseteq A$. *The* **generalized Liang entropy** *of* P *is defined by*

$$IE(P) = \sum_{i=1}^{|U|} \frac{1}{|U|}\left(1 - \frac{|S_P(u_i)|}{|U|}\right) = 1 - \frac{1}{|U|^2}\sum_{i=1}^{n} |S_P(u_i)| \tag{10}$$

where $|S_P(u)|$ *denotes the cardinality of* $S_P(u)$.

Obviously, we have $0 \leq IE(P) \leq 1 - \frac{1}{|U|}$. Function $IE(P)$ achieves the maximum value $1 - \frac{1}{|U|}$ if $K(P) = \omega$, and the minimum value 0 when $K(P) = \delta$.

Definition 3 (Conditional generalized Liang entropy). *Let* $\mathbb{S} = (U, A)$ *be an IIS and* $P, Q \subseteq A$. *The generalized Liang entropy of* Q *conditioned on* P *is defined by*

$$IE(Q|P) = \frac{1}{|U|}\sum_{i=1}^{|U|}\left(\frac{|S_P(u_i)| - |S_Q(u_i) \cap S_P(u_i)|}{|U|}\right) \tag{11}$$

It has been shown that Liang entropy $E(P)$ presented in [6] is a particular case of the generalized Liang entropy, and the conditional Liang entropy $E(Q|P)$ is a particular case of the conditional generalized Liang entropy $IE(Q|P)$. Moreover, let $\mathbb{S} = (U, A)$ be an IIS and $P, Q, R \subseteq A$, the following properties hold:

$P1$) If $K(P) \preceq K(Q)$ then $IE(P) \geq IE(Q)$ and $IE(P) = IE(Q)$ if and only if $K(P) = K(Q)$.
$P2$) If $K(P) \preceq K(Q)$ then $IE(P \cup Q) = IE(P)$.
$P3$) $IE(P \cup Q) \geq IE(P)$ and $IE(P \cup Q) \geq IE(Q)$.

$P4$) $IE(P \cup Q) = IE(P) + IE(Q|P) = IE(P) + IE(P|Q)$.
$P5$) $0 \leq IE(Q|P) \leq 1 - \frac{1}{|U|}$; the equality $IE(Q|P) = 0$ holds iff $K(P) \preceq K(Q)$ and the equality $IE(Q|P) = 1 - \frac{1}{|U|}$ holds iff $K(P) = \delta$ and $K(Q) = \omega$.
$P6$) If $U/_{SIM(P)} \preceq U/_{SIM(Q)}$ then $IE(R|Q) \geq IE(R|P)$.
$P7$) If $U/_{SIM(P)} \preceq U/_{SIM(Q)}$ then $IE(P|R) \geq IE(Q|R)$.
$P8$) $IE(Q|P) + IE(P|R) \geq IE(Q|R)$.

Let $\mathbb{D} = (U, A \cup \{d\})$ be an IDT, Huang Bing et al [4] defined the reducts based on information quantity as the minimal subsets of attributes B such that $IE(B|\{d\}) = IE(A|\{d\})$ They are, in fact, the μ-reducts with respect to the *conditional generalize Liang entropy* measure, defined by

$$\mu_{IE}(B) = IE(B|\{d\}) = IE(B \cup \{d\}) - IE(B) \tag{12}$$

4 Metric between Coverings and Properties

Recall that any map $d : X \times X \to [0, \infty)$ that satisfies the following conditions:

$M1$) $d(x, y) \geq 0$, $d(x, y) = 0$ if and only if $x = y$.
$M2$) $d(x, y) = d(y, x)$.
$M3$) $d(x, y) + d(y, z) \geq d(x, z)$.

for any $x, y, z \in X$ is called *a metric on* X [2].

The condition $M3$) is called the triangular inequality. The pair (X, d) is called a metric space. Based on the generalized Liang entropy, in this Section we establish a metric between coverings and study some properties of the proposed metric for attribute reduction in incomplete decision tables.

Theorem 1 (Metric). *For any incomplete information system* $\mathbb{S} = (U, A)$*, the map* $d_E : COVER(U) \times COVER(U) \to [0, \infty)$*, defined by*

$$d_E(K(P), K(Q)) = IE(P|Q) + IE(Q|P) \tag{13}$$

where $P, Q \subset A$*, is a metric on* $COVER(U)$.

Proof. We will show that d_E satisfies three properties of metric functions:

(M1) From Property $P5$) we have $d_E(K(P), K(Q)) \geq 0$ for any $P, Q \subset A$ and the equality holds if and only if $(IE(Q|P) = 0)$ and $(IE(P|Q) = 0)$, i.e.,

$$\left(U/_{SIM(P)} \preceq U/_{SIM(Q)}\right) \wedge \left(U/_{SIM(Q)} \preceq U/_{SIM(P)}\right) \Leftrightarrow K(P) = K(Q)$$

(M2) From the definition of d_E, it is easy to see that

$$d_E(K(P), K(Q)) = d_E(K(Q), K(P))$$

for any $K(P), K(Q) \in COVER(U)$.

(M3) For any $P, Q, R \subset A$, from Property $P5$) we have

$$IE(Q|P) + IE(P|R) \geq IE(Q|R) \text{ and } IE(R|P) + IE(P|Q) \geq IE(R|Q)$$

Thus we have $d_E(K(Q), K(P)) + d_E(K(P), K(R)) \geq d_E(K(Q), K(R))$

Therefore all conditions $(M1), (M2), (M3)$ are satisfied, we can conclude that d_E is a metric on $COVER(U)$

The following propositions present some properties of the metric d_E. The proofs of those facts are omitted due to lack of space.

Proposition 3. *Let $\mathbb{S} = (U, A)$ be an incomplete information system. For any subsets $B, C \subset A$:*

$$a) \quad d_E\left(K\left(B\right), K\left(C\right)\right) = \frac{1}{|U|}\sum_{i=1}^{|U|}\frac{|S_B(u_i)| - |S_C(u_i)|}{|U|} \tag{14}$$

$$b) \quad \text{if } B \subseteq C \text{ then } d_E\left(K\left(B\right), K\left(B \cup \{d\}\right)\right) \geq d_E\left(K\left(C\right), K\left(C \cup \{d\}\right)\right) \tag{15}$$

Proposition 3 b) states that the bigger the attribute set B is, the smaller the metric $d_E\left(K\left(B\right), K\left(B \cup \{d\}\right)\right)$ is, and vice versa. In other words, the metric decreases as tolerance classes become smaller through finer classification.

5 Metric Based Reducts in Incomplete Decision Tables

In next content, we define the reduct based on the proposed metric and prove that this reduct is the same as the reduct based on information quantity.

Definition 4. *If the set of attributes $R \subseteq A$ satisfies the following conditions:*

(1) $d_E\left(K\left(R\right), K\left(R \cup \{d\}\right)\right) = d_E\left(K\left(A\right), K\left(A \cup \{d\}\right)\right)$
(2) $\forall r \in R,\ d_E\left(K\left(R - \{r\}\right), K\left(\left(R - \{r\}\right) \cup \{d\}\right)\right) \neq d_E\left(K\left(A\right), K\left(A \cup \{d\}\right)\right)$

then R is called a reduct of A based on metric.

Proposition 4. *Let $\mathbb{D} = (U, A \cup \{d\})$ be an incomplete decision table and $B \subseteq A$. Then $d_E(K(B), K(B \cup \{d\})) = d_E(K(A), K(A \cup \{d\}))$ if and only if*

$$IE\left(B\,|\{d\}\right) = IE\left(A\,|\{d\}\right).$$

Proof. Let us consider $U = \{u_1, ..., u_n\}$ and $B \subseteq A$. Since $B \subset B \cup \{d\}, A \subset A \cup \{d\}$, and $d_E\left(K\left(B\right), K\left(B \cup \{d\}\right)\right) = d_E\left(K\left(A\right), K\left(A \cup \{d\}\right)\right)$, it follows from Proposition 3 that

$$\frac{1}{|U|}\sum_{i=1}^{|U|}\frac{|S_B(u_i)| - \left|S_{B\cup\{d\}}(u_i)\right|}{|U|} = \frac{1}{|U|}\sum_{i=1}^{|U|}\frac{|S_A(u_i)| - |S_{A\cup\{d\}}(u_i)|}{|U|} \Leftrightarrow$$

$$\Leftrightarrow \left(1 - \frac{1}{|U|^2}\sum_{i=1}^{|U|}\left|S_{B\cup\{d\}}(u_i)\right|\right) - \left(1 - \frac{1}{|U|^2}\sum_{i=1}^{|U|}|S_B(u_i)|\right)$$

$$= \left(1 - \frac{1}{|U|^2}\sum_{i=1}^{|U|}\left|S_{A\cup\{d\}}(u_i)\right|\right) - \left(1 - \frac{1}{|U|^2}\sum_{i=1}^{|U|}|S_A(u_i)|\right)$$

According to Equation 12, the last equation is equivalent to

$$IE\left(B \cup \{d\}\right) - IE\left(B\right) = IE\left(A \cup \{d\}\right) - IE\left(A\right)$$

which is equivalent to $IE\left(B\,|\{d\}\right) = IE\left(A\,|\{d\}\right)$. This completes the proof.

Therefore, we can conclude from Proposition 4 that the reduct based on proposed metric is the same as that based on information quantity in incomplete decision tables.

Definition 5. *Let $\mathbb{D} = (U, A \cup \{d\})$ be an incomplete decision table and $B \subseteq A$. The significance of attribute $b \in A - B$ is defined as*

$$SIG_B(b) = d_E(K(B), K(B \cup \{d\})) - d_E(K(B \cup \{b\}), K(B \cup \{b\} \cup \{d\})),$$

where $S_\emptyset(u_i) = U$ for any $u_i \in U, i = 1, ..., |U|$.

Definition 5 implies that the significance of attribute $b \in A - B$ is measured by the changes of the metric $d_E\left(K\left(B\right), K\left(B \cup \{d\}\right)\right)$ when b is added to B, the bigger the value of $SIG_B(b)$, the more important the attribute b. This significance of attribute will be treated as the attribute selection criterion in our heuristic algorithm for attribute reduction

The heuristic search for short metric based reducts in incomplete decision tables is presented in Algorithm 1 (Algorithm **MBR**). In order to find the best reduct, the algorithm begins with $R = \emptyset$, then the most important attribute is chosen from searching space and added into R. The above processes are done until we get the best reduct.

Algorithm 1. MBR: metric-based reduct for incomplete decision table

Data: An incomplete decision table $\mathbb{D} = (U, A \cup \{d\})$;
Output: The short metric-based reduct R of $\mathbb{D}$;

$R = \emptyset$;
Calculate $d_E\left(K\left(R\right), K\left(R \cup \{d\}\right)\right)$ and $T = d_E\left(K\left(A\right), K\left(A \cup \{d\}\right)\right)$;
`// Iterative insertion of the most important attribute to` R
while $d_E(K(R), K(R \cup \{d\})) \neq T$ **do**
  for *each* $a \in A - R$ **do**
    Calculate $S = d_E(K(R \cup \{a\}), K(R \cup \{a\} \cup \{d\}))$;
    $SIG_R(a) = d_E(K(R), K(R \cup \{d\})) - S$;
  $R = R \cup \left\{ \underset{a \in A-R}{ArgMax} \left\{SIG_R\left(a\right)\right\} \right\}$;
  Calculate $d_E\left(K\left(R\right), K\left(R \cup \{d\}\right)\right)$;
`// Deleting redundant attributes in` R
for *each* $a \in R$ **do**
  Calculate $d_E\left(K\left(R - \{a\}\right), K\left(R - \{a\} \cup \{d\}\right)\right)$;
  if $d_E\left(K\left(R - \{a\}\right), K\left(R - \{a\} \cup \{d\}\right)\right) = T$ **then** $R = R - \{a\}$
return R;

Let us consider While loop from command line 3 to 8. To calculate $SIG_R(a)$, we need to calculate $S_{R\cup\{a\}}(u_i)$, $S_{R\cup\{a\}\cup\{d\}}(u_i)$ because $S_R(u_i), S_{R\cup\{d\}}(u_i)$ have already calculated in the previous step. According to Zhang et al [21], the time complexity to calculate $S_{R\cup\{a\}}(u_i)$ for $\forall u_i \in U$ when $S_R(u_i)$ calculated is $O\left(|U|^2\right)$. So the time complexity to calculate all $SIG_E(a)$ is

$$(|A| + (|A| - 1) + ... + 1) * |U|^2 = (|A| * (|A| - 1)/2) * |U|^2 = O\left(|A|^2|U|^2\right),$$

where $|A|$ is the number of conditional attributes and $|U|$ is the number of objects. The time complexity to choose the attribute with maximum significance is $|A| + (|A| - 1) + ... + 1 = |A| * (|A| - 1)/2 = O\left(|A|^2\right)$. Hence, the time complexity of While loop is $O\left(|A|^2|U|^2\right)$. Similarly, the time complexity of For loop from command line 10 to 12 is $O\left(|A|^2|U|^2\right)$. Consequently, the time complexity of Algorithm 1 is $O\left(|A|^2|U|^2\right)$, which is less than that of [3], [4], [22]. However, the time complexity of Algorithm 1 is the same as that of [21].

5.1 Example

Table 1. Car descriptions

Car	$Price$	$Mileage$	$Size$	$Max\text{-}speed$	d
u_1	High	High	Full	Low	Good
u_2	Low	*	Full	Low	Good
u_3	*	*	Compact	High	Poor
u_4	High	*	Full	High	Good
u_5	*	*	Full	High	Excellent
u_6	Low	High	Full	*	Good

In this Section we consider the descriptions of cars as in Table 1 [4]. This is an incomplete decision table $\mathbb{D} = (U, A \cup \{d\})$, where

$$U = \{u_1, u_2, u_3, u_4, u_5, u_6\} \text{ and } A = \{Car, Price, Mileage, Size, Max\text{-}speed\}.$$

For simplification we will denote the attributes by a_1, a_2, a_3, a_4 respectively. Firstly, let us calculate the knowledge bases of the following sets of attributes:

$$\begin{aligned}
K(\{a_1\}) =& \{\{u_1, u_3, u_4, u_5\}, \{u_2, u_3, u_5, u_6\}, U, \{u_1, u_3, u_4, u_5\}, U,\\
& \{u_2, u_3, u_5, u_6\}\}\\
K(\{a_2\}) =& \{U, U, U, U, U, U\}\\
K(\{a_3\}) =& \{\{u_1, u_2, u_4, u_5, u_6\}, \{u_1, u_2, u_4, u_5, u_6\}, \{u_3\}, \{u_1, u_2, u_4, u_5, u_6\},\\
& \{u_1, u_2, u_4, u_5, u_6\}, \{u_1, u_2, u_4, u_5, u_6\}\}\\
K(\{a_4\}) =& \{\{u_1, u_2, u_6\}, \{u_1, u_2, u_6\}, \{u_3, u_4, u_5, u_6\}, \{u_3, u_4, u_5, u_6\},\\
& \{u_3, u_4, u_5, u_6\}, U\}
\end{aligned}$$

$$K(A) = \{\{u_1\}, \{u_2, u_6\}, \{u_3\}, \{u_4, u_5\}, \{u_4, u_5, u_6\}, \{u_2, u_5, u_6\}\}$$
$$K(\{d\}) = \{\{u_1, u_2, u_4, u_6\}, \{u_1, u_2, u_4, u_6\}, \{u_3\}, \{u_1, u_2, u_4, u_6\}, \{u_5\}, \{u_1, u_2, u_4, u_6\}\}$$

According to lines 1 and 2 of Algorithm 1, we set $R = \emptyset$ and calculate

$$T = d_E\left(K\left(A\right), K\left(A \cup \{d\}\right)\right) = \frac{1}{|U|^2} \sum_{i=1}^{6} \left(\left|S_A(u_i) - \left(S_A(u_i) \cap S_{\{d\}}(u_i)\right)\right|\right) = \frac{4}{36}.$$

Now, we start the first iteration of the While loop by the calculation of attribute significance:

$$SIG_{\emptyset}\left(a_1\right) = \frac{1}{|U|^2} \sum_{i=1}^{|U|} \left(\left|S_{\emptyset}(u_i) - S_{\{d\}}(u_i)\right| - \left|S_{\{a_1\}}(u_i) - S_{\{a_1,d\}}(u_i)\right|\right) = 0.$$

Similarly, $SIG_{\emptyset}\left(a_2\right) = 0$, $SIG_{\emptyset}\left(a_3\right) = \frac{10}{36}$, $SIG_{\emptyset}\left(a_4\right) = \frac{8}{36}$. Choose a_3 which haves the most significance and $R = \{a_3\}$. After calculation of

$$d_E\left(K\left(\{a_3\}\right), K\left(\{a_3, d\}\right)\right) = \frac{8}{36},$$

we can see that $d_E\left(K\left(\{a_3\}\right), K\left(\{a_3, d\}\right)\right) \neq d_E\left(K\left(A\right), K\left(A \cup \{d\}\right)\right)$. Thus we have to perform the second loop.

$$SIG_{\{a_3\}}(a_1) = \frac{2}{36}, SIG_{\{a_3\}}(a_2) = 0, SIG_{\{a_3\}}(a_4) = \frac{4}{36}.$$

Choose a_4 which haves the most significance and $R = \{a_3, a_4\}$. Calculate

$$d_E\left(K\left(\{a_3, a_4\}\right), K\left(\{a_3, a_4, d\}\right)\right) = \frac{4}{36} = d_E\left(K\left(A\right), K\left(A \cup \{d\}\right)\right).$$

Hence, go to For loop. We can see that

$$d_E\left(K\left(\{a_3\}\right), K\left(\{a_3, d\}\right)\right) = \frac{8}{36} \neq T; \quad d_E\left(K\left(\{a_4\}\right), K\left(\{a_4, d\}\right)\right) = \frac{10}{36} \neq T.$$

As a consequence, the algorithm finishes and returns $R = \{a_3, a_4\}$ as the best reduct of A. This result is the same as the result in the example in reference [4].

6 Experiments

The experiments on PC (Pentium Dual Core 2.13 GHz, 1GB RAM, WINXP) are performed on 6 data sets obtained from UCI Machine Learning Repository [20]. We choose information quantity based attribute reduction algorithm [4] (IQBAR for short) to compare with the proposed algorithm. The results of experiments are showm in Table 2 and Table 3, where $|U|$, $|A|$, $|R|$ are the numbers of objects, primal condition attributes, and after reduction respectively, and t is the time of operation (calculated by second). Condition attributes will be denoted by $1, 2, ..., |A|$. The results show that the reduct of the proposed algorithm is the same as that of the IQBAR algorithm. However, the time of operation in the proposed algorithm is less than that in the IQBAR algorithm.

Table 2. The results of the proposed algorithm and IQBAR algorithm

Seq.	Data sets	$\|U\|$	$\|A\|$	Algorithm *IQBAR*		Algorithm **MBR**	
				$\|R\|$	Comp. time	$\|R\|$	Comp. time
1	Hepatitis	155	19	4	1.296	4	0.89
2	Lung-cancer	32	56	4	0.187	4	0.171
3	Automobile	205	25	5	3	5	1.687
4	Anneal	798	38	9	179	9	86.921
5	Voting Records	435	16	15	25.562	15	16.734
6	Credit Approval	690	15	7	29.703	7	15.687

Table 3. The reducts of the proposed algorithm and IQBAR algorithm

Seq	Data sets	The reducts of Alg. *IQBAR*	The reducts of Alg. **MBR**
1	Hepatitis	$\{1, 2, 4, 17\}$	$\{1, 2, 4, 17\}$
2	Lung-cancer	$\{3, 4, 9, 43\}$	$\{3, 4, 9, 43\}$
3	Automobile	$\{1, 13, 14, 20, 21\}$	$\{1, 13, 14, 20, 21\}$
4	Anneal	$\{1, 3, 4, 5, 8, 9, 33, 34, 35\}$	$\{1, 3, 4, 5, 8, 9, 33, 34, 35\}$
5	Voting Records	$\{1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$	$\{1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$
6	Credit Approval	$\{1, 2, 3, 4, 5, 6, 8\}$	$\{1, 2, 3, 4, 5, 6, 8\}$

7 Conclusion

Attribute reduction is one of the crucial problems in both rough set theory for complete information systems and tolerance rough set for incomple information systems. In this paper, a generalized Liang entropy is proposed based on Liang entropy [6] and some of its properties are considered in incomplete information systems. Based on the generalized Liang entropy, a metric is established between coverings and a metric based attribute reduction method in incomplete decision tables is proposed. To construct the metric based attribute reduction method, we define the reduct based on metric, the significance of an attribute based on metric. We use the significance of an attribute as heuristic information to design and implemement an efficient attribute reduction algorithm in incomplete decision tables. We also prove theoretically and experimentally that the reduct based on metric is the same as that base on information quantity [4] and the time complexity of the proposed algorithm is less than that of the information quantity based algorithm [4].

References

1. Dai, X.P., Xiong, D.H.: Research on Heuristic Knowledge Reduction Algorithm for Incomplete Decision Table. In: 2010 International Conference on Internet Technology and Applications, pp. 1–3. IEEE (2010)
2. Deza, M.M., Deza, E.: Encyclopedia of Distances. Springer (2009)
3. Huang, B., He, X., Zhou, X.Z.: Rough Computational methods based on tolerance matrix. Acta Automatica Sinica 30, 363–370 (2004)

4. Huang, B., Li, H.X., Zhou, X.Z.: Attribute Reduction Based on Information Quantity under Incomplete Information Systems. Systems Application Theory and Practice 34, 55–60 (2005)
5. Kryszkiewicz, M.: Rough set approach to incomplete information systems. Information Science 112, 39–49 (1998)
6. Liang, J.Y., Chin, K.S., Dang, C.Y., Richard, C.M.Y.: New method for measuring uncertainty and fuzziness in rough set theory. International Journal of General Systems 31, 331–342
7. Liang, J.Y., Qian, Y.H.: Axiomatic approach of knowledge granulation in information system. In: Sattar, A., Kang, B.-H. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 1074–1078. Springer, Heidelberg (2006)
8. Liang, J.Y., Qian, Y.H.: Information granules and entropy theory in information systems. Information Sciences 51, 1–18 (2008)
9. Liang, J.Y., Shi, Z.Z., Li, D.Y., Wierman, M.J.: The information entropy, rough entropy and knowledge granulation in incomplete information system. International Journal of General Systems 35(6), 641–654 (2006)
10. Nguyen, L.G.: Metric Based Attribute Reduction in Decision Tables. In: The 2012 International Workshop on Rough Sets Applications (RSA 2012), FedCSIS Proceedings, pp. 333–338 (2012), http://fedcsis.org/proceedings/fedcsis2012/
11. Pawlak, Z.: Rough sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishers (1991)
12. Qian, Y.H., Liang, J.Y.: Combination Entropy and Combination Granulation in Incomplete Information System. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS (LNAI), vol. 4062, pp. 184–190. Springer, Heidelberg (2006)
13. Qian, Y.H., Liang, J.Y.: New method for measuring uncertainty in incomplete information systems. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (2008)
14. Qian, Y.H., Liang, J.Y., Dang, C.Y.: Knowledge structure, knowledge granulation and knowledge distance in a knowledge base. International Journal of Approximate Reasoning 50, 174–188 (2009)
15. Qian, Y.H., Liang, J.Y., Dang, C.Y., Wang, F., Xu, W.: Knowledge distance in information systems. Journal of Systems Science and Systems Engineering 16, 434–449 (2007)
16. Mantaras, R.L.: A Distance-Based Attribute Selection Measure for Decision Tree Induction. Machine Learning 6(1), 81–92 (1991)
17. Shifei, D., Hao, D.: Research and Development of Attribute Reduction Algorithm Based on Rough Set. In: IEEE, CCDC 2010, pp. 648–653 (2010)
18. Simovici, D.A., Jaroszewicz, S.: Generalized conditional entropy and decision trees. In: Proceeding of EGC, Lyon, France, pp. 369–380 (2003)
19. Simovici, D.A., Jaroszewicz, S.: A new metric splitting criterion for decision trees. International Journal of Parallel Emergent and Distributed Systems 21(4), 239–256 (2006)
20. The UCI machine learning repository, http://archive.ics.uci.edu/ml/datasets.html
21. Zhang, Q.G., Zheng, X.F., Xu, Z.Y.: Efficient Attribute Reduction Algorithm Based on Incomplete Decision Table. In: 2009 Second International Conference on Intelligent Computation Technology and Automation, pp. 192–195. IEEE (2009)
22. Zhou, X.Z., Huang, B.: Rough set-based attribute reduction under incomplete Information Systems. Journal of Nanjing University of Science and Technology 27, 630–636 (2003)

The Completion Algorithm in Multiple Decision Tables Based on Rough Sets

Na Jiao

Department of Information Science and Technology,
East China University of Political Science and Law, Shanghai 201620, P.R. China
jiaonaecupl@gmail.com

Abstract. It is well known that data mining and knowledge discovery on incomplete data are difficult and inevitable. However, the previous analysis on Rough Sets has been developed under a single decision table, but not under multiple decision tables. In this paper, introducing a general significance of attributes in multiple decision tables, a completion algorithm in multiple decision tables based on Rough Sets is proposed. Through the experiments, it is shown that the algorithm is effective to process incomplete multiple decision tables.

Keywords: rough sets, multiple decision tables, completion algorithm, missing attributes.

1 Introduction

Rough set theory [1, 2], introduced by Z.Pawlak in the early 1980s, is a mathematical tool to deal with vagueness and uncertainty. This theory has been applied to machine learning, data mining, decision analysis, and pattern recognition, etc [3]. It is known that data mining and knowledge discovery on incomplete data are difficult and inevitable. The previous methods [4-6] to deal with incomplete data are developed under a single decision table, but not under multiple decision tables.

In order to treat the problem, several approaches have been proposed in papers [7, 8]. Inuiguchi et al. [7] have discussed an approach to complete missing objects in multiple decision tables when attributes are common among decision tables.

However, we may have multiple decision tables when the information comes from multiple information sources or when objects are evaluated by multiple decision makers. When each decision table is obtained from a decision maker, it can be regarded as partial information about the opinion of the decision maker. The decision table possibly lacks certain attributes. These decision tables may have common attributes and different attributes. It will impact efficiency of data mining. In this paper, we introduce a new completion algorithm in multiple decision tables based on Rough Sets. Through the experiments, it is shown that the algorithm is effective to process incomplete multiple decision tables.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 111–118, 2013.

2 Definitions

A decision table is defined as a four-tuple $< U, C \cup \{d\}, V, f >$, where $U = \{x_1, x_2, ..., x_n\}$ is a universe of elements; C is a set of condition attributes and d is a decision attribute; $V = \cup_{a \in C \cup \{d\}} V_a$ is a set of values for all attributes, where V_a is a set of values for each attribute $a \in C$; and $f : U \times (C \cup \{d\}) \to V$ is an information function such that for every element $x \in U$, $a \in U \cup \{d\}$, $f(x, a) \in V_a$ is the value of attribute a for element x. This paper uses a representation of decision table in [7]. A decision table is formally defined as follows.

Definition 1. *Let d be a decision attribute, V_d be the set of values for decision attribute d. Then we can rewrite a decision table described by a four-tuple $< W, C \cup \sum, V, f >$, where $W = \{w_1, ..., w_t\}$ is a set of patterns, $\mathrm{w_i} = \cup_{a \in C} \{\langle a, a(x)\rangle\}$ is a pattern ($i = 1, ..., t$), $a(x)$ is the value of attribute $a \in C$ for the object $x \in U$; C is a set of condition attributes; $\Sigma = \{\sigma_C(w_i, v_d), \forall v_d \in V_d\}$ is a set of frequencies of each pattern w_i , where σ_C is a frequency function, $\sigma_C(w_i, v_d)$ is frequency for each pattern w_i and decision attribute value v_d ; $V = \cup_{a \in C} V_a$ is a set of values for all condition attributes, where V_a is a set of values for each attribute a; $f : W \times C \to V$ is an information function; For $w \in W$, $a \in C$, $f(w, a) \in V_a$ is the value of attribute a for pattern w.*

For example, the decision table shown in Table 1 can be rewritten as the table shown in Table 2.

Table 1. An example of decision table

U	Design	Function	Size	Dec
x_{11}	Classic	Simple	Compact	Accept
x_{12}	Classic	Simple	Compact	Accept
x_{13}	Classic	Simple	Compact	Reject
x_{21}	Classic	Multiple	Normal	Accept
x_{22}	Classic	Multiple	Normal	Accept

Table 2. A decision table rewritten

W	Design	Function	Size	Σ
w_1	Classic	Simple	Compact	(2,1)
w_2	Classic	Multiple	Normal	(2,0)

According to definition 1, for any decision table $T_i =< W_i, C_i \cup \sum_i, V_i, f_i >$, where C_i is a set of condition attributes with respect to decision table T_i. Then, we obtain the following definitions.

Definition 2. *According to definition 1, a set of decision tables is defined as*

$$\Gamma = \{T_i, i = 1, ..., h\},\ T_i =< W_i, C_i \cup \sum_i, V_i, f_i >,$$

where C_i is a set of condition attributes with respect to T_i;

The set of all condition attributes in Γ is defined as

$$C(\Gamma) = \bigcup_{i=1}^{h} C_i;$$

The set of missing attributes in T_i is defined as

$$A_i = \{a | a \notin C_i \wedge a \in C\};$$

The set of all missing attributes in Γ is defined as

$$A(\Gamma) = \bigcup_{i=1}^{h} A_i;$$

The set of decision tables which include attribute a in Γ is defined as

$$Z(\Gamma, a) = \{T_i | a \in C_i, \forall T_i \in \Gamma\};$$

The set of decision tables which lack attribute a in Γ is defined as

$$Y(\Gamma, a) = \{T_i | a \notin C_i, \forall T_i \in \Gamma\}.$$

Definition 3. *Let w_i be a pattern in one decision table T_i and W_j be the set of patterns in another decision table T_j, $i \neq j$, the set of values of common condition attributes between w_i and W_j denoted by $B(w_i, W_j)$ is defined as*

$$B(w_i, W_j) = \{< a, a(w_i) > | \forall a \in C_i \cap C_j, \exists w \in W_j, a(w_i) = a(w)\};$$

The set of common patterns between W_j and $B(w_i, W_j)$ denoted by $F(w_i, W_j)$ is defined as

$$F(w_i, W_j) = \{w | \forall < a, v_a > \in B(w_i, W_j), \forall w \in W_j, v_a = a(w)\};$$

The set of common patterns between $F(w_i, W_j)$ and $< b, v_b >$ denoted by $L(< b, v_b >, w_i, W_j)$ is defined as

$$L(< b, v_b >, w_i, W_j) = \{w | \forall w \in F(w_i, W_j), b(w) = v_b\}.$$

3 The Completion Algorithm in Multiple Decision Tables Based on Rough Sets

In multiple decision tables, the missing attributes in different decision tables may be entirely different. For example, a decision table possibly lacks several attributes, or several decision tables all lack an identical attribute, and furthermore, the significance of attributes may be different even though there is an identical missing attribute in different decision tables, the general significance of attributes in multiple decision tables should be defined.

3.1 The General Significance of Attributes in Multiple Decision Tables

In order to treat the error caused by human evaluation as well as to accommodate disagreements among decision tables, we introduce significance as defined in Variable Precision Rough Sets. In definition 5, the general significance of attributes is heuristic information for the following algorithm, which is used to avoid the formidable computational load.

Definition 4. [9, 10] *To $T_i \in \Gamma$, for each attribute $a \in C_i$, the significance of a denoted by $SGF(a, W_i)$ is defined as*

$$SGF\left(a,W_i\right)=\frac{\left|\mathrm{POS}_{C_i}^{\beta}\left(\sum_i\right)\right|-\left|\mathrm{POS}_{C_i-\{a\}}^{\beta}\left(\sum_i\right)\right|}{|W_i|},$$

where $\beta\in[0,0.5)$ *is an allowable error ratio and* $\left|POS_{C_i}^{\beta}\left(\sum_i\right)\right|=\sum_{\forall w\in W_i,\exists v_d\in V_d,\sigma_{C_i}(w,v_d)}\Big/\sum_{\forall v_d\in V_d}\sigma_{C_i}(w,v_d)\geq 1-\beta\,\sigma_{C_i}\left(w,v_d\right)$.

Definition 5. *In* Γ*, the general significance of a denoted by* $S\left(a\right)$ *is defined as*

$$S\left(a\right)=\frac{\sum_{i=1}^{h}|W_i|SGF(a,W_i)}{\sum_{i=1}^{h}|W_i|}.$$

The larger $S\left(a\right)$ *is, the more important the attribute a is. According to the heuristic information* $S\left(a\right)$*, we may reduce the searching space.*

3.2 The Completion Algorithm for Missing Attributes in Multiple Decision Tables

We introduce the method which simultaneously adds a_m to all the decision tables $Y(\Gamma,a_m)$ missing this attribute.

Firstly, suppose the domain of a_m is $V_{a_m}=\{v_{a_m}^1,...,v_{a_m}^r\}$. After a_m is added, we replace the original pattern w_u with r patterns.

$$\begin{array}{l} w_{u1}=\left\{w_u\left(c_1\right),w_u(c_2),...,w_u(c_q),v_{a_m}^1\right\}\\ \vdots\\ w_{ur}=\left\{w_u\left(c_1\right),w_u(c_2),...,w_u(c_q),v_{a_m}^r\right\}\end{array},$$

where $\left\{w_u\left(c_1\right),w_u(c_2),...,w_u(c_q)\right\}$ is values of pattern w_u.

Frequency of the original pattern is redistributed by r patterns. We define the ratio of frequency of every new pattern to frequencies of r patterns as follows.

$$\mu\left(w_u,v_d,W_j,\left\langle a_m,v_{a_m}^i\right\rangle\right)=\begin{cases}\sum_{w\in L}\sigma(w,v_d)\Big/\sum_{w\in F}\sigma(w,v_d), if\ F\neq\phi\\ 0, if\ F=\phi\end{cases},$$

where $i=1,\cdots,r;v_d$ is a value of decision attribute d; $F=F\left(w_u,W_j\right)$; $L=L\left(\left\langle a_m,v_{a_m}^i\right\rangle,w_u,W_j\right)$.

The average of the ratio is defined as follows.

$$\sigma\left(w_{ui},v_d\right)=\sigma\left(w_u,v_d\right)\times\frac{\sum_{j=1}^{|Z(\Gamma,a_m)|}\mu\left(w_u,v_d,W_j,\left\langle a_m,v_{a_m}^i\right\rangle\right)}{|Z(\Gamma,a_m)|},$$

where $i=1,\cdots,r$; v_d is a value of decision attribute d.

Delete the patterns whose frequencies are 0, we obtain the new decision tables.

It is shown that the completion algorithm in multiple decision tables in Algorithm 1.

Algorithm 1. Completion Algorithm in Multiple Decision Tables

Iutput: A set of incomplete decision tables $\Gamma = \{T_i, i = 1, ..., h\}$, where $T_i = \langle W_i, C_i \cup \Sigma_i, V_i, f_i \rangle$; $C = \bigcup_{i=1}^{h} C_i$ is the set of all condition attributes; $A_i = \{a | a \notin C_i \wedge a \in C\}$ is the set of missing attributes in T_i; $A = \bigcup_{i=1}^{h} A_i$ is the set of all missing attributes; A given threshold $\beta (0 \leq \beta < 0.5)$.

Output: Complete decision tables $\Gamma' = \{T'_i, i = 1, ..., h\}$, where $T'_i = \langle W'_i, C \cup \Sigma'_i, V'_i, f'_i \rangle$.

Step 1 Set $\Gamma' = T, C' = C, A' = A$.

Step 2 If $A' = \emptyset$, then go to step 5, otherwise step 3.

Step 3 According to definition 5, calculate the most important missing attribute $a_m = \arg \max_{a \in A'} \{S(a)\}$ in Γ'.

Step 4 The attribute a_m is added to all the decision tables $Y(\Gamma, a_m)$ which miss this attribute.

1. Add the attribute a_m to all the decision tables which miss this attribute and make sure that new condition attributes and new attribute values are filled in new patterns.
2. Calculate frequency of every new pattern.
3. Delete the patterns whose frequencies are 0.
4. Set $C' = C' \cup \{a_m\}$, $A' = A' - \{a_m\}$, go to step 2.

Step 5 Γ' is the result.

4 An Illustrative Example and Analysis

4.1 An Illustrative Example

In order to test the validity of the algorithm, Solar-Flare dataset (the dataset can be downloaded at ftp://ftp.ics.uci.edu/pub/machine-learning-databases) from UCI database is used. We select {Activity, Evolution, Historically-complex, Did region become, Area, C-class flares} 5 condition attributes and 1 decision attribute to construct 4 decision tables. $V_d = \{0, 1\}$, $\Sigma = (\sigma_C(w_i, 0), \sigma_C(w_i, 1))$, $\Gamma = \{T_1, T_2, T_3, T_4\}$ shown in Table 3. Suppose that 4 decision tables are evaluated by 4 decision makers who have their own partial opinion. These decision tables shown in Table 4 with missing attributes are generated from the 4 complete tables $T_1 \sim T_4$ which 4 attributes are removed from at random. T_1 and T_2 lack a_2. T_3 lacks a_4. T_4 lacks a_5. $C_1 = \{a_1, a_3, a_4, a_5\}$, $C_2 = \{a_1, a_3, a_4, a_5\}$, $C_3 = \{a_1, a_2, a_3, a_5\}$, $C_4 = \{a_1, a_2, a_3, a_4\}$, $C(\Gamma) = \{a_1, a_2, a_3, a_4, a_5\}$; $A_1 = \{a_2\}$, $A_2 = \{a_2\}$, $A_3 = \{a_4\}$, $A_4 = \{a_5\}$, $A(\Gamma) = \{a_2, a_4, a_5\}$.

Set β=0.2, as definition 5 the attribute a_2 is the missing attribute in decision tables T_1 and T_2, the general significance of a_2 is 0.0815. Similarly, the general significance of missing attributes a_4 and a_5 is 0. We obtain the most important missing attribute a_2 and add it to T_1 and T_2 . The domain of a_2 is $V_{a_2} = \{1, 2\}$, then, after the attribute a_2 is added, we replace each pattern with 2 patterns.

Table 3. The set of original decision tables

Table 3.1.

W_1	a_1	a_2	a_3	a_4	a_5	Σ
w_1	1	1	1	2	1	(12,1)
w_2	1	2	2	2	1	(20,1)
w_3	1	2	2	1	1	(15,0)
w_4	1	2	1	2	1	(24,12)
w_5	2	2	1	2	1	(3,10)
w_6	2	2	2	2	1	(25,3)
w_7	2	2	2	2	2	(11,0)

Table 3.2.

W_2	a_1	a_2	a_3	a_4	a_5	Σ
w_1	1	1	1	2	1	(13,1)
w_2	1	2	2	2	1	(15,0)
w_3	1	2	2	1	1	(11,0)
w_4	1	2	1	2	1	(20,10)
w_5	2	2	1	2	1	(4,15)
w_6	2	2	2	2	1	(23,2)
w_7	2	2	2	2	2	(12,0)

Table 3.3.

W_3	a_1	a_2	a_3	a_4	a_5	Σ
w_1	1	1	1	2	1	(10,0)
w_2	1	2	2	2	1	(22,1)
w_3	1	2	2	1	1	(20,1)
w_4	1	2	1	2	1	(28,14)
w_5	2	2	1	2	1	(2,9)
w_6	2	2	2	2	1	(33,5)
w_7	2	2	2	2	2	(14,0)

Table 3.4.

W_4	a_1	a_2	a_3	a_4	a_5	Σ
w_1	1	1	1	2	1	(12,1)
w_2	1	2	2	2	1	(18,0)
w_3	1	2	2	1	1	(18,0)
w_4	1	2	1	2	1	(15,7)
w_5	2	2	1	2	1	(1,8)
w_6	2	2	2	2	1	(20,1)
w_7	2	2	2	2	2	(10,0)

Then, calculate frequencies of new patterns. According to definition 3 we get $B\left(w_{11}, W_3\right) = \{\langle a_1, 1\rangle, \langle a_3, 1\rangle, \langle a_5, 1\rangle\}$, $F(w_{11}, W_3) = \{w_{31}, w_{33}\}$, $L(< a_2, 1 >, w_{11}, W_3) = \{w_{31}\}$, $L(< a_2, 2 >, w_{11}, W_3) = \{w_{33}\}$. $\mu(w_{11}, 0, W_3, \langle a_2, 1\rangle) = \frac{\sigma(w_{31},0)}{\sigma(w_{31},0)+\sigma(w_{33},0)} = \frac{10}{10+28}$ =0.263, similarly, $\mu(w_{11}, 0, W_4, \langle a_2, 1\rangle)$ =0.444, frequency of pattern w_{111} on the decision attribute value 0 is $\sigma_{C_1'}(w_{111}, 0)$=$36 * (0.263 + 0.444)/2$ =12.74, in the same way, $\sigma_{C_1'}(w_{111}, 1)$ =0.82. Frequencies of the pattern w_{111} on the domain $V_d = \{0, 1\}$ of decision attribute d are (12.74,0.82). At last, delete the patterns whose frequencies are (0,0). The result is given in Table 4.

After completing the 4 decision tables, we round the frequencies of all patterns. Compare $T_1' \sim T_4'$ to $T_1 \sim T_4$, the error ratio of T_1' is $1/137 * 100\% = 0.73\%$, the error ratio of T_2' is 0.79%, the error ratio of T_3' is 1.26%, and the error ratio of T_4' is 0.9%. The average error ratio is 0.94%. The illustrative example test the validity and feasibility of our completion algorithm.

4.2 Experiment and Analysis

Experiment 1: We use datasets Dermatology, Car, Postoperative-Patient and Nursery from UCI database as test dataset. We select different objects and different condition attributes and 1 decision attribute to construct 4, 5, 3, 6 decision tables respectively and remove attributes at random. Multiple decision

Table 4. The set of final decision tables

Table 4.1. T_1' after completing a_2

W_1'	a_1	a_2	a_3	a_4	a_5	Σ
g_1	1	1	1	2	1	(12.74,0.82)
g_2	1	2	2	2	1	(20,1)
g_3	1	2	2	1	1	(15,0)
g_4	1	2	1	2	1	(23.26,12.18)
g_5	2	2	1	2	1	(3,10)
g_6	2	2	2	2	1	(25,3)
g_7	2	2	2	2	2	(11,0)

Table 4.2. T_2' after completing a_2

W_2'	a_1	a_2	a_3	a_4	a_5	Σ
g_1	1	1	1	2	1	(11.68,0.69)
g_2	1	2	2	2	1	(15,0)
g_3	1	2	2	1	1	(11,0)
g_4	1	2	1	2	1	(21.32,10.31)
g_5	2	2	1	2	1	(4,15)
g_6	2	2	2	2	1	(23,2)
g_7	2	2	2	2	2	(12,0)

Table 4.3. T_3' after completing a_4

W_3'	a_1	a_2	a_3	a_4	a_5	Σ
g_1	1	1	1	2	1	(10,0)
g_2	1	2	2	2	1	(23.08,2)
g_3	1	2	2	1	1	(18.92,0)
g_4	1	2	1	2	1	(28,14)
g_5	2	2	1	2	1	(2,9)
g_6	2	2	2	2	1	(33,5)
g_7	2	2	2	2	2	(14,0)

Table 4.4. T_4' after completing a_5

W_4'	a_1	a_2	a_3	a_4	a_5	Σ
g_1	1	1	1	2	1	(12,1)
g_2	1	2	2	2	1	(18,0)
g_3	1	2	2	1	1	(18,0)
g_4	1	2	1	2	1	(15,7)
g_5	2	2	1	2	1	(1,8)
g_6	2	2	2	2	1	(20.54,1)
g_7	2	2	2	2	2	(9.46,0)

Table 5. Experiment results for different datasets

Dataset	N1	N2	N3	N4	The average error ratio
Dermatology	4	5	300	4	0.75%
Car	5	6	928	4	1.35%
Postoperative-Patient	3	4	90	3	1.1%
Nursery	6	7	2400	8	2.1%

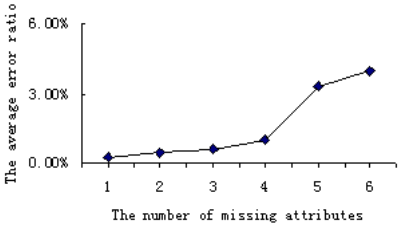


Fig. 1. Contrast the number of missing attributes with the average error ratio

tables are completed with the above algorithm. The experiment results are shown in Table 5 (N1: Number of decision tables; N2: Number of condition attributes; N3: Number of all objects; N4: Number of missing attributes).

Experiment 2: The dataset is Solar-Flare from UCI database. The attributes selected are the same as the above example. We select 1000 objects to construct 4 decision tables which remove 1, 2, 3, 4, 5, 6 attributes randomly. The 4 decision tables are completed with the above algorithm. The results (Fig. 1) illustrate that the larger number of missing attributes is, the higher average error ratio is. Missing attributes in each decision table should be small in number, otherwise, it will lead to appreciable error.

5 Conclusions

In this paper, a new algorithm is proposed to complete multiple decision tables. It keeps the integrity and consistency of all multiple decision tables, which provides a new idea for pre-process of multiple decision tables.

Acknowledgements. This paper is supported by the National Social Science Fund (Granted No. 13CFX049), Shanghai University Young Teacher Training Program (Granted No. hdzf10008) and the Research Fund for East China University of Political science and Law (Granted No. 11H2K034).

References

1. Pawlak, Z.: Rough Sets. International Journal of Information Computer Science 11(5), 341–356 (1982)
2. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: A tutorial. In: Pal, S.K., Skowron, A. (eds.) Rough Fuzzy Hybridization: A New Trend in Decision-Making, pp. 3–98. Springer (1999)
3. Yao, Y.Y.: The superiority of three-way decisions in probabilistic rough set models. Information Sciences 181(6), 1080–1096 (2011)
4. Kryszkiewicz, M.: Rough set approach to incomplete information system. Information Sciences 112, 39–49 (1998)
5. Pei, Z.: Rational decision making models with incomplete weight information for production line assessment. Information Sciences 222(10), 696–716 (2013)
6. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research 129(1), 1–47 (2001)
7. Inuiguchi, M., Miyajima, T.: Variable Precision Rough Set Approach to Multiple Decision Tables. In: Ślezak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3641, pp. 304–313. Springer, Heidelberg (2005)
8. Jiao, N., Miao, D.Q., Zhou, J.: Two novel feature selection methods based on decomposition and composition. Expert Systems with Applications 37(12), 7419–7426 (2010)
9. Ziarko, W.: Variable Precision Rough Set Model. Journal of Computer and System Sciences 46, 39–59 (1993)
10. Wang, G.Y.: Rough Sets Theory and Knowledge Acquisition, pp. 51–52. Xian JiaoTong University Publishing Company, Xian (2001)

Multi-label Classification Using Rough Sets

Ying Yu[1,2,3,⋆], Duoqian Miao[1,2], Zhifei Zhang[1,2], and Lei Wang[1,2]

[1] Department of Computer Science and Technology, Tongji University, Shanghai 201804, P.R. China
[2] Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, P.R. China
[3] Software School, Jiangxi Agriculture University, Jiangxi 330045, P.R. China

Abstract. In multi-label classification, each instance may be associated with multiple labels simultaneously which is different from the traditional single-label classification where an instance is only associated with a single label. In this paper, we propose two types of approaches to deal with multi-label classification problem based on rough sets. The first type of approach is to transform the multi-label problem into one or more single-label problems and then use the classical rough set model to make decisions. The second type of approach is to extend the classical rough set model in order to handle multi-label dataset directly, where the new model considers the correlations among labels. The effectiveness of multi-label rough set model is presented by a series of experiments completed for two multi-label datasets.

Keywords: rough sets, multi-label classification, correlation.

1 Introduction

Multi-label classification problems [1] widely exist in various applications where each instance is normally associated with multiple labels and the classes encountered in the problem are not mutually exclusive but may overlap.

There exists uncertainty during the process of multi-label classification due to the finite number of training instances and the ambiguity of concept themselves, which impacts the precision of the prediction. However, there is a lack of study on the uncertainty existing in the multi-label classification. Rough sets form a conceptual vehicle to deal with ambiguous, vague, and uncertain knowledge [2]. In this paper, several methods based on rough sets are proposed for the multi-label decision system.

The rest of this paper is organized as follows. Section 2 briefly reviews the related studies about rough sets and multi-label learning. In Section 3, two types of approaches for multi-label problem are proposed, which are respectively based

⋆ This paper is partially supported by the National Natural Science Foundation of China (Serial No. 61075056, 61273304, 61075056, 61103067, 61202170), and the State Scholarship Fund of China (File No. 201206260047).

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 119–126, 2013.

on classical rough set model and multi-label rough set model. Section 4 illustrates the effectiveness of multi-label rough set model through some experiments. Finally, Section 5 concludes the studies.

2 Related Works

This section briefly reviews some existing works on rough sets and multi-label learning that are pertinent to our study.

2.1 Rough Sets

Rough set theory, proposed in 1982 by Pawlak [2], is regarded as a tool to process inexact, uncertain or vague knowledge. Indiscernibility relation and Approximations are two important concepts in Pawlak rough set theory.

Rough set theory has attracted worldwide attention of many researchers and practitioners, who have contributed essentially to its development and applications. For example, in order to deal with incomplete information system, some researchers extend the equivalence relations to non-equivalence relations such as tolerance relation [3], similarity relation [4], limited tolerance relation [5], etc.. In order to support numerical attributes, Yao [6] and Hu [7] proposed the neighborhood rough set model based on the neighborhood relations.

2.2 Multi-label Learning

Multi-label classification is different from the traditional task of single-label classification where each instance is only associated with a single class label. An intuitive approach to multi-label learning is to decompose the task into a number of binary classification problems and each for one class. This kind of approaches include binary relevance method (BR) [1], binary pairwise classification approach (PW) [8] and label combination or label power-set method (LC) [9]. Such an approach, however, usually suffers from the deficiency that the correlation among the labels is not taken into account.

There are also numbers of multi-label classification algorithms derived from traditional machine learning methods. For example, Boostexter system [10] provides two boosting algorithms, Adaboost.MH and Adaboost.MR, which are two extensions of Adaboost for multi-label classification. Comit et al. [11] extended the alternating decision tree learning algorithm for multi-label classification. In addition, a number of multi-label methods are based on the popular k Nearest Neighbors (kNN) lazy learning algorithm [12].

3 Rough Sets Based Approaches for Multi-label Classification

In multi-label decision table, an object is associated with a subset of labels and different classes may overlap by definition in the feature space. Fig. 1(a) shows a

multi-label dataset which includes five instances with four labels *grass*, *tree*, *sky* and *water*. If we transfer Fig. 1(a) into Fig. 1(b), we find it looks like a single-label inconsistent decision table, where two objects with the same conditional features belong to different decision classes. In single-label classification system, the classes are mutually exclusive and the inconsistent problem was considered to be caused by noise, such as mistakes in recording process [13], which is in conflict with the definition of multi-label classification. We cannot directly cope with multi-label problem using the existing single-label inconsistent approaches. In this paper, we will present two types of rough sets based approaches for multi-label classification problem.

object	grass	tree	sky	water
1	X		X	
2		X	X	
3	X		X	X
4		X		
5			X	X

(a)

object	label
1	grass
1	sky
2	tree
2	sky
3	grass
3	sky
3	water
4	tree
5	sky
5	water

(b)

Fig. 1. Example of multi-label dataset and its transformation

Before introducing the methods, we present the formal notation in this paper. Let $MDT = \langle U, A \rangle$ be a multi-label decision table, where U is a finite, nonempty set called the universe, and $A = C \cup D$; $C = \{c_1, \ldots, c_n\}$ is the set of conditional attributes and $D = \{l_1, \ldots, l_m\}$ is the set of labels.

The first type of approach is to directly transform the multi-label problem into one or more traditional single-label problems and then use the classical rough set theory to obtain rules. As for the methods of transformation, we can refer to literature [1]. Fig. 1(a) is used as an original example to briefly exemplify these transformations.

For example, we can learn binary classifiers from original dataset, and one for each different label $l_j \in D$. Each dataset contains all instances of original dataset. The instance is labeled as 1, if the original label l_j is included and as 0, otherwise. Fig. 2 shows the result of transformation of Fig. 1(a) using this method. For a new object, its prediction is a set of labels which are output by classifiers. However, the precision of the decision suffers from the imbalance problem existing in the dataset.

In addition, we also can consider each different set of labels that exists in the multi-label dataset as a single-label. Fig. 3 shows the result of transformation of Fig. 1(a) using this method. The new labels come from the power set of D. This method suffers from the sparse problem that the dataset has a large number of classes as well as few examples per class.

object	grass
1	1
2	0
3	1
4	0
5	0

object	tree
1	0
2	1
3	0
4	1
5	0

object	sky
1	1
2	1
3	1
4	0
5	1

object	water
1	0
2	0
3	1
4	0
5	1

Fig. 2. Four datasets with binary labels

object	label
1	grass&sky
2	tree&sky
3	grass&sky&water
4	tree
5	sky&water

Fig. 3. Transformed dataset with power set

The second type of approach is to extend specific rough set model in order to handle multi-label data directly. It can be noticed from Fig. 1(a) that in multi-label dataset, different labels often co-occur in practice. Namely, the labels are not independent with each other. Taking Fig. 1(a) as an example, the probability of an image being annotated with label *sky* would be high if we know it has label *grass*. Thus, effective exploitation of correlation information among labels is crucial for the success of multi-label rough sets.

Generally speaking, the co-occurrence of labels is related with the location of instance. Those instances with multiple labels are usually located in the overlapped region. Fig. 4 gives an example to illustrate the relation between location and co-occurrence. Two labels are respectively marked by '*' and '+' in a 2-D space and examples simultaneously belonging to l_1 and l_2 are denoted by 'X'. For convenience, we assume that the distribution of two classes is circular. There are several instances in example space such as a, b, c,d and we associate a neighborhood with five neighbors to each instance. It can be seen that the instances located in the non-overlapped region only have one label while the instances located in the overlapped region may have two labels simultaneously. Let $\delta(x)$ denote the neighborhood of instance x and $|\delta_j(x)|$ is the number of instance with label $l_j (j = 1, \ldots, m)$ in $\delta(x)$. Let $\Gamma(x)$ denote the sum of all kinds of neighbors in $\delta(x)$ and $|\Gamma(x)| = \sum_{q=1}^{m} |\delta_q(x)|$. The proportion that the neighbors with label l_j accounts for of all kinds of neighbors is represented as $\Upsilon_j(x) = |\delta_j(x)|/|\Gamma(x)|$. Taking instances a and c as examples, $\Upsilon_1(a) = 1$ and $\Upsilon_2(a) = 0$ while $\Upsilon_1(b) = 1/6$ and $\Upsilon_2(b) = 5/6$. The proportion $\Upsilon_j(x)$ varies along the changing of location of instances. A larger value for $\Upsilon_j(x)$ will increase the probability of instance x having label l_j. Here, we first introduce the inclusion degree and then give the definition of upper and lower approximations of multi-label decision table according to the proportion $\Upsilon_j(x)$.

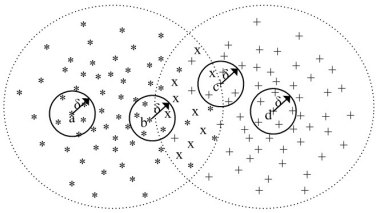


Fig. 4. Illustration of location estimation in multi-label system

Definition 1. *Given two sets A and B in the universe, the inclusion degree of A in B is defined as*

$$I(A,B) = \frac{Card(A \cap B)}{Card(A)} \tag{1}$$

where $Card(\Phi)$ stands for the number of elements in set Φ. The proportion $\Upsilon_j(x)$ can be described using inclusion degree as follows.

$$\Upsilon_j(x) = I(\Gamma(x), Y) = \frac{Card(\Gamma(x) \cap Y)}{Card(\Gamma(x))} \tag{2}$$

where Y represents the set of instances with label l_j in universe. Then the upper and lower approximations of decision class are defined as follows.

Definition 2. *Given a multi-label decision table $MDT = \langle U, A \rangle$, $X_i \in U$ and $A = C \cup D$; Y is the subset of instances with label $l_j (j = 1, \ldots, m)$ and $B \subseteq C$. Then the lower and upper approximations of decision class Y with respect to neighborhood relation R are denoted as $\underline{R_B^\beta}Y$ and $\overline{R_B^\alpha}Y$ respectively, and defined as follows.*

$$\underline{R_B^\beta}Y = \{x_i | I(\Gamma(x), Y) \geqslant \beta, x_i \in U\} \tag{3}$$

$$\overline{R_B^\alpha}Y = \{x_i | I(\Gamma(x), Y) \geqslant \alpha, x_i \in U\} \tag{4}$$

From the definition, we can see that just as decision-theoretic rough set models [14,15], the multi-label rough set model incorporates probabilistic approaches into rough set theory. For each label $l_j \in D$, inclusion degree β and $\alpha (0 \leqslant \alpha < \beta \leqslant 1)$ are different and they are estimated from the training dataset according to maximum posterior probability. Let l_j^1 denote the event of instance x_i having label l_j and l_j^0 denotes the event of instance x_i having no label l_j. $P(l_j^1 | \Upsilon_j(x_i))$ denotes the probability of instance x_i having label l_j, when the proportion is $\Upsilon_j(x_i))$ and $P(l_j^0 | \Upsilon_j(x_i))$ means just the opposite. Then according to Bayesian decision theory, if $P(l_j^1 | \Upsilon_j(x_i)) \geqslant P(l_j^0 | \Upsilon_j(x_i))$ then the instance x_i has label l_j, and otherwise the instance x_i has no relation with label l_j. The threshold β is determined when $P(l_j^1 | \Upsilon_j(x_i)) = P(l_j^0 | \Upsilon_j(x_i))$ and the threshold α is determined when $P(l_j^1 | \Upsilon_j(x_i))$ reaches a satisfied value. Taking Fig. 5 as an example, β is the threshold of lower approximation and α is selected as the threshold of upper approximation when $P(l_j^1 | \Upsilon_j(x_i))$ approaches zero.

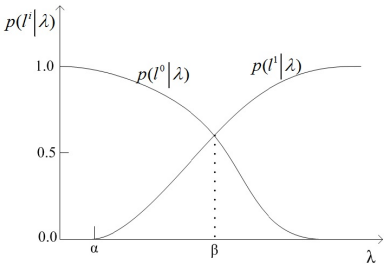


Fig. 5. Illustration of estimation of inclusion degree

For each label l_j, the multi-label rough set model divides the universe into three regions. Decision positive region is denoted by $POS_B(Y) = \underline{R_B^\beta} Y$ where the instances certainly belongs to class l_j. Negative region is denoted by $NEG_B(Y) = U - \overline{R_B^\alpha} Y$ where the instances have no relation with class l_j. The boundary region denoted by $BN_B(Y) = \overline{R_B^\alpha} Y - \underline{R_B^\beta} Y$ is a subset of instances that may have relation with class l_j.

After defining the upper and lower approximations of decision class, we will give the definition of multi-label decision function based on rough sets, which can be used for multi-label classification problem.

Definition 3. *Given a multi-label decision table* $MDT = \langle U, A \rangle$, $x_i \in U$. $\Upsilon_j(x_i)$ $(j = 1, \ldots, m)$ *is the proportion that the neighbors with label* l_j *in* $\delta(x_i)$ *have of all kinds of neighbors in* $\delta(x_i)$. *The multi-label decision function of* x_i *for label* l_j *is defined as* $MD_j(x_i) = l_j^1$, *if* $\Upsilon_j(x_i) \geqslant \beta$ *or* $MD_j(x_i) = l_j^0$, *if* $\Upsilon_j(x_i) \leqslant \alpha$.

$MD_j(x_i)$ is the result assigned to x_i according to the inclusion degree. Obviously, $MD_j(x_i) = l_j^1$ if x_i is located in the positive region of class l_j, or $MD_j(x_i) = l_j^0$ if x_i is located in the negative region of class l_j, or if x_i is located in the boundary region of class l_j, we will assign it a probability of having label l_j.

4 Experiments

To test the effectiveness of the multi-label rough set model(MLRS) presented in this paper, we apply it to two multi-label datasets which come from the the open source Mulan library [1] and Table 1 shows their associated properties. We compare MLRS with various state-of-art multi-label algorithms including the classifier chains algorithm CC, the random k label-set method for multi-label classification RAkEL and back-propagation multi-label learning (BPMLL) learner.

Experimental results of ten-fold cross-validation in terms of *Hamming loss*, *average precision*, *coverage*, *one-error* and *ranking loss* are shown in Table 2 and Table 3. The value following ± gives the standard deviation and the best result on each metric is shown in bold face. The number of the nearest neighbors is set as 10.

It can be seen from Table 2 and Table 3 that MLRS performs well on most evaluation criteria when it applied to the multi-label classification problem. With the enormous increasing of the amount of instances and labels, MLRS still can performs well compared to other multi-label algorithms. It shows that MLRS has some scalability.

Table 1. Multi-label datasets used for experiments

name	instances	attribute	labels	cardinality	density
Scene	2407	294	6	1.074	0.179
Corel5k	5000	499	374	3.522	0.009

Table 2. MLNRS vs. other multi-label algorithms over *Scene*

performance	RAkEL	BPMLL	CC	MLRS
hloss	0.1012±0.0075	0.2667±0.0508	0.1444±0.0164	**0.0912±0.0082**
avgprec	0.8379±0.0156	0.6852±0.0235	0.7176±0.0354	**0.8652±0.0153**
cov	0.5862±0.0593	0.9405±0.0855	1.3504±0.2002	**0.4818±0.0539**
one-error	0.2663±0.0258	0.5450±0.0381	0.3914±0.0453	**0.2255±0.0248**
rloss	0.0999±0.0121	0.1714±0.0165	0.3914±0.0453	**0.0790±0.0116**

Table 3. MLNRS vs. other multi-label algorithms over *Corel5k*

performance	RAkEL	BPMLL	CC	MLRS
hloss	**0.0097±0.0001**	0.5547±0.0213	0.0099±0.0001	0.0105±0.0001
avgprec	0.1075±0.0080	0.0563±0.0097	0.2364±0.0102	**0.2463±0.0092**
cov	336.0374±2.6687	169.0732±4.6338	165.3946±5.8193	**132.1238±5.4093**
one-error	0.7734±0.0201	0.9974±0.0025	**0.7076±0.0172**	0.7398±0.0154
rloss	0.6565±0.0116	0.2273±0.0096	0.1869±0.0083	**0.1513±0.0049**

5 Conclusion

We study the problem of classification under multi-label dataset in this paper. Based on rough set theory, we propose two kinds of approaches to deal with the multi-label problem and present a multi-label rough set model. After applying the model to multi-label datasets, we obtain promising results compared with other well-known multi-label algorithms. Future work will focus on the dimension reduction of multi-label dataset which can improve the accuracy and efficiency of prediction.

References

1. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining (IJDWM) 3, 1–13 (2007)
2. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
3. Kryszkiewicz, M.: Rough set approach to incomplete information systems. Information Sciences 112, 39–49 (1998)
4. Stefanowski, J., Tsoukias, A.: Incomplete information tables and rough classification. Computational Intelligence 17, 545–566 (2001)
5. Wang, G.: Extension of rough set under incomplete information systems. Journal of Computer Research and Development 10, 1–9 (2002)
6. Yao, Y.Y.: Relational interpretations of neighborhood operators and rough set approximation operators. Information Sciences 111, 239–259 (1998)
7. Hu, Q., Yu, D., Xie, Z.: Neighborhood classifiers. Expert Systems with Applications 34, 866–876 (2008)
8. Hllermeier, E., Frnkranz, J., Cheng, W., Brinker, K.: Label ranking by learning pairwise preferences. Artificial Intelligence 172, 1897–1916 (2008)
9. Tsoumakas, G., Vlahavas, I.P.: Random k-labelsets: An ensemble method for multilabel classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 406–417. Springer, Heidelberg (2007)
10. Schapire, R.E., Singer, Y.: BoosTexter: A boosting-based system for text categorization. Machine Learning 39, 135–168 (2000)
11. De Comit, F., Gilleron, R., Tommasi, M.: Learning multi-label alternating decision trees from texts and data. In: Perner, P., Rosenfeld, A. (eds.) MLDM 2003. LNCS, vol. 2734, pp. 251–274. Springer, Heidelberg (2003)
12. Zhang, M.L., Zhou, Z.H.: ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition 40, 2038–2048 (2007)
13. Meng, Z., Shi, Z.: Extended rough set-based attribute reduction in inconsistent incomplete decision systems. Information Sciences 204, 44–69 (2012)
14. Yao, Y.Y., Wong, S.K.M.: A decision theoretic framework for approximating concepts. International Journal of Man-Machine Studies 37(6), 793–809 (1992)
15. Yao, Y.Y., Wong, S.K.M., Lingras, P.: A decision-theoretic rough set model, Methodologies for Intelligent Systems. In: Ras, Z.W., Zemankova, M., Emrichm, M.L. (eds.) Methodologies for Intelligent Systems, vol. 5, pp. 17–25. North-Holland, New York (1990)

Enhancing Rough Clustering with Outlier Detection Based on Evidential Clustering

Manish Joshi[1] and Pawan Lingras[2]

[1] School of Computer Sciences, North Maharashtra University, Jalgaon, India
[2] Department of Mathematics and Computing Science,
Saint Mary's University, Halifax, Canada
joshmanish@gmail.com,
pawan@cs.smu.ca

Abstract. Soft clustering plays an important role in many real world applications. Fuzzy clustering, rough clustering, evidential clustering and many other approaches are used effectively to overcome the rigidness of crisp clustering. Each approach has its own unique features that set it apart from others. In this paper, we propose an enhanced rough clustering approach by combining the strengths of rough clustering and evidential clustering. The rough K-means algorithm is augmented with an ability to determine outliers in datasets using the concepts from the Evidential c-means algorithm. Different experiments are carried on various datasets and it is found that the modified rough K-means can effectively detect outliers with relatively smaller computational complexity.

Keywords: Rough Clustering, Fuzzy Clustering, Rough k-means, Fuzzy c-means, belief functions, Evidential c-means.

1 Introduction

The process of grouping objects into separate clusters is one of the first data mining techniques applied in a knowledge discovery process. Researchers have developed a number of clustering algorithms over the years. The conventional crisp clustering techniques group objects into separate clusters. Each object is assigned to only one cluster. The term crisp clustering refers to the fact that the cluster boundaries are strictly defined and object's cluster membership is unambiguous.

Such a requirement is found to be too restrictive in many data mining applications [1]. In practice, an object may display characteristics of different clusters. In such cases, an object should belong to more than one cluster, and as a result, cluster boundaries necessarily overlap.

A conventional clustering algorithm such as K-means categorizes an object into precisely one cluster. Whereas, fuzzy clustering [2, 3] and rough set clustering [4–6] provide an ability to specify the membership of an object to multiple clusters, which can be useful in real world applications.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 127–137, 2013.

Fuzzy set representation of clusters, using algorithms such as fuzzy c-means (FCM), make it possible for an object to belongs to multiple clusters with a degree of membership between 0 and 1 [3]. Evidential C-means (ECM) proposed by [7] is an extension of FCM and noise clustering algorithm proposed by Dave et al. [8]. It clearly identifies objects that belong to one or more clusters by the virtue of their position in the problem space. Basic belief assignment (bba) values are computed for all possible combinations of k clusters (2^k partitions), which are used to determine cluster membership. Rough K-means algorithm (RKM) [4] groups objects into lower and upper regions of clusters and an object can belongs to an upper region of multiple clusters.

Outlier detection is a very well explored research area with applicability in several real life applications. Outliers affect badly on the overall quality of knowledge obtained from the dataset. Hawkins et al. [9] defined an outlier as a much deviated observation from other observations. Acuna et al. [10] demonstrated the effect of outliers on the mis-classification error rate. Eskin et al. [11] proposed clustering based outlier detection using the distance to closest cluster. Mahoney et al. [12] and He et al. [13] proposed variation in clustering based outlier detection. A detailed survey of various outlier detection techniques is given by Chandola et al. [14] However, all these techniques detect outliers from crisp clustering. Jaruskulchai et al. proposed outlier detection for non-crisp clustering. They integrated possibilistic approach with FCM to propose PXFCM to detect outlier in fuzzy clustering. ECM is also able to detect outliers while generating flexible non-crisp clustering.

Joshi et al. [15] put forth strengths of RKM and ECM clustering algorithms after evaluating both the algorithms on various datasets. ECM was found to be good at outlier detection whereas RKM was good at dealing with high dimensional data. In this paper, we present our proposal to enhance RKM using the concepts used in ECM. We test our results with a synthetic and some standard datsets.

The rest of the paper is organized as follows. The description of ECM and RKM is presented in section 2. The details of the proposed enhancements are given in section 3, followed by experimental results and observations in section 4. Conclusions in section 5 mark the end of the paper.

2 Algorithms: RKM and ECM

The following subsections provide a brief review of the RKM and the ECM algorithms.

2.1 Rough K-means Algorithm

Lingras and West [4] provided RKM algorithm based on an extension of the K-means algorithm [16, 17]. Peters [5] discussed various refinements of Lingras and West's original proposal. These included calculation of rough centroids and the use of ratios of distances as opposed to differences between distances similar

Input:

k: the number of clusters,
$D(n, m)$: a data set containing n objects where each object has m dimensions,
p: a roughness parameter (threshold value = 1.4),
w_lower: relative importance assigned to lower bound (0.75),
w_upper: relative importance assigned to upper bound (0.25),
δ:an input parameter that stands for a small acceptable change in the subsequent centroid values,
$iter$: an input parameter indicating number of consecutive iterations for which difference in subsequent centroid values should be less than δ,

Output:

A set of clusters. Each cluster is represented by the objects in the lower region and in boundary region (upper bound)

Steps:

arbitrarily choose k objects from D as the initial cluster centers (centroids);
repeat
arbitrarily choose k objects from D as the initial cluster centers (centroids);
repeat
(re)assign each object to lower/upper bounds of appropriate clusters by determining its distance from each cluster centroid,
update the cluster means (centroids) using the number of objects assigned and relative importance assigned to lower bound and upper bound of the cluster;
until no change;

Fig. 1. The Rough K-means algorithm

to those used in the rough set based Kohonen algorithm described in [18]. The rough K-means [4] and its various extensions [5] have been found to be effective in distance based clustering. A comparative study of crisp, rough and evolutionary clustering depicts how rough clustering outperforms crisp clustering [19].

In RKM, each cluster $c_i, 1 \leq i \leq k$, is represented using its lower $\underline{A}(c_i)$ and upper $\overline{A}(c_i)$ approximations [20]. All objects that are clustered using the algorithm follow basic properties of rough set theory such as:

(P1) An object $\boldsymbol{x}$ can be part of a lower approximation of at most one cluster
(P2) $\boldsymbol{x} \in \underline{A}(\boldsymbol{c}_i) \Longrightarrow \boldsymbol{x} \in \overline{A}(\boldsymbol{c}_i)$
(P3) An object $\boldsymbol{x}$ is not part of any lower approximation

$$\Updownarrow$$

$\boldsymbol{x}$ belongs to upper approximation of two or more clusters.

Fig. 1 depicts the general idea of how rough K-means algorithm works.

An object is assigned to lower and/or upper approximation of one or more clusters. For each object vector, $\boldsymbol{v}$, let $d(\boldsymbol{v}, \boldsymbol{c}_j)$ be the distance between itself and the centroid of cluster $\boldsymbol{c}_j$. Let $d(\boldsymbol{v}, \boldsymbol{c}_i) = \min_{1 \leq j \leq k} d(\boldsymbol{v}, \boldsymbol{c}_j)$. The ratios $d(\boldsymbol{v}, \boldsymbol{c}_j)/d(\boldsymbol{v}, \boldsymbol{c}_i)$, $1 \leq i, j \leq k$, are used to determine the membership of $\boldsymbol{v}$. Let $T = \{j : d(\boldsymbol{v}, \boldsymbol{c}_j)/d(\boldsymbol{v}, \boldsymbol{c}_i) \leq threshold \text{ and } i \neq j\}$.

1. If $T \neq \emptyset$, $\boldsymbol{v} \in \overline{A}(\boldsymbol{c}_i)$ and $\boldsymbol{v} \in \overline{A}(\boldsymbol{c}_j), \forall j \in T$. Furthermore, $\boldsymbol{v}$ is not part of any lower approximation. The above criterion guarantees that property (P3) is satisfied.
2. Otherwise, if $T = \emptyset$, $\boldsymbol{v} \in \underline{A}(\boldsymbol{c}_i)$. In addition, by property (P2), $\boldsymbol{v} \in \overline{A}(\boldsymbol{c}_i)$.

It should be emphasized that the approximation space A is not defined based on any predefined relation on the set of objects. The lower and upper approximations are constructed based on the criteria described above. The value of a threshold is finalized based on the experiments described in [21].

2.2 Evidential C-means Algorithm

ECM is a credal partition based approach that generates 2^k values for each object to determine cluster membership of an object i by a bba m_i; where k is the number of clusters and bba is a basic belief assignment value. As compared to the earlier versions [22] where only k numbers of clusters were considered for membership assignment, 2^k clusters are proposed in the ECM algorithm to model all situations ranging from complete ignorance to full certainty concerning the cluster of i.

As mentioned earlier, the ECM algorithm first obtains a credal partition followed by a separate treatment of an empty set of the credal partition in order to obtain bba for all 2^k clusters. The similarity between an object and a cluster is measured using the Euclidean metric. In order to obtain the final solution matrix, the problem is represented as an unconstrained optimization problem and solved using an iterative algorithm.

Hence, for a dataset that has three clusters, ECM for each object generates eight different bba values m_i. These eight values correspond to the knowledge regarding the cluster membership of an object i. If a bba value of hundredth object for second cluster is 1, then we can say that cluster of object 100 is known with certainty. Likewise we can claim to have partial knowledge about cluster membership of an object ($0 < bba < 1$); no knowledge about cluster membership of an object (object belongs to all clusters), and importantly outlier characteristics of an object.

ECM builds on Fuzzy c-means to obtain an initial clustering. They further apply the noise-clustering proposed by Dave [8] for fine-tuning the initial solution.

3 RKM Enhancement for Outlier Detection

In this section, we describe our approach to enhance RKM algorithm that enables it to determine outlier objects. We define a term *'degreeOfOutlier'* and describe it in this section.

As discussed earlier, this enhancement is motivated and based on the concepts of Evidential clustering. We first apply RKM to determine cluster memberships for all objects, followed by calculations of *bba* values for each object. The proposed enhancement is as follows.

Let $sim(x_i, x_j)$ be similarity between two vectors. It can be inverse of the distance between these two vectors, i.e. $sim(x_i, x_j) = 1/distance(x_i, x_j)$. Furthermore, let us define similarity between a vector x_j and a set of vectors A as $\sum_{x_i \in A} sim(x_i, x_j)/\|A\|$, where $\|A\|$ is the cardinality of A. In the classical belief functions, $m(\emptyset) = 0$. We can derive the classical belief functions as follows.

Let $C = \{c_1, c_2, c_3, ..., c_k\}$ be a set of clusters, where c_i is the centroid of cluster i. Furthermore, m_i be the bba for an object x_i that belongs to upper bounds of clusters $c \in B$ such that $B \subseteq C$, then

$$m_i(B) = \frac{sim(B, x_i)}{sim(B, x_i) + sim(C - B, x_i)} \tag{1}$$

and the residual m should be assigned to C:

$$m_i(C) = 1 - m_i(B). \tag{2}$$

It should be noted that B and C are the only two focal elements of m_i. This covers the special case where the object x_i belongs to a single cluster c_h, i.e. when it is in the lower bound of c_h:

$$m_i(\{c_h\}) = \frac{sim(c_h, x_i)}{sim(c_h, x_i) + sim(C - \{c_h\}, x_i)} \tag{3}$$

and

$$m_i(C) = 1 - m_i(\{c_h\}). \tag{4}$$

The above formulation does not allow for outliers. If we drop the restriction that $m(\emptyset) = 0$, then we should modify the above bba as follows.

Let m_i be the bba for an object x_i that belongs to upper bounds of clusters $c \in B$ such that $B \subseteq C$, then

$$m_i(B) = \frac{sim(B, x_i)}{sim(B, x_i) + sim(C - B, x_i)} \tag{5}$$

The residual m can either be assigned to C or $\emptyset$ depending on how similar the object is to centers of the clusters in B. Typically, $sim(B, x_i)$ should be less than or equal to half of the maximum distance between cluster centers.

Let us define $farApart$ as the similarity between least similar cluster centroids:

$$farApart = \min_{c_i, c_j \in C \text{ and } c_i \neq c_j} sim(c_i, c_j) \tag{6}$$

If similarity between x_i and cluster centers in B is greater than or equal to half of $farApart$, it is not an outlier. If x_i is less similar to cluster centers in B than half of $farApart$, it could be potentially an outlier. Therefore, we can define:

$$degreeOfOutlier(x_i) = max\left(0, \frac{farApart}{2} - sim(B, x_i)\right) \tag{7}$$

This degree of outlier will be used to assign the residual m_i using the following two equations:

$$\frac{m_i(\emptyset)}{m_i(C)} = \frac{degreeOfOutlier(x_i)}{farApart/2}, \ and \tag{8}$$

$$m_i(C) + m_i(\emptyset) = (1 - m_i(B)) \tag{9}$$

Equations 1 to 4 are straightforward and these formulas are used to calculate *bba* values for all objects and *bba* for Ω as defined in [7]. We have used C instead of Ω. The value of *bba* of an object can be obtained using equation 1 if the object belongs to only one cluster. However, equation 3 can be used to obtain *bba* of an object that belongs to multiple clusters.

Equations 5 onward correspond to the situation where *bba* for an empty set can be non-zero ($m(\emptyset) \neq 0$). It means that, these equations can be used to detect outlier objects (if any) in the dataset. We define a measure 'degreeOfOutlier' that can be 0 or any positive real number. If 'degreeOfOutlier' value is zero it means that object is not an outlier. We can elaborate the formulation of 'degreeOfOutlier' using Figure 2.

The term 'farApart' gives the maximum distance among all centroids (least similar centroids are obtained using equation 6). Half of the 'farApart' distance that is $\frac{farApart}{2}$ is a threshold value that determines whether an object is an outlier. Hereafter, we refer to this threshold as an *'outlier determination threshold'*. If an object lie in the circumference of a circle drawn from cluster centroids with a radius of 'outlier determination threshold' value then that object is not an outlier. But if an object is beyond the circumference of all cluster centroid circles then that object is treated as an outlier. The value of $sim(B, X_{in})$ is greater than the threshold value of $\frac{farApart}{2}$, hence X_{in} is not an outlier (equation 7, 'degreeOfOutlier = 0'). Whereas X_{out} is an outlier because the value of $sim(B, X_{out})$ is smaller than the threshold resulting in positive value of 'degreeOfOutlier' as shown in Figure 2.

4 Experimental Results and Observations

We applied our enhanced RKM algorithm to various data sets to check if it can detect outliers properly. The data sets used for experimental purposes are presented in one subsection followed by results and observations in another subsection.

Synthetic Data Set: We used the synthetic data set developed by Lingras et al. [23] to experiment with the enhanced RKM algorithms. Sixty objects from

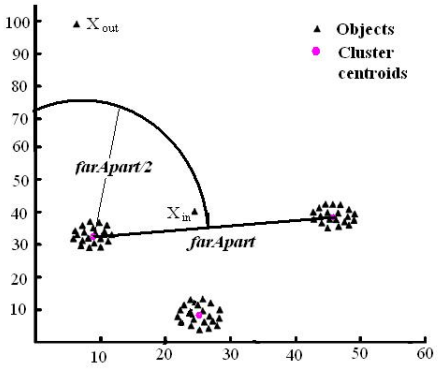


Fig. 2. Synthetic data with outlier

a total of 65 objects are distributed over three distinct clusters. However, five objects do not belong to any particular cluster.

Synthetic Data Set with Outlier: We modified the original synthetic data set to include an outlier as shown in Figure 2. We retained the 60 objects from original synthetic data set that are clustered into three distinct clusters and added two objects X_{in}, X_{out} in the data set. This modified data set have 62 objects where the 61^{st} object is an outlier as shown in Figure 2.

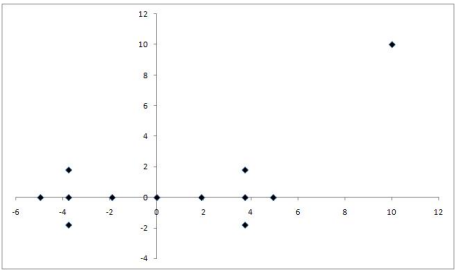


Fig. 3. Diamond data

Diamond Data Set: We used the classical Diamond data set [24] for experimentation. It is composed of twelve objects as represented in Figure 3. Objects 1 to 11 are part of Windhams data whereas object 12 is an outlier.

Glass Identification Data Set: We used another standard data set namely Glass Identification data set [25]. This data set has 214 instances of seven different types of glasses. Nine attributes including refractive index, amount of sodium, magnesium deposited are used for classification. In order to reduce the number of clusters we retained 163 instances corresponding to the first three types of

glasses. We added two outliers to the data set in order to test the performance of our algorithm.

Expected results are obtained after applying the enhanced RKM algorithm on both the Synthetic and modified Synthetic data sets. The enhanced RKM algorithm generated satisfactory results for the Diamond data set and for the Glass data sets too. The results and the observations are discussed in the next subsection.

4.1 Results and Observations

Our objectives of experiments on various data sets were two fold. Firstly, we want to check whether our enhanced RKM algorithm is able to appropriately detect outliers from the data sets. Moreover, we also checked whether our proposed algorithm incorrectly tags any object as an outlier. The results of applying enhanced RKM to the various data sets are as follows.

When we applied enhanced RKM to original synthetic dataset we get 'degreeOfOutlier' as 0 for all 65 objects. This is what we expected as the synthetic data set does not contain any outliers. For modified synthetic data set, X_{out} (7, 98) and X_{in} (25, 40) are the additional objects. Our enhanced RKM algorithm successfully detected the X_{out} as an outlier with a positive 'degreeOfOutlier' for X_{out}, whereas for X_{in} 'degreeOfOutlier' is 0 indicating that it is not an outlier. Table 1 shows how the value of 'degreeOfOutlier' changes for an object placed at different coordinates. The results for the object x_{in} and for the object x_{out} are also included in Table 1.

Table 1. Changing 'degreeOfOutlier' for varying coordinates in Synthetic data set

Coordinates	Cluster membership	degreeOfOutlier
(0, 0)	$\overline{c1}$, $\overline{c2}$	0
(0, 30)	$\underline{c1}$	0
(0, 70)	$\underline{c1}$	0
(0, 100)	Outlier	0.11
(0, 130)	Outlier	0.30
(0, 150)	Outlier forms its own cluster	0.6
(30, 0)	$\underline{c2}$	0
(70, 0)	$\overline{c2}$, $\overline{c3}$	0
(100, 0)	Outlier	0.18
(150, 0)	Outlier forms its own cluster	0.6
X_{in} (25, 40)	$\overline{c1}$, $\overline{c3}$	0
X_{out} (7, 98)	Outlier	0.15

Similarly, enhanced RKM algorithm is able to detect the outlier object from Diamond data set correctly. ECM algorithm detects the outlier from this data set whereas original RKM algorithm was not able to detect the outlier from this data set [15].

Both of these Synthetic and Diamond data sets have low dimensionality. Hence, we decided to test our algorithm on another standard data set with relatively higher dimensions. The standard Glass data set has nine dimensions. Initially, we verified that our algorithm does not falsely point at any object as an outlier. We added two outliers and our algorithm correctly identified both of these newly added instances as outliers.

For all these experiments we calculated, similarity among centroids as well as similarity between objects and cluster centroids. We experimented with two different types of similarity measures namely, Euclidean distance based similarity and Cosine similarity measures to verify which measure suits well. We have obtained Euclidean distance based similarity by taking inverse of conventional Euclidean distance using formula mentioned in equation 10, whereas cosine similarity is obtained as mentioned in equation 11.

$$sim(x_i, x_j) = \frac{1}{\sqrt{\sum_{k=1}^{m}(x_{ik} - x_{jk})^2}} \tag{10}$$

$$sim(x_i, x_j) = \sqrt{\frac{\sum_{k=1}^{m}(x_{ik} \times x_{jk})}{\sum_{k=1}^{m}(x_{ik})^2 \times \sum_{k=1}^{m}(x_{jk})^2}} \tag{11}$$

Where both x_i and x_j are m dimensional vectors.

All the above mentioned experimental results are obtained using Euclidean distance based similarity measure. With cosine similarity measure we could not obtain similar results as shown in Table 1. However, we find cosine similarity suitable when an object is stretched too far to form its own cluster. In this case, the outlier object forms its own cluster with only single object as shown in Table 1 for an object (0, 150). Euclidean distance based similarity results infinity (∞) and ultimately assigns 0 to 'degreeOfOutlier'. Use of Cosine similarity in such cases results in proper identification of outliers. Moreover, a case in which two outliers forming their own cluster is handled better by Cosine similarity as compared to Euclidean based similarity measure.

We performed various experiments and analyzed the results. We further summarize our observations as follows.

1. The algorithm correctly detects outliers, if any, from a data set.
2. The value of 'degreeOfOutlier' is directly proportional to how far an outlier is from 'outlier determination threshold'.
3. The above observation holds true till an object is not too far to form its own cluster with that sole object.
4. How far an object can be stretched without being labeled as an outlier depends upon 'outlier determination threshold'.
5. In the proposal discussed in section 3, we have mentioned 'outlier determination threshold' to be $farApart \times 0.5$. However, we have observed that in some situations we get better results when we used $farApart \times 0.6$ as 'outlier determination threshold'. More experiments shall be performed in future to determine best suitable threshold value for different data sets in different working conditions.

5 Conclusions

Our evaluation of strengths and limitations of RKM as compared to various other non-crisp clustering algorithms like Fuzzy c-means (FCM), Interval set K-means (IKM), Evidential c-means (ECM) suggests that RKM lacks the capability of outlier detection. Hence, we proposed an enhanced RKM algorithm. The enhancement is motivated by the theoretical foundation of belief function theory and its practical implementation in the form of ECM.

We incorporated the concept of basis belief assignment for an object and proposed use of 'outlier determination threshold' to determine whether an object is an outlier. We have evaluated the correctness of the proposed algorithm using a synthetic data set and two standard data sets.

Further studies may reveal if 'outlier determination threshold' is dependent on a given dataset.

References

1. Joshi, A., Krishnapuram, R.: Robust fuzzy clustering methods to support web mining. In: Proc. Workshop in Data Mining and knowledge Discovery, SIGMOD, pp. 15–22 (1998)
2. Bezdek, J.C., Hathaway, R.J.: Optimization of fuzzy clustering criteria using genetic algorithms. In: International Conference on Evolutionary Computation, pp. 589–594 (1994)
3. Pedrycz, W., Waletzky, J.: Fuzzy clustering with partial supervision. IEEE Transactions on Systems, Man, and Cybernetics, Part B 27(5), 787–795 (1997)
4. Lingras, P., West, C.: Interval set clustering of web users with rough k-means. Journal of Intelligent Information Systems 23, 5–16 (2004)
5. Peters, G.: Some refinements of rough k-means clustering. Pattern Recognition 39(8), 1481–1491 (2006)
6. Peters, J.F., Skowron, A., Suraj, Z., Rzasa, W., Borkowski, M.: Clustering: A rough set approach to constructing information granules. In: Soft Computing and Distributed Processing, pp. 57–61 (2002)
7. Masson, M., Denoeux, T.: Ecm: An evidential version of the fuzzy c-means algorithm. Pattern Recognition 41, 1384–1397 (2008)
8. Dave, R.N.: Clustering relational data containing noise and outliers. Pattern Recogn. Lett. 12, 657–664 (1991)
9. Hawkins, D.: Identification of outliers (1980)
10. Saad, M.F., Alimi, A.M.: Modified fuzzy possibilistic c-means. In: Proceedings of the International Multi Conference of Engineers and Computer Scientists (2009)
11. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection detecting intrusions in unlabeled data. Data Mining for Security Applications 19 (2002)
12. Mahoney, M.V., Chan, P.K.: Learning rules for anomaly detection of hostile network traffic. In: Proceedings of the 3rd IEEE International Conference on Data Mining, vol. 601. IEEE Computer Society (2003)
13. He, Z., Xu, X., Deng, S.: Discovering cluster-based local outliers 24, 1641–1650 (2003)

14. Varun, B.A., Vipin, K.: Anomaly detection: A survey. ACM Computing Surveys 41(3), 1641–1650 (2003)
15. Joshi, M., Lingras, P.: Evidential clustering or rough clustering: The choice is yours. In: Li, T., Nguyen, H.S., Wang, G., Grzymala-Busse, J., Janicki, R., Hassanien, A.E., Yu, H. (eds.) RSKT 2012. LNCS, vol. 7414, pp. 123–128. Springer, Heidelberg (2012)
16. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics) 28(1), 100–108 (1979), http://dx.doi.org/10.2307/2346830
17. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Cam, L.M.L., Neyman, J. (eds.) Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press (1967)
18. Lingras, P., Hogo, M., Snorek, M.: Interval set clustering of web users using modified kohonen self-organizing maps based on the properties of rough sets. In: Web Intelli. and Agent Sys., vol. 2 (August 2004)
19. Joshi, M., Lingras, P.: Evolutionary and iterative crisp and rough clustering ii: Experiments. In: Chaudhury, S., Mitra, S., Murthy, C.A., Sastry, P.S., Pal, S.K. (eds.) PReMI 2009. LNCS, vol. 5909, pp. 621–627. Springer, Heidelberg (2009)
20. Lingras, P.: Evolutionary rough k-means clustering. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 68–75. Springer, Heidelberg (2009)
21. Lingras, P., Chen, M., Miao, D.: Precision of rough set clustering. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 369–378. Springer, Heidelberg (2008)
22. Denoeux, T., Masson, M.: Evclus: Evidential clustering of proximity data. IEEE Transactions on Systems Man and Cybernetics 34(1), 95–109 (2004)
23. Lingras, P., Chen, M., Miao, D.: Rough multi-category decision theoretic framework. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 676–683. Springer, Heidelberg (2008)
24. Windham, M.P.: Numerical classification of proximity data with assignment measures. Journal of Classification 2, 157–172 (1985)
25. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml

On Dual Intuitionistic Fuzzy Rough Approximation Operators Determined by an Intuitionistic Fuzzy Implicator

Wei-Zhi Wu, Cang-Jian Gao, Tong-Jun Li, and You-Hong Xu

School of Mathematics, Physics and Information Science,
Zhejiang Ocean University, Zhoushan, Zhejiang, 316004, P.R. China
{wuwz,litj,xyh}@zjou.edu.cn, chinagcjian@126.com

Abstract. In this paper, dual intuitionistic fuzzy rough approximation operators determined by an intuitionistic fuzzy implication operator $\mathcal{I}$ in infinite universes of discourse are investigated. Lower and upper approximations of intuitionistic fuzzy sets with respect to an intuitionistic fuzzy approximation space in infinite universes of discourse are first introduced. Properties of $\mathcal{I}$-intuitionistic fuzzy rough approximation operators are then examined. Relationships between special types of intuitionistic fuzzy relations and properties of $\mathcal{I}$-intuitionistic fuzzy rough approximation operators are further established.

Keywords: Approximation operators, Intuitionistic fuzzy implicators, Intuitionistic fuzzy rough sets, Intuitionistic fuzzy sets, Rough sets.

1 Introduction

One of the main directions in the research of rough set theory [6] is naturally the generalization of concepts of Pawlak rough set approximation operators. Many authors have generalized the notion of rough set approximations into the fuzzy environment, and the results are called rough fuzzy sets (fuzzy sets approximated by a crisp approximation space) and fuzzy rough sets (fuzzy or crisp sets approximated by a fuzzy approximation space). As a more general case of fuzzy sets, the concept of intuitionistic fuzzy (IF for short) sets, which was originated by Atanassov [1], has played a useful role in the research of uncertainty theories. Unlike a fuzzy set, which gives a degree of which element belongs to a set, an IF set gives both a membership degree and a nonmembership degree. Obviously, an IF set is more objective than a fuzzy set to describe the vagueness of data or information. The combination of IF set theory and rough set theory is a new hybrid model to describe the uncertain information and has become an interesting research issue over the years (see e.g. [2, 3, 5, 7–12]).

It is well-known that the dual properties of lower and upper approximation operators are of particular importance in the analysis of mathematical structures in rough set theory. The dual pairs of lower and upper approximation operators in the rough set theory are strongly related to the interior and closure operators

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 138–146, 2013.

in topological space, the necessity (box) and possibility (diamond) operators in modal logic, and the belief and plausibility functions in the Dempster-Shafer theory of evidence. The main objective of this paper is to present the study of IF rough sets determined by an IF implicator $\mathcal{I}$ in infinite universes of discourse. We will define a dual pair of lower and upper $\mathcal{I}$-IF rough approximation operators and examine their essential properties.

2 Preliminaries

In this section we recall some basic notions and previous results which will be used in the later parts of this paper.

2.1 Intuitionistic Fuzzy Logical Operators

Throughout this paper, U will be a nonempty set called the universe of discourse. The class of all subsets (respectively, fuzzy subsets) of U will be denoted by $\mathcal{P}(U)$ (respectively, by $\mathcal{F}(U)$). In what follows, 1_y will denote the fuzzy singleton with value 1 at y and 0 elsewhere; 1_M will denote the characteristic function of a crisp set $M \in \mathcal{P}(U)$. For $\alpha \in [0,1]$ (where $[0,1]$ is the unit interval) , $\widehat{\alpha}$ will denote the constant fuzzy set: $\widehat{\alpha}(x) = \alpha$, for all $x \in U$. For any $A \in \mathcal{F}(U)$, the complement of A will be denoted by $\sim A$, i.e. $(\sim A)(x) = 1 - A(x)$ for all $x \in U$.

We first review a lattice on $[0,1] \times [0,1]$ originated by Cornelis *et al.* [4].

Definition 1. *Denote*

$$L^* = \{(x_1, x_2) \in [0,1] \times [0,1] \mid x_1 + x_2 \leq 1\}. \tag{1}$$

A relation $\leq_{L^*}$ *on* L^* *is defined as follows:* $\forall (x_1, x_2), (y_1, y_2) \in L^*$,

$$(x_1, x_2) \leq_{L^*} (y_1, y_2) \iff x_1 \leq y_1 \text{ and } x_2 \geq y_2. \tag{2}$$

The relation $\leq_{L^*}$ is a partial ordering on L^* and the pair $(L^*, \leq_{L^*})$ is a complete lattice with the smallest element $0_{L^*} = (0,1)$ and the greatest element $1_{L^*} = (1,0)$. The meet operator $\wedge$ and the join operator $\vee$ on $(L^*, \leq_{L^*})$ linked to the ordering $\leq_{L^*}$ are, respectively, defined as follows: $\forall (x_1, x_2), (y_1, y_2) \in L^*$,

$$\begin{aligned} (x_1, x_2) \wedge (y_1, y_2) &= (\min(x_1, y_1), \max(x_2, y_2)), \\ (x_1, x_2) \vee (y_1, y_2) &= (\max(x_1, y_1), \min(x_2, y_2)). \end{aligned} \tag{3}$$

Meanwhile, an order relation $\geq_{L^*}$ on L^* is defined as follows: $\forall x = (x_1, x_2), y = (y_1, y_2) \in L^*$,

$$(y_1, y_2) \geq_{L^*} (x_1, x_2) \iff (x_1, x_2) \leq_{L^*} (y_1, y_2), \tag{4}$$

and

$$x = y \iff x \leq_{L^*} y \text{ and } y \leq_{L^*} x. \tag{5}$$

Definition 2. *An IF negator on L^* is a decreasing mapping $\mathcal{N} : L^* \to L^*$ satisfying $\mathcal{N}(0_{L^*}) = 1_{L^*}$ and $\mathcal{N}(1_{L^*}) = 0_{L^*}$. If $\mathcal{N}(\mathcal{N}(x)) = x$ for all $x \in L^*$, then $\mathcal{N}$ is called an involutive IF negator.*

The mapping $\mathcal{N}_s$, defined as $\mathcal{N}_s(x_1, x_2) = (x_2, x_1), \forall (x_1, x_2) \in L^*$, is called the *standard IF negator*.

Definition 3. *An IF t-norm on L^* is an increasing, commutative, associative mapping $\mathcal{T} : L^* \times L^* \to L^*$ satisfying $\mathcal{T}(1_{L^*}, x) = x$ for all $x \in L^*$.*

Definition 4. *An IF t-conorm on L^* is an increasing, commutative, associative mapping $\mathcal{S} : L^* \times L^* \to L^*$ satisfying $\mathcal{S}(0_{L^*}, x) = x$ for all $x \in L^*$.*

Obviously, the greatest IF t-norm (respectively, the smallest IF t-conorm) with respect to (w.r.t.) the ordering $\leq_{L^*}$ is min (respectively, max), defined by $\min(x, y) = x \wedge y$ (respectively, $\max(x, y) = x \vee y$) for all $x, y \in L^*$.

An IF t-norm $\mathcal{T}$ and an IF t-conorm $\mathcal{S}$ on L^* are said to be *dual* w.r.t. an IF negator $\mathcal{N}$ if

$$\begin{aligned} \mathcal{T}\big(\mathcal{N}(x), \mathcal{N}(y)\big) &= \mathcal{N}\big(\mathcal{S}(x, y)\big), \forall x, y \in L^*, \\ \mathcal{S}\big(\mathcal{N}(x), \mathcal{N}(y)\big) &= \mathcal{N}\big(\mathcal{T}(x, y)\big), \forall x, y \in L^*. \end{aligned} \tag{6}$$

Definition 5. *A mapping $\mathcal{I} : L^* \times L^* \to L^*$ is referred to as an IF implicator on L^* if it is decreasing in its first component (left monotonicity), increasing in its second component (right monotonicity), and satisfies following conditions:*

$$\mathcal{I}(0_{L^*}, 0_{L^*}) = 1_{L^*}, \mathcal{I}(1_{L^*}, 0_{L^*}) = 0_{L^*}, \mathcal{I}(0_{L^*}, 1_{L^*}) = 1_{L^*}, \mathcal{I}(1_{L^*}, 1_{L^*}) = 1_{L^*}. \tag{7}$$

Remark 1. According to the left monotonicity of $\mathcal{I}$, it is easy to verify that $\mathcal{I}\big((\alpha, \beta), 1_{L^*}\big) = 1_{L^*}$ for all $(\alpha, \beta) \in L^*$, similarly, by the right monotonicity of $\mathcal{I}$, one can conclude that $\mathcal{I}\big(0_{L^*}, (\alpha, \beta)\big) = 1_{L^*}$ for all $(\alpha, \beta) \in L^*$.

Definition 6. *Let $\mathcal{S}$ be an IF $t-$conorm and $\mathcal{N}$ an IF negator on L^*. An IF S-implicator generated by the $\mathcal{S}$ and $\mathcal{N}$ is a mapping $\mathcal{I}_{\mathcal{S},\mathcal{N}}$ defined as follows:*

$$\mathcal{I}_{\mathcal{S},\mathcal{N}}(x, y) = \mathcal{S}\big(\mathcal{N}(x), y\big), \quad \forall x, y \in L^*. \tag{8}$$

Definition 7. *Let $\mathcal{T}$ be an IF t-norm on L^*. An IF R-implicator generated by the $\mathcal{T}$ is a mapping $\mathcal{I}_{\mathcal{T}}$ defined as follows:*

$$\mathcal{I}_{\mathcal{T}}(x, y) = \sup\{\gamma \in L^* \mid \mathcal{T}(x, \gamma) \leq_{L^*} y\}, \quad \forall x, y \in L^*. \tag{9}$$

Definition 8. [4] *A mapping $\mathcal{I} : L^* \times L^* \to L^*$ is said to be satisfied, respectively, axiom*

(A1) *if $\mathcal{I}(\cdot, y)$ is decreasing in L^* for all $y \in L^*$ and $\mathcal{I}(x, \cdot)$ is increasing in L^* for all $x \in L^*$ (monotonicity laws);*

(A2) *if $\mathcal{I}(1_{L^*}, x) = x$ for all $x \in L^*$ (neutrality principle);*

(A3) *if $\mathcal{I}(x, y) = \mathcal{I}\big(\mathcal{N}_{\mathcal{I}}(y), \mathcal{N}_{\mathcal{I}}(x)\big)$ for all $x, y \in L^*$ (contrapositivity);*

(A4) *if $\mathcal{I}(x, \mathcal{I}(y, z)) = \mathcal{I}(y, \mathcal{I}(x, z))$ for all $x, y, z \in L^*$ (exchangeability principle);*

(A5) *if $x \leq_{L^*} y \Longleftrightarrow \mathcal{I}(x, y) = 1_{L^*}$ for all $x, y \in L^*$ (confinement principle);*

(A6) *if $\mathcal{I} : L^* \times L^* \to L^*$ is a continuous mapping (continuity).*

Remark 2. In axiom (A3), the mapping $\mathcal{N}_{\mathcal{I}}$, defined by $\mathcal{N}_{\mathcal{I}}(x) = \mathcal{I}(x, 0_{L^*})$, $x \in U$, is an IF negator on L^*, and it is called the negator induced by $\mathcal{I}$. Moreover, it can be easily verified that if axioms (A2) and (A3) hold, then necessarily $\mathcal{N}_{\mathcal{I}}$ is involutive. An IF implicator $\mathcal{I}$ on L^* is called a border IF implicator (resp. EP, CP) if it satisfies axiom (A2) (resp. (A4), (A5)); an IF implicator $\mathcal{I}$ on L^* is called a *model IF implicator* if it satisfies axioms (A2), (A3) and (A4); an IF implicator on L^* is called a *Łukasiewicz IF implicator* if it satisfies axioms (A2)–(A6).

Theorem 1. [4] *An IF S-implicator $\mathcal{I}_{\mathcal{S},\mathcal{N}}$ on L^* defined by Definition 6 is a model IF implicator on the condition that $\mathcal{N}$ is an involutive IF negator; An IF R-implicator $\mathcal{I}_{\mathcal{T}}$ on L^* defined by Definition 7 is a border IF implicator.*

Given an IF negator $\mathcal{N}$ and a border IF implicator $\mathcal{I}$, we define an $\mathcal{N}$-dual operator of $\mathcal{I}$, $\theta_{\mathcal{I},\mathcal{N}} : L^* \times L^* \to L^*$ as follows:

$$\theta_{\mathcal{I},\mathcal{N}}(x, y) = \mathcal{N}(\mathcal{I}(\mathcal{N}(x), \mathcal{N}(y))), \quad x, y \in L^*. \tag{10}$$

According to Eq. (10), we can conclude following

Theorem 2. *For a border IF implicator $\mathcal{I}$ and an IF negator $\mathcal{N}$, we have*

(1) $\theta_{\mathcal{I},\mathcal{N}}(1_{L^*}, 0_{L^*}) = \theta_{\mathcal{I},\mathcal{N}}(1_{L^*}, 1_{L^*}) = \theta_{\mathcal{I},\mathcal{N}}(0_{L^*}, 0_{L^*}) = 0_{L^*}$.

(2) $\theta_{\mathcal{I},\mathcal{N}}(0_{L^*}, 1_{L^*}) = 1_{L^*}$.

(3) *If $\mathcal{N}$ is involutive, then $\theta_{\mathcal{I},\mathcal{N}}(0_{L^*}, x) = x$ for all $x \in L^*$.*

(4) *$\theta_{\mathcal{I},\mathcal{N}}$ is left monotonic (resp. right monotonic) whenever $\mathcal{I}$ is left monotonic (resp. right monotonic).*

(5) *If $\mathcal{I}$ is left monotonic, then $\theta_{\mathcal{I},\mathcal{N}}(x, 0_{L^*}) = 0_{L^*}$ for all $x \in L^*$; and if $\mathcal{I}$ is right monotonic, then $\theta_{\mathcal{I},\mathcal{N}}(1_{L^*}, x) = 0_{L^*}$ for all $x \in L^*$.*

(6) *If $\mathcal{I}$ is an EP IF implicator, then $\theta_{\mathcal{I},\mathcal{N}}$ satisfies the exchange principle, i.e.*

$$\theta_{\mathcal{I},\mathcal{N}}(x, \theta_{\mathcal{I},\mathcal{N}}(y, z)) = \theta_{\mathcal{I},\mathcal{N}}(y, \theta_{\mathcal{I},\mathcal{N}}(x, z)), \quad \forall x, y, z \in L^*. \tag{11}$$

(7) *If $\mathcal{I}$ is a CP IF implicator, then $x \leq y$ iff $\theta_{\mathcal{I},\mathcal{N}}(x, y) = 0_{L^*}$.*

2.2 Intuitionistic Fuzzy Sets

Definition 9. [1] *Let a set U be fixed. An IF set A in U is an object having the form*

$$A = \{\langle x, \mu_A(x), \gamma_A(x)\rangle \mid x \in U\},$$

where $\mu_A : U \to [0, 1]$ and $\gamma_A : U \to [0, 1]$ satisfy $0 \leq \mu_A(x) + \gamma_A(x) \leq 1$ for all $x \in U$, and $\mu_A(x)$ and $\gamma_A(x)$ are, respectively, called the degree of membership and the degree of non-membership of the element $x \in U$ to A. The family of all IF subsets in U is denoted by $\mathcal{IF}(U)$. The complement of an IF set A is defined by $\sim A = \{\langle x, \gamma_A(x), \mu_A(x)\rangle \mid x \in U\}$.

It can be observed that an IF set A is associated with two fuzzy sets μ_A and γ_A. Here, we denote $A(x) = (\mu_A(x), \gamma_A(x))$, then it is clear that $A \in \mathcal{IF}(U)$ iff $A(x) \in L^*$ for all $x \in U$. Obviously, a fuzzy set $A = \{\langle x, \mu_A(x)\rangle \mid x \in U\}$ can be identified with the IF set of the form $\{\langle x, \mu_A(x), 1 - \mu_A(x)\rangle \mid x \in U\}$. Thus an IF set is indeed an extension of a fuzzy set.

Some basic operations on $\mathcal{IF}(U)$ are introduced as follows [1]: for $A, B, A_i \in \mathcal{IF}(U)$, $i \in J$, J is an index set,

- $A \subseteq B$ iff $\mu_A(x) \leq \mu_B(x)$ and $\gamma_A(x) \geq \gamma_B(x)$ for all $x \in U$,
- $A \supseteq B$ iff $B \subseteq A$,
- $A = B$ iff $A \subseteq B$ and $B \subseteq A$,
- $A \cap B = \{\langle x, \min(\mu_A(x), \mu_B(x)), \max(\gamma_A(x), \gamma_B(x))\rangle \mid x \in U\}$,
- $A \cup B = \{\langle x, \max(\mu_A(x), \mu_B(x)), \min(\gamma_A(x), \gamma_B(x))\rangle \mid x \in U\}$,
- $\bigcap\limits_{i\in J} A_i = \{\langle x, \bigwedge\limits_{i\in J} \mu_{A_i}(x), \bigvee\limits_{i\in J} \gamma_{A_i}(x)\rangle \mid x \in U\}$,
- $\bigcup\limits_{i\in J} A_i = \{\langle x, \bigvee\limits_{i\in J} \mu_{A_i}(x), \bigwedge\limits_{i\in J} \gamma_{A_i}(x)\rangle \mid x \in U\}$.

For $(\alpha, \beta) \in L^*$, $\widehat{(\alpha, \beta)}$ will be denoted by the constant IF set: $\widehat{(\alpha, \beta)}(x) = (\alpha, \beta)$, for all $x \in U$. For any $y \in U$ and $M \in \mathcal{P}(U)$, IF sets 1_y, $1_{U-\{y\}}$, and 1_M are, respectively, defined as follows: for $x \in U$,

$$\mu_{1_y}(x) = \begin{cases} 1, & \text{if} \quad x = y, \\ 0, & \text{if} \quad x \neq y. \end{cases} \qquad \gamma_{1_y}(x) = \begin{cases} 0, & \text{if} \quad x = y, \\ 1, & \text{if} \quad x \neq y. \end{cases}$$

$$\mu_{1_{U-\{y\}}}(x) = \begin{cases} 0, & \text{if} \quad x = y, \\ 1, & \text{if} \quad x \neq y. \end{cases} \qquad \gamma_{1_{U-\{y\}}}(x) = \begin{cases} 1, & \text{if} \quad x = y, \\ 0, & \text{if} \quad x \neq y. \end{cases}$$

$$\mu_{1_M}(x) = \begin{cases} 1, & \text{if} \quad x \in M, \\ 0, & \text{if} \quad x \notin M. \end{cases} \qquad \gamma_{1_M}(x) = \begin{cases} 0, & \text{if} \quad x \in M, \\ 1, & \text{if} \quad x \notin M. \end{cases}$$

The IF universe set is $U = 1_U = \widehat{(1, 0)} = \widehat{1_{L^*}} = \{\langle x, 1, 0\rangle \mid x \in U\}$ and the IF empty set is $\emptyset = \widehat{(0, 1)} = \widehat{0_{L^*}} = \{\langle x, 0, 1\rangle \mid x \in U\}$.

By using L^*, IF sets on U can be represented as follows: for $A, B, A_j \in \mathcal{IF}(U) (j \in J, J$ is an index set), $x, y \in U$, and $M \in \mathcal{P}(U)$

- $A(x) = (\mu_A(x), \gamma_A(x)) \in L^*$,
- $U(x) = (1, 0) = 1_{L^*}$,
- $\emptyset(x) = (0, 1) = 0_{L^*}$,
- $x = y \Longrightarrow 1_y(x) = 1_{L^*}$ and $1_{U-\{y\}}(x) = 0_{L^*}$,
- $x \neq y \Longrightarrow 1_y(x) = 0_{L^*}$ and $1_{U-\{y\}}(x) = 1_{L^*}$,
- $x \in M \Longrightarrow 1_M(x) = 1_{L^*}$,
- $x \notin M \Longrightarrow 1_M(x) = 0_{L^*}, M \in \mathcal{P}(U)$,
- $A \subseteq B \Longleftrightarrow A(x) \leq_{L^*} B(x), \forall x \in U \iff B(x) \geq_{L^*} A(x), \forall x \in U$,
- $\big(\bigcap\limits_{j\in J} A_j\big)(x) = \bigwedge\limits_{j\in J} A_j(x) = \big(\bigwedge\limits_{j\in J} \mu_{A_j}(x), \bigvee\limits_{j\in J} \gamma_{A_j}(x)\big) \in L^*$,
- $\big(\bigcup\limits_{j\in J} A_j\big)(x) = \bigvee\limits_{j\in J} A_j(x) = \big(\bigvee\limits_{j\in J} \mu_{A_j}(x), \bigwedge\limits_{j\in J} \gamma_{A_j}(x)\big) \in L^*$.

Given an IF implicator $\mathcal{I}$, an involutive IF negator $\mathcal{N}$, and two IF sets A, $B \in \mathcal{IF}(U)$, we define two IF sets $A \Rightarrow_{\mathcal{I}} B$ and $\theta_{\mathcal{I},\mathcal{N}}(A,B)$ as follows:

$$\begin{aligned}(A \Rightarrow_{\mathcal{I}} B)(x) &= \mathcal{I}(A(x), B(x)),\ x \in U,\\ \theta_{\mathcal{I},\mathcal{N}}(A,B)(x) &= \theta_{\mathcal{I},\mathcal{N}}(A(x), B(x)),\ x \in U.\end{aligned} \tag{12}$$

It can easily be verified that

$$\theta_{\mathcal{I},\mathcal{N}}(A,B) = \sim_{\mathcal{N}} ((\sim_{\mathcal{N}} A) \Rightarrow_{\mathcal{I}} (\sim_{\mathcal{N}} B)). \tag{13}$$

3 $\mathcal{I}$-Intuitionistic Fuzzy Rough Approximation Operators

In this section, by employing an IF implicator $\mathcal{I}$ on L^*, we will define the lower and upper approximations of IF sets w.r.t. an arbitrary IF approximation space and discuss properties of $\mathcal{I}$-IF rough approximation operators.

Definition 10. *Let U and W be two nonempty universes of discourse. A subset $R \in \mathcal{IF}(U \times W)$ is referred to as an IF binary relation from U to W, namely, R is given by*

$$R = \{\langle (x,y), \mu_R(x,y), \gamma_R(x,y)\rangle \mid (x,y) \in U \times W\}, \tag{14}$$

where $\mu_R : U \times W \to [0,1]$ and $\gamma_R : U \times W \to [0,1]$ satisfy $0 \le \mu_R(x,y) + \gamma_R(x,y) \le 1$ for all $(x,y) \in U \times W$. We denote the family of all IF relations from U to W by $\mathcal{IFR}(U \times W)$. An IF relation $R \in \mathcal{IFR}(U \times W)$ is said to be serial if $\bigvee_{y \in W} R(x,y) = 1_{L^*}$ for all $x \in U$. If $U = W$, $R \in \mathcal{IFR}(U \times U)$ is called an IF binary relation on U. $R \in \mathcal{IFR}(U \times U)$ is said to be reflexive if $R(x,x) = 1_{L^*}$ for all $x \in U$. R is said to be symmetric if $R(x,y) = R(y,x)$ for all $x, y \in U$. R is said to be $\mathcal{T}$-transitive if $\bigvee_{y \in U} \mathcal{T}(R(x,y), R(y,z)) \le_{L^*} R(x,z)$ for all $x, z \in U$, where $\mathcal{T}$ is an IF t-norm.

Definition 11. *Let U and W be two non-empty universes of discourse and R an IF relation from U to W. The triple (U,W,R) is called a generalized IF approximation space.*

Definition 12. *Assume that $\mathcal{I}$ is an IF implicator and $\mathcal{N}$ an IF negator on L^*. Let (U,W,R) be a generalized IF approximation space and $A \in \mathcal{IF}(W)$, the $\mathcal{I}$-lower and $\mathcal{I}$-upper approximations of A, denoted as $\underline{R}_{\mathcal{I}}(A)$ and $\overline{R}_{\mathcal{I}}(A)$, respectively, w.r.t. the approximation space (U,W,R) are IF sets of U and are, respectively, defined as follows:*

$$\begin{aligned}\underline{R}_{\mathcal{I}}(A)(x) &= \bigwedge_{y \in U} \mathcal{I}\big(R(x,y), A(y)\big), x \in U,\\ \overline{R}_{\mathcal{I}}(A)(x) &= \bigvee_{y \in W} \theta_{\mathcal{I},\mathcal{N}}\big(\mathcal{N}(R(x,y)), A(y)\big), x \in U.\end{aligned} \tag{15}$$

The operators $\underline{R}_{\mathcal{I}}$, $\overline{R}_{\mathcal{I}} : \mathcal{IF}(W) \to \mathcal{IF}(U)$ are, respectively, referred to as $\mathcal{I}$-lower and $\mathcal{I}$-upper IF rough approximation operators of (U,W,R), and the pair $(\underline{R}_{\mathcal{I}}(A), \overline{R}_{\mathcal{I}}(A))$ is called the $\mathcal{I}$-IF rough set of A w.r.t. (U,W,R).

The following theorem shows that the lower and upper $\mathcal{I}$-IF rough approximation operators determined by an IF implicator $\mathcal{I}$ and an involutive IF negator $\mathcal{N}$ are dual with each other.

Theorem 3. *Assume that $\mathcal{I}$ is an IF implicator and $\mathcal{N}$ an involutive IF negator on L^*. Let (U, W, R) be a generalized IF approximation space, then*

$$\begin{array}{ll} \text{(DIFL)} & \underline{R}_{\mathcal{I}}(A) = \sim_{\mathcal{N}} \overline{R}_{\mathcal{I}}(\sim_{\mathcal{N}} A), \forall A \in \mathcal{IF}(W), \\ \text{(DIFU)} & \overline{R}_{\mathcal{I}}(A) = \sim_{\mathcal{N}} \underline{R}_{\mathcal{I}}(\sim_{\mathcal{N}} A), \forall A \in \mathcal{IF}(W). \end{array} \tag{16}$$

The next theorem gives some basic properties of $\mathcal{I}$-IF rough approximation operators.

Theorem 4. *Let (U, W, R) be an IF approximation space. Assume that $\mathcal{I}$ is a continuous, hybrid monotonic and border IF implicator on L^* and $\mathcal{N}$ an involutive IF negator on L^*, then the lower and upper $\mathcal{I}$-IF rough approximation operators have the following properties: For all $A, B \in \mathcal{IF}(W)$, $A_j \in \mathcal{IF}(W)(\forall j \in J, J$ is an index set), $M \subseteq W$, $(x, y) \in U \times W$ and all $(\alpha, \beta) \in L^*$,*

(IFL1) $\underline{R}_{\mathcal{I}}(\widehat{(\alpha, \beta)} \Rightarrow_{\mathcal{I}} A) = \widehat{(\alpha, \beta)} \Rightarrow_{\mathcal{I}} \underline{R}_{\mathcal{I}}(A)$, *provided that $\mathcal{I}$ is an EP IF implicator.*

(IFU1) $\overline{R}_{\mathcal{I}}(\theta_{\mathcal{I},\mathcal{N}}(\widehat{(\alpha, \beta)}, A)) = \theta_{\mathcal{I},\mathcal{N}}(\widehat{(\alpha, \beta)}, \overline{R}_{\mathcal{I}}(A))$, *provided that $\mathcal{I}$ is an EP IF implicator.*

(IFL2) $\underline{R}_{\mathcal{I}}(\bigcap\limits_{j \in J} A_j) = \bigcap\limits_{j \in J} \underline{R}_{\mathcal{I}}(A_j)$.

(IFU2) $\overline{R}_{\mathcal{I}}(\bigcup\limits_{j \in J} A_j) = \bigcup\limits_{j \in J} \overline{R}_{\mathcal{I}}(A_j)$.

(IFL3) $\underline{R}_{\mathcal{I}}(\widehat{(\alpha, \beta)}) \supseteq \widehat{(\alpha, \beta)}$.

(IFU3) $\overline{R}_{\mathcal{I}}(\widehat{(\alpha, \beta)}) \subseteq \widehat{(\alpha, \beta)}$.

(IFL4) $\underline{R}_{\mathcal{I}}(W) = U$.

(IFU4) $\overline{R}_{\mathcal{I}}(\emptyset_W) = \emptyset_U$, *where $\emptyset_W$ and $\emptyset_U$ are the empty sets in W and U respectively.*

(IFL5) $\underline{R}_{\mathcal{I}}(\widehat{(\alpha, \beta)} \Rightarrow_{\mathcal{I}} \emptyset_W) = \widehat{(\alpha, \beta)} \Rightarrow_{\mathcal{I}} \emptyset_U \Longleftrightarrow \underline{R}_{\mathcal{I}}(\emptyset_W) = \emptyset_U$, *provided that $\mathcal{I}$ is an EP IF implicator.*

(IFU5) $\overline{R}_{\mathcal{I}}(\theta_{\mathcal{I},\mathcal{N}}(\widehat{(\alpha, \beta)}, W)) = \theta_{\mathcal{I},\mathcal{N}}(\widehat{(\alpha, \beta)}, U) \Longleftrightarrow \overline{R}_{\mathcal{I}}(W) = U$, *provided that $\mathcal{I}$ is an EP IF implicator.*

(IFL6) $A \subseteq B \Longrightarrow \underline{R}_{\mathcal{I}}(A) \subseteq \underline{R}_{\mathcal{I}}(B)$.

(IFU6) $A \subseteq B \Longrightarrow \overline{R}_{\mathcal{I}}(A) \subseteq \overline{R}_{\mathcal{I}}(B)$.

(IFL7) $\underline{R}_{\mathcal{I}}(\bigcup\limits_{j \in J} A_j) \supseteq \bigcup\limits_{j \in J} \underline{R}_{\mathcal{I}}(A_j)$.

(IFU7) $\overline{R}_{\mathcal{I}}(\bigcap\limits_{j \in J} A_j) \subseteq \bigcap\limits_{j \in J} \overline{R}_{\mathcal{I}}(A_j)$.

(IFL8) $\underline{R_{\mathcal{I}}}(1_y \Rightarrow_{\mathcal{I}} (\widehat{\alpha,\beta}))(x) = \mathcal{I}(R(x,y),(\alpha,\beta))$.

(IFU8) $\overline{R}_{\mathcal{I}}(\theta_{\mathcal{I},\mathcal{N}}(1_{W-\{y\}},(\widehat{\alpha,\beta})))(x) = \theta_{\mathcal{I},\mathcal{N}}(\mathcal{N}(R(x,y)),(\alpha,\beta))$.

(IFL9) $\underline{R_{\mathcal{I}}}(1_{W-\{y\}})(x) = \mathcal{I}(R(x,y),0_{L^*})$.

(IFU9) $\overline{R}_{\mathcal{I}}(1_y)(x) = \theta_{\mathcal{I},\mathcal{N}}(\mathcal{N}(R(x,y)),1_{L^*})$.

(IFL10) $\underline{R_{\mathcal{I}}}(1_M)(x) = \bigwedge\limits_{y\notin M} \mathcal{I}(R(x,y),0_{L^*})$.

(IFU10) $\overline{R}_{\mathcal{I}}(1_M)(x) = \bigvee\limits_{y\in M} \theta_{\mathcal{I},\mathcal{N}}(\mathcal{N}(R(x,y)),1_{L^*})$.

Theorems 5-8 below show the relationships between some special IF relations and properties of $\mathcal{I}$-IF rough approximation operators.

Theorem 5. *Let (U,W,R) be an IF approximation space, $\mathcal{I}$ a continuous border and CP IF implicator, and $\mathcal{N}$ an involutive IF negator. Then*

$$\begin{aligned} R \text{ is serial} &\iff \text{(IFL0) } \underline{R_{\mathcal{I}}}((\widehat{\alpha,\beta})) = (\widehat{\alpha,\beta}),\ \forall(\alpha,\beta)\in L^*. \\ &\iff \text{(IFU0) } \overline{R}_{\mathcal{I}}((\alpha,\beta)) = (\widehat{\alpha,\beta}),\ \forall(\alpha,\beta)\in L^*. \end{aligned}$$

Theorem 6. *Let (U,R) be an IF approximation space (i.e. R is an IF relation on U), $\mathcal{I}$ a border and CP IF implicator, and $\mathcal{N}$ an involutive IF negator. Then*

$$\begin{aligned} R \text{ is reflexive} &\iff \text{(IFLR) } \underline{R_{\mathcal{I}}}(A)\subseteq A, \quad \forall A\in\mathcal{IF}(U), \\ &\iff \text{(IFUR) } A\subseteq \overline{R}_{\mathcal{I}}(A), \quad \forall A\in\mathcal{IF}(U). \end{aligned}$$

Theorem 7. *Let (U,R) be an IF approximation space, $\mathcal{I}$ a border and CP IF implicator, and $\mathcal{N}$ an involutive IF negator. Then*

$$\begin{aligned} &R \text{ is symmetric} \\ &\iff \text{(IFLS)}\underline{R_{\mathcal{I}}}(1_x \Rightarrow_{\mathcal{I}} (\widehat{\alpha,\beta}))(y) = \underline{R_{\mathcal{I}}}(1_y \Rightarrow_{\mathcal{I}} (\widehat{\alpha,\beta}))(x), \forall x,y\in U, \forall(\alpha,\beta)\in L^*, \\ &\iff \text{(IFUS)}\overline{R}_{\mathcal{I}}(\theta_{\mathcal{I},\mathcal{N}}(1_{U-\{y\}},(\widehat{\alpha,\beta})))(x) = \overline{R}_{\mathcal{I}}(\theta_{\mathcal{I},\mathcal{N}}(1_{U-\{x\}},(\widehat{\alpha,\beta})))(y), \\ &\qquad \forall x,y\in U, \forall(\alpha,\beta)\in L^*. \end{aligned}$$

Theorem 8. *Let (U,R) be an IF approximation space, $\mathcal{N}$ an involutive IF negator, and $\mathcal{I}$ an IF implicator and $\mathcal{T}$ an IF t-norm satisfying*

$$\mathcal{I}(a,\mathcal{I}(b,c)) = \mathcal{I}(\mathcal{T}(a,b),c), \quad \forall a,b,c\in L^*. \tag{17}$$

(1) *If R is IF $\mathcal{T}$-transitive, then*

$$\begin{aligned} &\text{(IFLT) } \underline{R_{\mathcal{I}}}(A)\subseteq \underline{R_{\mathcal{I}}}(\underline{R_{\mathcal{I}}}(A)), \quad \forall A\in\mathcal{IF}(U). \\ &\text{(IFUT) } \overline{R}_{\mathcal{I}}(\overline{R}_{\mathcal{I}}(A))\subseteq \overline{R}_{\mathcal{I}}(A), \quad \forall A\in\mathcal{IF}(U). \end{aligned}$$

(2) *If $\mathcal{I}$ is a CP and border IF implicator, then*

$$\text{(IFLT)} \iff \text{(IFUT)} \implies R \text{ is } \mathcal{T}\text{-transitive.} \tag{18}$$

4 Conclusion

We have investigated a general type of relation-based $\mathcal{I}$-intuitionistic fuzzy rough sets determined by an IF implicator $\mathcal{I}$. We have defined $\mathcal{I}$-lower and $\mathcal{I}$-upper approximations of IF sets with respect to a generalized IF approximation space. We have examined properties of $\mathcal{I}$-lower and $\mathcal{I}$-upper IF rough approximation operators and established relationships between some special types of IF binary relations and properties of $\mathcal{I}$-IF rough approximation operators. For further study, we will investigate other mathematical structures of the $\mathcal{I}$-IF approximation operators.

Acknowledgement. This work was supported by grants from the National Natural Science Foundation of China (Nos. 61272021, 61075120, 11071284, and 61173181), the Zhejiang Provincial Natural Science Foundation of China (No. LZ12F03002).

References

1. Atanassov, K.: Intuitionistic Fuzzy Sets: Theory and Applications. Physica-Verlag, Heidelberg (1999)
2. Chakrabarty, K., Gedeon, T., Koczy, L.: Intuitionistic fuzzy rough set. In: Proceedings of 4th Joint Conference on Information Sciences (JCIS), pp. 211–214. Durham, NC (1998)
3. Cornelis, C., Cock, M.D., Kerre, E.E.: Intuitionistic fuzzy rough sets: at the crossroads of imperfect knowledge. Expert Systems 20, 260–270 (2003)
4. Cornelis, C., Deschrijver, G., Kerre, E.E.: Implication in intuitionistic fuzzy and interval-valued fuzzy set theory: construction, classification, application. International Journal of Approximate Reasoning 35, 55–95 (2004)
5. Jena, S.P., Ghosh, S.K.: Intuitionistic fuzzy rough sets. Notes on Intuitionistic Fuzzy Sets 8, 1–18 (2002)
6. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Boston (1991)
7. Radzikowska, A.M.: Rough approximation operations based on IF sets. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 528–537. Springer, Heidelberg (2006)
8. Rizvi, S., Naqvi, H.J., Nadeem, D.: Rough intuitionistic fuzzy set. In: Proceedings of the 6th Joint Conference on Information Sciences (JCIS), pp. 101–104. Durham, NC (2002)
9. Samanta, S.K., Mondal, T.K.: Intuitionistic fuzzy rough sets and rough intuitionistic fuzzy sets. Journal of Fuzzy Mathematics 9, 561–582 (2001)
10. Zhou, L., Wu, W.-Z.: On generalized intuitionistic fuzzy approximation operators. Information Sciences 178, 2448–2465 (2008)
11. Zhou, L., Wu, W.-Z.: Characterization of rough set approximations in Atanassov intuitionistic fuzzy set theory. Computers and Mathematics with Applications 62, 282–296 (2011)
12. Zhou, L., Wu, W.-Z., Zhang, W.-X.: On characterization of intuitionistic fuzzy rough sets based on intuitionistic fuzzy implicators. Information Sciences 179, 883–898 (2009)

Discernibility Matrix Based Attribute Reduction in Intuitionistic Fuzzy Decision Systems

Qinrong Feng and Rui Li

School of Mathematics and Computer Science,
Shanxi Normal University, 041004, Linfen, Shanxi, P.R. China

Abstract. Based on the theory of rough sets and intuitionistic fuzzy sets, this paper researches attribute reduction in intuitionistic fuzzy decision systems (IFDS). Firstly, we establish an intuitionistic fuzzy rough set model based on the similarity relation. Secondly, the discernibility matrix based on the maximal consistent block is constructed and an algorithm of attribute reduction is designed, which can eliminate the redundant information from the given IFDS. Finally, an illustrative example is employed to show the validity of the algorithm in this paper.

Keywords: intuitionistic fuzzy decision systems, $\alpha-$similarity relation, discernibility matrix, attribute reduction.

1 Introduction

Rough set theory, proposed by Pawlak [1] in the early 1980s, is a mathematical tool to deal with uncertain, imprecise or incomplete knowledge in information systems. In 1986, Atanassov gave the definiton of intuitionistic fuzzy sets. As an intuitively straightforward extension of Zadeh's fuzzy set theory [2], intuitionistic fuzzy sets is defined by a pair of membership function: a membership degree and a non-membership degree, which depicted the essence of the fuzziness [3,4].

In recent years, many research results have been obtained with intuitionistic fuzzy rough sets [5-12]. For example, Zhou and Wu [6,7] described the rough approximations of intuitionistic fuzzy sets. The algorithms of attribute reduction based on intuitionistic fuzzy rough sets were proposed in [13-17]. Huang [13,14] constructed a dominance-based rough set model in intuitionistic fuzzy information systems, defined the lower and upper approximation discernibility matrices to find out all the lower and upper approximation reducts, and applied it in computing audit risk assessment. Zhang and Tian [15] gave a systematic study on attribute reduction with intuitionistic fuzzy rough sets. Wang and Shu [16] proposed an attribute reduction algorithm based on dependence degree and nondependence degree with intuitionistic fuzzy similarity relations. Chen [17] proposed a new method of attribute reduction integrating the information entropy with intuitionistic fuzzy equivalence relations.

In intuitionistic fuzzy decision systems (IFDS), the key step of constructing intuitionistic fuzzy rough set model is the classification of the universe. In the classification of the universe induced by dominance relations or intuitionistic

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 147–156, 2013.

fuzzy similarity relations, some articles did not take into account the degree of hesitancy.

This paper copes with IFDS based on the similarity relation. The degree of hesitancy will be introduced to the similarity relation. Investigations about the similarity measures of intuitionistic fuzzy sets have been carried out in [18-21]. A new weighted Euclidean distance and a parameter α will be used to define the similarity relation in IFDS. An algorithm of attribute reduction based on the discernibility matrix will be designed in IFDS.

The rest of this paper is organized as follows. In section 2, the basic preliminaries are briefly reviewed. In section 3, the intuitionistic fuzzy rough set model based on similarity relation is proposed. In section 4, the discernibility matrix is constructed and an algorithm of attribute reduction is designed in IFDS. In section 5, conclusions summarize the paper.

2 Preliminaries

In this section, we will review some basic concepts.

Definition 1. [13] *Let $\langle \mu, \gamma \rangle$ be an order pair, where $0 \leq \mu, \gamma \leq 1$ and $0 \leq \mu + \gamma \leq 1$. Then we call $\langle \mu, \gamma \rangle$ an intuitionistic fuzzy value.*

Definition 2. [13] *Let U be the universe of discourse. An intuitionistic fuzzy set A in U is an object having the form $A = \{\langle x, \mu_A(x), \gamma_A(x) \rangle \,|x \in U\}$, where $\mu_A : U \to [0,1]$ and $\gamma_A : U \to [0,1]$ satisfy $0 \leq \mu_A(x) + \gamma_A(x) \leq 1$ for all $x \in U$ and $\mu_A(x)$ and $\gamma_A(x)$ are, respectively, called the degree of membership and the degree of non-membership of the element $x \in U$ to A.*

$\pi_A(x) = 1 - \mu_A(x) - \gamma_A(x)$ denotes the degree of hesitancy of x to A or the degree of uncertainty of x to A. Evidently, $0 \leq \pi_A(x) \leq 1$ for all x.

It is obvious that any fuzzy set $A = \{\langle x, \mu_A(x) \rangle \,|x \in U\}$ can be identified with the intuitionistic fuzzy set in the form $\{\langle x, \mu_A(x), 1 - \mu_A(x) \rangle \,|x \in U\}$. Thus an intuitionstic fuzzy set is an extension of a fuzzy set.

Definition 3. *An intuitionistic fuzzy information system (IFIS) is a quadruple (U, A, V, f), where U is a non-empty and finite set of objects called the universe, A is a non-empty and finite set of attributes. V is the set of all intuitionistic fuzzy values. The information function f is a map from $U \times A$ to V, such that $f(x, a) = \langle \mu_a(x), \gamma_a(x) \rangle \in V$ for all $a \in A$.*

When $A = C \cup D$, and $C \cap D = \emptyset$, then $(U, C \cup D, V, f)$ is called an intuitionistic fuzzy decision system (IFDS).

3 Similarity Relation in IFDS

In this section, we construct the intuitionistic fuzzy rough set model based on similarity relation in IFDS.

3.1 Similarity Degree

In IFDS, the partition of U which are induced by an equivalence relation is too much, which is disadvantage to extract knowledge. Thus, we introduce a similarity degree to express the degree of similarity of two intuitionistic fuzzy values.

Firstly, we introduce a distance between two intuitionistic fuzzy values.

Xu and Yager [18] defined a distance between two intuitionistic fuzzy values as follows

Let $\delta_1 = \langle \mu_1, \gamma_1 \rangle$ and $\delta_2 = \langle \mu_2, \gamma_2 \rangle$ are two intuitionistic fuzzy values, then

$$d_H(\delta_1, \delta_2) = \frac{1}{2}\left(|\mu_1 - \mu_2| + |\gamma_1 - \gamma_2| + |\pi_1 - \pi_2|\right)$$

is called the normalized Hamming distance between δ_1 and δ_2.

But the following example shows that the normalized Hamming distance is not reasonable for some intuitionistic fuzzy values.

Example 1. Let $\delta_1 = \langle 0, 0 \rangle, \delta_2 = \langle 0.5, 0.5 \rangle, \delta_3 = \langle 0.2, 0.8 \rangle, \delta_4 = \langle 0.9, 0.1 \rangle$, we have

$$d_H(\delta_1, \delta_2) = d_H(\delta_1, \delta_3) = d_H(\delta_1, \delta_4) = 1$$

Obviously, this is unreasonable.

In order to solve this problem, we use the following the Euclidean distance between the intuitionistic fuzzy values.

$$d_E(\delta_1, \delta_2) = \sqrt{|\mu_1 - \mu_2|^2 + |\gamma_1 - \gamma_2|^2 + |\pi_1 - \pi_2|^2}$$

The following example shows that the Euclidean distance is also unreasonable for some intuitionistic fuzzy values.

Example 2. Let $\delta_1 = \langle 0.4, 0.5 \rangle, \delta_2 = \langle 0, 0.9 \rangle, \delta_3 = \langle 0.4, 0.1 \rangle$, then we have $d_E(\delta_1, \delta_2) = d_E(\delta_1, \delta_3) = 0.57$, $d_E(\delta_2, \delta_3) = 0.98$, so

$$d_E(\delta_1, \delta_2) = d_E(\delta_1, \delta_3) < d_E(\delta_2, \delta_3)$$

However, in many applications, we may have $d(\delta_1, \delta_3) < d(\delta_1, \delta_2)$. Because the degree of membership is more important than the degree of hesitancy in reality.

Thus, we give the following weighted Euclidean distance between the intuitionistic fuzzy values.

Definition 4. *Let $\delta_1 = \langle \mu_1, \gamma_1 \rangle$ and $\delta_2 = \langle \mu_2, \gamma_2 \rangle$ are two intuitionistic fuzzy values, then*

$$d(\delta_1, \delta_2) = \sqrt{a|\mu_1 - \mu_2|^2 + b|\gamma_1 - \gamma_2|^2 + c|\pi_1 - \pi_2|^2}$$

where a, b, c are weighted factors.

Proposition 1. *Let* $\delta_1 = \langle \mu_1, \gamma_1 \rangle$ *and* $\delta_2 = \langle \mu_2, \gamma_2 \rangle$ *are two intuitionistic fuzzy values, then* $d(\delta_1, \delta_2)$ *is a metric. That is, for any intuitionistic fuzzy values* δ_1, δ_2, δ_3, *we have*

(1) $d(\delta_1, \delta_2) \geq 0$, *and* $d(\delta_1, \delta_2) = 0$ *if and only if* $\delta_1 = \delta_2$.

(2) $d(\delta_1, \delta_2) = d(\delta_2, \delta_1)$

(3) $d(\delta_1, \delta_2) \leq d(\delta_1, \delta_3) + d(\delta_3, \delta_2)$.

Example 3. We recalculate the distance in example 2 using the new distance. By taking $a = b = 0.4$, $c = 0.2$. We have $d(\delta_1, \delta_2) = 0.36$, $d(\delta_1, \delta_3) = 0.31$, $d(\delta_2, \delta_3) = 0.59$, so

$$d(\delta_1, \delta_3) < d(\delta_1, \delta_2) < d(\delta_2, \delta_3).$$

It is obvious that the new distance is different from the previous one.

Next, we will use the new distance to define a similarity degree between two objects in IFDS.

Definition 5. *Let* $IFDS = (U, C \cup D, V, f)$, *for any* $x_i, x_j \in U$, $c_k \in C$, *the two intuitionistic fuzzy values* $f(x_i, c_k) = \langle \mu_{c_k}(x_i), \gamma_{c_k}(x_i) \rangle$ *and* $f(x_j, c_k) = \langle \mu_{c_k}(x_j), \gamma_{c_k}(x_j) \rangle$, *the similarity degree based on the weighted Euclidean distance is defined as follows*

$$sim_{c_k}(x_i, x_j) = 1 - d(x_i, x_j)$$
$$= 1 - \sqrt{a|\mu_{c_k}(x_i) - \mu_{c_k}(x_j)|^2 + b|\gamma_{c_k}(x_i) - \gamma_{c_k}(x_j)|^2 + c|\pi_{c_k}(x_i) - \pi_{c_k}(x_j)|^2}$$

where a, b, c are weighting factors.

Remark 1. In IFDS, weighting factors can be given according to the need of different users. In general, $a \geq b > c$ and $a + b + c = 1$, $0 \leq a, b, c \leq 1$.

Property 1. Let $IFDS = (U, C \cup D, V, f)$, for any $x_i, x_j \in U$, $c_k \in C$, the similarity degree based on the weighted Euclidean distance satisfies

(1) $0 \leq sim_{c_k}(x_i, x_j) \leq 1$

(2) $sim_{c_k}(x_i, x_j) = sim_{c_k}(x_j, x_i)$

(3) $f(x_i, c_k) = f(x_j, c_k) \Leftrightarrow sim_{c_k}(x_i, x_j) = 1$

(4) if $f(x_i, c_k) = \langle 1, 0 \rangle$, $f(x_j, c_k) = \langle 0, 1 \rangle$, and $a = b = 0.5$, then $sim_{c_k}(x_i, x_j) = 0$, that is, x_i and x_j are completely dissimilarity in terms of c_k.

3.2 $\alpha-$ Similarity Relation

Based on the similarity degree defined above, we can define $\alpha-$ similarity relation in IFDS.

Definition 6. *Let* $IFIS = (U, C, V, f)$, $A \subseteq C$, $\alpha \in [0,1]$, *then* $\alpha-$*similarity relation* $T^{\alpha}(A)$ *in IFIS is defined as follows*

$$T^{\alpha}(A) = \{(x_i, x_j) \in U \times U | sim_{c_k}(x_i, x_j) \geq \alpha, \forall c_k \in A\}.$$

Obviously, $T^{\alpha}(A)$ is reflexive, symmetric and non-transitive in general.

Remark 2. Different values of α will determine different similarity relations in IFDS, which will yield different classifications of the universe U. In general, we can take the appropriate α according to the distribution characteristics of data sets.

(1) $0 \leq \alpha \leq 1$. When $\alpha = 0$, the classification of U is trivial, consisting of a unique block. When $\alpha = 1$, the classification of U is discrete, consisting of all singletons from U.

(2) With the value of α increases, the classification of U gets finer.

(3) When discrete degree of data sets is greater, we can take smaller the value of α, and vice versa.

Property 2. Let $IFIS=(U,C,V,f)$, $A \subseteq C$, $\alpha \in [0,1]$, then

$$T^{\alpha}(A) = \bigcap_{c_k \in A} T^{\alpha}(c_k).$$

Property 3. Let $IFIS=(U,C,V,f)$, $A \subseteq C$, $\alpha \in [0,1]$, then $T^{\alpha}(C) \subseteq T^{\alpha}(A)$.

Definition 7. *Let $IFDS=(U,C \cup D,V,f)$, $A \subseteq C$, $\alpha \in [0,1]$, then $\alpha-$ relative similarity relation $T^{\alpha}(A|D)$ in IFDS is defined as follows*

$$T^{\alpha}(A|D) = \{(x_i,x_j) \in U \times U | \forall c_k \in A, sim_{c_k}(x_i,x_j) \geq \alpha \vee f_d(x_i) = f_d(x_j)\}$$

3.3 $\alpha-$ Maximal Consistent Block

Li [22] introduced the concept of maximal consistent block in incomplete information systems. Zhang [23] introduced the maximal consistent block in interval-valued information systems. In this section, the concept of $\alpha-$ maximal consistent block is introduced in IFDS.

Definition 8. *Let $IFIS=(U,C,V,f)$, $A \subseteq C$, $\alpha \in [0,1]$, $\alpha-$similarity class is defined as $S_A^{\alpha}(x_i) = \{x_j \in U | (x_i,x_j) \in T^{\alpha}(A)\}$.*

Definition 9. *Let $IFIS=(U,C,V,f)$, $A \subseteq C$, $\alpha \in [0,1]$, $M \subseteq U$, M is called $\alpha-$ maximal consistent block if and only if it satisfies*

(1) for any x_i,$x_j \in M$, if $(x_i,x_j) \in T^{\alpha}(A)$, then M is $\alpha-$ similarity class;

(2) if for any $x_k \in U-M$, $\exists x_i \in M$, such that $(x_i,x_k) \notin T^{\alpha}(A)$.

A $\alpha-$ maximal consistent block describes the maximal set of objects in which all objects are similar to certain extent. The set of $\alpha-$ maximal consistent block constitutes a completely covering of the universe U, which can be represented by $\xi^{\alpha}(A) = \{M_A^{\alpha}(x_1), M_A^{\alpha}(x_2), \cdots, M_A^{\alpha}(x_n)\}$, where $M_A^{\alpha}(x_i)$ is $\alpha-$ maximal consistent block of x_i in terms of A.

Remark 3. According to the need of different users and the distribution characteristics of data sets, we can adjust the value of α to get different maximal consistent block.

Property 4. Let $IFIS=(U,C,V,f)$, $A \subseteq C$, $\alpha \in [0,1]$, then for any $x \in U$, $M_C^{\alpha}(x) \subseteq M_A^{\alpha}(x)$.

Property 5. Let $IFIS=(U,C,V,f)$, $A \subseteq C$, $0 \leq \alpha \leq \beta \leq 1$, then for any $x \in U$, $M_A^{\beta}(x) \subseteq M_A^{\alpha}(x)$.

From the property 4 and 5, with the number of condition attributes in IFDS or the value of α increases, the classification of the universe U gets finer, and the granularity of knowledge decreases.

Proposition 2. *Let $IFIS=(U,C,V,f)$, $A \subseteq C$, $\alpha \in [0,1]$, then for any $x \in U$, we have $T^{\alpha}(A)=T^{\alpha}(C) \Leftrightarrow M_A^{\alpha}(x)=M_C^{\alpha}(x)$.*

4 Attribute Reduction Based on Discernibility Matrix in IFDS

Miao [24] stressed a fact that the definition of a discernibility matrix should be tied to a certain property. In this section, based on the $\alpha-$ maximal consistent block, the discernibility matrix is constructed to find out all the relative reducts, which preserve the relative similarity relation unchanged in IFDS.

Definition 10. *Let $IFDS=(U,C \cup D,V,f)$, $R \subseteq C$ is a reduct of C with respect to D if it satisfies the following two conditions*
(1) $T^{\alpha}(R|D)=T^{\alpha}(C|D)$
(2) $\forall R' \subset R, T^{\alpha}(R'|D) \neq T^{\alpha}(C|D)$

Definition 11. *Let $IFDS=(U,C \cup D,V,f)$, the function $\partial_C: U \rightarrow P(V_d)$ is called generalized decision, where for any $x \in U$, $\partial_C(x)=\{f_d(y)|y \in S_C^{\alpha}(x)\}$.*

In the following, we construct the discernibility matrix to serve as a tool for discussing and analyzing attribute reduction in IFDS.

Definition 12. *Let $IFDS=(U,C \cup D,V,f)$, $\alpha \in [0,1]$, its discernibility matrix $M=M(x,y)$ is a $|U| \times |U|$ matrix, in which the element $M(x,y)$ for an object pair (x,y) is defined by*

$$M(x,y)=\begin{cases} \{c \in C|\forall M_c^{\alpha}(x_i) \in \xi^{\alpha}(c), \{x,y\} \not\subseteq M_c^{\alpha}(\mathrm{x}_i)\}, & \partial_C(x) \neq \partial_C(y) \\ \mathrm{C}, & otherwise \end{cases}$$

Theorem 1. *Let $IFDS=(U,C \cup D,V,f), R \subseteq C$, then*

$$T^{\alpha}(R|D)=T^{\alpha}(C|D) \Leftrightarrow \forall x,y \in U, M(x,y) \neq \emptyset, R \cap M(x,y) \neq \emptyset.$$

Proof. $\Rightarrow$: If $f_d(x)=f_d(y)$, then $M(x,y)=C$, the conclusion is true. If $f_d(x) \neq f_d(y)$, then $T^{\alpha}(R)=T^{\alpha}(C)$ according to $T^{\alpha}(R|D)=T^{\alpha}(C|D)$. $T^{\alpha}(R)=T^{\alpha}(C) \Leftrightarrow M_R^{\alpha}(x)=M_C^{\alpha}(x), \forall x \in U$. $\forall y \in U$, we have (1) if $y \notin M_C^{\alpha}(x)$, then $y \notin M_R^{\alpha}(x)$. $y \in M_R^{\alpha}(y)$, so $M_R^{\alpha}(x) \neq M_R^{\alpha}(y)$. $R \cap M(x,y) \neq \emptyset$. (2) if $y \in M_C^{\alpha}(x)$, then $M(x,y)=C$. $R \cap M(x,y) \neq \emptyset$.

$\Leftarrow$: $\forall x, y \in U$, if $f_d(x) = f_d(y)$, then $T^\alpha(R|D) = T^\alpha(C|D)$ hold. If $f_d(x) \neq f_d(y)$, we show the proof for $M_R^\alpha(x) = M_C^\alpha(x)$. Proof by contradiction. Suppose $M_R^\alpha(x) \neq M_C^\alpha(x)$, then $M_C^\alpha(x) \subset M_R^\alpha(x)$. According to $M(x,y) \neq \emptyset$, x and y are discernible. $y \notin M_C^\alpha(x)$. Therefore $y \in M_R^\alpha(x), y \notin M_C^\alpha(x)$. (Suppose $y \notin M_R^\alpha(x)$, then $M_R^\alpha(x) \subseteq M_C^\alpha(x)$, Therefore $M_R^\alpha(x) = M_C^\alpha(x)$, contradiction). That is x and y are maximal consistent in terms of R, but x and y are not maximal consistent in terms of C. $\forall a \in R, M_a^\alpha(x) = M_a^\alpha(y)$. According to $M(x,y) \neq \emptyset$, $R \cap M(x,y) = \emptyset$, contradiction. □

Theorem 2. $CORE(C|D) = \{a \in C | M(x,y) = \{a\}\ x, y \in U\}$.

Proof. $\Rightarrow$: $\forall a \in CORE(C|D), T^\alpha(C|D) \neq T^\alpha(C - \{a\}|D)$, thus $T^\alpha(C|D) \subset T^\alpha(C - \{a\}|D)$. It indicates that there exists $(x_1, y_1) \in T^\alpha(C - \{a\}|D)$, but $(x_1, y_1) \notin T^\alpha(C|D)$. Thus $(\forall b \in C - \{a\}, sim_b(x_1, y_1) > \alpha) \vee (f_d(x_1) = f_d(y_1))$ and $(\exists a \in C, sim_a(x_1, y_1) \leq \alpha) \wedge (f_d(x_1) \neq f_d(y_1))$. There two results imply that $\forall b \in C - \{a\}, sim_b(x_1, y_1) > \alpha$. Thus $M(x_1, y_1) \cap (C - \{a\}) = \emptyset$ and $M(x_1, y_1) \cap C \neq \emptyset$. Thus, $M(x_1, y_1) = \{a\}$.

$\Leftarrow$: For any $a \in C$ such that $\{a\} \in M$, there exists $(x_1, y_1) \in U \times U$ satisfying $M(x_1, y_1) = \{a\}$. This indicates that $(x_1, y_1) \notin T^\alpha(C|D), (x_1, y_1) \in T^\alpha(C - \{a\}|D)$. These imply that $T^\alpha(C|D) \neq T^\alpha(C - \{a\}|D)$. Thus, $a \in CORE(C|D)$. □

Definition 13. *The discernibility function of the discernibility matrix $M(x,y)$ is defined by* $f(\mathrm{M}) = \wedge \{\vee(\mathrm{M(x,y)}) | \forall x,y \in U, M(\mathrm{x,y}) \neq \emptyset\}$.

Theorem 3. *The reduct set problem is equivalent to the problem of transforming the discernibility function to a reduced disjunctive form. Each conjunctor of the reduced disjunctive form is a reduct of IFDS.*

Proof. This is a direct result from theorem 1 and the definition of minimal disjunction form. □

Algorithm 1: Attribute reduction based on discernibility matrix in IFDS.

Input: $IFDS = (U, C \cup D, V, f)$

Output: all relative reducts in IFDS

Step 1. Compute the $\alpha-$ maximal consistent block of every condition attribute;

Step 2. Compute the $\alpha-$ similarity class of every object in the universe in terms of C;

Step 3. Compute the generalized decision of every object in the universe;

Step 4. Construct the discernibility matrix M in IFDS;

Step 5. Construct the discernibility function $f(M)$;

Step 6. Convert $f(M)$ into a disjunctive normal form;

Step 7. Each disjunctive item in $f(M)$ corresponds to a relative reduct.

Example 4. Table 1 shows an information system security audit risk judgement decision table in [14]. $U = \{x_1, x_2, \cdots, x_{10}\}$ includes ten audited objects. The condition attribute set $C = \{c_1, c_2, \cdots, c_5\}$.

Table 1. An information system security audit risk judgement decision table

U	c_1	c_2	c_3	c_4	c_5	d
x_1	$\langle 0.2, 0.4\rangle$	$\langle 0.1, 0.7\rangle$	$\langle 0.2, 0.6\rangle$	$\langle 0.6, 0.4\rangle$	$\langle 0.2, 0.8\rangle$	1
x_2	$\langle 0.1, 0.7\rangle$	$\langle 0.1, 0.8\rangle$	$\langle 0.3, 0.6\rangle$	$\langle 0.5, 0.2\rangle$	$\langle 0.2, 0.7\rangle$	2
x_3	$\langle 0.1, 0.8\rangle$	$\langle 0.1, 0.8\rangle$	$\langle 0.2, 0.8\rangle$	$\langle 0.5, 0.4\rangle$	$\langle 0.6, 0.4\rangle$	1
x_4	$\langle 0.1, 0.9\rangle$	$\langle 0.6, 0.3\rangle$	$\langle 0.2, 0.7\rangle$	$\langle 0.2, 0.8\rangle$	$\langle 0.6, 0.4\rangle$	1
x_5	$\langle 0.4, 0.6\rangle$	$\langle 0.2, 0.6\rangle$	$\langle 0.2, 0.8\rangle$	$\langle 0.2, 0.8\rangle$	$\langle 0.2, 0.8\rangle$	2
x_6	$\langle 0.1, 0.6\rangle$	$\langle 0.2, 0.6\rangle$	$\langle 0.2, 0.8\rangle$	$\langle 0.2, 0.4\rangle$	$\langle 0.2, 0.8\rangle$	1
x_7	$\langle 0.6, 0.4\rangle$	$\langle 0.6, 0.4\rangle$	$\langle 0.6, 0.4\rangle$	$\langle 0.7, 0.3\rangle$	$\langle 0.4, 0.6\rangle$	2
x_8	$\langle 0.6, 0.2\rangle$	$\langle 0.6, 0.2\rangle$	$\langle 0.8, 0.2\rangle$	$\langle 0.4, 0.6\rangle$	$\langle 0.4, 0.5\rangle$	2
x_9	$\langle 0.6, 0.2\rangle$	$\langle 0.6, 0.4\rangle$	$\langle 0.8, 0.2\rangle$	$\langle 0.1, 0.6\rangle$	$\langle 0.8, 0.2\rangle$	3
x_{10}	$\langle 0.6, 0.4\rangle$	$\langle 0.6, 0.4\rangle$	$\langle 0.8, 0.2\rangle$	$\langle 0.8, 0.2\rangle$	$\langle 0.6, 0.4\rangle$	3

Let $\alpha = 0.8, a = 0.4, b = 0.4, c = 0.2$

Step 1, we compute the $\alpha-$maximal consistent block of every condition attribute

$$\begin{aligned}
\xi^{0.8}(c_1) &= \{\{x_1, x_2, x_6\}, \{x_2, x_3, x_4, x_6\}, \{x_2, x_3, x_5, x_6\}, \{x_7, x_8, x_9, x_{10}\}, \\
&\quad \{x_5, x_7, x_{10}\}\} \\
\xi^{0.8}(c_2) &= \{\{x_1, x_2, x_3, x_5, x_6\}, \{x_4, x_7, x_8, x_9, x_{10}\}\} \\
\xi^{0.8}(c_3) &= \{\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{x_2, x_7\}, \{x_7, x_8, x_9, x_{10}\}\} \\
\xi^{0.8}(c_4) &= \{\{x_1, x_2, x_3, x_7, x_{10}\}, \{x_1, x_3, x_8\}, \{x_2, x_3, x_6\}, \{x_4, x_5, x_8, x_9\}, \\
&\quad \{x_6, x_9\}\} \\
\xi^{0.8}(c_5) &= \{\{x_1, x_2, x_5, x_6, x_7, x_8\}, \{x_3, x_4, x_9, x_{10}\}, \{x_3, x_4, x_7, x_8, x_{10}\}\}
\end{aligned}$$

Step 2, we compute the $\alpha-$similarity class of every object in the universe in terms of C

$$\begin{aligned}
&[x_1]_C^{0.8} = \{x_1, x_2\}, [x_2]_C^{0.8} = \{x_1, x_2, x_6\}, [x_3]_C^{0.8} = \{x_3\}, [x_4]_C^{0.8} = \{x_4\}, \\
&[x_5]_C^{0.8} = \{x_5\}, [x_6]_C^{0.8} = \{x_2, x_6\}, [x_7]_C^{0.8} = \{x_7, x_{10}\}, [x_8]_C^{0.8} = \{x_8\}, \\
&[x_9]_C^{0.8} = \{x_9\}, [x_{10}]_C^{0.8} = \{x_7, x_{10}\}
\end{aligned}$$

Step 3, we compute the generalized decision of every object in the universe. The result is represented in the Table 2.

Table 2. An information system security audit risk judgement generalized decision table

U	c_1	c_2	c_3	c_4	c_5	d	$\partial_C(x)$
x_1	$\langle 0.2, 0.4\rangle$	$\langle 0.1, 0.7\rangle$	$\langle 0.2, 0.6\rangle$	$\langle 0.6, 0.4\rangle$	$\langle 0.2, 0.8\rangle$	1	$\{1, 2\}$
x_2	$\langle 0.1, 0.7\rangle$	$\langle 0.1, 0.8\rangle$	$\langle 0.3, 0.6\rangle$	$\langle 0.5, 0.2\rangle$	$\langle 0.2, 0.7\rangle$	2	$\{1, 2\}$
x_3	$\langle 0.1, 0.8\rangle$	$\langle 0.1, 0.8\rangle$	$\langle 0.2, 0.8\rangle$	$\langle 0.5, 0.4\rangle$	$\langle 0.6, 0.4\rangle$	1	$\{1\}$
x_4	$\langle 0.1, 0.9\rangle$	$\langle 0.6, 0.3\rangle$	$\langle 0.2, 0.7\rangle$	$\langle 0.2, 0.8\rangle$	$\langle 0.6, 0.4\rangle$	1	$\{1\}$
x_5	$\langle 0.4, 0.6\rangle$	$\langle 0.2, 0.6\rangle$	$\langle 0.2, 0.8\rangle$	$\langle 0.2, 0.8\rangle$	$\langle 0.2, 0.8\rangle$	2	$\{2\}$
x_6	$\langle 0.1, 0.6\rangle$	$\langle 0.2, 0.6\rangle$	$\langle 0.2, 0.8\rangle$	$\langle 0.2, 0.4\rangle$	$\langle 0.2, 0.8\rangle$	1	$\{1, 2\}$
x_7	$\langle 0.6, 0.4\rangle$	$\langle 0.6, 0.4\rangle$	$\langle 0.6, 0.4\rangle$	$\langle 0.7, 0.3\rangle$	$\langle 0.4, 0.6\rangle$	2	$\{2, 3\}$
x_8	$\langle 0.6, 0.2\rangle$	$\langle 0.6, 0.2\rangle$	$\langle 0.8, 0.2\rangle$	$\langle 0.4, 0.6\rangle$	$\langle 0.4, 0.5\rangle$	2	$\{2\}$
x_9	$\langle 0.6, 0.2\rangle$	$\langle 0.6, 0.4\rangle$	$\langle 0.8, 0.2\rangle$	$\langle 0.1, 0.6\rangle$	$\langle 0.8, 0.2\rangle$	3	$\{3\}$
x_{10}	$\langle 0.6, 0.4\rangle$	$\langle 0.6, 0.4\rangle$	$\langle 0.8, 0.2\rangle$	$\langle 0.8, 0.2\rangle$	$\langle 0.6, 0.4\rangle$	3	$\{2, 3\}$

Step 4, we construct the discernibility matrix in IFDS as follows

$$M=\begin{pmatrix}
C & & & & & & & & & \\
C & C & & & & & & & & \\
\{1,5\} & \{5\} & C & & & & & & & \\
\{1,2,4,5\} & \{2,4,5\} & C & C & & & & & & \\
\{1,4\} & \{4\} & \{4,5\} & \{1,2,5\} & C & & & & & \\
C & C & \{5\} & \{2,4,5\} & \{4\} & C & & & & \\
\{1,2,3\} & \{1,2\} & \{1,2,3\} & \{1,3,4\} & \{2,3,4\} & \{1,2,3,4\} & C & & & \\
\{1,2,3\} & \{1,2,3,4\} & \{1,2,3\} & \{1,3\} & C & \{1,2,3,4\} & \{4\} & C & & \\
C & C & \{1,2,3,4\} & \{1,3\} & \{1,2,3,5\} & \{1,2,3,5\} & \{4,5\} & \{5\} & C & \\
\{1,2,3,5\} & \{1,2,3,5\} & \{1,2,3\} & \{1,3,4\} & \{2,3,4,5\} & C & C & \{4\} & \{4\} & C
\end{pmatrix}$$

Since M is symmetric, we represent M only by elements in the lower triangle of M, and we use i to denote c_i in M.

Step 5, we construct the discernibility function and convert into the disjunctive normal form as follows

$$\begin{aligned} f(M) &= c_4 \wedge c_5 \wedge (c_1 \vee c_3) \wedge (c_1 \vee c_2) \\ &= (c_1 \wedge c_4 \wedge c_5) \vee (c_2 \wedge c_3 \wedge c_4 \wedge c_5) \end{aligned}$$

Finally, we obtain all the relative reducts $\{c_1, c_4, c_5\}$, $\{c_2, c_3, c_4, c_5\}$. c_4 and c_5 are the core attributes which can be obtained through the discernibility matrix.

Remark 4. The space and the time complexity of the proposed algorithm based on discernibility matrix are, respectively, $O(|C||U|^2)$ and $O(|C|^2|U|^2)$. So the algorithm is suitable for some IFDS which the number of attributes and objects is less.

5 Conclusion

This paper focus on the classification of the universe and attribute reduction in IFDS. The new weighted Euclidean distance and a parameter α is used to define the similarity relation, and an algorithm of attribute reduction based on the discernibility matrix is designed in IFDS. In the future work, we may develop some fast algorithms to compute reduction in IFDS.

Acknowledgments. This work is partially supported by the Natural Science Foundation of Shanxi Province (No.2012011011-1) and the Natural Science Foundation of Shanxi Normal University (No.ZR1109).

References

1. Pawlak, Z.: Rough sets. International Journal of Information and Computer Sciences 11, 341–356 (1982)
2. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
3. Atanassov, K.: Intuitionistic fuzzy sets. Fuzzy Sets and Systems 20, 87–96 (1986)
4. Atanassov, K.: Intuitionistic Fuzzy Sets:Theory and Applications. Physica-Verlag, Heidelberg (1999)

5. Pekala, B.: Properties of Atanassov's intuitionistic fuzzy relations and Atanassov's operators. Information Sciences 213, 84–93 (2012)
6. Zhou, L., Wu, W.Z.: Characterization of rough set approximations in Atanassov intuitionistic fuzzy set theory. Computers and Mathematics with Applications 62, 282–296 (2011)
7. Zhou, L., Wu, W.Z.: On generalized intuitionistic fuzzy rough approximation operators. Information Sciences 178, 2448–2465 (2008)
8. Radzikowska, A.M.: Rough approximation operations based on IF sets. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 528–537. Springer, Heidelberg (2006)
9. Samanta, S.K., Mondal, T.K.: Intuitionistic fuzzy rough sets and rough intuitionistic fuzzy sets. Journal of Fuzzy Mathematics 9, 561–582 (2001)
10. Cornelis, C., Cock, M.D., Kerre, E.E.: Intuitionistic fuzzy rough sets:at the crossroads of imperfect knowledge. Expert Systems 20, 260–270 (2003)
11. Zhang, X.H., Zhou, B., Li, P.: A general frame for intuitionistic fuzzy rough sets. Information Sciences 216, 34–49 (2012)
12. Zhang, Z.M.: Generalized intuitionistic fuzzy rough sets based on intuitionistic fuzzy coverings. Information Sciences 198, 186–206 (2012)
13. Huang, B., Li, H.X., Wei, D.K.: Dominance-based rough set model in intuitionistic fuzzy information systems. Knowledge-Based Systems 28, 115–123 (2012)
14. Huang, B., Zhuang, Y.L., Li, H.X., et al.: A dominance intuitionistic fuzzy-rough set approach and its applications. Appl. Math. Modelling (2012), doi: http://dx.doi.org/10.1016/j.apm.2012.12.009
15. Zhang, Z.M., Tian, J.F.: On attribute reduction with intuitionistic fuzzy rough sets. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 20, 59–76 (2012)
16. Wang, X.M., Shu, L.: An attribute reduction algorithm based on similarity measure of intuitionistic fuzzy rough sets. Fuzzy Systems and Mathematics 26, 185–190 (2012) (in Chinese)
17. Chen, H., Yang, H.C.: One new algorithm for intuitiontistic fuzzy-rough attribute reduction. Journal of Chinese Computer Systems 32, 506–510 (2011) (in Chinese)
18. Xu, Z.S., Yager, R.R.: Intuitionistic and interval-valued intutionistic fuzzy preference relations and their measures of similarity for the evaluation of agreement within a group. Fuzzy Optimization and Decision Making 8, 123–139 (2009)
19. Liu, H.W.: New similarity measures between intuitionistic fuzzy sets and between elements. Mathematical and Computer Modelling 42, 61–70 (2005)
20. Hwang, C.M., Yang, M.S., et al.: A similarity measure of intuitionistic fuzzy sets based on the Sugeno integral with its application to pattern recognition. Information Sciences 189, 93–109 (2012)
21. Zeng, S., Su, W., Sun, L.: A method based on similarity measures for interactive group decision-making with intuitionistic fuzzy preference relations. Appl. Math. Modelling (2013), doi: http://dx.doi.org/10.1016/j.apm.2013.01.044
22. Leung, Y., Li, D.Y.: Maximal consistent block technique for rule acquisition in incomplete information systems. Information Sciences 153, 85–106 (2003)
23. Zhang, N., Miao, D.Q., Yue, X.D.: Approaches to knowledge reduction in interval-valued information systems. Journal of Computer Research and Development 47, 1362–1371 (2010) (in Chinese) ISSN 1000-1239/CN 11-1777/TP
24. Miao, D.Q., Zhao, Y., Yao, Y.Y., et al.: Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model. Information Sciences 179, 4140–4150 (2009)

A Fuzzy Rough Set Approach for Incrementally Updating Approximations in Hybrid Information Systems

Anping Zeng[1,2], Tianrui Li[1,⋆], Chuan Luo[1], Junbo Zhang[1], and Yan Yang[1]

[1] School of Information Science and Technology,
Southwest Jiaotong University, Chengdu 610031, China
[2] School of Computer and Information Engineering,
Yibin University, Yibin 644007, China
zengap@126.com, trli@swjtu.edu.cn, luochuan@my.swjtu.edu.cn,
junbozhang86@163.com, yyang@swjtu.edu.cn

Abstract. In real-applications, there may exist missing data and many kinds of data (e.g., categorical, real-valued and set-valued data) in an information system which is called as a Hybrid Information System (HIS). A new Hybrid Distance (HD) between two objects in HIS is developed based on the value difference metric. Then, a novel fuzzy rough set is constructed by using the HD distance and the Gaussian kernel. In addition, the information systems often vary with time. How to use the previous knowledge to update approximations in fuzzy rough sets is a key step for its applications on hybrid data. The fuzzy information granulation methods based on the HD distance are proposed. Furthermore, the principles of updating approximations in HIS under the variation of the attribute set are discussed. A fuzzy rough set approach for incrementally updating approximations is then presented. Some examples are employed to illustrate the proposed methods.

Keywords: Fuzzy Rough Set, Incrementally Learning, Hybrid Information Systems.

1 Introduction

Rough Set Theory(RST) is a powerful mathematical tool proposed by Pawlak [1] for processing inexact, uncertain, or vague information, and it has been widely used in several research areas including knowledge discovery, pattern recognition, artificial intelligence, and data mining [2–5].

In fact, categorical, real-valued and set-valued features usually coexist in real-world databases. A disadvantage of the Pawlak's rough set is that this model is concerned with categorical features assuming some discrete values. Some discretization algorithms can be used to divide the domain of the corresponding numerical feature into several intervals, but the discretization usually causes

⋆ Corresponding author.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 157–168, 2013.

information loss. Therefore, an extended model of RST, fuzzy rough set, was proposed to deal with these cases [6, 7].

When developing a fuzzy rough set model, one of important issues is generating fuzzy relations between the samples and inducing a set of fuzzy granules with the fuzzy relations. Combined with the Euclidean distance, Gaussian kernels are first introduced to acquire fuzzy relations between samples described by fuzzy or numeric attributes in order to generate fuzzy information granules in the approximation space [11]. But the Euclidean distance has difficult to deal with the categorial and set-valued data, this paper will introduces a new hybrid distance.

In real-life applications, information systems may be big data [12,13] and vary with time. In fuzzy rough sets, the generating of fuzzy relations between samples inevitably elapses a lot of time, and frequently computing the fuzzy relations will reduce efficiency of the algorithms. Incremental updating approximations is a feasible solution. In fact, in RST and its extensions, more and more serious problems are arising due to the big data and dynamic property. Some researchers have paid attention to the problem of updating approximations of RST and its extensions incrementally in dynamic information systems [14–26]. Under the variation of attribute set, Li et al. proposed some approaches for incremental updating approximations and extracting rules in RST [14–17]. However, the incremental approach for updating approximations based on fuzzy rough sets under the variation of attribute set has not been taken into account until now.

The rest of this paper is organized as follows. In Section 2, some preliminaries are introduced. In Section 3, the generating methods of fuzzy information granules in hybrid information systems are presented. In Section 4, the updating principles for lower and upper approximations are analyzed under the variation of attribute set. Some illustrative examples are conducted. In Section 5, we conclude the paper.

2 Preliminaries

The rough set theory describes a crisp subset of a universe by two definable subsets called lower and upper approximations [1]. By using the lower and upper approximations, the knowledge hidden in information systems can be discovered and expressed in the form of decision rules.

Definition 1. *Let (U, R) be a Pawlak approximation space. The universe $U \neq \emptyset$. $R \subseteq U \times U$ is an equivalence relation on U. U/R denotes the family of all equivalence classes R, and $[x]_R$ denotes an equivalence class of R containing an element $x \in U$. For any $X \subseteq U$, the lower approximation and upper approximation of X are defined respectively as follows:*

$$\begin{aligned} \underline{R}X &= \{x \in U | [x]_R \subseteq X\}; \\ \overline{R}X &= \{x \in U | [x]_R \cap X \neq \emptyset\}. \end{aligned} \tag{1}$$

The concept of fuzzy rough sets was first proposed by Dubois and Prade [6].

Definition 2. *Let R be a fuzzy equivalence relation on U and X be a fuzzy subset of U. The fuzzy lower and upper approximations of X were defined as*

$$\begin{aligned}\underline{R}X(x) &= \inf_{y\in U}\{max(1-R(x,y),X(y))\};\\ \overline{R}X(x) &= \sup_{y\in U}\{min(R(x,y),X(y))\}.\end{aligned} \tag{2}$$

More generally, Yeung et al. proposed a model of fuzzy rough sets with a pair of T-norm and S-norm in [10].

$$\begin{aligned}\underline{R}X(x) &= \inf_{y\in U}\{S(N(R(x,y)),X(y))\};\\ \overline{R}X(x) &= \sup_{y\in U}\{T(R(x,y),X(y))\}.\end{aligned} \tag{3}$$

In [11], based on the Gaussian kernel function, Hu et al. proposed a Gaussian kernelized fuzzy rough set model with a pair of T_{cos}-norm and S_{cos}-norm.

Definition 3. *Let R_G be a Gaussian kernelized T_{cos}-fuzzy equivalence relation on U and X be a fuzzy subset of U. The fuzzy lower and upper approximations of X are defined as*

$$\begin{aligned}\underline{R_G}X(x) &= \inf_{y\in U} S_{cos}(N(R_G(x,y)),X(y));\\ \overline{R_G}X(x) &= \sup_{y\in U} T_{cos}(R_G(x,y),X(y)).\end{aligned} \tag{4}$$

Where $\forall x,y\in U, R_G(x,y)=k(x,y), T_{cos}(a,b)=max\{ab-\sqrt{1-a^2}\sqrt{1-b^2},0\}$ is a T-norm, and its dual $S_{cos}(a,b)=min\{a+b-ab+\sqrt{2a-a^2}\sqrt{2b-b^2},1\}$. In [11], the Gaussian kernel function $k(x,y)$ is definiton as follow.

Let U be a finite universe, and $U\neq\emptyset$. The samples are m-dimension vectors. $\forall x_i,x_k\in U,\ x_i =< x_{i1},x_{i2},...,x_{im}>,\ x_k =< x_{k1},x_{k2},...,x_{km}>$. The gaussian kernel function

$$k(x_i,x_k)=\exp\left(-\frac{||x_i-x_k||^2}{2\delta^2}\right) \tag{5}$$

can be used to compute the similarity between samples x_i and x_k. $||x_i-x_k||$ is the Euclidean distance between x_i and x_k.

A disadvantage of the Euclidean distance is that it is concerned with real values. In fact, categorical, real-valued and set-valued attributes usually coexist in real-world databases. In next section, a new hybrid distance will be introduced.

3 Gaussian Kernelized Fuzzy Rough Set in Hybrid Information Systems

Definition 4. *A Hybrid Information System (HIS) can be written as $(U, C\cup D, V, f)$, where U is the set of objects, $C=C^r\cup C^s\cup C^c$, C^r is the real-valued*

attribute set, C^s is the set-valued attribute set, C^c is the categorical attribute set, D denotes the set of decision attributes, $C^r \cap C^s = \emptyset, C^r \cap C^c = \emptyset, C^s \cap C^c = \emptyset, C \cap D = \emptyset$.

Example 1. Table 1 is a HIS with two categorical attributes "Headache", "Muscle Pain", a real-valued attribute "Temperature", a set-valued attribute "Syndrome" (denoted as a_1, a_2, a_3, a_4, respectively), and a decision attribute d. "?" denotes the unknown value.

Table 1. A hybrid information systems

U	**Headache**(a_1)	**Muscle Pain**(a_2)	**Temperature**(a_3)	**Syndrome**(a_4)	d
x_1	Sick	Yes	40	{C, R, A}	Flu
x_2	Sick	Yes	39.5	{C, R, A}	Flu
x_3	Middle	?	39	{C}	Flu
x_4	Middle	Yes	36.8	{R}	Rhinitis
x_5	Middle	No	?	{R}	Rhinitis
x_6	No	No	36.6	{R, A}	Health
x_7	No	?	?	{A}	Health
x_8	No	Yes	38	{C, R, A}	Flu
x_9	?	Yes	37	{R}	Health

3.1 Hybrid Distance

In HIS, there are different type of attributes, to construct the distance among objects efficiently, a novel distance function should be presented. Firstly, value difference under different type of attribute should be defined.

In order to deal with the value difference under the categorical attributes, Stanfill and Waltz [27] introduced a Value Difference Metric (VDM). Based it, the normalized value difference under the categorical attributes is defined as:

Definition 5. *Let HIS=$< U, C \cup D, V, f >, \forall x, y \in U, \forall a \in C$ and a is a categorical attribute,*

$$vdm(a(x), a(y)) = \sqrt{\frac{1}{|U/D|} \sum_{d_i \in U/D} (\frac{|a(x) \cap d_i|}{|a(x)|} - \frac{|a(y) \cap d_i|}{|a(y)|})^2}. \tag{6}$$

Where $|.|$ denotes support degree, and it is clear that $vdm(a(x), a(y)) \in [0, 1]$.

In [27], Wilson et al. also defined value difference under real-valued attributes.

Definition 6. *Let HIS=$< U, C \cup D, V, f >, \forall x, y \in U, \forall a \in C$ and a is a real-valued attribute,*

$$vdr(a(x), a(y)) = \frac{|a(x) - a(y)|}{4\delta_a} \tag{7}$$

where δ_a is the standard deviation under the attribute a.

In order to deal with the unknown values (denoted by "?"), Wilson et al. also defined value difference as [27]:

Definition 7. *Let HIS=*$< U, C \cup D, V, f >, \forall x, y \in U, \forall a \in C$, $a(x) =?$ *or* $a(y) =?$ *and* $x \neq y$,

$$vdi(a(x), a(y)) = 1. \tag{8}$$

According to Definition 7, the value difference will be set as 1 between an unknown value and another one.

To set-valued attribute, it can be seen as a set of multiple categorical attributes. For example, to the set-valued attribute d in Table 1, the subset which has maximum cardinal number in the domain V_d is {C, R, A}. Therefore, attribute d can be divided to three categorical attributes (C, R, A, respectively). Therefore, set-value {C, R, A}={C=Yes, R=Yes,A=Yes}, {C, R}={C=Yes, R=Yes,A=?}. Because the value difference between "?" and other values is equal to 1, the value difference between {C, R, A} and {C, R} is 1/3. Therefore, the value difference of set-valued attributes is defined as follow:

Definition 8. *Let HIS=*$< U, C \cup D, V, f >, \forall x, y \in U, \forall a \in C$ *and* a *is a set-valued attribute. Let* V_a *be the domain of* a.

$$vds(a(x), a(y)) = 1 - \frac{|a(x) \cap a(y)|}{s} \tag{9}$$

where s *is the maximum cardinal number (cardinality) in the subset of* V_a.

In order to deal with the hybrid and incomplete attributes, according to Definitions 5, 6, 7 and 8, a novel Hybrid Distance (HD) can be defined as follows:

Definition 9. *Given a HIS, the Hybrid Distance (HD) is defined as:*

$$HD(x, y) = \sqrt{\sum_{a=1}^{m} d^2(a(x), a(y))} \tag{10}$$

where m is the number of attributes, and

$$d(a(x), a(y)) = \begin{cases} 1, & a(x) =? \text{ or } a(y) =? \text{ and } x \neq y \\ vdm(a(x), a(y)), & a \text{ is a categorical attribute} \\ vds(a(x), a(y)), & a \text{ is a set} - \text{valued attribute} \\ vdr(a(x), a(y)), & a \text{ is a real} - \text{valued attribute} \end{cases} \tag{11}$$

Example 2. Based on Example 1, we can compute the HD distance matrix. According to formula (10), the following results hold:

(1) Because attribute a_1 is categorical, $d(a_1(x_1), a_1(x_3)) = vdm(a_1(x_1), a_1(x_3))$ $= \sqrt{\frac{1}{3}((\frac{2}{2} - \frac{1}{3})^2 + (\frac{0}{2} - \frac{2}{3})^2 + (\frac{0}{2} - \frac{0}{3})^2)} = 0.54$.

(2) Because $a_2(x_3) =?$, $d(a_2(x_1), a_2(x_3)) = 1$.

(3) Because attribute a_3 is real-valued, $d(a_3(x_1), a_3(x_3)) = \frac{a_3(x_1)-a_3(x_3)}{4\delta_{a_3}} = (40-39)/(4\times 1.28) = 0.19$.

(4) Because attribute a_4 is set-valued, $d(a_4(x_1), a_4(x_3)) = vds(a_4(x_1), a_4(x_3)) = \frac{3-1}{3} = 0.67$.

$HD(x_1, x_3) = (\sum_{a=1}^{4} d^2(a(x), a(y)))^{1/2} = \sqrt{0.54^2 + 1^2 + 0.19^2 + 0.67^2} = 1.33.$

3.2 Generating Fuzzy Relations under the Hybrid Attributes

Based on the gaussian kernel function in Formula (5), the Euclidean distance is replaced by HD distance, the new gaussian kernel function

$$k_H(x_i, x_k) = \exp\left(-\frac{||x_i - x_k||^2}{2\delta^2}\right) \tag{12}$$

$||x_i - x_k||$ is the HD distance between x_i and x_k. We have

(1) $k_H(x_i, x_k) \in [0, 1]$;

(2) $k_H(x_i, x_k) = k_H(x_k, x_i)$;

(3) $k_H(x_i, x_i) = 1$.

Using the new Gaussian kernel function, we can compute the T_{cos}-equivalence relation R_G in HIS. Furthermore, we can construct a Gaussian fuzzy rough set model.

Example 3. Base on Table 1, let $\delta^2 = 0.8$, each sample is a 4-D vector, the fuzzy relation between each two samples can be computed by Formula (12). For example, $R_G(x_1, x_3) = k_H(x_1, x_3) = exp(-\frac{HD^2(x_1,x_3)}{2\times 0.8}) = exp(-1.33^2/1.6) = 0.33$. Therefore,

$$R_G = \begin{pmatrix} 1 & 0.99 & 0.33 & 0.49 & 0.30 & 0.53 & 0.18 & 0.76 & 0.33 \\ & 1 & 0.33 & 0.53 & 0.30 & 0.57 & 0.18 & 0.79 & 0.35 \\ & & 1 & 0.26 & 0.15 & 0.29 & 0.13 & 0.33 & 0.14 \\ & & & 1 & 0.48 & 0.40 & 0.13 & 0.61 & 0.53 \\ & & & & 1 & 0.24 & 0.13 & 0.30 & 0.26 \\ & & & & & 1 & 0.22 & 0.80 & 0.26 \\ & & & & & & 1 & 0.22 & 0.08 \\ & & & & & & & 1 & 0.40 \\ & & & & & & & & 1 \end{pmatrix}.$$

3.3 Gaussian Kernelized Fuzzy Rough Set in HIS

Let HIS=$(U, C \cup D, V, f)$, $U/D = \{d_i\}, i = 1, 2, ..., |U/D|$. Here we suppose the following relationships hold: $\forall x \in d_i, d_i(x) = 1$; otherwise, $d_i(x) = 0$. Therefore, we can approximate the decision regions with the fuzzy granules induced by Gaussian function. Based on Definition 3, Hu et al. proposed the following proposition [11]:

Proposition 1. *HIS*=$(U, C \cup D, V, f)$, $\forall d_i \in U/D$,

$$\begin{aligned} \underline{R_G} d_i(x) &= \inf_{y \notin d_i} \sqrt{1 - {R_G}^2(x, y)}; \\ \overline{R_G} d_i(x) &= \sup_{y \in d_i} R_G(x, y). \end{aligned} \tag{13}$$

To simple the computing, we can generate the fuzzy lower and upper approximations by the follow proposition:

Proposition 2. *HIS*=$(U, C \cup D, V, f)$, $\forall x \in U, \forall d_i \in U/D$.

$$\begin{aligned} \underline{R_G} d_i(x) &= \sqrt{1 - (\sup_{y \notin d_i} R_G(x, y))^2}; \\ \overline{R_G} d_i(x) &= \sup_{y \in d_i} R_G(x, y). \end{aligned} \tag{14}$$

Proof. It is clear that function $y = \sqrt{1 - x^2}, x \in [0, 1]$ is a monotonically decreasing function. It is easy to prove that $\sqrt{1 - (\sup(x))^2} = \inf(\sqrt{1 - x^2}), x \in [0, 1]$. Therefore, $\underline{R_G} d_i(x) = \sqrt{1 - (\sup_{y \notin d_i} R_G(x, y))^2}$.

Example 4. Based on Examples 1 and 3, $U/D = \{d_1, d_2, d_3\}$, $d_1 = \{x_1, x_2, x_3, x_8\}$, $d_2 = \{x_4, x_5\}$, $d_3 = \{x_6, x_7, x_9\}$. According to Proposition 2,

$\underline{R_G} d_1(x_1) = \sqrt{1 - (\sup_{y \notin d_1} R_G(x_1, y))^2} = \sqrt{1 - (\sup\{0.49, 0.3, 0.53, 0.18, 0.33\})^2}$
$= \sqrt{1 - 0.53^2} = 0.85$.

Similarly, the other lower approximations can be computed as follows.
$\underline{R_G} d_1 = \{0.85/x_1, 0.82/x_2, 0.96/x_3, 0/x_4, 0/x_5, 0/x_6, 0/x_7, 0.84/x_8, 0/x_9\}$.
$\underline{R_G} d_2 = \{0/x_1, 0/x_2, 0/x_3, 0.79/x_4, 0.95/x_5, 0/x_6, 0/x_7, 0/x_8, 0/x_9\}$.
$\underline{R_G} d_3 = \{0/x_1, 0/x_2, 0/x_3, 0/x_4, 0/x_5, 0.61/x_6, 0.98/x_7, 0/x_8, 0.84/x_9\}$.

$\overline{R_G} d_1(x_1) = \sup_{y \in d_1} R_G(x_1, y) = \sup\{R_G(x_1, x_1), R_G(x_1, x_2), R_G(x_1, x_3),$
$R_G(x_1, x_8)\} = \sup\{1, 0.99, 0.33, 0.76\} = 1$.

Similarly, the other upper approximations can be computed as follows.
$\overline{R_G} d_1 = \{1/x_1, 1/x_2, 1/x_3, 0.61/x_4, 0.3/x_5, 0.8/x_6, 0.22/x_7, 1/x_8, 0.4/x_9\}$.
$\overline{R_G} d_2 = \{0.49/x_1, 0.53/x_2, 0.26/x_3, 1/x_4, 1/x_5, 0.4/x_6, 0.13/x_7, 0.61/x_8, 0.53/x_9\}$.
$\overline{R_G} d_3 = \{0.53/x_1, 0.57/x_2, 0.29/x_3, 0.53/x_4, 0.26/x_5, 1/x_6, 1/x_7, 0.8/x_8, 1/x_9\}$.

In next section, we apply the fuzzy rough set to design the incremental updating approximations under the variation of the attribute set.

4 A Fuzzy Rough Set Approach of Incrementally Updating Approximations under the Variation of the Attribute Set

We discuss the variation of approximations in HIS when the attribute set evolves over time. Given a HIS $= (U, C \cup D, V, f)$ at time t, $U \neq \emptyset$ and $C \cap D = \emptyset$. Suppose there are some attributes enter into HIS or get out of HIS at time $t+1$. The fuzzy equivalence relations will be changed. And then, the fuzzy lower and upper approximations will be changed too. Let $R_G^{(t)}$ be the fuzzy equivalence relation at time t. For each fuzzy set $X \subseteq U$, the fuzzy lower and upper approximations are denoted by $\underline{R_G^{(t)}}X$ and $\overline{R_G^{(t)}}X$ at time t, respectively. Let $P \subseteq C$ denote the attribute set at time t, R_G^P denotes the fuzzy equivalence relation under the attribute set P. Let $R_G^{(t+1)}$ be the fuzzy equivalence relation, Q_i be the immigrating attribute set and Q_e be the emigrating attribute set at time $t+1$. The fuzzy lower and upper approximations of X are denoted by $\underline{R_C^{(t+1)}}X$ and $\overline{R_G^{(t+1)}}X$, respectively. With these stipulations, we focus on the algorithms for updating approximations of the decision classes when (1) attributes enter into the HIS at time $t+1$; (2) attributes get out of the HIS at time $t+1$.

4.1 The Immigration of Attributes

Given a HIS $= (U, C \cup D, V, f)$, $\forall x_i, x_k \in U$. x_i, x_k can be seen as two m-dimension vectors, and $x_i =< x_i^{c_1}, x_i^{c_2}, ..., x_i^{c_m} >$, $x_k =< x_k^{c_1}, x_k^{c_2}, ..., x_k^{c_m} >$, $c_j \in C$, and $j = 1, ..., m, m = |C|$. $\forall P \subseteq C$, x_i, x_k can be seen as two m-dimension vectors denoted as x_i^P and x_k^P, respectively. $x_i^P =< x_i^{p_1}, x_i^{p_2}, ..., x_i^{p_l} >$, $x_k^P =< x_k^{p_1}, x_k^{p_2}, ..., x_k^{p_l} >$, $p_j \in P$, and $j = 1, ..., l, l = |P|$. According to formula (12), the following proposition holds.

Proposition 3. $\forall P \subseteq C, \forall x_i, x_k \in U$, $x_i \neq x_k$.

$$R_G^P(x_i, x_k) = \prod_{p_j \in P} R_G^{\{p_j\}}(x_i, x_k). \tag{15}$$

Proof. $R_G^P(x_i, x_k) = \exp(-\frac{||x_i^P - x_k^P||^2}{2\delta^2}) = \exp(-\frac{\sum_{j=1}^{|P|} d_{sa}^2(x_{ij}, x_{kj})}{2\delta^2})$
$= \prod_{p_j \in P} \exp(-\frac{d_{sa}^2(x_{ij}, x_{kj})}{2\delta^2}) = \prod_{p_j \in P} \exp(-\frac{||x_i^{\{p_j\}} - x_k^{\{p_j\}}||^2}{2\delta^2}) = \prod_{p_j \in P} R_G^{\{p_j\}}(x_i, x_k)$.

Proposition 4. *Let Q_i be an attribute set immigrating into HIS at time $t+1$. $\forall d_i \in U/D$, and $\forall x \in U$. The fuzzy approximations at time $t+1$ are as follows:*

$$\begin{aligned} \underline{R_G^{(t+1)}} d_i(x) &= \sqrt{1 - (\sup_{y \notin d_i} \{R_G^{(t)}(x, y) \times \prod_{q \in Q_i} R_G^{\{q\}}(x_i, x_k)\})^2}; \\ \overline{R_G^{(t+1)}} d_i(x) &= \sup_{y \in d_i} \{R_G^{(t)}(x, y) \times \prod_{q \in Q_i} R_G^{\{q\}}(x_i, x_k)\}. \end{aligned} \tag{16}$$

Table 2. Attribute a_5 is added into HIS

U	**Headache**(a_1)	**Muscle Pain**(a_2)	**Temperature**(a_3)	**Syndrome**(a_4)	**Cough**(a_5)	d
x_1	Sick	Yes	40	{C, R, A}	Yes	Flu
x_2	Sick	Yes	39.5	{C, R, A}	Yes	Flu
x_3	Middle	?	39	{C}	Yes	Flu
x_4	Middle	Yes	36.8	{R}	No	Rhinitis
x_5	Middle	No	?	{R}	No	Rhinitis
x_6	No	No	36.6	{R, A}	No	Health
x_7	No	?	?	{A}	No	Health
x_8	No	Yes	38	{C, R, A}	Yes	Flu
x_9	?	Yes	37	{R}	No	Health

Example 5. Based on Example 4, attribute a_5 is added into HIS (shown as Table 2). Therefore, $P = \{a_1, a_2, a_3, a_4\}$, $Q_i = \{a_5\}$.

According to Formula (12), we can compute the fuzzy relation between each two samples under the attribute set Q_i. For example, $R_G^{\{a_5\}}(x_1, x_4) = exp(-\frac{\frac{1}{3}(1+(\frac{2}{5})^2+(\frac{3}{5})^2)}{2\delta^2}) = 0.73$. Therefore,

$$R_G^{\{a_5\}} = \begin{pmatrix} 1 & 1 & 1 & 0.73 & 0.73 & 0.73 & 0.73 & 1 & 0.73 \\ & 1 & 1 & 0.73 & 0.73 & 0.73 & 0.73 & 1 & 0.73 \\ & & 1 & 0.73 & 0.73 & 0.73 & 0.73 & 1 & 0.73 \\ & & & 1 & 1 & 1 & 1 & 0.73 & 1 \\ & & & & 1 & 1 & 1 & 0.73 & 1 \\ & & & & & 1 & 1 & 0.73 & 1 \\ & & & & & & 1 & 0.73 & 1 \\ & & & & & & & 1 & 0.73 \\ & & & & & & & & 1 \end{pmatrix}.$$

Because $R_G^{(t)}$ has been generated in Example 3, $\forall x_i, x_k \in U, R_G^{(t+1)}(x_i, x_k) = R_G^{(t)}(x_i, x_k) \times R_G^{\{a_5\}}(x_i, x_k)$. Therefore,

$$R_G^{(t+1)} = \begin{pmatrix} 1 & 0.99 & 0.33 & 0.36 & 0.22 & 0.38 & 0.13 & 0.76 & 0.24 \\ & 1 & 0.33 & 0.39 & 0.22 & 0.41 & 0.13 & 0.79 & 0.25 \\ & & 1 & 0.19 & 0.11 & 0.21 & 0.09 & 0.33 & 0.10 \\ & & & 1 & 0.48 & 0.40 & 0.13 & 0.44 & 0.53 \\ & & & & 1 & 0.24 & 0.13 & 0.22 & 0.26 \\ & & & & & 1 & 0.22 & 0.58 & 0.26 \\ & & & & & & 1 & 0.16 & 0.08 \\ & & & & & & & 1 & 0.29 \\ & & & & & & & & 1 \end{pmatrix}.$$

According to Proposition 4, the approximations are as follows.

$\underline{R_G}d_1 = \{0.92/x_1, 0.91/x_2, 0.98/x_3, 0/x_4, 0/x_5, 0/x_6, 0/x_7, 0.81/x_8, 0/x_9\}$.
$\underline{R_G}d_2 = \{0/x_1, 0/x_2, 0/x_3, 0.84/x_4, 0.97/x_5, 0/x_6, 0/x_7, 0/x_8, 0/x_9\}$.
$\underline{R_G}d_3 = \{0/x_1, 0/x_2, 0/x_3, 0/x_4, 0/x_5, 0.81/x_6, 0.99/x_7, 0/x_8, 0.84/x_9\}$.
$\overline{R_G}d_1 = \{1/x_1, 1/x_2, 1/x_3, 0.44/x_4, 0.22/x_5, 0.58/x_6, 0.16/x_7, 1/x_8, 0.29/x_9\}$.
$\overline{R_G}d_2 = \{0.36/x_1, 0.39/x_2, 0.19/x_3, 1/x_4, 1/x_5, 0.4/x_6, 0.13/x_7, 0.44/x_8, 0.53/x_9\}$.
$\overline{R_G}d_3 = \{0.38/x_1, 0.41/x_2, 0.21/x_3, 0.53/x_4, 0.26/x_5, 1/x_6, 1/x_7, 0.58/x_8, 1/x_9\}$.

4.2 The Emigration of Attributes

Given two attribute sets $P, Q_e \subseteq C$, and $Q_e \subset P, Q_e \neq \emptyset$, fuzzy relation $R_G^{P-Q_e}(x_i, x_k)$ between x_i and x_k can be computed according to Proposition 3. And then, the following updating proposition of fuzzy approximations can be gotten.

Proposition 5. *Let $P \subseteq C$, and Q_e be the attributes emigrating from HIS at time $t+1$, and $Q_e \subset P$. $\forall d_i \in U/D$, and $\forall x \in U$. The fuzzy lower and upper approximations at time $t+1$ as follows:*

$$\begin{aligned}
\underline{R_G^{(t+1)}} d_i(x) &= \sqrt{1 - (\sup_{y \notin d_i}(R_G^{(t)}(x,y) / \prod_{q \in Q_e} R_G^{\{q\}}(x,y)))^2}; \\
\overline{R_G^{(t+1)}} d_i(x) &= \sup_{y \in d_i}(R_G^{(t)}(x,y) / \prod_{q \in Q_e} R_G^{\{q\}}(x,y)).
\end{aligned} \tag{17}$$

Table 3. The emigrating of attributes a_4, a_5

U	Headache(a_1)	Muscle Pain(a_2)	Temperature(a_3)	Syndrome(a_4)	Cough(a_5)	d
x_1	Sick	Yes	40	{C, R, A}	Yes	Flu
x_2	Sick	Yes	39.5	{C, R, A}	Yes	Flu
x_3	Middle	?	39	{C}	Yes	Flu
x_4	Middle	Yes	36.8	{R}	No	Rhinitis
x_5	Middle	No	?	{R}	No	Rhinitis
x_6	No	No	36.6	{R, A}	No	Health
x_7	No	?	?	{A}	No	Health
x_8	No	Yes	38	{C, R, A}	Yes	Flu
x_9	?	Yes	37	{R}	No	Health

Example 6. Based on Example 5, attribute set $\{a_4, a_5\}$ is deleted from HIS (shown as Table 3). Therefore $P = \{a_1, a_2, a_3, a_4, a_5\}$, $Q_e = \{a_4, a_5\}$. According to Proposition 3, we can compute the fuzzy relations under the attribute set Q_e. For example, $R_G^{Q_e}(x_1, x_4) = R_G^{\{a_4\}}(x_1, x_4) \times R_G^{\{a_5\}}(x_1, x_4) = 0.55$, $R_G^{(t+1)}(x_1, x_4) = R_G^{(t)}(x_1, x_4) / R_G^{Q_e}(x_1, x_4) = 0.36/0.55 = 0.65$. Therefore,

$$R_G^{(t+1)} = \begin{pmatrix}
1 & 0.99 & 0.43 & 0.65 & 0.40 & 0.56 & 0.24 & 0.76 & 0.43 \\
 & 1 & 0.44 & 0.70 & 0.40 & 0.61 & 0.24 & 0.79 & 0.46 \\
 & & 1 & 0.48 & 0.29 & 0.39 & 0.24 & 0.43 & 0.26 \\
 & & & 1 & 0.48 & 0.74 & 0.24 & 0.80 & 0.53 \\
 & & & & 1 & 0.44 & 0.24 & 0.40 & 0.27 \\
 & & & & & 1 & 0.29 & 0.85 & 0.48 \\
 & & & & & & 1 & 0.29 & 0.15 \\
 & & & & & & & 1 & 0.52 \\
 & & & & & & & & 1
\end{pmatrix}.$$

According to Proposition 5, the approximations are as follows:
$\underline{R_G}d_1 = \{0.76/x_1, 0.72/x_2, 0.88/x_3, 0/x_4, 0/x_5, 0/x_6, 0/x_7, 0.52/x_8, 0/x_9\}$.
$\underline{R_G}d_2 = \{0/x_1, 0/x_2, 0/x_3, 0.6/x_4, 0.9/x_5, 0/x_6, 0/x_7, 0/x_8, 0/x_9\}$.
$\underline{R_G}d_3 = \{0/x_1, 0/x_2, 0/x_3, 0/x_4, 0/x_5, 0.52/x_6, 0.96/x_7, 0/x_8, 0.84/x_9\}$.
$\overline{R_G}d_1 = \{1/x_1, 1/x_2, 1/x_3, 0.8/x_4, 0.4/x_5, 0.85/x_6, 0.29/x_7, 1/x_8, 0.52/x_9\}$.
$\overline{R_G}d_2 = \{0.65/x_1, 0.7/x_2, 0.48/x_3, 1/x_4, 1/x_5, 0.74/x_6, 0.24/x_7, 0.8/x_8, 0.53/x_9\}$.
$\overline{R_G}d_3 = \{0.56/x_1, 0.61/x_2, 0.39/x_3, 0.74/x_4, 0.44/x_5, 1/x_6, 1/x_7, 0.85/x_8, 1/x_9\}$.

5 Conclusions

In HIS, the attributes may be hybrid, and possible have unknown values. Based on this, a new HD formula was designed. Combined with the HD distance and the Gaussian kernel, a novel fuzzy rough set was constructed. In HIS, the attributes generally vary with time. The incremental updating principles of upper and lower approximations of fuzzy rough sets under the variation of the attribute set were discussed in this paper. Several examples were employed to illustrate the proposed methods. Our future research work will focus on the validation of the proposed algorithms in real data sets and the application on feature selection.

Acknowledgement. This work is supported by the National Science Foundation of China (Nos. 61175047, 61100117, U1230117), the Youth Social Science Foundation of the Chinese Education Commission (No. 11YJC630127), the Fundamental Research Funds for the Central Universities (SWJTU11ZT08, SWJTU12CX117, SWJTU12CX091), the Scientific Research Fundation of Sichuan Provincial Education Department (No. 13ZB0210), the 2013 Doctoral Innovation Funds of Southwest Jiaotong University.

References

1. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
2. Ananthanarayana, V.S., Narasimha, M.M., Subramanian, D.K.: Tree structure for efTcient data mining using rough sets. Pattern Recognit. Lett. 24, 851–862 (2003)
3. Skowron, A.: Extracting laws from decision tables: A rough set approach. Comput. Intell. 11, 371–388 (1995)
4. Peters, J.F., Skowron, A.: A rough set approach to knowledge discovery. International Journal of Computational Intelligence Systems 17(2), 109–112 (2002)
5. Tsumoto, S.: Mining diagnostic rules from clinical databases using rough sets and medical diagnostic model. Information Sciences 162(2), 65–80 (2004)
6. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. International Journal of General Systems 17(2-3), 191–209 (1990)
7. Yao, Y.Y.: Combination of rough and fuzzy sets based on α-level sets. In: Lin, T.Y., Cercone, N. (eds.) Rough Sets and Data Mining: Analysis for Imprecise Data, pp. 301–321. Kluwer Academic Publishers, Boston (1997)
8. Morsi, N., Mohamed, Y.: Axiomatics for Fuzzy Rough Set. Fuzzy Sets System 100(1-3), 327–342 (1998)

9. Mi, J., Zhang, W.: An Axiomatic Characterization of a Fuzzy Generalization of Rough Sets. Information Sciences 160, 235–249 (2004)
10. Yeung, D.S., Chen, D.G., Tsang, E.C.C., Lee, J.W.T., Wang, X.Z.: On the Generalization of Fuzzy Rough Sets. IEEE Trans. Fuzzy Systems 13(3), 343–361 (2005)
11. Hu, Q.H., Zhang, L., Chen, D.G., et al.: Gaussian Kernel based Fuzzy rough Sets: Model, Uncertainty Measures and Applications. International Journal of Approximate Reasoning 51, 453–471 (2010)
12. Zhang, J.B., Li, T.R., Ruan, D., Gao, Z.Z., Zhao, C.B.: A Parallel Method for Computing Rough Set Approximations. Information Sciences 194, 209–223 (2012)
13. Zhang, J.B., Li, T.R., Pan, Y.: Parallel Rough Set based Knowledge Acquisition using Map Reduce from Big Data. In: ACM SIGKDD 2012 Big Data Mining (BigMine 2012) Workshop, Beijing, China, pp. 20–27 (2012)
14. Li, T.R., Ruan, D., Wets, G., Song, J., Xu, Y.: A rough sets based characteristic relation approach for dynamic attribute generalization in data mining. Knowledge-Based Systems 20(5), 485–494 (2007)
15. Li, T.R., Xu, Y.: A generalized rough set approach to attribute generalization in data mining. Journal of Southwest Jiaotong University (English Edition) 8(1), 69–75 (2000)
16. Chan, C.C.: A rough set approach to attribute generalization in data mining. Information Sciences 107, 177–194 (1998)
17. Cheng, Y.: The incremental method for fast computing the rough fuzzy approximations. Data and Knowledge Engineering 70, 84–100 (2011)
18. Liu, D., Li, T.R., Ruan, D., Zhang, J.B.: Incremental learning optimization on knowledge discovery in dynamic business intelligent systems, Journal of Global Optimization. Journal of Global Optimization 51, 325–344 (2011)
19. Chen, H.M., Li, T.R., Hu, C.X., Ji, X.L.: An incremental updating principle for computing approximations in information systems while the object set varies with time. In: Proc. IEEE International Conference on Granular Computing, pp. 49–52. IEEE Press, Chengdu (2009)
20. Shusaku, T., Hiroshi, T.: Incremental learning of probabilistic rules from clinical database based on rough set theory. Journal of the American Medical Informations Association 4, 198–202 (1997)
21. Guan, Y., Wang, H.: Set-valued information systems. Information Sciences 176(17), 2507–2525 (2006)
22. Wang, L., Wu, Y., Wang, G.Y.: An incremental rule acquisition algorithm based on variable precision rough set model. Journal of Chongqing University of Posts and Telecommunications (Natural Science) 17(6), 709–713 (2005)
23. Zhang, J.B., Li, T.R., Ruan, D., Liu, D.: Neighborhood rough sets for dynamic data mining. International Journal of Intelligent Systems 27, 317–342 (2012)
24. Yong, L., Congfu, X., Yunhe, P.: A parallel approximate rule extracting algorithm based on the improved discernibility matrix. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 498–503. Springer, Heidelberg (2004)
25. Guo, S., Wang, Z.Y., Wu, Z.C., Yan, H.P.: A novel dynamic incremental rules extraction algorithm based on rough set theory. In: Proc. Fourth International Conference on Machine Learning and Cybernetics, pp. 1902–1907. IEEE Press, Guang Dong (2005)
26. Zheng, Z., Wang, G.Y.: A rough set and rule tree based incremental knowledge acquisition algorithm. Fundamenta Informaticae 59(2-3), 299–313 (2004)
27. Randall Wilson, D., Martinez Tony, R.: Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)

Implicator-Conjunctor Based Models of Fuzzy Rough Sets: Definitions and Properties

Lynn D'eer[1], Nele Verbiest[1], Chris Cornelis[1,2], and Lluis Godo[3]

[1] Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Krijgslaan 281 (S9), B-9000 Gent, Belgium
{Lynn.Deer,Nele.Verbiest}@UGent.be
[2] Department of Computer Science and Artificial Intelligence, University of Granada, Calle del Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain
chriscornelis@ugr.es
[3] IIIA - CSIC, Campus de la UAB s/n, 08193 Bellaterra, Spain
godo@iiia.csic.es

Abstract. Ever since the first hybrid fuzzy rough set model was proposed in the early 1990's, many researchers have focused on the definition of the lower and upper approximation of a fuzzy set by means of a fuzzy relation. In this paper, we review those proposals which generalize the logical connectives and quantifiers present in the rough set approximations by means of corresponding fuzzy logic operations. We introduce a general model which encapsulates all of these proposals, evaluate it w.r.t. a number of desirable properties, and refine the existing axiomatic approach to characterize lower and upper approximation operators.

Keywords: fuzzy sets, rough sets, hybridization, lower and upper approximation, implication, conjunction, axiomatic approach.

1 Introduction

Fuzzy sets [1] generalize classical or *crisp sets* in a sense that objects can be assigned intermediary membership degrees to a set or relation, drawn from a partially ordered set, typically $[0, 1]$. On the other hand, *rough sets* [2] characterize a set of objects by means of a lower and an upper approximation, taking into account an equivalence relation that represents indiscernibility between objects. Both theories have fostered broad research communities and have been applied in a wide range of settings. It was recognized early on that they are complementary, rather than competitive; a first hybrid fuzzy rough set model was proposed by Dubois and Prade [3] in 1990. Now, more than 20 years later, interest in fuzzy rough sets is thriving; this is mainly thanks to their proven application potential in machine learning, and in particular in feature selection [4–6] and instance selection [7].

Fuzzy-rough hybridization has been pursued in a variety of ways; in this paper, we focus on the most common approach, i.e., using fuzzy logical extensions of the Boolean implication and conjunction, along with infimum and supremum as extensions of the universal and existential quantifiers. This idea sparked the

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 169–179, 2013.

seminal proposal in [3], and since then many papers [8–18] have focused on the refinement of this model using both *constructive approaches*, which propose new definitions of approximation operators, and *axiomatic approaches*, which set forth a set of axioms or desirable properties, and characterize the operators that satisfy them.

It was found that through a deliberate choice of fuzzy logical operators, and the use of a similarity relation (also called fuzzy equivalence relation) to model approximate indiscernibility, most properties of the original rough set model can be maintained [8, 9]; on the other hand, from a practical point of view, the use of similarity relations is not always convenient (see e.g. [6]), and as De Cock et al. [16] argued, they cause part of the hybridization potential to remain unexplored. Moreover, also in the crisp case, various types of binary relations have been considered to replace the indiscernibility equivalence relation, see e.g. [19, 20]. For all of these reasons, several other authors considered fuzzy rough set models based on general fuzzy relations [10–15, 17].

In this paper, we unify all these approaches under the umbrella of a general implicator-conjunctor based fuzzy rough set model, imposing minimal restrictions on the approximations. After recalling some preliminaries in Section 2, we present the definitions of the approximations in Section 3, and give a chronological overview of special cases studied in the literature. In Section 4, we evaluate the model w.r.t. desirable properties, while in Section 5, we refine the axiomatic approach of Wu et al. [13], weakening some of its conditions and proposing an alternative characterization that caters specifically to residual implications. Finally, in Section 6, we conclude and outline future work.

2 Preliminaries

2.1 Fuzzy Logical Connectives

A *conjunctor* is a mapping $\mathcal{C}\colon [0,1]^2 \to [0,1]$ which is increasing in both arguments and which satisfies $\mathcal{C}(0,0) = \mathcal{C}(0,1) = \mathcal{C}(1,0) = 0$ and $\mathcal{C}(1,1) = 1$. It is called a *border conjunctor* if it satisfies $\mathcal{C}(1,x) = x$ for all x in $[0,1]$. A commutative, associative border conjunctor $\mathcal{T}$ is called a *t-norm*.

A *disjunctor* is a mapping $\mathcal{D}\colon [0,1]^2 \to [0,1]$ which is increasing in both arguments and which satisfies $\mathcal{D}(1,0) = \mathcal{D}(0,1) = \mathcal{D}(1,1) = 1$ and $\mathcal{D}(0,0) = 0$. It is called a *border disjunctor* if it satisfies $\mathcal{D}(0,x) = x$ for all x in $[0,1]$. A commutative, associative border disjunctor $\mathcal{S}$ is called a *t-conorm*.

A *negator* is a decreasing mapping $\mathcal{N}\colon [0,1] \to [0,1]$ which satisfies $\mathcal{N}(0) = 1$ and $\mathcal{N}(1) = 0$. It is *involutive* if for all $x \in [0,1]$, $\mathcal{N}(\mathcal{N}(x)) = x$. The standard negator $\mathcal{N}_s$ is defined by, for x in $[0,1]$, $\mathcal{N}_s(x) = 1 - x$.

Given an involutive negator $\mathcal{N}$, a conjunctor $\mathcal{C}$ and a disjunctor $\mathcal{D}$, the *$\mathcal{N}$-dual of $\mathcal{C}$* is a disjunctor $\mathcal{D}_{\mathcal{C},\mathcal{N}}$, defined by $\mathcal{D}_{\mathcal{C},\mathcal{N}}(x,y) = \mathcal{N}(\mathcal{C}(\mathcal{N}(x),\mathcal{N}(y)))$, and the *$\mathcal{N}$-dual of $\mathcal{D}$* is a conjunctor $\mathcal{C}_{\mathcal{D},\mathcal{N}}$, defined by $\mathcal{C}_{\mathcal{D},\mathcal{N}}(x,y) = \mathcal{N}(\mathcal{D}(\mathcal{N}(x),\mathcal{N}(y)))$, for all x, y in $[0,1]$. It can be verified that the $\mathcal{N}$-dual of a t-norm is a t-conorm, and vice versa.

An *implicator* $\mathcal{I}$ is a mapping $\mathcal{I}\colon [0,1]^2 \to [0,1]$ satisfying $\mathcal{I}(1,0) = 0$, $\mathcal{I}(1,1) = \mathcal{I}(0,1) = \mathcal{I}(0,0) = 1$ which is decreasing in the first and increasing in the second argument. If $\mathcal{I}$ satisfies $\mathcal{I}(1,x) = x$ for all x in $[0,1]$, it is called a *border implicator*, and if it satisfies the exchange principle, $\mathcal{I}(x,\mathcal{I}(y,z)) = I(y,\mathcal{I}(x,z))$ for all x,y,z in $[0,1]$, it is called an *EP implicator*.

Let $\mathcal{C}$, $\mathcal{D}$ and $\mathcal{N}$ be a border conjunctor, a disjunctor and a negator respectively. The *S-implicator* $\mathcal{I}_{\mathcal{D},\mathcal{N}}$ based on $\mathcal{D}$ and $\mathcal{N}$ is defined by, for x,y in $[0,1]$, $\mathcal{I}_{\mathcal{D},\mathcal{N}}(x,y) = \mathcal{D}(\mathcal{N}(x),y)$. The *R-implicator* $\mathcal{I}_{\mathcal{C}}$ based on $\mathcal{C}$ is defined by, for x,y in $[0,1]$, $\mathcal{I}_{\mathcal{C}}(x,y) = \sup\{\gamma \in [0,1] \mid \mathcal{C}(x,\gamma) \leq y\}$. Both S-implicators and R-implicators are particular cases of border implicators.

Given an involutive negator $\mathcal{N}$ and an implicator $\mathcal{I}$, the *induced conjunctor of $\mathcal{I}$ and $\mathcal{N}$* is a conjunctor $\mathcal{C}_{\mathcal{I},\mathcal{N}}$ defined by, for $x,y \in [0,1]$, $\mathcal{C}_{\mathcal{I},\mathcal{N}}(x,y) = \mathcal{N}(\mathcal{I}(x,\mathcal{N}(y))$. It is not necessarily a t-norm.

2.2 Fuzzy Sets and Relations

A fuzzy set A in a non-empty universe set U is a mapping $A : U \to [0,1]$. The collection of all fuzzy sets in U is denoted by $\mathcal{F}(U)$.

Given α in $[0,1]$, the *constant (fuzzy) set* $\hat{\alpha}$ is defined by, for x in U, $\hat{\alpha}(x) = \alpha$. In the crisp case, the only constant sets are $\emptyset$ and U.

Let $A, B \in \mathcal{F}(U)$ and $x \in U$. Given a negator $\mathcal{N}$, the *$\mathcal{N}$-complement of A* is defined by $(co_{\mathcal{N}}(A))(x) = \mathcal{N}(A(x))$. Given a conjunctor $\mathcal{C}$ and a disjunctor $\mathcal{D}$, the *$\mathcal{C}$-intersection and $\mathcal{D}$-union of A and B* are defined by $(A \cap_{\mathcal{C}} B)(x) = \mathcal{C}(A(x),B(x))$ and $(A \cup_{\mathcal{D}} B)(x) = \mathcal{D}(A(x),B(x))$. If $\mathcal{C} = \min$ and $\mathcal{D} = \max$, we simply write $\cap$ and $\cup$. Given an implicator $\mathcal{I}$, the *$\mathcal{I}$-implication of A and B* is defined by $(A \Rightarrow_{\mathcal{I}} B)(x) = \mathcal{I}(A(x),B(x))$.

A binary fuzzy relation R in U is a fuzzy set in $U \times U$. We define its inverse fuzzy relation R' by $R'(x,y) = R(y,x)$ for x,y in U. R is called *reflexive* if $R(x,x) = 1$, *symmetric* if $R(x,y) = R(y,x)$ and *inverse serial* if $\sup_{x \in U} R(x,y) = 1$ for all y in U. For a symmetric binary fuzzy relation R, it obviously holds that $R = R'$.

Given a t-norm $\mathcal{T}$, R is called *$\mathcal{T}$-transitive* if for all x, y and z in U, $\mathcal{T}(R(x,y),R(y,z)) \leq R(x,z)$. If R is reflexive, symmetric and $\mathcal{T}$-transitive, it is called a *$\mathcal{T}$-similarity relation*. When $\mathcal{T} = \min$, we shortly speak about a *similarity relation*. Because the minimum operator is the largest t-norm, a similarity relation is a $\mathcal{T}$-similarity relation for every t-norm $\mathcal{T}$.

2.3 Lower and Upper Approximations in Rough Set Theory

A *classical* or *Pawlak approximation space* is a couple (U,R) consisting of a non-empty set U and an equivalence relation R in U. The rough approximation of a crisp set A in U by R is the pair of sets $(R{\downarrow}A, R{\uparrow}A)$ defined by, for $x \in U$,

$$x \in R{\downarrow}A \Leftrightarrow (\forall y \in U)((y,x) \in R \Rightarrow y \in A) \tag{1}$$

$$x \in R{\uparrow}A \Leftrightarrow (\exists y \in U)((y,x) \in R \wedge y \in A). \tag{2}$$

A pair (A_1, A_2) of sets in U is called a *rough set* in (U, R) if there is a set A in U such that $A_1 = R{\downarrow}A$ and $A_2 = R{\uparrow}A$. Some of the most important properties of lower and upper approximation in a Pawlak approximation space are listed in the left hand side of Table 2. Note that we denote the complement of a crisp set A by A^c.

3 Implicator-Conjunctor Based Model

Many definitions of fuzzy rough sets emerge by faithfully extending Eqs. (1) and (2) to the $[0, 1]$-valued case. In particular, Dubois and Prade worked with a similarity relation R, and replaced the Boolean implication and conjunction by the S-implicator $\mathcal{I}_{\max,\mathcal{N}_s}$ (Kleene-Dienes implicator) and the minimum t-norm, respectively. In this section, we consider a *fuzzy approximation space*, i.e., a couple (U, R) consisting of a non-empty set U and a binary fuzzy relation R in U, and define a general format for the approximations using implicators and conjunctors.

Definition 1. *Let (U, R) be a fuzzy approximation space, A a fuzzy set in U, $\mathcal{I}$ an implicator and $\mathcal{C}$ a conjunctor. The* $(\mathcal{I}, \mathcal{C})$-fuzzy rough approximation *of A by R is the pair of fuzzy sets $(R{\downarrow_{\mathcal{I}}}A, R{\uparrow_{\mathcal{C}}}A)$ defined by, for $x \in U$,*

$$(R{\downarrow_{\mathcal{I}}}A)(x) = \inf_{y \in U} \mathcal{I}(R(y, x), A(y)) \tag{3}$$

$$(R{\uparrow_{\mathcal{C}}}A)(x) = \sup_{y \in U} \mathcal{C}(R(y, x), A(y)). \tag{4}$$

A pair (A_1, A_2) of fuzzy sets in U is called a fuzzy rough set *in (U, R) if there is a fuzzy set A in U such that $A_1 = R{\downarrow_{\mathcal{I}}}A$ and $A_2 = R{\uparrow_{\mathcal{C}}}A$.*

In Table 1 we give a chronological overview of special cases of the general model. Some authors [8,15,18] actually require lower semicontintuity of $\mathcal{T}$ instead of left-continuity, but by a result from [21] these two notions are equivalent for t-norms. Also, some papers [10, 11, 13, 17] consider fuzzy relations from U to W, with both U and W non-empty, finite universes, but here we restrict ourselves to the case $U = W$. As can be seen, Wu et al. [10] were the first to consider general binary fuzzy relations, while Mi and Zhang [11] initiated the use of conjunctors that are not necessarily t-norms. Also note that the t-norm $\mathcal{T}_{cos}$ used in [18] is defined, for x, y in $[0, 1]$, by $\mathcal{T}_{\cos}(x, y) = \max(xy - \sqrt{(1 - x^2)(1 - y^2)}, 0)$. Its use is inspired by the fact that some commonly used kernel functions in machine learning are in fact $\mathcal{T}_{\cos}$-similarity relations.

4 Properties

In the following, we assume that (U, R), (U, R_1) and (U, R_2) are fuzzy approximation spaces, A and B are fuzzy sets in U, $\mathcal{I}$ is an implicator, $\mathcal{C}$ a conjunctor and $\mathcal{N}$ an involutive negator. In the right hand side of Table 2, we show the extensions of the classical rough set properties to a fuzzy approximation space. We can prove the following propositions, which mainly generalize known results obtained in a restricted setting, see e.g. [9].

Table 1. Overview of special cases of the general fuzzy rough set model

Model	Conjunctor	Implicator	Relation
[3] Dubois & Prade, 1990	min	$\mathcal{I}_{\max,\mathcal{N}_s}$	similarity
[8] Morsi & Yakout, 1998	left-cont. t-norm $\mathcal{T}$	$\mathcal{I}_{\mathcal{T}}$	$\mathcal{T}$-similarity
[9] Radzikowska & Kerre, 2002	t-norm $\mathcal{T}$	border implicator $\mathcal{I}$	similarity
[10] Wu et al., 2003	min	$\mathcal{I}_{\max,\mathcal{N}_s}$	general
[11] Mi & Zhang, 2004	$\mathcal{C}_{\mathcal{I}_{\mathcal{T}},\mathcal{N}_s}$; left-cont. t-norm $\mathcal{T}$	$\mathcal{I}_{\mathcal{T}}$	general
[13] Wu et al., 2005	cont. t-norm $\mathcal{T}$	implicator $\mathcal{I}$	general
[14] Pei, 2005	min	$\mathcal{I}_{\max,\mathcal{N}_s}$	general
[15] Yeung et al., 2005	left-cont. t-norm $\mathcal{T}$	$\mathcal{I}_{\mathcal{S}_{T,\mathcal{N}},\mathcal{N}}$, $\mathcal{N}$ involutive	general
[15] Yeung et al., 2005	$\mathcal{C}_{\mathcal{I}_{\mathcal{T}},\mathcal{N}_s}$; left-cont. t-norm $\mathcal{T}$	$\mathcal{I}_{\mathcal{T}}$	general
[16] De Cock et al., 2007	t-norm $\mathcal{T}$	border implicator $\mathcal{I}$	general
[17] Mi et al., 2008	cont. t-norm $\mathcal{T}$	$\mathcal{I}_{\mathcal{S}_{\mathcal{T},\mathcal{N}_s},\mathcal{N}_s}$	general
[18] Hu et al., 2010	left-cont. t-norm $\mathcal{T}$ eft-cont. t-norm $\mathcal{T}$	$\mathcal{I}_{\mathcal{S}_{\mathcal{T},\mathcal{N}_s},\mathcal{N}_s}$	$\mathcal{T}_{\cos}$-similarity
[18] Hu et al., 2010	$\mathcal{C}_{\mathcal{I}_{\mathcal{T}},\mathcal{N}_s}$; left-cont. t-norm $\mathcal{T}$	$\mathcal{I}_{\mathcal{T}}$	$\mathcal{T}_{\cos}$-similarity

Proposition 1. *If $\mathcal{C}$ is the induced conjunctor of $\mathcal{I}$ and $\mathcal{N}$, i.e., $\mathcal{C} = \mathcal{C}_{\mathcal{I},\mathcal{N}}$, then the duality property holds.*

Corollary 1. *Let $\mathcal{D}$ be the $\mathcal{N}$-dual disjunctor of $\mathcal{C}$. If the pair $(\mathcal{I},\mathcal{C})$ consists of the S-implicator $\mathcal{I}_{\mathcal{D},\mathcal{N}}$ and the conjunctor $\mathcal{C}$, then the duality property holds.*

Corollary 2. *Let $\mathcal{T}$ be a left-continuous t-norm and $\mathcal{N} = \mathcal{N}_{\mathcal{I}_{\mathcal{T}}}$. If the pair $(\mathcal{I},\mathcal{C})$ consists of the R-implicator $\mathcal{I}_{\mathcal{T}}$ and the t-norm $\mathcal{T}$, then the duality property holds.*

To see this corollary, note that $\mathcal{C}_{\mathcal{I}_{\mathcal{T}},\mathcal{N}} = \mathcal{T}$ indeed holds: for x, y in $[0,1]$, $\mathcal{C}_{\mathcal{I}_{\mathcal{T}},\mathcal{N}}(x,y) = \mathcal{N}(\mathcal{I}_{\mathcal{T}}(x,\mathcal{N}(y))) = \mathcal{N}(\mathcal{I}_{\mathcal{T}}(x,\mathcal{I}_{\mathcal{T}}(y,0))) = \mathcal{N}(\mathcal{I}_{\mathcal{T}}(\mathcal{T}(x,y),0)) = \mathcal{N}(\mathcal{N}(\mathcal{T}(x,y))) = \mathcal{T}(x,y)$.

Proposition 2. *If the pair $(\mathcal{I},\mathcal{C})$ consists of the R-implicator $\mathcal{I}_{\mathcal{T}}$ and the left-continuous t-norm $\mathcal{T}$, then the adjointness property holds.*

Note that in generalizing the adjointness condition to a fuzzy approximation space, we have replaced R in the right hand side of the equivalence by its inverse fuzzy relation R'. Clearly, if R is symmetric (which is the case for a Pawlak approximation space), this modification is redundant.

Proposition 3. *If R is reflexive, $\mathcal{I}$ is a border implicator and $\mathcal{C}$ is a border conjunctor, then the inclusion property holds.*

Corollary 3. *Let $\mathcal{T}$ and $\mathcal{S}$ be a t-norm and its $\mathcal{N}$-dual t-conorm. If R is reflexive, and $(\mathcal{I},\mathcal{C}) = (\mathcal{I}_{\mathcal{S},\mathcal{N}},\mathcal{T})$ or $(\mathcal{I},\mathcal{C}) = (\mathcal{I}_{\mathcal{T}},\mathcal{T})$, then the inclusion property holds.*

Table 2. Properties in a Pawlak approximation space and their corresponding extensions to a fuzzy approximation space

Name	Pawlak approximation space	Fuzzy approximation space
Duality	$R{\downarrow}A = (R{\uparrow}A^c)^c$ $R{\uparrow}A = (R{\downarrow}A^c)^c$	$R{\downarrow_{\mathcal{I}}}A = \mathrm{co}_{\mathcal{N}}(R{\uparrow_{\mathcal{C}}}(\mathrm{co}_{\mathcal{N}}(A)))$ $R{\uparrow_{\mathcal{C}}}A = \mathrm{co}_{\mathcal{N}}(R{\downarrow_{\mathcal{I}}}(\mathrm{co}_{\mathcal{N}}(A)))$
Adjointness	$R{\uparrow}A \subseteq B \Leftrightarrow A \subseteq R{\downarrow}B$	$R{\uparrow_{\mathcal{C}}}A \subseteq B \Leftrightarrow A \subseteq R'{\downarrow_{\mathcal{I}}}B$
Inclusion	$R{\downarrow}A \subseteq A$ $A \subseteq R{\uparrow}A$	$R{\downarrow_{\mathcal{I}}}A \subseteq A$ $A \subseteq R{\uparrow_{\mathcal{C}}}A$
Set monotonicity	$A \subseteq B \Rightarrow R{\downarrow}A \subseteq R{\downarrow}B$ $A \subseteq B \Rightarrow R{\uparrow}A \subseteq R{\uparrow}B$	$A \subseteq B \Rightarrow R{\downarrow_{\mathcal{I}}}A \subseteq R{\downarrow_{\mathcal{I}}}B$ $A \subseteq B \Rightarrow R{\uparrow_{\mathcal{C}}}A \subseteq R{\uparrow_{\mathcal{C}}}B$
Relation monotonicity	$R_1 \subseteq R_2 \Rightarrow R_2{\downarrow}A \subseteq R_1{\downarrow}A$ $R_1 \subseteq R_2 \Rightarrow R_1{\uparrow}A \subseteq R_2{\uparrow}A$	$R_1 \subseteq R_2 \Rightarrow R_2{\downarrow_{\mathcal{I}}}A \subseteq R_1{\downarrow_{\mathcal{I}}}A$ $R_1 \subseteq R_2 \Rightarrow R_1{\uparrow_{\mathcal{C}}}A \subseteq R_2{\uparrow_{\mathcal{C}}}A$
Intersection	$R{\downarrow}(A \cap B) = R{\downarrow}A \cap R{\downarrow}B$ $R{\uparrow}(A \cap B) \subseteq R{\uparrow}A \cap R{\uparrow}B$	$R{\downarrow_{\mathcal{I}}}(A \cap B) = R{\downarrow_{\mathcal{I}}}A \cap R{\downarrow_{\mathcal{I}}}B$ $R{\uparrow_{\mathcal{C}}}(A \cap B) \subseteq R{\uparrow_{\mathcal{C}}}A \cap R{\uparrow_{\mathcal{C}}}B$
Union	$R{\downarrow}(A \cup B) \supseteq R{\downarrow}A \cup R{\downarrow}B$ $R{\uparrow}(A \cup B) = R{\uparrow}A \cup R{\uparrow}B$	$R{\downarrow_{\mathcal{I}}}(A \cup B) \supseteq R{\downarrow_{\mathcal{I}}}A \cup R{\downarrow_{\mathcal{I}}}B$ $R{\uparrow_{\mathcal{C}}}(A \cup B) = R{\uparrow_{\mathcal{C}}}A \cup R{\uparrow_{\mathcal{C}}}B$
Idempotence	$R{\downarrow}(R{\downarrow}A) = R{\downarrow}A$ $R{\uparrow}(R{\uparrow}A) = R{\uparrow}A$	$R{\downarrow_{\mathcal{I}}}(R{\downarrow_{\mathcal{I}}}A) = R{\downarrow_{\mathcal{I}}}A$ $R{\uparrow_{\mathcal{C}}}(R{\uparrow_{\mathcal{C}}}A) = R{\uparrow_{\mathcal{C}}}A$
Constant sets	$R{\downarrow}\emptyset = \emptyset = R{\uparrow}\emptyset$ $R{\downarrow}U = U = R{\uparrow}U$	$R{\downarrow_{\mathcal{I}}}\hat{\alpha} = \hat{\alpha}$ $R{\uparrow_{\mathcal{C}}}\hat{\alpha} = \hat{\alpha}$

Proposition 4. *The properties of set and relation monotonicity, intersection and union always hold.*

Proposition 5. *If R is a reflexive and $\mathcal{T}$-transitive relation, where $\mathcal{T}$ is a left-continuous t-norm and the pair $(\mathcal{I}, \mathcal{C})$ consists of the R-implicator $\mathcal{I}_{\mathcal{T}}$ and the t-norm $\mathcal{T}$, then the idempotence property holds.*

Proposition 6. *If R is a reflexive relation, $\mathcal{I}$ a border implicator and $\mathcal{C}$ a border conjunctor, then the constant sets property holds.*

Summing up, in order to satisfy all properties in Table 2, $\mathcal{C}$ should be a left-continuous t-norm $\mathcal{T}$ and $\mathcal{I}$ its R-implicator, while R needs to be at least reflexive and $\mathcal{T}$-transitive. Propositions 2 and 5 do not hold in general for S-implicators, for instance, Dubois and Prade's model [3] does not satisfy them.

5 Axiomatic Approach

In the axiomatic approach, we work with unary operators on $\mathcal{F}(U)$ and some axioms to obtain a fuzzy relation R such that the operators behave as approximation operators with respect to R. Such an approach is useful to get insight in the logical structure of fuzzy rough sets.

As our starting point, we use the axiomatic approach developed by Wu et al. [13], who propose axioms to characterise lower and upper approximations, which are generalized here for an implicator-conjunctor pair.

Definition 2. *Let $H, L\colon \mathcal{F}(U) \to \mathcal{F}(U)$, $\mathcal{C}$ a conjunctor and $\mathcal{I}$ an implicator. H is a* $\mathcal{C}$-upper approximation *if it satisfies, for all $A, A_j \in \mathcal{F}(U)$, $\alpha \in [0,1]$,*

(H1) $H(\hat{\alpha} \cap_{\mathcal{C}} A) = \hat{\alpha} \cap_{\mathcal{C}} H(A)$

(H2) $H\left(\bigcup_{j\in J} A_j\right) = \bigcup_{j\in J} H(A_j)$

L is an $\mathcal{I}$-lower approximation *if it satisfies, for all $A, A_j \in \mathcal{F}(U)$, $\alpha \in [0,1]$,*

$(L1)$ $L(\hat{\alpha} \Rightarrow_{\mathcal{I}} A) = \hat{\alpha} \Rightarrow_{\mathcal{I}} L(A)$

$(L2)$ $L\left(\bigcap_{j\in J} A_j\right) = \bigcap_{j\in J} L(A_j)$

Wu et al. required $\mathcal{C}$ and $\mathcal{I}$ to be a continuous t-norm and implicator, resp., but these conditions can be slightly weakened. For this, we can use e.g. results from [22] obtained in the framework of fuzzy modal logics that can be easily adapted to approximation operators.

Proposition 7. *Let $H\colon \mathcal{F}(U) \to \mathcal{F}(U)$ and $\mathcal{T}$ a left-continuous t-norm. H is a $\mathcal{T}$-upper approximation if and only if for all $A \in \mathcal{F}(U)$, $H(A) = R{\uparrow_{\mathcal{T}}}A$, where $R(x,y) = H(\{x\})(y)$, for x, y in U.*

Proposition 8. *Let $L\colon \mathcal{F}(U) \to \mathcal{F}(U)$ and $\mathcal{I}$ an EP implicator that is left-continuous in its first argument and such that $\mathcal{N}_{\mathcal{I}}$ is continuous. L is an $\mathcal{I}$-lower approximation if and only if for all $A \in \mathcal{F}(U)$, $L(A) = R{\downarrow_{\mathcal{I}}}A$, where $R(x,y) = \mathcal{N}_{\mathcal{I}}(L(U \setminus \{x\})(y))$, for x, y in U.*

Adding more axioms to Definition 2, we can characterize specific properties of the fuzzy relation R, as the following propositions show.

Proposition 9. *Let $\mathcal{T}$ be a left-continuous t-norm and H a $\mathcal{T}$-upper approximation. There exists a fuzzy relation R in U such that $H = R{\uparrow_{\mathcal{T}}}$ that is*

1. *inverse serial* $\Leftrightarrow \forall \alpha \in [0,1] : H(\hat{\alpha}) = \hat{\alpha} \Leftrightarrow H(U) = U$
2. *reflexive* $\Leftrightarrow \forall A \in \mathcal{F}(U) : A \subseteq H(A)$
3. *symmetric* $\Leftrightarrow \forall x, y \in U : H(\{x\})(y) = H(\{y\})(x)$
4. *$\mathcal{T}$-transitive* $\Leftrightarrow \forall A \in \mathcal{F}(U) : H(H(A)) \subseteq H(A)$

Proposition 10. *Let $\mathcal{I}$ be a border and EP implicator that is left-continuous in its first argument such that $\mathcal{N}_{\mathcal{I}}$ is continuous, and L an $\mathcal{I}$-lower approximation. There exists a fuzzy relation R in U such that $L = R{\downarrow_{\mathcal{I}}}$ that is*

1. *inverse serial* $\Leftrightarrow \forall \alpha \in [0,1] : L(\hat{\alpha}) = \hat{\alpha}$ *and* $\mathcal{I}$ *satisfies* $x \leq y \Leftrightarrow \forall z \in [0,1] : \mathcal{I}(x,z) \geq \mathcal{I}(y,z)$
2. *reflexive* $\Leftrightarrow \forall A \in \mathcal{F}(U) : L(A) \subseteq A$
3. *symmetric* $\Leftrightarrow \forall x, y \in U, \alpha \in [0,1] : L(\{x\} \Rightarrow_{\mathcal{I}} \hat{\alpha})(y) = L(\{y\} \Rightarrow_{\mathcal{I}} \hat{\alpha})(x)$
4. $\mathcal{T}$*-transitive* $\Leftrightarrow \forall A \in \mathcal{F}(U) : L(L(A)) \subseteq L(A)$ *for all* A *in* $\mathcal{F}(U)$ *and* $\mathcal{I}$ *satisfies* $\mathcal{I}(x, \mathcal{I}(y,z)) = \mathcal{I}(\mathcal{T}(x,y),z)$ *for all* x, y, z *in* $[0,1]$

The above propositions characterize lower and upper approximations separately. If these operators are dual, we can link them together.

Proposition 11. *Let* $\mathcal{T}$ *be a left-continuous t-norm,* $\mathcal{I}$ *an EP implicator that is left-continuous in its first argument and such that* $\mathcal{N}_{\mathcal{I}}$ *is involutive,* H *a* $\mathcal{T}$*-upper approximation and* L *an* $\mathcal{I}$*-lower approximation. If* H *and* L *satisfy duality w.r.t.* $\mathcal{N}_{\mathcal{I}}$*, then there exists a binary fuzzy relation* R *in* U *such that* $H = R{\uparrow_{\mathcal{T}}}$ *and* $L = R{\downarrow_{\mathcal{I}}}$.

A drawback of the above approach is that it excludes some important operators. For instance, it can be verified that the R-implicator $\mathcal{I}_{\min}$ does not satisfy the conditions of Proposition 8, because $\mathcal{N}_{\mathcal{I}_{\min}}$ is not involutive. However, it satisfies all properties from Table 2. For this reason, below we introduce and characterize the alternative notion of a $\mathcal{T}$-coupled pair of approximations.

Definition 3. *Let* $\mathcal{T}$ *be a left-continuous t-norm,* $H, L : \mathcal{F}(U) \to \mathcal{F}(U)$*. We call* (H, L) *a* $\mathcal{T}$-coupled pair *of upper and lower approximations if the following conditions hold:*

(H1,H2) H *is a* $\mathcal{T}$*-upper fuzzy approximation operator*
(L2) $L\left(\bigcap_{j \in J} A_j\right) = \bigcap_{j \in J} L(A_j)$
(HL) $L(A \Rightarrow_{\mathcal{I}_{\mathcal{T}}} \hat{\alpha}) = H(A) \Rightarrow_{\mathcal{I}_{\mathcal{T}}} \hat{\alpha}$

Proposition 12. *Let* $\mathcal{T}$ *be a left-continuous t-norm,* $H, L : \mathcal{F}(U) \to \mathcal{F}(U)$*.* (H, L) *is a* $\mathcal{T}$*-coupled pair of upper and lower approximations if and only if there exists a binary fuzzy relation* R *in* U *such that* $H = R{\uparrow_{\mathcal{T}}}$ *and* $L = R{\downarrow_{\mathcal{I}_{\mathcal{T}}}}$.

Proof. Assume (H, L) is a $\mathcal{T}$-coupled pair and $A \in \mathcal{F}(U)$. By (H1, H2), H is a $\mathcal{T}$-upper approximation, so by Proposition 7, $H(A) = R{\uparrow_{\mathcal{T}}}A$, where $R(x,y) = H(\{x\})(y)$, for x, y in U. On the other hand, it can be verified that $A = \bigcap_{y \in U}(\{y\} \Rightarrow_{\mathcal{I}_{\mathcal{T}}} \widehat{A(y)})$, so by (L2) and (HL), we have $L(A) = \bigcap_{y \in U} L(\{y\} \Rightarrow_{\mathcal{I}_{\mathcal{T}}} \widehat{A(y)}) = \bigcap_{y \in U} H(\{y\}) \Rightarrow_{\mathcal{I}_{\mathcal{T}}} \widehat{A(y)} = R{\downarrow_{\mathcal{I}_{\mathcal{T}}}}A$.
Conversely, it is clear that $R{\uparrow_{\mathcal{T}}}$ and $R{\downarrow_{\mathcal{I}_{\mathcal{T}}}}$ are an upper and a lower approximation satisfying (H1, H2) and (L2), respectively. To see (HL), let $x \in U$, $\alpha \in [0,1]$, then $\left(R{\downarrow_{\mathcal{I}_{\mathcal{T}}}}(A \Rightarrow_{\mathcal{I}_{\mathcal{T}}} \hat{\alpha})\right)(x) = \inf_{y \in U} \mathcal{I}_{\mathcal{T}}(R(y,x), \mathcal{I}_{\mathcal{T}}(A(y), \alpha)) = \inf_{y \in U} \mathcal{I}_{\mathcal{T}}(\mathcal{T}(R(y,x), A(y)), \alpha) = \mathcal{I}_{\mathcal{T}}(\sup_{y \in U} \mathcal{T}(R(y,x), A(y)), \alpha) = \mathcal{I}_{\mathcal{T}}((R{\uparrow_{\mathcal{T}}}A)(x), \alpha) = (R{\uparrow_{\mathcal{T}}}A \Rightarrow_{\mathcal{I}_{\mathcal{T}}} \hat{\alpha})(x)$.

Proposition 13. *Let $\mathcal{T}$ be a left-continuous t-norm and let (H, L) be a $\mathcal{T}$-coupled pair of upper and lower fuzzy approximation operators. There exists a binary fuzzy relation R in $U \times U$ such that $H = R{\uparrow_{\mathcal{T}}}$ and $L = R{\downarrow_{\mathcal{I}_{\mathcal{T}}}}$ that is:*

1. *inverse serial* $\Leftrightarrow H(U) = U \Leftrightarrow \forall A \in \mathcal{F}(U)\colon L(A) \subseteq H(A)$
2. *reflexive* $\Leftrightarrow \forall A \in \mathcal{F}(U)\colon L(A) \subseteq A \Leftrightarrow \forall A \in \mathcal{F}(U)\colon A \subseteq H(A)$
3. *symmetric* $\Leftrightarrow \forall x, y \in U\colon H(\{x\})(y) = H(\{y\})(x) \Leftrightarrow \forall A \in \mathcal{F}(U)\colon H(L(A)) \subseteq A \Leftrightarrow \forall A \in \mathcal{F}(U)\colon A \subseteq L(H(A))$
4. *$\mathcal{T}$-transitive* $\Leftrightarrow \forall A \in \mathcal{F}(U)\colon L(A) \subseteq L(L(A)) \Leftrightarrow \forall A \in \mathcal{F}(U)\colon H(H(A)) \subseteq H(A)$

Proof. By Proposition 12, we know that there exists a relation R such that $H = R{\uparrow_{\mathcal{T}}}$ and $L = R{\downarrow_{\mathcal{I}_{\mathcal{T}}}}$.

1. The equivalence between inverse seriality and $H(U) = U$ can be proved as follows: $H(U)(x) = \sup_{y \in U} \mathcal{T}(R(y, x), U(y)) = \sup_{y \in U} \mathcal{T}(R(y, x), 1) = \sup_{y \in U} R(y, x)$. Hence, $U = H(U)$ iff $H(U)(x) = 1$ for all $x \in U$, iff $\sup_{y \in U} R(y, x) = 1$ for all $x \in U$. The equivalence with $L(A) \subseteq H(A)$ for all $A \in \mathcal{F}(U)$ corresponds to [22, Proposition 4].
2. This corresponds to [22, Proposition 5].
3. The first equivalence is proved as in Proposition 9, item 3. The second and third one correspond to [22, Proposition 9].
4. This corresponds to [22, Proposition 13].

6 Conclusion and Future Work

In this paper, we have studied a general implicator-conjunctor based model for the lower and upper approximation of a fuzzy set under a binary fuzzy relation. We reviewed models from the literature that can be seen as special cases, and enriched the existing axiomatic approach with a new notion of $\mathcal{T}$-coupled pairs of approximations, which characterize the operations satisfying all relevant properties of classical rough sets, i.e., left-continuous t-norms and their R-implicators.

An important challenge is to extend the formal treatment to noise-tolerant fuzzy rough set models, such as those studied in [23–29]. Observing that the implicator-conjunctor based approximations are sensitive to small changes in the arguments (for instance, because of their reliance on inf and sup operations), many authors have proposed models that are more robust against data perturbation. However, this normally goes at the expense of the properties the corresponding fuzzy rough set model satisfies.

Acknowledgment. This work was partially supported by the Spanish Ministry of Science and Technology under Project TIN2011-28488. Lluis Godo has been partially supported by the MINECO Project TIN2012-39348-C02-01.

References

1. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
2. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
3. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. International Journal of General Systems 17, 191–209 (1990)
4. Jensen, R., Shen, Q.: Fuzzy-rough sets assisted attribute selection. IEEE Transactions on Fuzzy Systems 15(1), 73–89 (2007)
5. Tsang, E., Chen, D., Yeung, D., Wang, X., Lee, J.: Attributes reduction using fuzzy rough sets. IEEE Transactions on Fuzzy Systems 16(5), 1130–1141 (2008)
6. Cornelis, C., Hurtado Martín, G., Jensen, R., Ślęzak, D.: Attribute selection with fuzzy decision reducts. Information Sciences 180(2), 209–224 (2010)
7. Verbiest, N., Cornelis, C., Herrera, F.: FRPS: A fuzzy rough prototype selection method. Pattern Recognition 46(10), 2770–2782 (2013)
8. Morsi, N., Yakout, M.: Axiomatics for fuzzy rough set. Fuzzy Sets Systems 100, 327–342 (1998)
9. Radzikowska, A., Kerre, E.: A comparative study of fuzzy rough sets. Fuzzy Sets and Systems 126, 137–155 (2002)
10. Wu, W., Mi, J., Zhang, W.: Generalized fuzzy rough sets. Information Sciences 151, 263–282 (2003)
11. Mi, J., Zhang, W.: An axiomatic characterization of a fuzzy generalization of rough sets. Information Sciences 160, 235–249 (2004)
12. Wu, W., Zhang, W.: Constructive and axiomatic approaches of fuzzy approximation operators. Information Sciences 159, 233–254 (2004)
13. Wu, W., Leung, Y., Mi, J.: On characterizations of (I, T)-fuzzy rough approximation operators. Fuzzy Sets and Systems 154, 76–102 (2005)
14. Pei, D.: A generalized model of fuzzy rough sets. International Journal of General Systems 34(5), 603–613 (2005)
15. Yeung, D., Chen, D., Tsang, E., Lee, J., Xizhao, W.: On the generalization of fuzzy rough sets. IEEE Transactions on Fuzzy Systems 13(3), 343–361 (2005)
16. De Cock, M., Cornelis, C., Kerre, E.: Fuzzy rough sets: the forgotten step. IEEE Transactions on Fuzzy Systems 15(1), 121–130 (2007)
17. Mi, J., Leung, Y., Zhao, H., Feng, T.: Generalized fuzzy rough sets determined by a triangular norm. Information Sciences 178, 3203–3213 (2008)
18. Hu, Q., Zhang, L., Chen, D., Pedrycz, W., Yu, D.: Gaussian kernel based fuzzy rough sets: model, uncertainty measures and applications. International Journal of Approximate Reasoning 51, 453–471 (2010)
19. Slowinski, R., Vanderpooten, D.: Similarity relations as a basis for rough approximations. In: Wang, P. (ed.) Advances in Machine Intelligence and Soft Computing, pp. 17–33 (1997)
20. Slowinski, R., Vanderpooten, D.: Generalized rough set models. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery, pp. 286–318 (1998)
21. Fodor, J.: Left-continuous t-norms in fuzzy logic: an overview. Journal of Applied Sciences at Budapest Tech Hungary 1(2) (2004)
22. Radzikowska, A., Kerre, E.: Characterisation of main classes of fuzzy relations using fuzzy modal operators. Fuzzy Sets and Systems 152, 223–247 (2005)
23. Salido, J.M.F., Murakami, S.: Rough set analysis of a general type of fuzzy data using transitive aggregations of fuzzy similarity relations. Fuzzy Sets and Systems 139, 635–660 (2003)

24. Mieszkowicz-Rolka, A., Rolka, L.: Fuzzy rough approximations of process data. International Journal of Approximate Reasoning 49, 301–315 (2008)
25. Cornelis, C., De Cock, M., Radzikowska, A.M.: Vaguely quantified rough sets. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS (LNAI), vol. 4482, pp. 87–94. Springer, Heidelberg (2007)
26. Zhao, S., Tsang, E.C.C., Chen, D.: The model of fuzzy variable precision rough sets. IEEE Transactions on Fuzzy Systems 17(2), 451–467 (2009)
27. Hu, Q., An, S., Yu, D.: Soft fuzzy rough sets for robust feature evaluation and selection. Information Sciences 180, 4384–4400 (2010)
28. Cornelis, C., Verbiest, N., Jensen, R.: Ordered weighted average based fuzzy rough sets. In: Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.) RSKT 2010. LNCS, vol. 6401, pp. 78–85. Springer, Heidelberg (2010)
29. Hu, Q., Zhang, L., An, S., Zhang, D., Yu, D.: On robust fuzzy rough set models. IEEE Transactions on Fuzzy Systems 20(4), 636–651 (2012)

OWA-FRPS: A Prototype Selection Method Based on Ordered Weighted Average Fuzzy Rough Set Theory

Nele Verbiest[1], Chris Cornelis[1,2], and Francisco Herrera[2]

[1] Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Krijgslaan 281 (S9), B-9000 Gent, Belgium
Nele.Verbiest@UGent.be

[2] Department of Computer Science and Artificial Intelligence, University of Granada, Calle del Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain
chriscornelis@ugr.es, herrera@decsai.ugr.es

Abstract. The Nearest Neighbor (NN) algorithm is a well-known and effective classification algorithm. Prototype Selection (PS), which provides NN with a good training set to pick its neighbors from, is an important topic as NN is highly susceptible to noisy data. Accurate state-of-the-art PS methods are generally slow, which motivates us to propose a new PS method, called OWA-FRPS. Based on the Ordered Weighted Average (OWA) fuzzy rough set model, we express the quality of instances, and use a wrapper approach to decide which instances to select. An experimental evaluation shows that OWA-FRPS is significantly more accurate than state-of-the-art PS methods without requiring a high computational cost.

Keywords: Ordered Weighted Average, Fuzzy Rough Sets, Prototype Selection, KNN.

1 Introduction

One of the most well-known and most widely used classification algorithms is Nearest Neighbors (NN,[1]). This method classifies a test instance t to the class of the nearest neighbor of t in the training set. Although NN has been proven to be very useful for many classification problems, it deals with some problems, among which its sensitivity to noise and its large storage requirements are the most important ones.

In this work we alleviate these problems by using Prototype Selection (PS,[2]). This technique removes redundant and/or noisy instances from the training set, such that the training set requires less storage and such that the NN algorithm is more accurate. PS techniques that mainly try to improve the classification accuracy are called edition methods, those that focus on reducing the required storage are condensation methods. Hybrid PS techniques try to tackle both problems simultaneously. In this work we develop an editing method.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 180–190, 2013.

Many PS methods have been proposed in the literature, a comprehensive overview can be found in [2]. When the algorithm does not make use of a specific classifier to classify the entire training set, the method is called a filter method. Condensation methods do use a specific classifier, the NN classifier in our case, to classify the entire training data to obtain a quality assessment of a certain prototype subset. Filter methods are generally faster and less accurate, while wrapper methods are slower and more accurate. Many wrapper PS algorithms are evolutionary based, like CHC [3], GGA [4,5] or SSMA [6], while others use other search heuristics like RMHC [7] or RNG [8]. Most of the filter methods are based on the NN algorithm itself, like AllKNN [9] or MENN [10]. The method that we develop is a wrapper.

Although many researchers have focused on developing fuzzy rough feature selection [11] algorithms, there is not much literature on fuzzy rough PS yet. Nevertheless, fuzzy rough set theory [12] is a good tool to model noisy data. To the best of our knowledge, the only fuzzy rough based PS method is FRIS [13]. This method selects those instances that have a fuzzy positive region higher than a certain threshold. This method has some problems, the main one being that the method's performance highly relies on a good threshold selection.

In this work, we propose a new fuzzy rough based PS method that assesses the quality of instances using Ordered Weighted Average (OWA) fuzzy rough set theory [14], a more robust version of fuzzy rough set theory, and automatically selects an appropriate threshold.

The remainder of this work is structured as follows. In Section 2, we first discuss three OWA fuzzy rough quality measures that can be used to assess the quality of instances, and then show how these measures can be used to carry out PS. In Section 3, we evaluate our algorithm, called OWA Fuzzy Rough Prototype Selection (OWA-FRPS), and we conclude in Section 4.

2 Ordered Weighted Average Based Fuzzy Rough Prototype Selection

In this section we present our new PS method. In the first subsection we define three measures to assess the quality of instances, and in the second subsection we demonstrate how we can use these measures to carry out PS.

2.1 Assessing the Quality of Instances Using OWA Fuzzy Rough Sets

First we introduce some notations. We consider a decision system $(X, \mathcal{A} \cup \{d\})$, consisting of n instances $X = \{x_1, \ldots, x_n\}$, m attributes $\mathcal{A} = \{a_1, \ldots, a_m\}$ and a decision attribute $d \notin \mathcal{A}$. We denote by $a(x)$ the value of an instance $x \in X$ for an attribute $a \in \mathcal{A}$. We assume that each continuous attribute $a \in \mathcal{A}$ is normalized, that is, $\forall x \in X : a(x) \in [0, 1]$. The categorical attributes can take values in a finite set. The decision attribute d is categorical too and assigns a class $d(x)$ to each instance $x \in X$.

We associate a fuzzy indiscernibility relation $R : X \times X \to [0,1]$ with the decision system as follows. First, we calculate the fuzzy indiscernibility R_a for each feature $a \in \mathcal{A}$ separately. When a is categorical, $R_a(x,y) = 1$ for $x, y \in X$ if $a(x) = a(y)$ and $R_a(x,y) = 0$ otherwise. When a is continuous, $R_a(x,y) = 1 - |a(x) - a(y)|$ for all $x, y \in X$.

Next, we combine these separate fuzzy indiscernibility relations using a t-norm $\mathcal{T}$ (the Lukasiewicz t-norm[1] in this paper):

$$\forall x, y \in X : R(x,y) = \underbrace{\mathcal{T}(R_a(x,y))}_{a \in \mathcal{A}} \tag{1}$$

This fuzzy indiscernibility relation is the keystone of fuzzy rough set theory. A fuzzy set S can be approximated by its fuzzy rough lower approximation

$$\forall x \in X : (R \downarrow S)(x) = \min_{y \in X} \mathcal{I}(R(x,y), S(y)) \tag{2}$$

with $\mathcal{I}$ the Lukasiewicz implicator[2] in this paper, and by its upper approximation

$$\forall x \in X : (R \uparrow S)(x) = \max_{y \in X} \mathcal{T}(R(x,y), S(y)) \tag{3}$$

The fuzzy lower approximation expresses to what extent instances similar to x also belong to S , while the upper approximation expresses to what extent there exist instances that are similar to x and belong to S.

These concepts can be used to assess the quality of instances. First, note that we can consider the class $[x]_d$ of an instance $x \in X$ as a fuzzy set in X:

$$\forall y \in X : [x]_d(y) = \begin{cases} 1 & \text{if } d(x) = d(y) \\ 0 & \text{else} \end{cases} \tag{4}$$

which can be considered as the crisp set that contains all instances that have the same class as x.

If we want to assess the quality of an instance x, we can use the lower approximation of $[x]_d$:

$$(R \downarrow [x]_d)(x). \tag{5}$$

This value expresses to what extent instances similar to x also belong to the same class as x. Another option is to use the upper approximation of $[x]_d$:

$$(R \uparrow [x]_d)(x) \tag{6}$$

which expresses to what extent there exist instances that are similar to x and that belong to the same class as x.

Both measures are particularly meaningful in the context of NN classification, because they rate instances highly if they are surrounded by neighbors of the

[1] The Lukasiewicz t-norm is the mapping $\mathcal{T} : [0,1]^2 \to [0,1]$, such that $\forall a, b \in [0,1], \mathcal{T}(a,b) = \max(0, a + b - 1)$.

[2] The Lukasiewicz implicator is the mapping $\mathcal{I} : [0,1]^2 \to [0,1]$, such that $\forall a, b \in [0,1], \mathcal{I}(a,b) = \min(1 - a + b, 1)$.

same class: the lower approximation measure is high for x if there are no instances from a different class that are near (similar) to x, while the upper approximation measure is high if there exist neighbors from the same class.

In [14] it was noted that the traditional fuzzy rough approximations are highly susceptible to noise, as they use the crisp min and max operators, such that single instances can drastically influence the approximation values. A solution to this problem is to use OWA fuzzy rough sets [14], which replace these crisp operators by softer OWA operators [15]. Recall that, given a weight vector $W = \langle w_1, \ldots, w_n \rangle$ for which $\sum_{i=1}^{n} w_i = 1$ and $\forall i \in 1, \ldots, n, w_i \in [0, 1]$, the OWA aggregation of n values $s_1, \ldots, s_n$ is given by:

$$OWA_W(s_1, \ldots, s_n) = \sum_{i=1}^{n} w_i t_i, \tag{7}$$

where $t_i = s_j$ if s_j is the ith largest value in $s_1, \ldots, s_n$.

When $\langle 0, \ldots, 0, 1 \rangle$ is used as weight vector, the minimum operator is retrieved, which is the operator that is used in the traditional fuzzy lower approximation. We replace this minimum by a less strict operator that still has the characteristics of a minimum operator, that is, we consider a weight vector with ascending weights, such that lower values get higher weights, and higher values get lower weights. In this work we use the weight vector $W_{min} = \langle w_1, \ldots, w_n \rangle$ where

$$\forall i \in 1, \ldots, n : w_i = \frac{i}{n(n+1)/2}. \tag{8}$$

Completely analogously, we can define the $OWA_{W_{max}}$ operator that softens the maximum operator. Its weights $W_{max} = \langle w_1, \ldots, w_n \rangle$ are defined as follows in this paper:

$$\forall i \in 1, \ldots, n : w_i = \frac{n - i + 1}{n(n+1)/2}. \tag{9}$$

Replacing the strict minimum and maximum operators in the traditional definitions of fuzzy lower and upper approximation leads to the following more robust definitions of OWA fuzzy rough sets:

$$\forall x \in X : (R \downarrow^{OWA} S)(x) = \underset{y \in X}{OWA_{W_{min}}} \mathcal{I}(R(x, y), S(y)) \tag{10}$$

$$\forall x \in X : (R \uparrow^{OWA} S)(x) = \underset{y \in X}{OWA_{W_{max}}} \mathcal{T}(R(x, y), S(y)) \tag{11}$$

We will use this OWA fuzzy rough set model, leading to the following three quality measures:

$$\forall x \in X : \gamma_L(x) = (R \downarrow^{OWA} [x]_d)(x), \tag{12}$$

$$\forall x \in X : \gamma_U(x) = (R \uparrow^{OWA} [x]_d)(x), \tag{13}$$

and

$$\forall x \in X : \gamma_{LU}(x) = (R \downarrow^{OWA} [x]_d)(x) + (R \uparrow^{OWA} [x]_d)(x) \tag{14}$$

2.2 OWA-FRPS

Based on the quality measures γ defined in the previous subsection, we can formulate an algorithm to find a good subset of instances. We obviously want to select the instances with a high γ value and remove those with a low γ value, but now the question raises what threshold to use.

The main idea of our approach is to use the γ values of all instances in X as threshold. We calculate the leave-one-out training accuracy of the corresponding reduced subsets of instances and select the threshold that corresponds to the highest accuracy. More specifically, we carry out the following steps:

1. Calculate the $\gamma(x)$ values for all instances $x \in X$.
2. Remove the duplicates among all these γ values, the final set of γ values, which will all be considered as thresholds, is $G = \{\tau_1, \ldots, \tau_p\}, p \leq n$.
3. For each of the thresholds $\tau \in G$, consider the following subset: $S_\tau = \{x \in X | \gamma(x) \geq \tau\}$.
4. Calculate the training leave-one-out accuracy of each of these subsets using the LOO procedure in Algorithm 1.
5. Select the subsets $S_{\tau_{i_1}}, \ldots, S_{\tau_{i_s}}$ with the highest leave-one-out accuracy. Note that multiple subsets can correspond to the same leave-one-out accuracy.
6. Return the subset $S_{median(\tau_{i_1}, \ldots, \tau_{i_s})}$.

Algorithm 1. LOO, procedure to measure the training accuracy of a subset of instances using a leave-one-out approach

Input: Reduced decision system $(S, \mathcal{A} \cup \{d\})$ $(S \subseteq X)$.
$acc \leftarrow 0$
for $x \in X$ **do**
  if $x \in S$ **then**
    Find the nearest neighbor nn of x in $S \setminus \{x\}$
  else
    Find the nearest neighbor nn of x in S
  end if
  if $d(nn) = d(x)$ **then**
    $acc \rightarrow acc + 1$
  end if
end for
Output: acc

We illustrate the algorithm with an example. Consider the decision system in Table 1, with ten instances, two continuous features and one categorical feature. The values γ_{LU} are given in the last column for each instance. There are no duplicates, so the set of thresholds consists of the ten values in the last column of Table 1. In Table 2, we show the corresponding subsets. In order to calculate the training leave-one-out accuracy, we need the Euclidean distances between

the instances, which are given in Table 3. In the last two columns of Table 2, the instances that are correctly classified using the subset S_τ are given, together with the training accuracy. The subsets corresponding to the highest LOO training accuracy are S_{τ_1}, S_{τ_3} and S_{τ_9}, and subset $S_{\tau_3} = \{x_1, x_3, x_5, x_9\}$ will be returned by the OWA-FRPS algorithm.

Table 1. Decision system with 2 continuous features (a_1 and a_2) and one categorical feature (a_3). The class is given in column d and the value for the γ_{LU} measure is shown in the last column.

	a_1	a_2	a_3	d	γ_{LU}
x_1	0.2	0.4	A	0	1.02
x_2	0.3	0.3	A	1	1.016
x_3	1	0	B	0	1.16
x_4	0.7	0.9	B	1	1.07
x_5	0.4	0.3	A	0	1.05
x_6	0.3	0.6	A	1	1.01
x_7	0.4	1	B	0	1.06
x_8	0.3	0.2	B	1	1.15
x_9	0.7	0.5	A	0	1.17
x_{10}	0	0.1	A	1	1.14

Table 2. Thresholds τ considered in the OWA-FRPS algorithm and corresponding subsets of instances S_τ

Threshold τ	Corresponding subset S_τ	Correctly classified instances	LOO training accuracy
1.02	$\{x_1, x_3, x_4, x_5, x_7, x_8, x_9, x_{10}\}$	$\{x_1, x_5, x_6, x_9, x_{10}\}$	0.5
1.016	$\{x_1, x_2, x_3, x_4, x_5, x_7, x_8, x_9, x_{10}\}$	$\{x_5\}$	0.1
1.16	$\{x_3, x_9\}$	$\{x_1, x_3, x_5, x_7, x_9\}$	0.5
1.07	$\{x_3, x_4, x_8, x_9, x_{10}\}$	$\{x_2, x_4, x_5\}$	0.3
1.05	$\{x_3, x_4, x_5, x_7, x_8, x_9, x_{10}\}$	$\{x_4, x_5, x_9\}$	0.3
1.01	$\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$	$\{x_5, x_9\}$	0.2
1.06	$\{x_3, x_4, x_7, x_8, x_9, x_{10}\}$	$\{x_2, x_5\}$	0.2
1.15	$\{x_3, x_9, x_{10}\}$	$\{x_3, x_5, x_7\}$	0.3
1.17	$\{x_9\}$	$\{x_1, x_3, x_5, x_7, x_9\}$	0.5
1.14	$\{x_3, x_9, x_{10}\}$	$\{x_3, x_5, x_7\}$	0.3

Table 3. Euclidean distance between the instances

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
x_1	0.000	0.082	0.775	0.707	0.129	0.129	0.683	0.592	0.294	0.208
x_2	0.082	0.000	0.726	0.712	0.058	0.173	0.707	0.580	0.258	0.208
x_3	0.775	0.726	0.000	0.548	0.695	0.785	0.673	0.420	0.668	0.819
x_4	0.707	0.712	0.548	0.000	0.695	0.645	0.183	0.465	0.622	0.843
x_5	0.129	0.058	0.695	0.695	0.000	0.183	0.705	0.583	0.208	0.258
x_6	0.129	0.173	0.785	0.645	0.183	0.000	0.624	0.622	0.238	0.337
x_7	0.683	0.707	0.673	0.183	0.705	0.624	0.000	0.465	0.668	0.810
x_8	0.592	0.580	0.420	0.465	0.583	0.622	0.465	0.000	0.645	0.606
x_9	0.294	0.258	0.668	0.622	0.208	0.238	0.668	0.645	0.000	0.465
x_{10}	0.208	0.208	0.819	0.843	0.258	0.337	0.810	0.606	0.465	0.000

3 Experimental Evaluation

In this section we carry out an experimental evaluation to demonstrate the benefits of OWA-FRPS over other PS methods.

3.1 Experimental Set-Up

We use 28 datasets from the KEEL dataset repository[3]. The characteristics of these datasets are listed in Table 4. As our main focus is to improve the accuracy of NN, we compare OWA-FRPS with 12 PS algorithms that are most accurate according to the study performed in [2]. Additionally, we also compare OWA-FRPS to FRIS [13] with parameter value $\alpha = 10$. In Table 5, we give an overview of the algorithms we consider with references to the literature. Note that we use three versions of the new OWA-FRPS algorithm, depending on which measure is used to rank the instances.

For each dataset and PS method, we carry out the following 10 fold cross validation procedure. For each fold, we apply the PS method to the remaining folds (the train data) and then let NN find the nearest neighbors of the test instances in this reduced training set. We report the average classification accuracy, reduction and running time over the 10 folds.

3.2 Results

In Table 6, we show the average accuracy, reduction (the percentage of removed instances) and running time (in seconds) over all datasets. First, we note that on average, the OWA-FRPS-LU algorithm is more accurate than the other versions, which shows that both the lower and upper approximation contribute to the quality assessment of the instances. All OWA-FRPS algorithms outperform the state-of-the-art PS algorithms. From now on, we continue the analysis with OWA-FRPS-LU, to which we simply refer to as OWA-FRPS. To test if the improvement is significant, we carry out the statistical Friedman test and Holm post hoc procedure [21]. The Friedman ranks and the adjusted p-values of the Holm post hoc procedure are listed in Table 7. The OWA-FRPS algorithm has the best (i.e. lowest) rank. The low adjusted p-values confirm that OWA-FRPS is significantly more accurate than the state-of-the-art PS algorithms.

The reduction rate of the OWA-FRPS algorithms is about 30 percent, which is not as high as some of the evolutionary PS methods, but as the focus of our method is to improve the accuracy rather than reducing the storage needs, this result is of less importance.

The running time is of more interest to us. OWA-FRPS is slower than 6 other methods, but these methods have considerably lower accuracy rates. The running time of OWA-FRPS is shorter than the running times of the most accurate PS methods, so although OWA-FRPS is a wrapper and obtains excellent accuracy results, it does not come with the extra computational cost that wrapper PS methods typically have.

[3] `www.keel.es`

Table 4. Datasets used in the experimental evaluation with their number of instances (#Inst.), number of features (#Feat.) and number of classes (#Cl.)

Name	#Inst.	#Feat.	#Cl.	Name	#Inst.	#Feat.	#Cl.
appendicitis	106	7	2	**housevotes**	232	16	2
australian	690	14	2	**iris**	150	4	3
automobile	150	25	6	**led7digit**	500	7	10
balance	625	4	3	**lymphography**	148	18	4
bands	365	19	2	**mammographic**	830	5	2
breast	277	9	2	**new thyroid**	215	5	3
bupa	345	6	2	**pima**	768	8	2
crx	653	15	2	**saheart**	462	9	2
dermatology	358	34	6	**sonar**	208	60	2
ecoli	336	7	8	**vehicle**	846	18	4
glass	214	9	7	**vowel**	990	13	11
haberman	306	3	2	**wine**	178	13	3
hayesroth	160	4	3	**wisconsin**	683	9	2
heart	270	13	2	**zoo**	101	16	7

Table 5. Overview of the algorithms evaluated in the experimental study

Name	Description	Reference
AllKNN	NN based filter method	[9]
CHC	Evolutionary based wrapper method	[3]
GGA	Evolutionary based wrapper method	[4,5]
HMNEI	Hit and miss network based filter method	[16]
MENN	NN based filter method	[10]
ModelCS	Tree-based filter method	[17]
MSS	Spatial-based filter method	[18]
POP	Spatial-based filter method	[19]
RMHC	Random mutation hill climbing wrapper method	[7]
RNG	Graph based wrapper method	[8]
RNN	NN based filter method	[20]
SSMA	Evolutionary wrapper method	[6]
FRIS	Fuzzy rough based filter method	[13]
OWA-FRPS-LU	New OWA-FRPS method based on the quality measure that takes into account both the lower and upper approximation	-
OWA-FRPS-L	New OWA-FRPS method based on the quality measure that takes into account the lower approximation	-
OWA-FRPS-U	New OWA-FRPS method based on the quality measure that takes into account the upper approximation	-

Table 6. Average results of the PS methods averaged over all datasets, ordered according to performance. Reduction is the ratio of removed instances, running time is given in seconds.

Accuracy		**Reduction**		**Running Time**	
OWA-FRPS-LU	0.8087	CHC	0.9681	POP	0.0083
OWA-FRPS-L	0.8053	GGA	0.9391	MSS	0.0297
OWA-FRPS-U	0.7948	SSMA	0.9356	ModelCS	0.0306
RNG	0.7901	RNN	0.9111	MENN	0.0474
CHC	0.7893	RMHC	0.9015	FRIS	0.0576
ModelCS	0.7892	HMNEI	0.5383	AllKNN	0.0580
GGA	0.7863	MENN	0.4723	HMNEI	0.0714
AllKNN	0.7837	MSS	0.4632	**OWA-FRPS-U**	0.1834
SSMA	0.7828	**OWA-FRPS-U**	0.3462	**OWA-FRPS-L**	0.1880
FRIS	0.7808	AllKNN	0.3377	**OWA-FRPS-LU**	0.2031
RMHC	0.7799	**OWA-FRPS-L**	0.3247	RNG	2.6473
HMNEI	0.7785	**OWA-FRPS-LU**	0.2766	RNN	6.3661
POP	0.7741	RNG	0.2323	SSMA	14.9963
MENN	0.7705	ModelCS	0.1152	CHC	16.3427
MSS	0.7674	FRIS	0.0799	RMHC	18.2093
RNN	0.7614	POP	0.0484	GGA	42.9252

Table 7. Values of the statistics of the Friedman test and Holm post hoc procedure that compares OWA-FRPS-LU to the state-of the-art algorithms. The second column shows the Friedman ranks, the third column the Holm adjusted p-values.

Method	**Friedman Rank**	**Adj. p-value**
RNN	10	0.003846
MSS	10	0.004167
POP	9	0.004545
RMHC	8	0.005
FRIS	8	0.005556
MENN	7.5	0.00625
HMNEI	7	0.007143
SSMA	7	0.008333
AllKNN	7	0.01
GGA	7	0.0125
ModelCS	7	0.016667
CHC	6.5	0.025
RNG	6	0.05
OWA-FRPS-LU	4	-

4 Conclusion and Future Work

In this paper, we proposed a new PS method based on the OWA fuzzy rough set model, called OWA-FRPS. In order to select a subset of instances from the training set that improves the classification of the NN classifier, OWA-FRPS ranks the instances according to a OWA fuzzy rough measure, and then

automatically selects a suitable threshold to select the final subset of instances. An experimental evaluation on several datasets shows that our method achieves accuracy rates that are better than those of state-of-the-art PS methods, and moreover, OWA-FRPS is considerably faster.

As future directions, we would like to expand the use of OWA-FRPS for other classifiers like SVM and to improve OWA-FRPS for imbalanced datasets, that is, datasets for which one class is significantly more present than the other [22,23].

Acknowledgment. This work was partially supported by the Spanish Ministry of Science and Technology under Project TIN2011-28488.

References

1. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967)
2. García, S., Derrac, J., Cano, F.: Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(3), 414–435 (2012)
3. Cano, J., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in kdd: an experimental study. IEEE Transactions on Evolutionary Computation 7(6), 561–575 (2003)
4. Kuncheva, L., Jain, L.: Nearest neighbor classifier: Simultaneous editing and feature selection. Pattern Recognition Letters 20, 1149–1156 (1999)
5. Kuncheva, L.: Editing for the k-nearest neighbors rule by a genetic algorithm. Pattern Recognition Letters 16(8), 809–814 (1995)
6. García, S., Cano, J., Herrera, F.: A memetic algorithm for evolutionary prototype selection: A scaling up approach. Pattern Recognition 41, 2693–2709 (2008)
7. Skalak, D.: Prototype and feature selection by sampling and random mutation hill climbing algorithms. In: Machine Learning: Proceedings of the Eleventh International Conference, pp. 293–301 (1994)
8. Sanchez, J., Pla, F., Ferri, F.: Prototype selection for the nearest neighbour rule through proximity graphs. Pattern Recognition Letters 18, 507–513 (1997)
9. Tomek, I.: An experiment with the edited nearest-neighbor rule. IEEE Transactions on Systems, Man and Cybernetics 6(6), 448–452 (1976)
10. Hattori, K., Takahashi, M.: A new edited k-nearest neighbor rule in the pattern classification problem. Pattern Recognition 32, 521–528 (2000)
11. Cornelis, C., Jensen, R., Hurtado, G., Slezak, G.: Attribute selection with fuzzy decision reducts. Information Sciences 180(2), 209–224 (2010)
12. Dubois, D., Prade, H.: Rough fuzzy sets and fuzzy rough sets. International Journal of General Systems 17, 191–209 (1990)
13. Jensen, R., Cornelis, C.: Fuzzy-rough instance selection. In: Proceedings of the 19th International Conference on Fuzzy Systems, pp. 1776–1782 (2010)
14. Cornelis, C., Verbiest, N., Jensen, R.: Ordered weighted average based fuzzy rough sets. In: Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.) RSKT 2010. LNCS, vol. 6401, pp. 78–85. Springer, Heidelberg (2010)
15. Yager, R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Transactions on Systems, Man and Cybernetics 18, 183–190 (1988)

16. Marchiori, E.: Hit miss networks with applications to instance selection. Journal of Machine Learning Research 9, 997–1017 (2008)
17. Brodley, C.: Recursive automatic bias selection for classifier construction. Machine Learning 20, 63–94 (1995)
18. Barandela, R., Ferri, F., Sanchez, J.: Decision boundary preserving prototype selection for nearest neighbor classification. International Journal of Pattern Recognition and Artificial Intellgence 19, 787–806 (2005)
19. Riquelme, J., Aguilar-Ruiz, J., Aguilar-Ruiz, J., Toro, M.: Finding representative patterns with ordered projections. Pattern Recognition 36(4), 1009–1018 (2003)
20. Gates, G.: The reduced nearest neighbor rule. IEEE Transactions on Information Theory 18(3), 431–433 (1972)
21. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 1(1), 3–18 (2011)
22. Ramentol, E., Caballero, Y., Bello, R., Herrera, F.: Smote-rsb*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory. Knowledge and Information Systems, 1–21 (2011)
23. Ramentol, E., Verbiest, N., Bello, R., Caballero, Y., Cornelis, C., Herrera, F.: Smote-frst: A new resampling method using fuzzy rough set theory. In: Proceedings of the 10th International FLINS Conference on Uncertainty Modeling in Knowledge Engineering and Decision Making, pp. 800–805 (2012)

Applications of IF Rough Relational Model to Deal with Diabetic Patients

Chhaya Gangwal[1], Rabi Nanda Bhaumik[2], and Shishir Kumar[3]

[1] Research Scholar, Tripura University
c_hhaya@hotmail.com
[2] Emeritus Fellow (UGC) and Professor of Mathematics(Rtd.)
bhaumik_r_n@yahoo.co.in
[3] Assistant Professor (Biostatistics), Department of Community Medicine,
Agartala Government Medical College, Agartala, Tripura
shishirpunia@gmail.com

Abstract. This paper presents an intuitionistic fuzzy (IF) rough relational database model. The IF rough relational database model extends the IF and rough relational database models along with an IF rough relational algebra for querying. The usefulness of this model was illustrated with the Diabetic patients of Tripura where the various types of uncertainties are presented. For this study, first we design our database with an IF rough E-R diagram, created our database schema using an IF rough data definition and manipulation language (DDL and DML). Using IF Rough SQL-like languages, we then illustrate how the IF rough relational database may be queried and how the results are better than those of conventional databases.

1 Introduction

The explosive growth in databases has generated an urgent need for new techniques and tools that can intelligently transform the processed data into useful information and knowledge. There has been lot of works on diabetic databases. Breault [7] used rough sets on Diabetic Databases to see the accuracy in predicting diabetic status. The diabetic databases contain mostly imprecise data. Significant work has been done in incorporating uncertainty management in databases using theories like probability, rough sets, fuzzy sets, and IF sets etc. We see that Wong [12] model can process only incomplete information, Bagai and Suderraman [2] pointed out that their model can process incomplete and inconsistent information. Beaubouef and Petry [3,4] model can process uncertainty. In this paper we apply "IF rough relational database model"[9], for handling impreciseness and uncertain data for diabetic databases. We utilize the notions of indiscernibility from rough set theory coupled with the idea of membership and non-membership values from IF set theory.

2 Preliminaries

2.1 IF Set [1]

An IF set A in a nonempty set X is $A = \{(x, \mu_A(x), \nu_A(x)) : x \in X \}$, where $\mu_A(x)$ and $\nu_A(x)$ are functions from X to I = [0, 1] such that $0 \leq \mu_A(x) + \nu_A(x) \leq 1$, $\forall\ x \in X$. The

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 191–199, 2013.

numbers $\mu_A(x)$ and $\nu_A(x)$ represent the degree of membership and degree of non-membership for each element $x \in X$ to A respectively. The quantity $\pi_A(x) = 1 - (\mu_A(x) + \nu_A(x))$ is called the degree of indeterminacy or hesitation of the element $x \in X$ to the IF set A.

2.2 Rough Set [11]

Let *R* be an indiscernibility relation on universal set *U*. The pair A = (*U*, *R*) is called a Pawlak approximation space. Then for any non-empty subset *X* of *U*, the sets $\underline{R}X = \{x \in U : [x]_R \subseteq X\}$ and $\overline{R}X = \{x \in U : [x]_R \cap X \neq \phi\}$ are respectively, called the lower and the upper approximations of *X* in *A*. The set approximation $\underline{R}X$, $(U - \overline{R}X)$ and $(\overline{R}X - \underline{R}X)$ are described as R-positive region, R-negative region and R-boundary region respectively, where $[x]_R$ denotes the equivalence class of the relation *R* containing the element *x*. *X* is said to be definable set, if $\overline{R(X)} = \underline{R(X)}$. Otherwise *X* is said to be rough set.

2.3 IF Rough Set [9]

Let U be a universe and X, a rough set in U. An IF rough set A in U is characterized by a membership function $\mu_A : U \to [0,1]$ and a non-membership function $\nu_A : U \to [0,1]$ such that $\mu_A(\underline{R}X) = 1, \nu_A(\underline{R}X) = 0$ or $[\mu_A(x), \nu_A(x)] = [1,0]$ if $x \in (\underline{R}X)$ and $\mu_A(U - \overline{R}X) = 0, \nu_A(U - \overline{R}X) = 1,$ or $[\mu_A(x), \nu_A(x)] = [0,1]$ if $x \in U - \overline{R}X$, $0 \leq \mu_A(\overline{R}X - \underline{R}X) + \nu_A(\overline{R}X - \underline{R}X) \leq 1$.

2.4 IF Rough Relational Database Model [9]

In this model, a tuple t_i takes the form $(d_{i1}, d_{i2}, \ldots, d_{im}, d_{i[\mu,\nu]})$ where d_{ij} is a domain value of a particular domain set D_j and $d_{i[\mu,\nu]} \in [0, 1]$, the domain for IF membership and non-membership values denoted as $d_{i[\mu,\nu]} = [d_{i\mu}, d_{i\nu}]$. In the relational database, $d_{ij} \in D_j$. In the IF rough relational database except for the membership and non-membership values $d_{ij} \subseteq D_j$ where $d_{ij} \neq \phi$.

Definition 1. Let $P(D_i)$ be the power set of D_i. *An IF rough relation R* is a subset of the product set $P(D_1) \times P(D_2) \times \ldots P(D_m) \times D_{[\mu,\nu]}$, where $D_{[\mu,\nu]}$ is the domain for membership and non-membership value of the closed interval [0,1] and $P(D_i) = P(D_i) - \phi$. *Example* For a specific relation, R, membership and non-membership are determined semantically. Given that D_1 is the set of names of patients, D_2 is the set 'description' attributes of Blood pressure, (Anil, Very Severe, [1,0]) ; (Gopal, {Mild, Severe}, [0.6, 0.2]) are elements of the relation R (Patient Name, Blood pressure, [μ, ν]).

Definition 2. Let $t_i = (d_{i1}, d_{i2}, \ldots, d_{im}, d_{i[\mu,\nu]})$ be an IF rough tuple. An interpretation of t_i is a tuple $\alpha = (a_1, a_2, \ldots, a_{m,n}; a_{[\mu,\nu]})$ where $a_j \in d_{ij}$ for each domain D_j.

Definition 3. Two tuples t_i and t_j are redundant if and only if they possess an identical interpretation.

Definition 4. Two sub-tuples X = (d_{x1}, d_{x2}, ..., $d_{xm,}$ $d_{x[\mu,\nu]}$) and Y = (d_{y1}, d_{y2}, ..., $d_{ym,}$ $d_{y[\mu,\nu]}$) are roughly-redundant, R if for some [p] $\subseteq$ [d_{xj}] and [q] $\subseteq$ [d_{yj}], [p] = [q] for all j = 1,2, ..., m.

2.5 Set Operations and Relational Operations [9]

The set operations and relational operations on subsets of tuples are shown below.

IF Rough Difference: The IF rough difference between T_1 and T_2 is an IF rough relation T = T_1- T_2

$\{t(d_1,..,d_n,[\mu_i,\nu_i]) \in \underline{RT}_1 : t(d_1,..,d_n,[\mu_i,\nu_i]) \notin \underline{RT}_2\} \cup t(d_1,..,d_n,[\mu_i,\nu_i]) \in \overline{RT}_1 \; and \{t(d_1,..,d_n,[\mu_i,\nu_i])$
$\in \overline{RT}_2 \; if \; \mu_i > \mu_j\} \cup \{t(d_1,..,d_n,[\mu_i,\nu_i]) \in \overline{RT}_1 \; and \; t(d_1,..,d_n,[\mu_i,\nu_i]) \notin \overline{RT}_2 \; if \; \mu_i = \mu_j \; and if \; \nu_i < \nu_j\}.$

IF Rough Union: The IF rough union between T_1 and T_2 is an IF rough relation T= $T_1 \cup T_2$, where $\underline{RT} = \{t : t \in \underline{RT}_1 \cup \underline{RT}_2\} and \quad \mu_{\underline{RT}}(t) = MAX[\mu_{\underline{RT}_1}(t), \mu_{\underline{RT}_2}(t)],$

$and \quad if \; \mu_{\underline{RT}_1}(t) = \mu_{\underline{RT}_2}(t), \nu_{\underline{RT}}(t) = MIN[\nu_{\underline{RT}_1}(t), \nu_{\underline{RT}_2}(t)], \overline{RT} = \{t : t \in \overline{RT}_1 \cup \overline{RT}_2\} \quad and$
$\mu_{\overline{RT}}(t) = MAX[\mu_{\overline{RT}_1}(t), \mu_{\overline{RT}_2}(t)], and \quad if \; \mu_{\overline{RT}_1}(t) = \mu_{\overline{RT}_2}(t), \nu_{\overline{RT}}(t) = MIN[\nu_{\overline{RT}_1}(t), \nu_{\overline{RT}_2}(t)].$

IF Rough Intersection: The IF rough intersection between T_1 and T_2 is an IF rough relation T= $T_1 \cap T_2$, where $\underline{RT} = \{t : t \in \underline{RT}_1 \cap \underline{RT}_2\} and \quad \mu_{\underline{RT}}(t) = MIN[\mu_{\underline{RT}_1}(t), \mu_{\underline{RT}_2}(t)],$

$and \; if \; \mu_{\underline{RT}_1}(t) = \mu_{\underline{RT}_2}(t), \nu_{\underline{RT}}(t) = MAX[\nu_{\underline{RT}_1}(t), \nu_{\underline{RT}_2}(t)], \overline{RT} = \{t : t \in \overline{RT}_1 \cap \overline{RT}_2\} \quad and$
$\mu_{\overline{RT}}(t) = MIN[\mu_{\overline{RT}_1}(t), \mu_{\overline{RT}_2}(t)], and \; if \; \mu_{\overline{RT}_1}(t) = \mu_{\overline{RT}_2}(t), \nu_{\overline{RT}}(t) = MAX[\nu_{\overline{RT}_1}(t), \nu_{\overline{RT}_2}(t)].$

IF Rough Select: The IF rough selection $\sigma_{A=a}(x)$, of tuples from T_1 is an IF rough relation T_2 having the same schema as T_1 and where

$\underline{RT}_2 = \{t \in T_1 : \cup_i [a_i] = \cup_j [b_j]\}, \overline{RT}_2 = \{t \in T_1 : \cup_i [a_i] \subseteq \cup_j [b_j]\}, \quad a_i, b_j \in dom(A)$

IF Rough Project: The IF rough projection of T_1 onto Y, $\pi_Y(T_1)$, is an IF rough relation T_2 with schema T_2(Y) where $T_2(Y) = \{t(Y) : t \in T_1\}$.

IF Rough Join: The IF rough join, T_1 join T_2, of two relations T_1 and T_2, is a relation

$T(C_1, C_2, .., C_{m+n}) \quad where \quad T = \{t : \exists t_{T_1} \in T_1, t_{T_2} \in T_2 \quad for \quad t_{T_1} = t(X), t_{T_2} = t(Y) \quad and$
$t_{T_1}(X \cap Y) = t_{T_2}(X \cap Y), \mu = 1, \nu = 0 \; for \; \underline{RT} \; and \quad t_{T_1}(X \cap Y) \subseteq t_{T_2}(X \cap Y), \quad or$
$t_{T_2}(X \cap Y) \subseteq t_{T_1}(X \cap Y), \mu = MIN(\mu_{T_1}, \mu_{T2}) \; and \; if \quad \mu_{T_1} = \mu_{T_2}, \nu = MAX(\nu_{T_1}, \nu_{T_2}), \; for \; \overline{RT}.$

3 Application: The Diabetic Patient Database

3.1 Extraction of Information

Diabetes Mellitus has become the most common chronic diseases among people of Tripura. The information is recorded by investigators from G.B. Pant hospital, I.G.M. hospital and private medical practitioners of Agartala. A proforma was filled up with the required information after a verbal interview of the patients and from the documents of the patient who were above 35 years of age. It is possible that some of the information may be uncertain or unavailable. Perhaps spoken words may be

unclear. Sometimes it is possible to make a ‘‘good guess’’ at the uncertain part, at the same time acknowledging the fact that uncertainty is present.

Another problem arises when there are more than one investigators recording data about patients. If there are inconsistencies in categorizing data, we may not know which observation, if either, is more correct than the other. Through, the IF rough relational database model, we can incorporate all of these types of uncertainty, rather than discarding the data as invalid.

A database system was designed based on a case study of diabetic patients via Entity Relationship Diagram (ERD), relational Model and Implementation in SQL server. The ERD is outlined below.

3.2 Database Design

Entity Relationship (ER) Diagram

First, we design a database by using some types of semantic model and create an IF rough ER diagram. There are five tables namely Patients, Personal and family history(PFH), Demographic, General physical examination(GPE) and Laboratory investigation(LI) with different attributes. Attributes that allow equivalent values are denoted by * . The ER diagram of our database is shown in Fig.1.

IF Rough Data Definition Language

We introduce an IF rough data definition language (IFRDDL) to define the IF rough relations and indiscernibility relation. First, we create a base table by IFRDDL command. A simple table containing three items of information about patients is formed [Table 1]. The table is named “Patients” and stores information about each

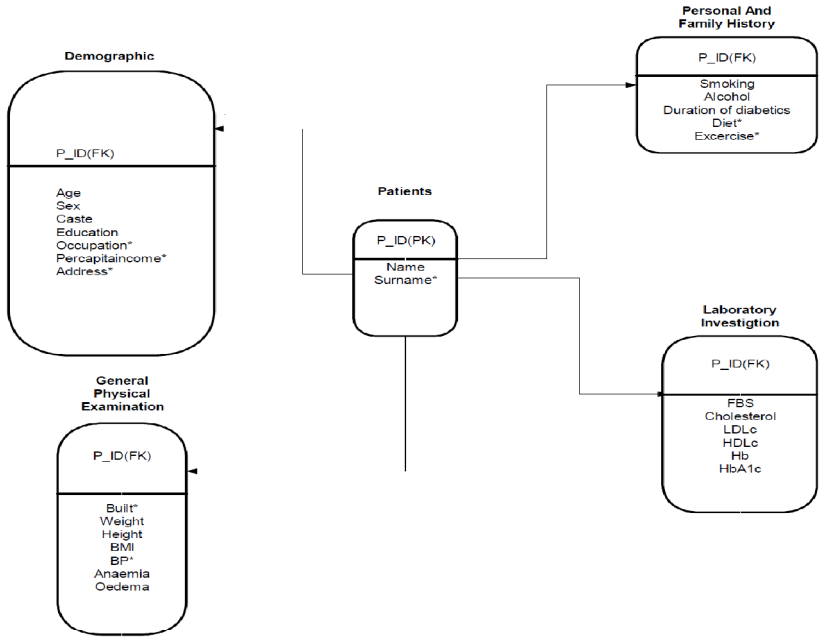


Fig. 1. An IF rough E-R diagram for the Diabetic Patients Database

patient's ID number, first name and surname. It also contains an attributes called [MU, NMU] which draws values from the range [0, 1]. Additionally, we specify whether or not we can allow indiscernibility values for each attribute. This is defined by including ''IND'' along with the attribute line of the table definition.

```
IFRCREATE TABLE PATIENTS (
PID                    DECIMAL(3),
FIRST NAME             CHAR(25),
SURNAME                CHAR(25)        IND,
MU, NMU                CHAR(10),
PRIMARY KEY (PID));
```

PATIENTS (Table 1): The patient table has attributes PID, First Name, Surname and MU, NMU. The membership and non-membership value is to manage the attribute 'Surname'.

Similarly, **PERSONAL AND FAMILY HISTORY[Table-3], DEMOGRAPHIC [Table-4], GENERAL PHYSICAL EXAMINATION[Table-5], LABORATORY INVESTIGATION[Table-6]** and **INDISCERNIBILITY[Table-2]** tables are created**.**

Now, the database schema has been defined where actual data is stored. Data values for all attributes including the membership and non-membership value are inserted into the specified relation. If a value for [MU,NMU] is not included, it is automatically assigned a default value from [1,0]. This saves considerable data entry time.

The IF rough counterpart to SQL's INSERT is **IFRINSERT**:

```
IFRINSERT
INTO DEMOGRAPHIC
VALUES (2, 44, M, Govt.Service,{Low, Normal},{Shamali Bagar,
Abhoynagar}, [0.4, 0.5] );
```

The IFRINSERT command is used to enter tuples in the INDISCERNIBILITY relation. The IFRINDISCERNIBILITY relation is a special one, used only for the grouping of similar attribute values into equivalence classes. Therefore, we introduce some new commands that will facilitate the creation of classes of equivalent values. All membership and non-membership values for tuples in this relation are automatically set to [1,0]. The indiscernibility identifier PID serves to specify values that are indiscernible. The actual value of PID is irrelevant, as long as all tuples belonging to a given class have identical values for the attribute PID. Therefore, it is needed to specify the values grouped into a class, and the system can set up the tuples in the INDISCERNIBILITY relation.

Special Commands

1	IFRCLASS	To create a new equivalence class
2	IFRREMOVE CLASS	To remove a value from a class.
3	IFRDELETE CLASS	To delete an entire class including all its member.
4	IFR ADD CLASS	To add a value to a class, rather than create a new class.
5	IFR DELETE	To delete tuples.
6	IFR UPDATE	To update tuples
7	IFR DROP	To drop Tables

4 Implementation in SQL Server

SQL server is software where we can store huge amount of information via a database. In this server we can execute the queries conveniently by SQL query language.

SQL queries for the IF Rough Relational Database

Based on our data definition language for the IF rough relational database on SQL, we present some SQL like queries to our diabetic database.

Question1: List all BP categories from General Physical Examination.

IFR Query: SELECT (GPE.BP)
FROM GPE

Output1:

BP	MU	NMU
high normal	1.0	0.0
normal	1.0	0.0
mild, severe	0.3	0.5
low normal, low	0.8	0.1
high normal, high	0.5	0.3
high normal, very severe	0.4	0.4
mild	1.0	0.0
high normal, slightly above normal	0.8	0.1
high normal, normal	0.7	0.3
normal, low normal	0.8	0.1
low normal	1.0	0.0
mild, severe	0.5	0.3

Question 2: Find names and surname of all female patients who are house wife and taken non-veg.

IFR Query: SELECT (Demographic.PID),(Patients.Name),(Patients.Surname), (Demographic.sex),(Demographic.Occupation),(PFH.diet)
FROM PFH, Demographic, Patients
WHERE ((demographic.sex= 'F')and (Demographic.Occupation ='House wife') and (PFH.Diet ='Non-veg')) and (PFH.PID =Demographic.PID) and (PFH.PID = Patients.PID));

Output 2:

PID	Name	Surname	sex	occupation	diet	MU	NMU
7	Laxmi Rani	Roy,Biswas	F	House Wife	Non-veg, Veg	0.2	0.6
19	Malati	Deb	F	House Wife	Non-veg, Veg	0.5	0.4
58	Nakul Rani	Deb	F	House Wife	Non-veg, Veg	0.5	0.4
117	Rani Bala	Dey	F	House Wife	Non-veg, Veg	0.4	0.5

This IF rough relational database model can help the medical experts for the queries of impreciseness on diet and surname as shown above.

5 Conclusion

This paper concerns the modeling of imprecision and vagueness in diabetic databases of Tripura through the IF rough relational database model which is easy to understand and to use. The IF rough E-R diagram is also shown here. It is more efficient model

of the uncertainty through the use of indiscernibility and membership and non-membership values. Finally, this IF rough relational model can serve the better purpose of medical experts.

References

1. Atanassov, K.: Intuitionistic Fuzzy Sets. Fuzzy Sets and Systems 20, 87–96 (1986)
2. Bagai, R.S.: A paraconsistent relational data model. International Journal of Computer Mathematics 55 (1995)
3. Beaubouef, T., Petry, F.E.: Uncertainty modeling for database design using intuitionistic and rough set theory, Jour. Intelligent and Fuzzy Systems 20(3), 105–117 (2009)
4. Beaubouef, T., Petry, F.E.: Intuitionistic rough sets applied to databases. Transactions on Rough Sets 7, 26–30 (2007)
5. Beaubouef, T., Petry, F.E.: Extension of the relational database and its algebra with rough set techniques. Comput. Intell. 11, 233–245 (1995)
6. Beaubouef, T., Petry, F.E.: Fuzzy rough set techniques for uncertainty processing in a relational database. Intl. Journal of Intelligent Systems 15, 389–424 (2000)
7. Breault, J.L.: Data mining diabetic databases:Are rough sets a useful addition? Computing Science and Statistics 34 (2001)
8. Codd, E.F.: A relational model of data for large shared data banks. Comm. ACM 13, 377–387 (1970)
9. Gangwal, C., Bhaumik, R.N.: Intuitionistic fuzzy rough relational database model. International Journal of Database Theory and Application 5(3), 91–102 (2012)
10. Michel, C., Beguin, C.: Using a database to query for diabetes mellitus. Stud. Health Technol. Inform. 14, 178–182 (1994)
11. Pawlak, Z.: Rough sets Internat. Internat. J. Comput. Inform. Sci. 11, 341–356 (1982)
12. Wong, E.: A statistical approach to incomplete information in database systems. ACM Trans. on Database Systems 7, 470–488 (1982)
13. Zadeh, L.A.: Fuzzy sets. Information Control 18, 338–353 (1965)

Appendices

Table 1. Patients

PID	Name	Surname	MU	NMU
1	Saroj	Ambuly	1	0
2	Bishnu Pada	Saha	1	0
3	Rekho	Bhawmik	1	0
…				
199	Narayan	Debnath	1	0
200	Gori	{Chakrbroty, Bhattacharjee}	0.5	0.5

Table 2. Indiscernibility

IndClass	Description
1	Low
1	Below normal
1	Slightly below normal
2	Slightly above normal
2	Mild
3	High
3	Moderate
4	Very high
4	Severe
4	Very severe

Table 3. Personal and Family History

PID	Smoking	Alcohol	Duration	Diet	Exercise	MU	NMU
1	Yes	No	Small	Non-veg	Yes	1	0
2	No	No	Small	Non-veg	Yes	1	0
3	No	No	Small	Non-veg	No	1	0
…							
199	No	No	Small	Non-veg	Yes	1	0
200	No	No	Small	Non-veg	Yes	1	0

Table 4. Demographic

PID	Age	Sex	Occupation	Income	Address	MU	NMU
1	58	M	Retired	High	Palace compound,Agartala	1	0
2	44	M	Govt.service	{Low,Normal}	{ShamaliBagar, Aboynagar}	0.4	0.5
3	43	F	Housewife	High	Krishnanagar	1	0
…							
199	52	M	Govt.service	High	{Gourabasti,Jogendranagar}	0.6	0.3
200	36	F	Housewife	Normal	Krishnanagar	1	0

Table 5. General Physical Examination

PID	Built	BMI	BP	Anaemia	Oedema	MU	NMU
1	Average	Normal	High normal	No	No	1	0
2	Average	Normal	Normal	Yes	Yes	1	0
3	Average	Obesity	{Normal,high normal}	No	No	0.7	0.2
…..							
199	Average	Normal	High normal	No	No	1	0
200	Average	Normal	Normal	No	No	1	0

Table 6. Laboratory Investigation

PID	FBS	Cholesterol	LDL	HDL	Hb%	HbA1c	MU	NMU
1	High	Normal	85	42	Normal	Low	1	0
2	High	Normal	101	50	Normal	Low	1	0
3	High	Normal	77	41	Low	High	1	0
…								
199	Normal	Normal	61	30	Normal	High	1	0
200	High	Medium	122	52	Normal	Low	1	0

Table 7. Summary of Attributes

Attributes	Description	Attributes	Description
History of smoking	Yes No	Anaemia	Yes No
History of alcohol	Yes No	Oedema	Yes No
Duration of Diabetics (Years)	Small :<5 Large: <10 Very large: >10	HbA1c(%)	7 = Low 7-8 = Normal >8 = High
Diet	Veg Non-Veg	Hemoglobin (Hb %)	<12: Low >12 :Normal
Exercise	Yes No	Built	Average Fatty Very fatty Lean
Sex	M= male F = female		
Caste	general Sched. Caste Schedule tribes others	Body Mass Index= Weight in kilograms /(Height in Meters x Height in Meters)	Under weight = <15 Normal= 15-23 Over weight= 23-25 Obesity= >25
Education	Illiterate Primary Secondary Graduate Master degree Technical	Blood Pressure(in mm of Hg) Diastolic-Systolic	Diastolic / Systolic Very severe= >110 / >185 Severe= 100-110 /165-185 Mild= 90-99 / 140-164 High normal= 80-89/125-139 Normal= 70-79 /105-124 Low normal= 60-69/ 90-104 Low= 50-59 / 70-89 Very low = 40-49 / 55-69
Occupation	Govt.service Private serv. Business Retired Housewife	Lipid Profile Total cholesterol,[LDL-C, HDL-C and TGs]	Normal= <120/<80 Medium= 120-129/80-85 High= 130-159/86-99 Very high = ≥160/≥100
Income - Per capita (Rs.)	Low= 1000-2000 Normal= 2000-4000 High= >4000	Fasting Blood Sugar (mg/dl)	Normal = <100 High = 101-150 Very High = >150

Matrix Representation of Parameterised Fuzzy Petri Nets

Zbigniew Suraj

Institute of Computer Science, University of Rzeszów, Poland
zbigniew.suraj@ur.edu.pl

Abstract. There are many representations of Petri nets. One of the most known and applicable representations is the matrix one. Such representation enables an easy implementation of different kinds of concurrent algorithms for Petri nets. The aim of this paper is to present a matrix representation of parameterised fuzzy Petri nets. Recently, this net model has been proposed as a new class of fuzzy Petri nets. It extends, in a natural way, the fuzzy Petri nets by introducing two parameterised families of sums and products.

Keywords: matrix representation, parameterised fuzzy Petri nets, knowledge representation, approximate reasoning, decision support systems.

1 Introduction

Petri nets are widely used in both theoretical analysis and practical modelling of concurrent systems. Recently, Petri nets have been gaining a growing interest among researchers engaged in Artificial Intelligence field due to its adequacy for knowledge representation and the approximate reasoning process [1],[3],[4],[10],[11],[12],[14],[15]. Several extensions have been proposed for Petri nets in the last four decades improving different aspects: hierarchical nets, high level nets, temporal nets [2],[5],[6],[9]. An additional improvement comes with the investigation of the connection between logic and Petri nets. Logical propositions can be associated with Petri nets allowing for logical reasonings about the modelled system and its behaviour. In all these Petri net models, though, only well-known pieces of information are taken into account. The collected book [1] focuses on the current state-of-the-art in the use of fuzziness in Petri nets.

Petri nets can be represented in many ways. One of the most known and applicable representations of Petri nets is the matrix one. The development of the matrix Petri net theory provides a useful tool for dealing with many problems in the analysis of Petri nets. Matrix representation of classical Petri nets has been presented earlier in the literature, e.g. [2],[8],[9]. The matrix approach to the representation and analysis of extended fuzzy Petri nets presented in [3] seems promising but also has some drawbacks concerning, in particular, the matrix operations which are very complicated and unnatural.

The aim of this paper is to present the matrix representation of parameterised fuzzy Petri nets ($PFPNs$) introduced in the paper [10]. The application

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 200–207, 2013.

of $PFPNs$ in knowledge representation and fuzzy reasoning process has been presented in [11]. $PFPNs$ extend the existing fuzzy Petri nets [1] by introducing two parameterised families of sums and products, which are supposed to function as substitute for the *min* and *max* operators appearing in the classical fuzzy Petri nets. $PFPNs$ are more flexible than the traditional ones, as in the former class the user has the chance to define the parameterised input/output operators. The choice of suitable operators for a given fuzzy reasoning process and the speed of reasoning process are very important, especially in real-time decision support systems.

The matrix representation is one of the most convenient representations of Petri nets in the domain of modern computer programming. Moreover, there exist the automated computation systems, e.g. Matlab, Mathematica, Maple, which make it possible to solve many computing problems, especially those with matrix and vector formulations. Our approach enables us to carry out a fuzzy reasoning process using, for example, the computation systems mentioned above (cf. [3]). However, this issue is not considered in the paper. Here, we present only formal background for the matrix representation of $PFPNs$. Using the matrix representation, we can view any $PFPN$ as a collection of matrices and vectors whose components are real numbers, strings or triples of real functions. However, its behaviour can be characterised by means of simple matrix equations or inequalities.

There are several possibilities for increasing the usefulness of $PFPNs$. They concern different ways of a net operating. In this paper we assume that a $PFPN$ can operate in two main modes: *single firings* or *steps*. Steps are a generalisation of net work in the mode of single firings. The net work in the mode of steps can be treated as a simultaneous firing of a selected set of enabled transitions or a single firing of them in any order.

The proposed matrix representation of $PFPNs$ allows to implement parallel firing of independent transitions in one reasoning step easily by using natural operations on matrices. Moreover, it is significantly simpler than the matrix representation provided for the extended fuzzy Petri nets in [3]. The matrix net representation discussed here can also be applied immediately to the classical fuzzy Petri nets as well as to the extended fuzzy Petri nets in order to obtain both more convenient representation of these nets and quite natural operations on matrices.

The paper is organised as follows. In Sect. 2 we give a brief introduction to *PFPNs*. Sect. 3 describes the matrix representation of $PFPNs$. It is the main contribution of the paper. In Sect. 4 we provide some conclusions related to our approach and further investigations.

2 Preliminaries

Basic operations in the classical fuzzy set theory such as the intersection and the union, are defined by using the minimum and maximum operators. However, some other definitions of these operations are often employed, too. In particular,

for the intersection and the union parameterised families of sums and products are used. As the parameterised ones are also used for defining *PFPNs*, we give the example of parameterised family of sum $S(a,b,v)$ and product $T(a,b,v)$ used in this paper, where: $S(a,b,v) = \frac{a+b-(2-v)*a*b}{1-(1-v)*a*b}$, $T(a,b,v) = \frac{a*b}{v+(1-v)*(a+b-a*b)}$, and $v \in (0,\infty)$.

For more details about parameterised families of sums and products one shall refer to [7].

A parameterised fuzzy Petri net (*PFP*-net) is a tuple $N = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M0)$ where: $P = \{p_1, \ldots, p_n\}$ is a finite set of *places*; $T = \{t_1, \ldots, t_m\}$ is a finite set of *transitions*; $S = \{s_1, \ldots, s_n\}$ is a finite set of *statements*; $I : T \to 2^P$ is the *input function*; $O : T \to 2^P$ is the *output function*; $\alpha : P \to S$ is the *statement binding function*; $\beta : T \to [0,1]$ is the *truth degree function*; $\gamma : T \to [0,1]$ is the *threshold function*; Op is a finite set of *parameterised operators*; the sets P, T, S, Op are pairwise disjoint and $\mathrm{card}(P) = \mathrm{card}(S)$; $\delta : T \to Op \times Op \times Op$ is the *operator binding function*; $M0 : P \to [0,1]$ is the *initial marking*.

We say that the place p is an *input place* of a transition t if $p \in I(t)$. Analogously, we say that the place p' is an *output place* of a transition t if $p' \in O(t)$.

Let N be a *PFP*-net. A *marking* of N is a function $M : P \to [0,1]$.

Example 1. Consider a *PFP*-net such that: $P = \{p_1, \ldots, p_5\}$; $T = \{t_1, t_2\}$; $S = \{s_1, \ldots, s_5\}$; $I(t_1) = \{p_1, p_2\}$, $I(t_2) = \{p_2, p_3\}$; $O(t_1) = \{p_4\}$, $O(t_2) = \{p_5\}$; $\alpha(p_i) = s_i$ for $i = 1, \ldots, 5$; $\beta(t_1) = 0.7$, $\beta(t_2) = 0.8$; $\gamma(t_1) = 0.4$, $\gamma(t_2) = 0.3$; $Op = \{S(.), T(.)\}$; $\delta(t_1) = (S(.), T(.), S(.))$, $\delta(t_2) = (T(.), T(.), S(.))$; and $M0 = (0.6, 0.4, 0.7, 0, 0)$. If we take parameterised families of sums and products mentioned above and a parameter value $v = 1$, then $S(a,b,1) = s_P(a,b) = a+b-a*b$ (the probabilistic sum) and $T(a,b,1) = t_P(a,b) = a*b$ (the algebraic product).

For more detailed information about *PFPNs* the reader is referred to [10].

3 Matrix Representation

Using the matrix representation, we can view a structure of any $PFPN$ as a collection of matrices and vectors whose components are real numbers, strings or triples of real functions. However, its dynamics can be characterised by means of simple matrix equations or inequalities. Before introducing the description we recall some concepts and auxiliary notation.

3.1 Basic Concepts and Notation

For any vector function $g, g' : X \to \Re$, where X is a nonempty set, and $\Re$ is the set of all real numbers, we denote in a classical way: $g + g'$ (the sum), $g - g'$ (the difference), $g = g'$ (the equality relation), and $g \leq g'$ (the inequality relation). By Y^T we denote the transposed matrix of a matrix Y.

Let $N = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M0)$ be a *PFP*-net, $t \in T$, $I(t) = \{p_{i1}, \ldots, p_{ik}\}$ be a set of input places of a transition t, $\beta(t), \gamma(t) \in [0,1]$, M be a marking of N, and v be a parameter value for a parameterised family of sums and

products. Moreover, let In^v be an input parameterised operator belonging to one of the classes: parameterised sums or products, and Out_1^v, Out_2^v be output parameterised operators belonging to the class of parameterised products and the class of parameterised sums, respectively. These three operators correspond to a transition t.

In order to define enabling and firing rules for a PFP-net by means of matrices, at first we introduce four auxiliary one column n-vector functions t^γ, t^-, t^0, t^+ (vectors with n-coordinates) as follows:

$t^\gamma : P \to [0,1]$ for $p \in P$, and $t^\gamma(p) = \begin{cases} \gamma(t) & \text{for } p \in I(t), \\ 0 & \text{otherwise.} \end{cases}$

The function t^γ attaches a threshold value, i.e., the number $\gamma(t)$, to each input place of a transition t. It is called *a selecting transition function.*

$t^- : P \to [0,1]$ for $p \in P$, and $t^-(p) = \begin{cases} In^v(M(p_{i1}), \ldots, M(p_{ik})) & \text{for } p \in I(t), \\ 0 & \text{otherwise.} \end{cases}$

The function t^- describes aggregating tokens from the input places of a transition t. It is called *an aggregating token function.*

$t^0 : P \to [0,1]$ for $p \in P$, and $t^0(p) = \begin{cases} M(p) & \text{for } p \in I(t), \\ 0 & \text{otherwise.} \end{cases}$

The function t^0 describes memorizing tokens residing in the input places of a transition t by a marking M. It is called *a memorizing token function.*

$t^+ : P \to [0,1]$ for $p \in P$, and

$$t^+(p) = \begin{cases} Out_1^v(In^v(M(p_{i1}), \ldots, M(p_{ik}), \beta(t)) & \text{if } p \in O(t), \\ 0 & \text{otherwise.} \end{cases}$$

The function t^+ describes transferring tokens to the output places of a transition t after its firing by a marking M. It is called *a transferring token function.*

It is worth to observe that values of the vector function t^γ do not depend on actual marking M of a net N, whereas the values of three remaining functions do.

3.2 Structure

The alternative for the definition of $PFPNs$ as a structure $N = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M0)$ is defining a tuple of matrices $(N_{in}, N_{out}, S_A, S_B, S_C, S_D, M0)$ representing input function I, output function O, statement binding function α, truth degree function β, threshold function γ, operator binding function δ and initial marking $M0$, respectively. Each of the two first matrices, i.e., N_{in}, N_{out}, consists of n rows (each row corresponds to one place $p \in P$, $n = card(P)$) and m columns (each column corresponds to one transition $t \in T$, $m = card(T)$). The elements of these matrices are defined as follows: *an input incidence matrix* $N_{in}(p,t) = t^-(p)$, *an output incidence matrix* $N_{out}(p,t) = t^+(p)$. The remaining five matrices are one-row vectors. They are defined as follows: one row n-vector S_A of statements such that $S_A(p) = \alpha(p)$ for $p \in P$; one row m-vector S_B of truth degree values such that $S_B(t) = \beta(t)$ for $t \in T$; one row m-vector S_C of threshold values such that $S_C(t) = \gamma(t)$ for $t \in T$; one row m-vector S_D of triples of parameterised operators such that $S_D(t) = \delta(t)$ for $t \in T$; one row n-vector $M0$ of initial marking.

We assume that sets P and T have been ordered in the following way: $P = \{p_1, \ldots, p_n\}$ and $T = \{t_1, \ldots, t_m\}$.

For $PFPNs$ one can determine all components of a structure $N = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M0)$ in a one-to-one way on the basis of matrices $N_{in}, N_{out}, S_A, S_B, S_C, S_D, M0$. This means that both representations of $PFPNs$ (set-theoretical and matrix) are equivalent in this sense.

3.3 Dynamics

There exist several possibilities for increasing the usefulness of Petri nets. They concern different ways of a net operating. A Petri net can operate in two main modes: *single firings* or *steps*. Now, we define these modes for the $PFPNs$ using matrix notation.

Single Firings. Let Op be the set of parameterised operators defined as in the definition of the PFP-net, and $In^v, Out_1^v, Out_2^v \in Op$ be the parameterised input/output operators with a given parameter value v. Moreover, let $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ be the n-dimensional vectors, where $x_i, y_i \in [0, 1]$. By $\mathbf{op}_p(X, Y)$ we denote the n-dimensional vector $Z = (z_1, \ldots, z_n)$ such that $z_i = op_p(x_i, y_i)$, where $p \in \{In^v, Out_1^v, Out_2^v\}$, $i = 1, \ldots, n$. In other words, the components of the vector Z are the result of parameterised input/output operations for the corresponding components of the vectors X and Y.

Let $N = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M0)$ be a PFP-net, $t \in T$, and t^γ, t^-, t^0, t^+ be vectors defined in subsection 3.1, corresponding to a transition t. Moreover, let M be a marking of N with a parameter value v.

A transition $t \in T$ is *enabled* for marking M and a parameter value v, if the value of parameterised input operator In^v for the transition t is greater than, or equal to, the value of threshold function γ corresponding to t, i.e., $t^- \geq t^\gamma$.

Mode 1. If M with a parameter value v is a marking of N enabling a transition t and M' is the marking derived from M by firing t, then

$$M' = \begin{cases} \mathbf{op}_{Out_2^v}(M^T - t^0, t^+) & \text{if } t \text{ fires by } M, \\ undefined & \text{otherwise.} \end{cases}$$

In this mode, a procedure for computing the marking M' is as follows. If the transition t is enabled by M with a parameter value v, then at first the difference of the two vectors M^T and t^0 is computed and then the output operation $\mathbf{op}_{Out_2^v}$ for the value of the difference and the vector t^+ is determined (the first condition from M' definition). In other case, i.e., if the transition t is not enabled for M with a parameter value v, a new marking M' is not determined (the second condition from M' definition).

Mode 2. If M with a parameter value v is a marking of N enabling a transition t and M' is the marking derived from M by firing t, then

$$M' = \begin{cases} \mathbf{op}_{Out_2^v}(M^T, t^+) & \text{if } t \text{ fires by } M, \\ undefined & \text{otherwise.} \end{cases}$$

The main difference in the definition of the marking M' presented above (*Mode 2*) concerns input places of the fired transition t. In *Mode 1* numbers are

removed from all input places of the fired transition t (*cf.* the first definition condition of *Mode 1*), whereas in *Mode 2* all numbers are copied from input places of the fired transition t (the first definition condition of *Mode 2*).

We say that t fires from a marking M with a parameter value v to M'.

Example 2. Let $N = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M0)$ be a *PFP*-net from Example 1. Describe this net by means of matrices: N_{in}, N_{out}, S_A, S_B, S_C, S_D, $M0$, where: $N_{in} = ((0.76, 0), (0.76, 0.28), (0, 0.28), (0, 0), (0, 0))$, $N_{out} = ((0, 0), (0, 0), (0, 0), (0.53, 0), (0, 0.22))$, $S_A = (s_1, s_2, s_3, s_4, s_5)$, $S_B = (0.7, 0.8)$, $S_C = (0.4, 0.3)$, $S_D = ((s_P, t_P, s_P), (t_P, t_P, s_P))$, and $M0 = (0.6, 0.4, 0.7, 0, 0)$. The vectors ${t_1}^{\gamma}$, ${t_2}^{\gamma}$, ${t_1}^{-}$, ${t_2}^{-}$, ${t_1}^{0}$, ${t_2}^{0}$, ${t_1}^{+}$, ${t_2}^{+}$ have the form by the initial marking $M0$ as follows: ${t_1}^{\gamma} = (0.4, 0.4, 0, 0, 0)^T$, ${t_2}^{\gamma} = (0, 0.3, 0.3, 0, 0)^T$, ${t_1}^{-} = (0.76, 0.76, 0, 0, 0)^T$, ${t_2}^{-} = (0, 0.28, 0.28, 0, 0)^T$, ${t_1}^{0} = (0.6, 0.4, 0, 0, 0)^T$, ${t_2}^{0} = (0, 0.4, 0.7, 0, 0)^T$, ${t_1}^{+} = (0, 0, 0, 0.53, 0)^T$, ${t_2}^{+} = (0, 0, 0, 0, 0.22)^T$. It is easy to see that the transition t_1 is enabled by the initial marking $M0$, but the transition t_2 is not. This follows from the fact that: ${t_1}^{-} \geq {t_1}^{\gamma}$ and ${t_2}^{-} < {t_2}^{\gamma}$. After firing the transition t_1 in *Mode 1* by the marking $M0$ we obtain a new marking $M' = \mathbf{s_P}(M0^T - {t_1}^{0}, {t_1}^{+}) = (0, 0, 0.7, 0.53, 0)$.

Remark. It is also worth pointing out that the matrix representation of $PFPNs$ proposed in this paper is significantly simpler than the one provided for the extended fuzzy Petri nets in [3] and more general than that presented in [13].

Steps. Steps are a generalisation of net work in the mode of single firings. In the paper we consider two kinds of steps: *simple* and *generalised*. The definitions of these concepts are presented below. The net work in the mode of steps can be treated as a simultaneous firing of a selected set of enabled transitions or a single firing of them in any order.

Let $N = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M0)$ be a *PFP*-net, $U \subseteq T$ and M be a marking of N with a parameter value v.

A nonempty set U of transitions is called *a simple step* by a marking M (regarding to transitions concurrency) with a parameter value v if they are enabled by M and pair-wise concurrent (i.e., there are no transitions which have joint input and output places).

A nonempty set U of transitions is called *a generalised step* (or simply *a step*) by a marking M with a parameter value v if they are enabled by M and can be fired simultaneously.

Remark. In the definition of a step we do not demand the concurrency of transitions with a step U, but we demand only the possibility of its simultaneous firing. This means that if the sets of input places and output places for transitions belonging to the step U are not pairwise disjoint, thus simultaneous firing of those transitions will be possible only in *Mode 2*. This definition is a natural generalisation of the simple step definition.

Before formulating a definition of next marking after firing a (simple) step we still need additional concepts and notation.

Let $N = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M0)$ be a *PFP*-net, $U = \{t_{i1}, \ldots, t_{ik}\} \subseteq T$ be a step (a simple step), and let ${t_{ij}}^{\gamma}$ (${t_{ij}}^{-}$, ${t_{ij}}^{0}$, t_{ij}^{+}) be a selecting transition (an aggregating token, a memorizing token, a transferring token) function

corresponding to the transition t_{ij}, $j = 1, \ldots, k$. By U^{γ}, U^{-}, U^{0}, U^{+} we denote vectors such that: $U^{\gamma} = \mathbf{max}({t_{i1}}^{\gamma}, \ldots, {t_{ik}}^{\gamma})$, $U^{-} = \mathbf{max}({t_{i1}}^{-}, \ldots, {t_{ik}}^{-})$, $U^{0} = \mathbf{max}({t_{i1}}^{0}, \ldots, {t_{ik}}^{0})$, $U^{+} = \mathbf{op}_{Out_2^v}({t_{i1}}^{+}, \ldots, {t_{ik}}^{+})$, where Out_2^v is any second output operator defined in *PFP*-net definition. In particular situation this output operator can be replaced by the generalised maximum operator **max**, i.e., the maximum operation regarded to vectors.

Let $N = (P, T, S, I, O, \alpha, \beta, \gamma, Op, \delta, M0)$ be a *PFP*-net, $U \subseteq T$ be a step (a simple step), U^0 be a step memorizing token function corresponding to transitions from U, U^+ be a step transferring token function corresponding to transitions from U.

Mode 1. If M is a marking of N with a parameter value v enabling a step U and $M^{'}$ the marking derived from M by firing transitions from U, then

$$M^{'} = \begin{cases} \mathbf{op}_{Out_2^v}(M^T - U^0, U^+) & \text{if } U \text{ fires by } M, \\ undefined & \text{otherwise.} \end{cases}$$

In this mode, a procedure for computing the marking $M^{'}$ is similar to appropriate procedure corresponding to $PFPNs$ and *Mode 1* presented above. The difference is that present procedure uses steps instead of single transitions. Remaining stages of the procedure are analogous to the previous procedure concerning *Mode 1*.

Mode 2. If M is a marking of N with a parameter value v enabling a step U and $M^{'}$ the marking derived from M by firing transitions from U, then

$$M^{'} = \begin{cases} \mathbf{op}_{Out_2^v}(M^T, U^+) & \text{if } U \text{ fires by } M, \\ undefined & \text{otherwise.} \end{cases}$$

The difference in the definitions of marking $M^{'}$ presented above (*Mode 2*) and *Mode 1* is analogous to the $PFPNs$ concerning single transitions instead of steps.

We say that a step (a simple step) U fires from a marking M with a parameter value v to $M^{'}$.

A step (a simple step) U by a marking M with a parameter value v is called *maximal*, if there is no step (simple step) U' by M with the parameter value v such that $U' \supset U$.

These definitions are illustrated by the following example.

Example 3. Consider a *PFP*-net in Example 1. A set of transitions $U = \{t_1, t_2\}$ is a step by the marking $M = (0.6, 0.5, 0.7, 0, 0)$ with a parameter value $v = 1$ and the same parameterised families of sums and products as in Example 1 in *Mode* 2 for this net. The vectors U^{γ}, U^{-}, U^{0}, U^{+} have the form: $U^{\gamma} = (0.4, 0.4, 0.3, 0, 0)^T$, $U^{-} = (0.8, 0.8, 0.35, 0, 0)^T$, $U^{0} = (0.6, 0.5, 0.7, 0, 0)^T$, $U^{+} = (0, 0, 0, 0.56, 0.28)^T$ for the step U by M with $v = 1$ for this net. It is easy to see that the step U by M with $v = 1$ is enabled, because $U^{-} \geq U^{\gamma}$. After firing U by M with $v = 1$ in *Mode* 2 we obtain a new marking $M' = (0.6, 0.5, 0.7, 0.56, 0.28)$.

4 Conclusions

In this paper we have proposed a matrix representation of $PFPNs$. This representation enables an easy implementation of different concurrent algorithms for $PFPNs$ in modern programming languages or computational environments. In particular, taking into account parallel firing rules in a step we can speed up the reasoning process represented by a given $PFPN$. In further investigations we will consider this representation of $PFPNs$ in order to show its practical use in fuzzy reasoning process as well as in dynamical systems taking into account fuzzy control [16],[17] among others.

Acknowledgment. The author is grateful to anonymous referees for their helpful comments.

References

1. Cardoso, J., Camargo, H. (eds.): Fuzziness in Petri Nets. Springer (1999)
2. David, R., Alla, H.: Petri Nets & Grafcet. Tools for Modelling Discrete Event Systems. Prentice-Hall (1992)
3. Fryc, B., Pancerz, K., Peters, J.F., Suraj, Z.: On fuzzy reasoning using matrix representation of extended fuzzy Petri nets. Fundamenta Informaticae 60(1-4), 143–157 (2004)
4. Gao, M., Zhou, M.: Fuzzy reasoning Petri nets. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 33(3), 314–324 (2003)
5. Jensen, K., Rozenberg, G.: High-level Petri Nets. Springer (1991)
6. Desel, J., Reisig, W., Rozenberg, G. (eds.): Lectures on Concurrency and Petri Nets. Springer (2004)
7. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer (2000)
8. Murata, T.: Petri nets: properties, analysis and applications. Proc. of the IEEE 77, 541–580 (1989)
9. Peterson, J.L.: Petri net theory and the modeling of systems. Prentice-Hall (1981)
10. Suraj, Z.: Parameterised fuzzy Petri nets for approximate reasoning in decision support systems. In: Hassanien, A.E., Salem, A.-B.M., Ramadan, R., Kim, T.-H. (eds.) AMLTA 2012. CCIS, vol. 322, pp. 33–42. Springer, Heidelberg (2012)
11. Suraj, Z.: Parameterised fuzzy Petri nets for knowledge representation and reasoning. In: Proc. of the 2nd Int. Conf. on Data Management Technologies and Applications (DATA 2013), Reykjavik, Iceland, July 29-31, pp. 29–31 (to appear, 2013)
12. Suraj, Z.: A new class of fuzzy Petri nets for knowledge representation and reasoning. Fundamenta Informaticae (in print)
13. Suraj, Z.: Linear-Algebraic Representation of Generalised Fuzzy Petri Nets. In: Proc. of the 2013 IFSA World Congress, Edmonton, Canada, June 24-28, pp. 226–231 (2013)
14. Wang, H., Jiang, C., Liao, S.: Concurrent reasoning of fuzzy logical Petri nets based on multi-task schedule. IEEE Transactions on Fuzzy Systems 9(3), 444–449 (2001)
15. Yuan, J., Shi, H., Liu, C., Shang, W.: Backward Concurrent Reasoning Based on Fuzzy Petri Nets. In: Proc. of the 2008 IEEE Int. Conf. on Fuzzy Systems (FUZZ 2008), Hong Kong, China, June 1-6, pp. 832–837 (2008)
16. Zimmermann, H.J.: Fuzzy Set Theory and Its Applications. Kluwer, Boston (1993)
17. Yager, R.R., Filev, D.P.: Essentials of fuzzy modeling and control. Wiley (1994)

A Mathematical Theory of Fuzzy Numbers
Granular Computing Approach

Tsau Young Lin

Department of Computer Science
San Jose State University
San Jose, CA 95192
and
GrC Society

Abstract. The essence of granular computing (GrC) is to replace the concept of points in classical mathematics by that of granules. Usual fuzzy number systems are obtained by using type I fuzzy sets as granules. These fuzzy number systems have a common weakness - lack of existence theorem. Let R be the real number system, the trapezoidal membership functions at $r \in R$ is a base of fuzzified topological neighborhood system FNS(r). By taking FNS(r) as the granule, a new (but not type I) fuzzy number system $\mathcal{F}$ is formed. Surprisingly, we have found that such a new $\mathcal{F}$ is abstractly isomorphic to the classical real number system.

Keywords: Fuzzy numbers, Fuzzified topological neighborhood systems, Granular computing, Qualitative fuzzy set, Topology.

1 Introduction

What is a real number, a vector, a point in Euclidean plane or etc.? Due to the nature of mathematics, these questions are answered in a "whole sale" style. Namely, mathematicians have to define first the real number system, the vector spaces, Euclidean planes, and etc. (often axiomatically), then answer to the question by saying that an element of them is a real number, a vector, a point and etc. respectively.

What is a fuzzy number? There are plenty of answers. What is a fuzzy number system? There are no very clear cut answers. The "standard" constructions of fuzzy numbers (of type I fuzzy sets) are, more or less, given in the following ways: Let R be the set of real numbers.

1. Type I hypothesis: For each real number $r \in R$, there is associated a unique membership function $f_r : R \longrightarrow I$ that represents a very special fuzzy set, namely, the fuzzy number associated to r. Let $F_R = \{f_r \mid r \in R\}$ be the collection of such membership functions.
2. Constructions of mathematical structure on F_R: Binary operations, such as "addition" and "multiplication", are then introduced into F_R (and may be some other structures). The collection F_R, together with such a mathematical structure, forms the "standard" fuzzy number system.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 208–215, 2013.

However, the constructions of mathematical structure on F_R are incomplete: It needs to show that there exists (can be constructed) a set F_R of membership functions such that (1) for each real number there is a unique membership function in F_R, and (2) the "sum" and "product" of "addition" and "multiplication" of two members in F_R are some members of F_R again, in the jargon of mathematics, there is a construction of F_R such that F_R is closed under algebraic operations. (1) is satisfied by hypothesis, but (2) has not be shown in literature.

For example, by Type I hypothesis, membership functions f_6, f_2, f_3, f_4 and $f_{1.5}$ have been selected. But, there are no proofs for the following equalities:

$$f_6 = f_2 \odot f_3 = f_4 \odot f_{1.5} = f_5 \odot f_{1.2} = \ldots$$

We shall call these equalities consistency conditions. In other words, without proving the consistency condition, F_R, as a mathematical system, may not exists; see the second paragraph of Section 4.

The primary purpose of this paper is to reformulate these "standard" fuzzy numbers into a mathematical system that meets the consistency conditions. This system is, however, not in Type I theory, but in a very general Type II, called qualitative fuzzy set theory [8].

At first, the final theorem is a surprise: the new fuzzy number is a copy of the classical real number system.

However, if we do a little deeper analysis, we can see that this answer should be the expected one. A trapezoidal fuzzy number (its membership function contains an non-empty open crisp interval) is a "real world" approximation of a real number. So the collection of such approximations should converge to the the real number system; we show this in Section 4.

We also illustrate the idea in a more "commonly" used approximations. The collection of n-digits decimals $\forall\ n \in Z^+$ (positive integers) has been used as approximations for centuries. The system of such n-digits $\forall\ n \in Z^+$ does converge to real number system; this is proved by the concept of topology; see Section 3.

What is granular computing (GrC)? In 1996, granular computing (GrC) [12] was coined to label Zadeh's idea: GrC is a new mathematics, in which the concept of points in classical mathematics, is replaced by that of granules. Note that Zadeh's idea actually appeared in [10]. In fact, a more formal example in model theory of such an idea did exist a few years ahead [4]; the non-standard real number system (hyperreal) could be viewed as a "new" system, in which each real number r is replaced by a granule of "r + infinitesimals".

By taking a granule at p as the largest neighborhood system LNS(p) among all topologically equivalent NS(p) (see Example 1, Item 4), in GrC 2011 [9], LNS was axiomatized. That means Zadeh's idea is formally realized:

- This set of axioms defines the mathematics of GrC.

Using Zadeh's style of expressing, the granule, that has been axiomatized, is a granular variable that takes neighborhood $N \in$ LNS(p) as values. The concept of neighborhood systems (NS) was introduced in [7,6], and LNS(p) can also be regarded as a NS(p) that meets the super set condition (see Definition 1).

This paper is organized as follows: In Section 2, we review the concept of topological neighborhood system TNS(p), $p \in R$, then in Section 3, the algebraic operations are introduced among TNS(p) $\in 2^{2^R}$. In Section 4, we fuzzify the idea, and in Section 5, we state our conclusions.

2 Topological Neighborhood System (TNS)

First, we need to recall the concept of topology. Next few paragraphs are taken from [5] and the axioms are from (Chapter 1, Exercise B).

Definition 1. *The pair* $(U, TNS(U))$ *is called a topological space (or TNS-space), if* $TNS(U) = \{ TNS(p) : p \in U\}$ *is defined as follows: For each* $p \in U$, *let* $TNS(p)$ *be the family of all subsets, called neighborhoods, that satisfies the following axioms:*

1. *If* $N \in TNS(p)$, *then* $p \in N$;
2. *If* N *and* M *are members of* $TNS(P)$, *then* $N \cap M \in TNS(p)$;
3. *superset condition: If* $N \in TNS(p)$ *and* $N \subset M$, *then* $M \in TNS(p)$;
4. *If* $N \in TNS(p)$, *then there is a member* M *of* $TNS(p)$ *such that* $M \subset N$ *and* $M \in TNS(y)$ *for each* y *in* M *(that is,* M *is a neighborhood of each of its points).*

Definition 2. *A base* $\mathcal{B}(p)$ *of* $TNS(p)$ *of a point* p *is a family of neighborhoods such that every neighborhood* $N \in TNS(p)$ *contains a member of the family* $\mathcal{B}(p)$.

Example 1. (Bases of TNS of R)

1. $\forall\ p \in R$, *let us consider the collection* $\mathcal{B}(p) = \{N_1(p) = (p - 1/10^n, p + 1/10^n), n \in Z^+\}$. *This collection* $\mathcal{B}(p)$ *is a base of the* $TNS(p)$. N_1 *is the uncertainty region of the n-digits decimal number of* p.
2. *Let* $TNS(p)$ *be the maximal collection of subsets, in which each subset contains an* $N_1(p)$ *for some* $n \in Z^+\}$, *where* Z^+ *is positive integers. Then* $TNS(U) = \{ TNS(p)\}$ *is the TNS of real number system; it is routine to verify that the four axioms in Definition 1 are satisfied.*
3. *Another base of TNS of real numbers can also be defined by using base other than 10 the collection* $\{N_2(p) | N_2(p) = (p-\epsilon, p+\epsilon)$, *where* $\epsilon = 1/b^n$,*where* b *is any positive integer. This base leads to the same maximal collection* $TNS(U)$.
4. *In the theory of neighborhood system (NS), which is the ultimate generalization of topology, the notation for maximal collection is* $LNS(U)$. *So, if we interpret the topology as a special kind of NS, then* $TNS(U) = LNS(U)$. *This may explain a bit more on LNS that is mentioned in the Section of Introduction.*

3 The Real Number System $\mathcal{R}_T$ Defined by Topology

Let the universe R of discourse in this section be the real number system. The main idea is to introduce the algebraic structure into $\mathcal{R}_T \subset 2^{2^R}$, for example, $\text{TNS}(p) \oplus \text{TNS}(q) \ \in 2^{2^R}$ and $\text{TNS}(p) \otimes \text{TNS}(q) \ \in 2^{2^R}$, between two TNS.

Definition 3. *The real number system R is a complete order field.*

The real number system can be defined in many ways; we take the axiomatic approach ([1] p.98). Here "order field" refers to the usual "college algebra" (four kinds of operations and order relations that satisfy various kinds of laws), and term "complete" means: Any bounded below set has the greatest lower bound. For example, the rational number system Q is not a complete order field because the set $A = \{x \in Q | x > \sqrt{2}\} \subset Q$ dose not have the greatest lower bound, while R is a complete order field. For R, the greatest lower bound of its subset $B = \{x \in R | x > \sqrt{2}\} \subset R$ is $\sqrt{2}$.

Before, we give the new definition of fuzzy numbers, we will show in this section that the real number system can be defined by the collection of topological neighborhood system (TNS).

Definition 4. (GrC based real number system $\mathcal{R}_T$) *GrC based real number system is:*

$$\mathcal{R}_T = \{\bar{p} \mid \bar{p} = TNS(p), p \in R\},$$

with appropriate algebraic structure that will be introduced below.

For mathematical students, this is a fairly routine to verify that $\mathcal{R}_T$ defined above, indeed, forms the complete ordered field. Since this is in computer science paper, we shall sketch few key points.

Definition 5. (Subset operations in an algebraic system) *Let $X, Y \subseteq E$ be two subsets of an algebraic system $(E, \cdot)$. The operator $\circ$ between X and Y is defined as*

$$X \circ Y = \{x \cdot y | \ \forall \ x \in X, \ \forall \ y \in Y\},$$

or in terms of the convolution of characteristic functions

$$\chi_{X \circ Y}(z) = \max_{x \cdot y = z} \min(\chi_X(x), \chi_Y(y)), \ (x, y, z) \in R^3.$$

where $\cdot$ is a binary operator in E.

Proposition 1. *If $\cdot$ is commutative or associative, then $\circ$ is commutative or associative respectively.*

In practice (for example, in the coset multiplication in group theory), we do not use new notation $\circ$, but (by abuse of notation) use the notation $\cdot$ of the given binary operations. By applying Definition 5 twice (to neighborhoods ($\subseteq R$), then to TNS ($\subseteq 2^R$), we have

Definition 6. (External algebraic operations) $\forall\ \bar{p},\ \bar{q} \in \mathcal{R}_T$,

$$\bar{p} \oplus' \bar{q} \equiv \{N(p) + N(q) \mid N(p) \in TNS(p), N(q) \in TNS(q)\}$$

$$\bar{p} \odot' \bar{q} \equiv \{N(p) \cdot N(q) \mid N(p) \in TNS(p), N(q) \in TNS(q)\},$$

where (in the following formulas, we use "$\cdot$", instead of "$\circ$"),

$$N(p) + N(q) = \{r_1 + r_2 | r_1 \in N(p), r_2 \in N(q)\},$$

$$(\chi_{N(p)+N(q)}(z) = \max_{x+y=z} \min(\chi_{N(p)}(x), \chi_{N(q)}(y)),\ (x, y, z) \in R^3).$$

$$N(p) \cdot N(q) = \{r_1 \cdot r_2 | r_1 \in N(p), r_2 \in N(q)\},$$

$$(\chi_{N(p)\cdot N(q)}(z) = \max_{x\cdot y=z} \min(\chi_{N(p)}(x), \chi_{N(q)}(y)),\ (x, y, z) \in R^3),$$

These 2 external operations induce the following 2 inclusions.

Proposition 2. $\forall\ \bar{p},\ \bar{q} \in \mathcal{R}_T$,

$$\bar{p} \oplus' \bar{q} \subseteq \overline{p + q}; \quad \bar{p} \odot' \bar{q} \subseteq \overline{p \cdot q}.$$

We shall explain the first inclusion: $\bar{p} \oplus' \bar{q}$ consists of all possible $\{N(p) + N(q)\}$. From Example 1, there are bases (of $1/10^t$-neighborhoods) for TNS(p) and TNS(q). Namely, there are $(p - 1/10^n, p + 1/10^n) \subseteq N(p)$, for some integer n, and $(q - 1/10^m, q + 1/10^m) \subseteq N(q)$, for some integer m. Obviously, we can find a $1/10^s$-neighborhood of $p + q$ so that $(p + q - 1/10^s, p + q + 1/10^s) \subseteq (p - 1/10^n, p + 1/10^n) + (q - 1/10^m, q + 1/10^m)$. This inclusion implies that $N(p) + N(q) \in \text{TNS}(p+q)$, by the Axiom of supper set condition in Definition 1. Similar proof works for the other inclusion too.

The two inclusions induce two following internal operations in $\mathcal{R}_T$

Definition 7. $\forall\ \bar{p},\ \bar{q} \in \mathcal{R}_T$,

$$\bar{p} \oplus \bar{q} \equiv \overline{p + q}: \quad \bar{p} \odot \bar{q} \equiv \overline{p \cdot q}$$

Definition 8. *Algebraic system with two operators $(E, \circ_1, \circ_2)$ is called a bi-operator algebra.*

For example $(\mathcal{R}_T, \oplus, \odot)$ just introduced and $(R, +, \cdot)$ are bi-operator algebras.

Let us side track a little bit. The existence of the 2 internal operations imply the consistent conditions. For example $\bar{6} = \overline{2 \cdot 3} \equiv \bar{2} \odot \bar{3} = \ldots \quad \bar{6} = \overline{1 + 5} \equiv \bar{1} \oplus \bar{5} = \ldots$.

Next, we introduce the order relation into $(\mathcal{R}_T, \oplus, \odot)$ by

$$\bar{p} < \bar{q} \Leftrightarrow p < q.$$

Now, we have $(\mathcal{R}_T, \oplus, \odot, >)$. With these, we shall prove the following main theorem.

Theorem 1. $(\mathcal{R}_T, \oplus, \odot, >)$ *is a complete order field.*

Let $p \in R$, then the map: $\bar{p} \longrightarrow p$ is a one-to-one onto map, because R is a Hausdorff space. In Definition 7, we have defined $\bar{p} \oplus \bar{q} \equiv \overline{p+q}$, and $\bar{p} \otimes \bar{q} \equiv \overline{p \cdot q}$, so the two compositions below,

$$\bar{p} \oplus \bar{q} \longrightarrow \overline{p+q} \longrightarrow p+q; \qquad \bar{p} \otimes \bar{q} \longrightarrow \overline{p \cdot q} \longrightarrow p \cdot q,$$

imply that the map from $(\mathcal{R}_T, \oplus, \odot)$ to $(R, +, \cdot)$ is an isomorphism of bi-operator algebras. Since this isomorphism (and its inverse) preserves all the identities and inequalities among elements, the isomorphism actually is a complete order field isomorphism. QED.

4 Fuzzy Number System $\mathcal{F}$

This section is the main subject of this paper. A new mathematic system called "fuzzy number system" will be formally defined.

Let $N(0)$ be a neighborhood of 0 in R. We claim that $N(0) \neq N(0) + N(0)$: From group theory, the equality holds only if $NS(0)$ is a abelian subgroup of R. There is no bounded abeliam subgroup in R, so we proved the claim. This shows that for characteristic functions, and hence for membership functions, F_R cannot be closed under the "addition" that is defined by convolution; similar conclusion can be drawn for "multiplications". This counter example implies that F_R, together with the algebraic operations defined by convolutions, such as [3], are not closed under algebraic operations. To show that F_R with some algebraic operations is a well-defined mathematical system, an explicit proof of closed-ness is needed; that seems lacking in the literature.

4.1 The Universe of Membership Functions

First, we have to specify the membership functions. In Type I fuzzy control, the outputs are control functions. So in most cases, they are continuous functions. Therefore the membership functions used in control are likely the continuous functions. Unfortunately, this choice will exclude out the classical sets from fuzzy set theory (characteristic functions are not continuous functions). So we choose the functions of continuous almost everywhere (a.e.) as our universe of discourse, where "continuous a.e." means a function whose continuous points are almost everywhere, in other words, whose set of discontinuous points has measure zero [2].

4.2 Fuzzification of Topology

Definition 9. (fuzzification of neighborhood system) *The fuzzification FNS(p), called fuzzy neighborhood system, of topological neighborhood system TNS(p) consists of all membership functions f_i defined on R that contain (as "inclusion" of fuzzy sets), at least, one subset N that is an element TNS(p).*

Observe that in this case all membership functions have some "flat top"; we call them "trapezoidal" membership functions.

Definition 10. (GrC Based Fuzzy numbers) *Fuzzy number system is:*

$$\mathcal{F} = \{\tilde{p} \mid \tilde{p} = FNS(p), p \in R\},$$

with appropriate algebraic structure that will be introduced below.

Here are the external operations:

Definition 11. $\forall \tilde{p},\ \tilde{q} \in \mathcal{F}$,

$$\tilde{p} \oplus' \tilde{q} \equiv \{f_p \boxplus f_q | f_p \in \tilde{p}, f_q \in \tilde{q}\},$$

$$f_p \boxplus' f_q(z) \equiv \max_{x+y=z} \min(f_p(x), f_q(y)),\ (x, y, z) \in R^3.$$

$$\tilde{p} \otimes' \tilde{q} \equiv \{f_p \boxtimes f_q | f_p \in \tilde{p}, f_q \in \tilde{q}\}.$$

$$f_p \boxtimes' f_q \equiv \max_{x \cdot y=z} \min(f_p(x), f_q(y)),\ (x, y, z) \in R^3.$$

The two external operations induce the following fuzzy-inclusions

Proposition 3. $\forall \tilde{p},\ \tilde{q} \in \mathcal{F}$,

$$\tilde{p} \oplus' \tilde{q} \subseteq \widetilde{p+q}; \qquad \tilde{p} \otimes' \tilde{q} \subseteq \widetilde{p \cdot q}.$$

Note that $1/10^t$-neighborhoods are also a base for $\tilde{p}$ (as well as $\bar{p}$), so the same reasoning for $\bar{p}$ (Proposition 2) does work for $\tilde{p}$ mathematically.

The 2 inclusions induce the following 2 internal operations

Definition 12. $\forall \tilde{p},\ \tilde{q} \in \mathcal{F}$,

$$\widetilde{p+q} \equiv \tilde{p} \oplus \tilde{q}; \qquad \widetilde{p \cdot q} \equiv \tilde{p} \otimes \tilde{q}.$$

These operations are simple generalizations of the convolutions defined on characteristic functions (the generalized formulas were used in [3]). As in $\mathcal{R}_T$, we will introduce the order relation into $(\mathcal{F}, \oplus, \odot)$ by

$$\tilde{p} < \tilde{q} \Leftrightarrow p < q.$$

So we have $(\mathcal{F}, \oplus, \odot, >)$.

Again, let us have some side tracks, the 2 internal operations imply the consistent conditions: $\tilde{r} = \tilde{p} \otimes \tilde{q} = \tilde{r} \oplus \tilde{s} \quad \forall\ p,\ q,\ r,\ s \in R$ for all possible decompositions of the real number r with respect to multiplication and additions respectively. For example, $\tilde{6} = \widetilde{2 \cdot 3} \equiv \tilde{2} \odot \tilde{3} = \ldots \quad \tilde{6} = \widetilde{1+5} \equiv \tilde{1} \oplus \tilde{5} = \ldots$.

With these, we shall prove the following main theorem

Theorem 2. $(\mathcal{F}, \oplus, \odot, >)$ *is a complete order field.*

The same proof for Theorem 1 will work for $\mathcal{F}$ by considering a similar map $\tilde{p} \longrightarrow p$.

5 Conclusion – The Meanings of Computing in $\mathcal{F}$ and $\mathcal{R}_T$

In applications, we often use n-digits, say n=2, decimals, instead of the real numbers R, to do the computation. Then, the infinitely many numbers in the interval [0, 1] have reduced to 100 2-digits representations, $0.00, 0.01, \ldots 0.99$; each number represents a crisp interval (a neighborhood). So 2-digits computation is an interval or neighborhood computing. The theory in Section 3 (TNS or GrC computing) says, if n increases, $1/10^n$-interval computing will converge to real number computing.

If we fuzzified the 2-digits numbers, then it means we are computing in trapezoidal fuzzy numbers, the theory in Section 4 (LNS or GrC computing) guarantees that, if we decrease the length of the flat top, the computations will be eventually converge to the real number computations.

This paper gives n-digits and fuzzy n-digits computing some theoretical foundations.

References

1. Birkhof, G., MacLane, S.: A Survey of Modern Algebra. MacMillan Publishers (1970)
2. Delillo, N.J.: Adanced Calculus with Applications. Macmillan Publishing Co. (1982)
3. Dobois, D., Prade, H.: Fuzzy Sets and System: Theory and Applications. Academic (1980)
4. Robinson, A.: Non-standard analysis. North-Holland Pub. Co. (1966)
5. Kelley, J.: General topology. Springer, New York (1975) ISBN 0387901256
6. Lin, T.Y.: Chinese Wall security policy -an aggressive model. In: Proceedings of the Fifth Aerospace Computer Security Application Conference, December 4-8 (1989)
7. Lin, T.Y.: Neighborhood Systems and Approximation in Database and Knowledge Base Systems. In: Proceedings of the Fourth International Symposium on Methodologies of Intelligent Systems, Poster Session, October 12-15 (1989)
8. Lin, T.Y.: Qualitative Fuzzy Sets: A comparison of three approaches. In: Proceeding of Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Vancouver, Canada, July 25-28, pp. 2359–2363 (2001)
9. Lin, T.Y., Syau, Y.-R.: Granular Mathematics foundation and current state. GrC (2011)
10. Zadeh, L.: Fuzzy sets and Information Granularity. In: Gupta, M., Ragade, R., Yager, R. (eds.) Advances in Fuzzy Set Theory and Applications, pp. 3–18. North-Holland, Amsterdam (1979)
11. Zadeh, L.: The Key Roles of Information Granulation and Fuzzy logic in Human Reasoning. In: 1996 IEEE International Conference on Fuzzy Systems, September 8-11, p. 1 (1996)
12. Zadeh, L.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft Comput. 2(1), 23–25 (1998)

Network Performance Analysis Based on Quotient Space Theory

Ling Zhang, Yuan-ting Yan, Shu Zhao, and Yan-ping Zhang*

School of Computer Science and Technology, Anhui University
Key Laboratory of Intelligent Computing and
Signal Processing of Ministry of Education,
Hefei, Anhui Province, 230039, China
zhangyp2@gmail.com

Abstract. Some performance analyses in complex network (e.g., shortest path, etc.) are complicated. Generally, human have natural ability to solve complex problems by approximating the optimal solution step by step. The granular computing model based on QST (Quotient Space Theory) provides not only a hierarchical description from fine to coarse but also an effective approach from coarse to fine to solve these complex problems. This paper proposes some methods on complex network performance analysis based on QST. Firstly, maximum cover network chain is used to solve the shortest path problem. Then, a method to find the optimal path of a weighted network is put forward. Finally, dynamic network is decomposed into a series of static networks to solve the maximum flow problem in dynamic network. Theoretical proofs and experimental results show that QST is an effective tool for complex problem solving.

Keywords: Quotient Space, Performance Analysis, Shortest Path, Optimal Path, Dynamic Network.

1 Introduction

The basic idea of QST is that "one of the basic characteristics in human problem solving is the ability to conceptualize the world at different granularities and translate from one abstraction level to the others easily" [1]. In recent years, we have further studied the relationships among the quotient space theory, fuzzy set theory, and rough set theory [3, 4, 12-19]. A unified representation of these theories is given to describe fuzzy. The solving of complex problem is simplified by a hierarchical description based on QST. We developed a set of its applications in path planning, temporal planning, robot motion planning, heuristic search [1, 2], etc. There are many complicated problems in complex networks, such as the shortest path, optimal path [10-11], etc. Recently, we have applied QST into complex network analysis and made some progresses [5, 6, 8, 9]. This paper summarizes the achievements we have made, and presents a method for dynamic network performance analysis.

* Corresponding author.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 216–224, 2013.

The rest of this paper is organized as follows. Section 2 briefly proposes the quotient space theory and discusses how to apply it to network analysis. Section 3 and Section 4 puts forward the shortest path solving and optimal path finding based on QST, respectively. Section 5 presents the dynamic network analysis. Conclusions are given in Section 6.

2 Method of Quotient Space

A problem is described as a triplet (X, f, T), where X is the universal, f is the attribute and T is the structure (or topology). Given an equivalence relation R on X, the quotient set corresponding to X, f and T is denoted by$[X]$,$[f]$ and $[T]$, respectively. $([X], [f], [T])$ is called the quotient space of (X, f, T). Then, the principles "truth preserving" and "falsity preserving" in quotient spaces is established [1, 20]. These principles are used to speed up the problem solving.

First of all, we discuss how to apply QST into network analysis.

Let $N = (V, E)$ be an unweighted network, where V is the set of nodes and E is the set of edges. R_1 is an equivalence relation on V, and $N_1 = (V_1, E_1)$ is the quotient space corresponding to R_1 which is denoted by $N > N_1$.

Definition 1. Let $N > N_1 > ... > N_n$ be a quotient space chain, for $x \in N$, $(x, [x_1], [x_2], ...[x_n])$ is called hierarchical coordinates of x with respect to quotient space chain, for short: hierarchical coordinates of x.

Quotient space theory of problem solving construct a quotient space chain firstly, then choosing a certain quotient space and solving the corresponding problem in N_i, after that, choosing a finer quotient space N_j , $j < i$ and solving the corresponding problem in N_j. The above steps are repeated until the whole problem solving is completed.

3 Method of Quotient Space for the Shortest Path

Given a network $N = (V, E)$ and $a, b \in V$, the shortest path between a and b is denoted by the minimal number of nodes from a to b .

We use the method of QST to solve shortest path. First, we construct a proper quotient space.

Definition 2. Suppose a complete subgraph C of network N is a maximum complete subgraph if and only if C is not a proper subset of other complete subgraphs.

Definition 3. Given a network $N = (V, E)$ and a set of maximum complete subgraphs $\{C_i | i = 1, ..., k\}$, if $\bigcup V_i = V$, $\bigcup E_i = E$, then $\{C_i | i = 1, ..., k\}$ is called maximum complete cover of $N = (V, E)$.

Quotient space of complex network is constructed by maximum complete cover.

Definition 4. Given a network N and a maximum complete cover $\{C_i | i = 1, ..., k\}$, a maximum complete subgraph C_i is regarded as a node, if two

maximum complete subgraphs have common points, then the corresponding two nodes are connected. The network obtained by this method is called the first level maximum cover network, denoted by N_1 (the quotient network of N). Similarly, suppose N_i is the i^{th} level cover network and N_{i+1} is the first level maximum cover network of N_i, then N_{i+1} is called the $(i+1)^{th}$ maximum cover network of N .

Definition 5. Given a maximum cover network chain $(N > N_1 > ... > N_i)$, if a node a in N_i could cover all nodes in V ,where V is the set of nodes in N. a is called a complete node. If N_i have complete node and N_{i-1} does not contains any complete node. Then we say $(N > N_1 > ... > N_i)$ is the maximum complete cover network chain of N, for short, cover network chain.

Definition 6. Given a maximum complete cover network chain $(N > N_1 > ... > N_i)$ of network N, suppose $a_0 \in V, a_0^i \in V_i$, and $a_0 \in a_0^i$, then the hierarchical coordinates of a_0 with respect to $(N > N_1 > ... > N_i)$ is $(a_0^0 = a_0, a_0^1, ..., a_0^i)$, where $a_0^{j-1} \in a_0^j, 0 < j \leq i$, a_0^j is a node in quotient space N_j.

Definition 7. Given a network N and its first level maximum cover network N_1, suppose L is the shortest path from a to b in N, and $L = \{a_0 = a, a_1, ..., a_k = b\}$. We construct a path $L_1 = \{a_0^1, ..., a_{k-1}^1\}$ in N_1, where $a_i, a_{i+1} \in a_i^1, i = 0, 1, ...k-1$, then $a_i^1 \neq a_j^1, i \neq j$. Hence, the path L in N was mapped into $L_1 = \{a_0^1, ..., a_{k-1}^1\}$ in N_1, where $p_1(a_0) = a_0^1$, $p_1(a_i) = a_{i-1}^1 \cap a_i^1, i \geq 1$, $p_1(b) = a_{k-1}^1$,L_1 is called the projection of L in N_1, denoted by $L_1 = p_1(L)$. Generally, suppose L_{i-1} is the projection of L in N_{i-1}, the projection of L_{i-1} in N_i is L_i, L_i is called the projection of L in N_i.

Given a network N, L is supposed to be the shortest path from a to b in N, the necessary and sufficient condition of $L = \{a = a_0, a_1, ..., a_k = b\}$ is that: $L_i = p_i(L)$ is the shortest path from $a_0^i (a_0 \in a_0^i)$ to $a_{k-i}^i (a_k \in a_{k-i}^i)$ in N_i, where L_i is the projection of L in the i^{th} level cover network. It is the truth preserving principle of shortest path. We can translate the problem of shortest path L into the corresponding shortest path problem L_i in N_i . After that, we are backtracking to L_{i-1}, so L could be got by $i-1$ steps.

The algorithm about finding the shortest path for two given nodes by using maximum cover network chain is illustrated as follows, which is called pyramid algorithm.

Algorithm 1. pyramid algorithm

Input: $(N > N_1 > ... > N_i)$, $(a = a_0^0, a_1^0, ..., a_0^i)$,$(b = b_0^0, b_0^1, ..., b_0^i = a_0^i)$
Output: L_0

1: Choosing a quotient space N_i satisfies: for $j \leq i$, a, b do not belong to the same a_k^j simultaneously.
2: Construct path $L_{i-j} = (a_0^{i-j}, a_1^{i-j}, ..., a_j^{i-j} = b_0^{i-j})$(all nodes are different).
3: If $i \neq j$, then
4: Construct path $L_{i-j-1} = (a_0^{i-j-1}, a_1^{i-j-1}, ..., a_{j+1}^{i-j-1} = b_0^{i-j-1})$.
5: else return $L_0 = (a = a_0^0, a_1^0, ..., a_i^0 = b)$

Given $a, b \in V$, the shortest path between a and b can be solved by follow steps.

1) Suppose $(N > N_1 > ... > N_i)$ is the maximum cover network chain of N. Choosing a quotient space N_i. N_i satisfies the following conditions: $\exists a_0^i \in V_i, a, b \in a_0^i$, for arbitrary $j \leq i$, a, b do not belong to the same a_k^j simultaneously, where a, b are two nodes in V. Let $(a = a_0^0, a_1^0, ..., a_0^i)$,$(b = b_0^0, b_0^1, ..., b_0^i = a_0^i)$ be the hierarchical coordinates of a, b.

2) Because a, b belong to the same node in N_i .There exist $a_0^{i-1}, a_1^{i-1}, a \in a_0^{i-1}, b \in a_1^{i-1}$ and $a_0^{i-1}, b_0^{i-1} \in a_0^i$. Construct path $L_{i-1} = (a_0^{i-1}, a_1^{i-1})$.

3) Suppose we have a path $L_{i-j} = (a_0^{i-j}, a_1^{i-j}, ..., a_j^{i-j} = b_0^{i-j})$(all nodes are different). If $i \neq j$, take $a_{k-1}^{i-j-1} \in a_{k-a}^{i-j} \cap a_k^{i-j}, k = 1, ..., j$, moreover, there exist a_0^{i-j-1}, $a_{j+1}^{i-j-1}, a \in a_0^{i-j-1}, b \in a_{j+1}^{i-j-1}, a_0^{i-j-1} \in a_0^{i-j}, a_{j+1}^{i-j-1} \in a_j^{i-j}$. Construct path $L_{i-j-1} = (a_0^{i-j-1}, a_1^{i-j-1}, ..., a_{j+1}^{i-j-1} = b_0^{i-j-1})$.

4) Finally, $L_0 = (a = a_0^0, a_1^0, ..., a_i^0 = b)$ is got until $i = j$. L_0 is the shortest path from a to b with the length of i.

Let the hierarchical coordinates of a, b be two hypotenuse of the triangle. $a_1^{i-2} \in a_0^{i-1} \cap b_0^{i-1}$, so we have $(a_0^{i-2}, a_1^{i-2}, ..., b_0^{i-2})$. Generally, suppose we have $(a_0^j, a_1^j, ..., b_0^j), j < i$. Taking $a_k^{j-1} \in a_{k-1}^j \cap a_k^j$, then we got $(a_0^{j-1}, a_1^{j-1}, ..., a_{i-j}^j = b_0^j)$.

By the above method, finally we got $L_0 = (a = a_0^0, a_1^0, ..., a_i^0 = b)$, where $a_j^{i-t} \in a_{j-1}^{i-t+1} \cap a_j^{i-t+1}, j = 1, ..., t-2, a_{t-1}^{i-t} \in a_{t-2}^{i-t+1} \cap b_0^{i-t+1}$.

Shortest path:$L(a, b) = (a, a_1^0, a_2^0, ..., a_{i-1}^0, b)$.

$$a_i^0$$

$$a_0^{i-1}, b_0^{i-1}$$

$$a_0^{i-2}, a_0^{i-2}, b_0^{i-2}$$

$$a_0^{i-3}, a_1^{i-3}, a_2^{i-3}, b_0^{i-3}$$

$$a_0^{i-4}, a_1^{i-4}, a_2^{i-4}, a_3^{i-4}, b_0^{i-4}$$

.....

$$a_0^0 = a, a_1^0, a_2^0, ..., a_{i-1}^0, b_0^0 = b$$

Suppose $(N > N_1 > ... > N_m)$ is the maximum cover network chain of N. $\forall a, b \in V$,where V is the set of nodes in N, the length of shortest path $= j \Longleftrightarrow \exists N_j$, and $a_k^j \in V_j, a, b \in a_k^j$, for arbitrary $i < j, a, b$ do not belong to the same node of N_k, where $j < m$ and a_k^j denote the k^{th} element in N_j.

4 Optimal Path of Weighted Network

We already have the method for shortest path finding which is solved by finding the path with minimal number of nodes. It is common in daily life, for example, in

order to avoid the traffic light, we hope to avoid more crossroads as possible as we can. Actually, there also exist many other factors, such as road condition, traffic condition, etc. These factors can be regarded as weights of the network. An optimal path is the path that connects any pair of nodes with the maximal weight. We will discuss the optimal path problem of weighted network in this section.

Complex network was represented by weighted graph. Suppose $N = (V, E, w(e))$ is the weighted network. Where V is a finite set of nodes, E is the set of edges, $w : E \to R^+, w(e) \in [0, d]$, is the weight number(bandwidth, flow, etc.).

Suppose the set of weights of all edges is $\{d_1 > d_2 > ... > d_k\}$. An equivalence relation $R(d_i)$ is given to acquire the corresponding quotient space for weighted network.

Definition 8. Equivalence relation $R(d_i)$:

$a \sim b \Longleftrightarrow \exists a = a_1, a_2, ..., a_m = b, w(a_j, a_{j+1}) \geq d_i, a_j, a_{j+1} \in V, j = 1, ..., m-1, i = 1, ..., k$

The quotient space corresponding to $R(d_i)$ is $N_i = \{a_1^i, ..., a_n^i\}, i = 1, ..., k$ and $a_1^i, ..., a_n^i \in V_i$. Obviously, $N > N_1 > ... > N_k$ is a hierarchical quotient space chain.

The elements in N is represented by hierarchical structure:

Suppose $z \in V$ and the hierarchical coordinates of z is $(z_0, z_1, ..., z_k)$ where $z_i \in \mathbb{N}$, and $\mathbb{N}$ is the set of natural number. Projection $p_i : X \to X_i$, if $p_i(z)$ belongs to the i^{th} element of X_i, then, let the i^{th} element z_i of z be t.

Definition 9. $\forall a, b \in V, d_i, a, b$is d_i connected$\Longleftrightarrow$There exists a path from s to b with weight number $\geq d_i$.

Theorem 1. $\forall a = (a_0, a_1, a_2, ..., a_k), b = (b_0, b_1, b_2, ..., b_k) \in V, d_i, a, b$ is d_i connected$\Longleftrightarrow$ $a_i = b_i$,where $a = (a_0, a_1, a_2, ..., a_k), b = (b_0, b_1, b_2, ..., b_k)$ denote the hierarchical coordinates.

The proof of theorem 1 is in [5].

Using quotient space chain and hierarchical coordinates, the problem of shortest path in original space could be transform into the problem of shortest path in different quotient spaces. The solving process is from coarse to fine which greatly reduces the complexity of problem solving [4].In [5], we presented an approximate algorithm to finding the optimal path. We also carried out some comparison experiments. Table 1 is one of the experiment results. More details about the algorithm and experiment are in [5].

Table 1. Total CPU time (in seconds) in the random network

	100	200	300	400	500
optimal path	0.397	1.356	3.112	6.634	12.171
Dijkstra	1.719	9.797	91.141	656.391	1002.125
Floyd	0.940	4.220	118.630	212.030	511.560

5 Analysis of Dynamic Network

This section analyzes the maximum flow in dynamic network base on QST.

Let $N = (V, E, d_e(v,t))$ be the given dynamic network, where V is the set of nodes (also regard as domain of definition), E is the set of edges (regard as the topological structure T), $d_e(v,t), e \in E, v$ is one vertex of edge e and $t \in [t_0, t_1]$ represents the trafficability in x of edge e at the time of t (regard as attribute f). Given a source node a, sink node b, initial moment t_0. Suppose the direction of fluid is from a to b . Given a time period $[t_0, t_2]$, there are two problems. The first problem is the maximum flow of b during the period $[t_0, t_2]$. And the second is how to calculate the maximum flow of node b at t_2 .The goal of dynamic network analysis in this section is to calculate the maximum flow.

The main theorem about maximum flow in static network is the minimum cut theorem. Extending minimum cut theorem to dynamic situations is not only the theory needs, but also has great significance in applications. For example, for a big river, how to predict the downstream flood peak with the flow conditions measured from upstream hydrological station when the river basin has heavy rains. This is a typical relationship problem between minimum cut and maximum flow. The properties of minimum cut and maximum flow in dynamic network are discussed as follows.

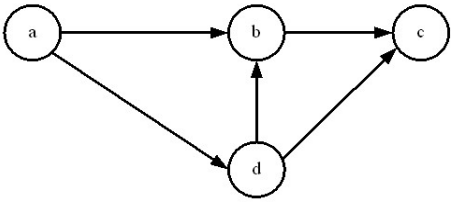


Fig. 1. Example of dynamic network

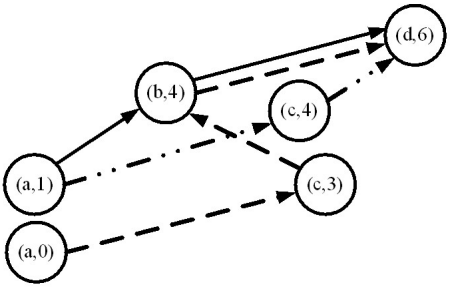


Fig. 2. Streamlines of (d,6)

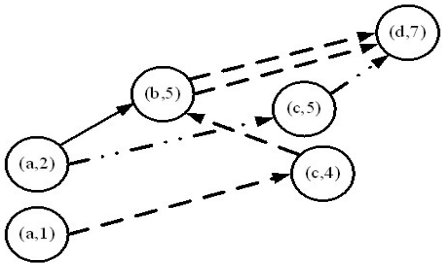


Fig. 3. Streamlines of (d,7)

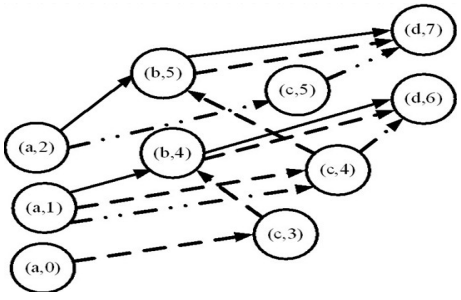


Fig. 4. Merging (d,6) and (d,7)

Definition 10. Given a dynamic network N, source node a, sink node b and the moment t, the maximum flow of b in moment t is the most probable maximum outflow of b.

Maximum flow defined in definition 10 is not additivity as illustrated as follows.

Example 1. Maximum flow defined in definition 10 is not additivity. Suppose the maximum flow of all edges is 1 with except for edge (bd) which is 2. Obviously, static maximum flow from a to d is 2.

Here we discuss the dynamic maximum flow problem. The length of edges are given: $(ab) = 4, (cd) = 2, (ac) = 3, (cb) = 1, (bd) = 2$. (Suppose time is discrete and flow rate $c = 1$ for convenience).

Question: The maximum flow of (d,6).

First of all, calculating all streamlines of (d,6):

((a,0),(b,4),(d,6)), ((a,1),(c,4),(d,6)), ((a,0),(c,3),(b,4),(d,6)). Streamline ((a,0), (b,4), (d,6)) represents a path: start from point a at moment 0, arrived point b at moment 4 and at moment 6 arrived point d. The other paths have the similar meaning.

From Fig. 2, a unit of fluid start from point a at the moment of 0, flow through (acbd), arrived point b at moment 6(as the dashline shows). Another unit of fluid start from point a, flow throughabd), arrived point d at t=6 (as the solid arrow shows). At t=1,a unit of fluid start from point a, flow throughacd,arrived point b at t=6(as dotted arrow shows). Hence, the maximum flow of b at t=6 are 3.

Similarly, the maximum flow of (d,7) is 3 (as Fig. 3 shows).

The two networks are merged as Fig. 4. Edge ((a,1), (c,4)) have two streamlines with flow equals to 1, but the trafficability is 1. So only one path could choose between ((a,1),(c,4),(d,6)) and ((a,1),(c,4),(b,5)(d,7)). Hence, the maximum flow of b during the period $[6, 7]$ is $5 < 6$. This example shows that maximum flow is not additivity.

The concepts in static network are not suitable in dynamic network. Therefore, we propose two definitions with additivity to decompose the dynamic net-work into a combination of static networks.

Definition 11. Suppose that the fluid always flow along the shortest path between source node a and sink node b (if the shortest path is congested, then flow along the second short path). According to this rule, the maximum outflow of b at moment t is the maximum flow from a to b at moment t, denoted by $t - maximum$.

Maximum flow by definition 11 have the properties as follows.

Theorem 2. Maximum flow has the property of additivity.

proof. Suppose time is discrete for convenience. Let the maximum flow of b in moment t, $t+1$ are $S(t), S(t+1)$. Suppose the maximum flow from t to $t+1$ is S, the goal is to prove that $S = S(t) + S(t+1)$.

Obviously, $S \leq S(t) + S(t+1)$, so we need to prove '$<$' is false. We can use the proof by contradiction to prove it.

Suppose '<' is true. So there exist a flow arrived b in moment t flow along a path, and it also arrived b in moment $t+1$ flow along another path (the increase of flow is caused by double counting). It is contradict with definition 11. So we have $S = S(t) + S(t+1)$.

Definition 12. Total flow is maximal. Suppose the flow is defined by some specific methods. From initial moment t_0 to arbitrary moment $t(t_0 < t)$, if the total outflow of b is maximal. We say that the flow has the property: total flow is maximal.

Theorem 3. Maximum flow has the property that total flow is maximal.

proof. Suppose the maximum flow defined by definition 11(denoted by method 1) of exit node b from t_0 to t_1 is s_1. And the maximum flow defined by another definition (denoted by method 2) of exit node b from t_0 to t_1 is $s_2, s_2 > s_1$. Also suppose the time is discrete for convenience. Let $S_1(t), S_2(t)$ be the total flow of method 1 and method 2 from moment t_0 to moment t.

Because $S_1(t_0) = S_2(t_0) = 0$, we suppose $S_1(k-1) = S_2(k-1)$, the goal is to prove $S_1(k) = S_2(k)$. Suppose $S_2(k) > S_1(k)$, let $S_2(k) - S_2(k-1) = s_2, S_1(k) - S_1(k-1) = s_1$, so, the flow in moment k by method 2 is at least s_2, that is to say, at least s_2 units of flow with the shortest path equals to k. From this view, there exist at least s_2 units of flow arrived at b by method 1. It is contradict with the hypothesis $s_1 < s_2$.

So we have $S_1(k) = S_2(k)$.

For an arbitrary period $[t_m, t_n]$, we treat the $t_m - maximum$ and the $t_n - maximum$ as a granule, respectively. And then we get a coarser granule by merging the two granules. The maximum flow of $[t_m, t_n]$ can be acquired by the property of additivity. The analysis of maximum flow in dynamic network could be decomposed into the analysis of the corresponding static networks by the concept of t-maximum and its additivity.

6 Conclusion

QST is an effective theory for complex problem solving, it provides a coarse to fine hierarchical description of complex problem. The applying of QST in complex network analysis is proposed in this paper. Based on QST, the method for solving shortest path by maximum cover network chain is proposed. For weighted network, the method of optimal path finding is proposed according to the characteristics of weighted network. In order to overcome the drawback that some of the static network concepts cannot be extended to dynamic network directly. We first decompose the dynamic network into a series of static networks. Then, analyze the static network and got the corresponding maximum flow ($t - maximum$) respectively. The maximum flow of dynamic network can be easily obtained by the property of additivity.

Acknowledgement. The work is supported by the National science Foundation of China (Nos.61175046 and 61073117).

References

1. Zhang, L., Zhang, B.: Theory and Application of Problem Solving Quotient space granular computing theory and methods. Tsinghua University Press, Beijing (2007) (in Chinese)
2. Zhang, L., Zhang, B.: Dynamic Quotient Space Model and Its Basic Properties. Artificial Intelligence and Pattern Recognition 25(2), 181–186 (2012) (in Chinese)
3. Zhang, L., Zhang, B.: Fuzzy tolerance quotient spaces and fuzzy subsets. Chinese science (Information Science) 53(4), 704–714 (2010) (in Chinese)
4. Zhang, L., Zhang, B., Zhang, Y.: The structural analysis of fuzzy measures. Chinese science (Information Science) 41(7), 820–832 (2011) (in Chinese)
5. Zhang, L., He, F.G., Zhang, Y.P., Zhao, S.: A New Algorithm for Optimal Path Finding in Complex Networks Based on the Quotient Space. Fundamenta Informaticae 93, 459–469 (2009)
6. He, F.-G., Zhang, Y.-P., Chen, J., Zhang, L.: Path queries on massive graphs based on multi-granular graph partitioning. In: Yao, J., Ramanna, S., Wang, G., Suraj, Z. (eds.) RSKT 2011. LNCS, vol. 6954, pp. 569–578. Springer, Heidelberg (2011)
7. He, F.G., Zhang, Y.P., Zhang, L.: Network granular storage and its application to path finding. Computer Applications and Software 28(11), 99–101,144 (2011) (in Chinese)
8. Zhao, S., Xu, X.S., Hua, B., Zhang, Y.P.: Contraction Network for Solving Maximum Flow Problem. In: Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, pp. 12–16. ACM (2012)
9. Zhang, Y.P., Xu, X.S., Hua, B., Zhao, S.: Contracting Community for Computing Maximum Flow. In: 2012 IEEE International Conference on Granular Computing, pp. 773–778. IEEE, China (2012)
10. Lee, D.S., Rieger, H.: Maximum flow and topological structure of complex networks. EPL (Europhysics Letters) 73(3), 471 (2006)
11. Kao, K.H., Chang, J.M., Wang, Y.L.: A quadratic algorithm for finding next-to-shortest paths in graphs. Algorithmica 61(2), 402–418 (2011)
12. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
13. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
14. Yao, Y.Y., Yao, B.: Covering based rough set approximations. Information Sciences 200(1), 91–107 (2012)
15. Cheng, Y., Miao, D.Q., Feng, Q.R.: Positive approximation and converse approximation in interval-valued fuzzy rough sets. Information Sciences 181(11), 2086–2110 (2011)
16. Yao, Y.Y., Zhang, N., Miao, D.Q.: Set-theoretic Approaches to Granular Computing. Fundamenta Informaticae 115(2), 247–264 (2012)
17. Xu, J., Shen, J., Wang, G.: Rough set theory analysis on decision subdivision. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 340–345. Springer, Heidelberg (2004)
18. Zhang, Q.H., Wang, G.Y., Xiao, Y.: Approximation sets of rough sets. Journal of Software 23(7), 1745–1759 (2012) (in Chinese)
19. Zhang, L., Zhang, B.: The quotient space theory of problem solving. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS (LNAI), vol. 2639, pp. 11–15. Springer, Heidelberg (2003)

Comparative Study between Extension of Covering Approximation Space and Its Induction through Transversal Matroid

Yanfang Liu and William Zhu*

Lab of Granular Computing
Minnan Normal University, Zhangzhou 363000, China
williamfengzhu@gmail.com

Abstract. Extension of a covering approximation space has been successfully applied to attribute reduction of covering-based rough sets. While the algorithms to solve attribute reduction are almost greedy ones. As a generalization of linear algebra and graph theory, matroids provide well-established platforms for greedy algorithms. In this paper, we introduce induction of a covering approximation space through transversal matroids, and then study its relationship with extension of the covering approximation space. Generally, the induced space of a covering approximation space generates more exact approximations than itself. Based on this, we investigate the relationship between induction of a covering approximation space and its extension. In fact, the induced space of a covering approximation space generates a bigger covering lower approximation and smaller covering upper approximation than the extended space. These interesting results demonstrate the potential for studying attribute reduction of covering-based rough sets by matroidal approaches.

Keywords: Covering-based rough set, transversal matroid, approximation operators, closure operator, attribute reduction.

1 Introduction

Rough set theory was proposed by Pawlak [10] to deal with granularity and vagueness in data analysis. The advantage of rough set theory is that it does not need any additional information about data, it has been successfully applied to various fields such as process control, economics, medical diagnosis, biochemistry, environmental science, biology, chemistry, psychology, and conflict analysis. However, the classical rough set theory is based on an equivalence relation or a partition on a universe, which is too restrictive for many applications. Scholars have proposed several interesting and important extensions of the rough set model [6, 9, 11–14, 17, 22, 23, 25, 26, 30, 31].

Particularly, through a covering instead of a partition on a universe, Zakowski [26] defined the concepts of covering lower and upper approximation operators and introduced covering-based rough sets. A covering information system, in which each attribute induces a covering rather than a partition, is emerged. Since then, many authors studied properties of covering lower and upper approximation operators [11, 27–29, 32]

* Corresponding author.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 225–235, 2013.

and employed covering-based rough sets to deal with covering information/decision systems [1, 4, 17–19]. In order to obtain a smaller reduction than the existing methods, Wang et al. [19] introduced extension of a covering approximation space and successfully applied it to attribute reduction of covering decision systems. The key idea of [19] is that the approximation ability of a covering is improved by extension since the extended space of a covering approximation space generates more exact approximations than itself. While the algorithms to solve the problem of attribute reduction are almost greedy ones. As a generalization of linear algebra and graph theory, matroids provide well-established platforms for greedy algorithms. Recently, many authors have combined rough sets and matroids [2, 5, 7, 8, 15, 16, 20, 21, 33] and employed matroidal approaches to attribute reduction [20]. In this paper, we introduce induction of a covering approximation space through transversal matroids and compare it to extension of the space.

On the one hand, for a covering of a universe, it can induce a transversal matroid. We present an expression of the closure of any single point set with respect to this transversal matroid through the covering itself. Through the closure of any single point set with respect to this transversal matroid, a new covering from the matroid is generated. We call the ordered pair with the universe and the new covering the induced space of the original covering approximation space. Moreover, we prove the induced space of a covering approximation space generates a bigger covering lower approximation and a smaller covering upper approximation than itself. On the other hand, based on the above results, we study the relationship between induction of a covering approximation space and its extension. Generally, the induced space of a covering approximation space generates more exact approximations than the extended space. These interesting results suggest the potential for employing matroids to study some problems of covering-based rough sets, such as attribute reduction.

The remainder of this paper is organized as follows. In Section 2, some basic definitions and related results about covering-based rough sets and matroids are introduced. Section 3 proposes the induced space of a covering approximation space through transversal matroids and studies its relationship with the original covering. In Section 4, for a covering approximation space, we compare the induced space and the extended one on approximation ability. Section 5 concludes this paper.

2 Preliminaries

In this section, we review some basic definitions and related results of covering-based rough sets and matroids.

2.1 Covering-Based Rough Set Model

Some relevant concepts of covering-based rough sets will be introduced in this subsection [30].

Definition 1. *(Covering) Let U be a universe of discourse and $\mathbf{C}$ a family of subsets of U. If none of subsets in $\mathbf{C}$ is empty and $\cup\mathbf{C} = U$, then $\mathbf{C}$ is called a covering of U.*

Definition 2. *(Covering approximation space) Let U be a universe and $\mathbf{C}$ a covering of U. We call the ordered pair $(U, \mathbf{C})$ a covering approximation space.*

Neighborhood is an important concept in covering-based rough sets and has been widely applied to knowledge classification and feature selection.

Definition 3. *(Neighborhood) Let $(U, \mathbf{C})$ be a covering approximation space. For all $x \in U$, $N_{\mathbf{C}}(x) = \cap\{K \in \mathbf{C} : x \in K\}$ is called the neighborhood of x with respect to $\mathbf{C}$.*

A pair of covering lower and upper approximation operators were proposed through the concept of neighborhood.

Definition 4. *(Covering lower and upper approximation operators) Let $(U, \mathbf{C})$ be a covering approximation space. For any $X \subseteq U$,*

$$L_{\mathbf{C}}(X) = \{x \in U : N_{\mathbf{C}}(x) \subseteq X\},$$
$$H_{\mathbf{C}}(X) = \{x \in U : N_{\mathbf{C}}(x) \cap X \neq \emptyset\},$$

where $L_{\mathbf{C}}, H_{\mathbf{C}}$ are covering lower, upper approximation operators, respectively.

2.2 Matroid Model

There are many different but equivalent ways to define a matroid. In the following definition, we introduce a matroid from the viewpoint of independent sets.

Definition 5. *(Matroid [3]) A matroid is a pair $\mathcal{M} = (U, \mathcal{I})$ consisting of a finite universe U and a collection $\mathcal{I}$ of subsets of U called independent sets satisfying the following three properties:*
(I1) $\emptyset \in \mathcal{I}$;
(I2) If $I \in \mathcal{I}$ and $I' \subseteq I$, then $I' \in \mathcal{I}$;
(I3) If $I_1, I_2 \in \mathcal{I}$ and $|I_1| < |I_2|$, then there exists $u \in I_2 - I_1$ such that $I_1 \cup \{u\} \in \mathcal{I}$, where $|I|$ denotes the cardinality of I.

Since the above definition of matroids focuses on independent sets, it is also called the independent set axioms of matroids. In a matroid, the rank function generalizes the maximal independence in vector subspaces.

Definition 6. *(Rank function [3]) Let $\mathcal{M} = (U, \mathcal{I})$ be a matroid and $X \subseteq U$.*

$$r_{\mathcal{M}}(X) = max\{|I| : I \subseteq X, I \in \mathcal{I}\},$$

where $r_{\mathcal{M}}$ is called the rank function of $\mathcal{M}$.

Through the dependency between an element and a subset of a universe, the closure operator of a matroid is introduced.

Definition 7. *(Closure operator [3]) Let $\mathcal{M} = (U, \mathcal{I})$ be a matroid and $X \subseteq U$. For any $u \in U$, if $r_{\mathcal{M}}(X) = r_{\mathcal{M}}(X \cup \{u\})$, then u depends on X. The subset of all elements depending on X of U is called the closure of X with respect to $\mathcal{M}$ and denoted by $cl_{\mathcal{M}}(X)$:*

$$cl_{\mathcal{M}}(X) = \{u \in U : r_{\mathcal{M}}(X) = r_{\mathcal{M}}(X \cup \{u\})\},$$

where $cl_{\mathcal{M}}$ is called the closure operator of $\mathcal{M}$.

The closure operator uniquely determines the matroid and vice versa. That is, we can introduce a matroid in terms of closure operators.

Proposition 1. *(Closure axioms [3]) Let* $cl : 2^U \to 2^U$ *be an operator. Then there exists a matroid* $\mathcal{M}$ *such that* $cl = cl_{\mathcal{M}}$ *iff* cl *satisfies the following conditions:*
(CL1) For all $X \subseteq U$, $X \subseteq cl(X)$;
(CL2) For all $X, Y \subseteq U$, *if* $X \subseteq Y$, *then* $cl(X) \subseteq cl(Y)$;
(CL3) For all $X \subseteq U$, $cl(cl(X)) = cl(X)$;
(CL4) For all $X \subseteq U, x, y \in U$, *if* $y \in cl(X \cup \{x\}) - cl(X)$, *then* $x \in cl(X \cup \{y\})$.

3 Induction of Covering Approximation Space through Transversal Matroid

As a branch of matroid theory, transversal theory reflects the relationships between collections of subsets of a nonempty set and their matroidal structures. It presents how to induce a matroid, namely, transversal matroid, by a family of subsets of a set. We first introduce the notion of transversal matroid.

Definition 8. *(Transversal [3]) Let* $\mathbf{F} = \mathbf{F}(J) = \{F_j : j \in J\}$ *be a family of subsets of* U. *A transversal of* $\mathbf{F}$ *is a set* $T \subseteq U$ *for which there exists a bijection* $\pi : T \to J$ *such that* $t \in F_{\pi(t)}$. *A partial transversal of* $\mathbf{F}$ *is a transversal of its subfamily.*

In order to illustrate tansversals of any family of a nonempty set, we present the following example.

Example 1. Suppose $U = \{a, b, c, d\}$, $\mathbf{F} = \mathbf{F}(J) = \{K_1, K_2, K_3\}$ and its index set $J = \{1, 2, 3\}$, where $K_1 = \{a, b, c\}$, $K_2 = \{c, d\}$ and $K_3 = \{b, d\}$. Then $T = \{a, b, c\}$ is a transversal of $\mathbf{F}$ since there exists a bijection $\pi : T \to J$ such that $t \in K_{\pi(t)}$, where $\pi(a) = 1, \pi(c) = 2$ and $\pi(b) = 3$. Similarly, suppose $J' = \{1, 2\} \subseteq J$. Then $T' = \{c, d\}$ is a transversal of $\mathbf{F}(J')$, so it is also a partial transversal of $\mathbf{F}$.

For a family of subsets of a universe, all its partial transversals satisfy the independent set axioms of matroids. Therefore, a matroid, called transversal matroid, is introduced based on a family of subsets of a universe.

Definition 9. *(Transversal matroid [3]) Let* $\mathbf{F} = \mathbf{F}(J) = \{F_j : j \in J\}$ *be a family of subsets of* U. *We call* $\mathcal{M}(\mathbf{F}) = (U, \mathcal{I}(\mathbf{F}))$ *the transversal matroid induced by* $\mathbf{F}$, *where* $\mathcal{I}(\mathbf{F})$ *is the family of all the partial transversals of* $\mathbf{F}$.

As a special family of a universe, a covering can generate a transversal matroid. An example is presented to illustrate transversal matroids based on coverings.

Example 2. Suppose $U = \{a, b, c, d\}$ and $\mathbf{C} = \{K_1, K_2\}$ is a covering of U, where $K_1 = \{a, b, c\}, K_2 = \{c, d\}$. According to Definition 8, $\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, c\}$, $\{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}$ are partial transversals of $\mathbf{C}$. Therefore $\mathcal{I}(\mathbf{C}) = \{\emptyset, \{a\}$, $\{b\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}\}$, where $\mathcal{I}(\mathbf{C})$ is the family of independent sets of the transversal matroid induced by $\mathbf{C}$.

From the above definition, a transversal matroid can be induced by a covering through transversal theory. Conversely, a covering is also induced by a matroid.

Proposition 2. *Let $\mathcal{M} = (U, \mathcal{I})$ be a matroid. Then $\{cl_{\mathcal{M}}(\{x\}) : x \in U\}$ is a covering of U.*

Proof. According to (CL1) of Proposition 1, we have $x \in cl_{\mathcal{M}}(\{x\})$ for any $x \in U$. Then $\bigcup_{x \in U} cl_{\mathcal{M}}(\{x\}) = U$. Therefore, $\{cl_{\mathcal{M}}(\{x\}) : x \in U\}$ is a covering of U.

For a matroid $\mathcal{M}$, we denote the covering $\{cl_{\mathcal{M}}(\{x\}) : x \in U\}$ as $\mathbf{C}(\mathcal{M})$.

Proposition 3. *Let $\mathcal{M} = (U, \mathcal{I})$ be a matroid. For all $x \in U$,*

$$N_{\mathbf{C}(\mathcal{M})}(x) = cl_{\mathcal{M}}(\{x\}).$$

Proof. According to (CL1) of Proposition 1, $x \in cl_{\mathcal{M}}(\{x\})$ for all $x \in U$. According to (CL2) and (CL3) of Proposition 1, for any $x, y \in U$, if $x \in cl_{\mathcal{M}}(\{y\})$, then $cl_{\mathcal{M}}(\{x\}) \subseteq cl_{\mathcal{M}}(\{y\})$. According to Definition 3 and Proposition 2, $N_{\mathbf{C}(\mathcal{M})}(x) = \bigcap_{x \in cl_{\mathcal{M}}(\{y\})} cl_{\mathcal{M}}(\{y\}) = cl_{\mathcal{M}}(\{x\})$.

Through the above two inductions between a covering and a matroid, an induced space of a covering approximation space is obtained.

Definition 10. *Let $(U, \mathbf{C})$ be a covering approximation space and $\mathcal{M}(\mathbf{C})$ the transversal matroid. The ordered pair $(U, \mathbf{C}(\mathcal{M}(\mathbf{C})))$ is called the induced space of $(U, \mathbf{C})$ through the transversal matroid, where $\mathbf{C}(\mathcal{M}(\mathbf{C}))$ is called the induction of $\mathbf{C}$ through the transversal matroid.*

In order to investigate the relationships between covering lower approximations of a covering approximation space and ones of its induced space, we study the closure of any single point set in the following theorem. First, we introduce the concept of repeat degree with respect to a covering.

Definition 11. *(Repeat degree [24]) Let $(U, \mathbf{C})$ be a covering approximation space. For all $X \subseteq U$, $|\{K \in \mathbf{C} : X \subseteq K\}|$ is called the repeat degree of X with respect to $\mathbf{C}$ and denoted as $d_{\mathbf{C}}(X)$.*

We give an example to illustrate the notion of repeat degree as follows.

Example 3. (Continued from Example 2) Suppose $X = \{a, c, d\}$ and $Y = \{b, c\}$. Then $d_{\mathbf{C}}(X) = |\{K \in \mathbf{C} : X \subseteq K\}| = |\emptyset| = 0$ and $d_{\mathbf{C}}(Y) = |\{K \in \mathbf{C} : Y \subseteq K\}| = |\{K_1\}| = 1$.

Theorem 1. *Let $(U, \mathbf{C})$ be a covering approximation space and $\mathcal{M}(\mathbf{C})$ the transversal matroid. For all $x \in U$,*

$$cl_{\mathcal{M}(\mathbf{C})}(\{x\}) = \{x\} \cup \{u \in U : d_{\mathbf{C}}(\{x\}) = d_{\mathbf{C}}(\{u\}) = d_{\mathbf{C}}(\{x, u\}) = 1\}.$$

Proof. According to Definition 6, we need to prove that $r_{\mathcal{M}(\mathbf{C})}(\{x\}) = r_{\mathcal{M}(\mathbf{C})}(\{x, u\}) \Leftrightarrow d_{\mathbf{C}}(\{x\}) = d_{\mathbf{C}}(\{u\}) = d_{\mathbf{C}}(\{x, u\}) = 1$. According to Definitions 6, 8, 9 and 11, $r_{\mathcal{M}(\mathbf{C})}(\{x\}) = r_{\mathcal{M}(\mathbf{C})}(\{x, u\}) \Leftrightarrow \{x, u\} \notin \mathcal{I}(\mathbf{C}) \Leftrightarrow \{x, u\}$ is not a partial transversal of $\mathbf{C} \Leftrightarrow d_{\mathbf{C}}(\{x\}) = d_{\mathbf{C}}(\{u\}) = d_{\mathbf{C}}(\{x, u\}) = 1$.

Example 4. (Continued from Example 2) Since $d_{\mathbf{C}}(\{a\}) = 1, d_{\mathbf{C}}(\{b\}) = 1, d_{\mathbf{C}}(\{c\}) = 2, d_{\mathbf{C}}(\{d\}) = 1, d_{\mathbf{C}}(\{a,b\}) = 1, d_{\mathbf{C}}(\{a,c\}) = 1, d_{\mathbf{C}}(\{a,d\}) = 0, d_{\mathbf{C}}(\{b,c\}) = 1, d_{\mathbf{C}}(\{b,d\}) = 0$ and $d_{\mathbf{C}}(\{c,d\}) = 1$. Therefore,
$cl_{\mathcal{M}(\mathbf{C})}(\{a\}) = \{a,b\}$; $cl_{\mathcal{M}(\mathbf{C})}(\{c\}) = \{c\}$;
$cl_{\mathcal{M}(\mathbf{C})}(\{b\}) = \{a,b\}$; $cl_{\mathcal{M}(\mathbf{C})}(\{d\}) = \{d\}$.

In fact, the covering induced by a matroid is a partition of the universe. The following proposition is presented to confirm this.

Proposition 4. *Let $(U, \mathbf{C})$ be a covering approximation space and $\mathcal{M}(\mathbf{C})$ the transversal matroid. Then $\mathbf{C}(\mathcal{M}(\mathbf{C}))$ is a partition of U.*

Proof. According to Proposition 2, we have $\mathbf{C}(\mathcal{M}(\mathbf{C})) = \{cl_{\mathcal{M}(\mathbf{C})}(\{x\}) : x \in U\}$ is a covering of U. Then we need to prove only for any $x, y \in U$, $cl_{\mathcal{M}(\mathbf{C})}(\{x\}) \cap cl_{\mathcal{M}(\mathbf{C})}(\{y\}) = \emptyset$ if $cl_{\mathcal{M}(\mathbf{C})}(\{x\}) \neq cl_{\mathcal{M}(\mathbf{C})}(\{y\})$.
Suppose $cl_{\mathcal{M}(\mathbf{C})}(\{x\}) \cap cl_{\mathcal{M}(\mathbf{C})}(\{y\}) \neq \emptyset$. Then there exists $z \in U$ such that $z \in cl_{\mathcal{M}(\mathbf{C})}(\{x\})$ and $z \in cl_{\mathcal{M}(\mathbf{C})}(\{y\})$. According to Theorem 1, we obtain $d_{\mathbf{C}}(\{x\}) = d_{\mathbf{C}}(\{z\}) = d_{\mathbf{C}}(\{x,z\}) = 1$ and $d_{\mathbf{C}}(\{y\}) = d_{\mathbf{C}}(\{z\}) = d_{\mathbf{C}}(\{z,y\}) = 1$. Therefore, $x \in cl_{\mathcal{M}(\mathbf{C})}(\{z\})$ and $y \in cl_{\mathcal{M}(\mathbf{C})}(\{z\})$. That is, $cl_{\mathcal{M}(\mathbf{C})}(\{x\}) = cl_{\mathcal{M}(\mathbf{C})}(\{z\})$ and $cl_{\mathcal{M}(\mathbf{C})}(\{y\}) = cl_{\mathcal{M}(\mathbf{C})}(\{z\})$, i.e., $cl_{\mathcal{M}(\mathbf{C})}(\{x\}) = cl_{\mathcal{M}(\mathbf{C})}(\{y\})$ which is contradictory with $cl_{\mathcal{M}(\mathbf{C})}(\{x\}) \neq cl_{\mathcal{M}(\mathbf{C})}(\{y\})$. Hence for any $x, y \in U$, if $cl_{\mathcal{M}(\mathbf{C})}(\{x\}) \neq cl_{\mathcal{M}(\mathbf{C})}(\{y\})$, then $cl_{\mathcal{M}(\mathbf{C})}(\{x\}) \cap cl_{\mathcal{M}(\mathbf{C})}(\{y\}) = \emptyset$.

Covering lower and upper approximation operators of this paper are based on neighborhoods. The following proposition presents the relationship between the neighborhoods of a covering approximation space and the ones of its induced space.

Proposition 5. *Let $(U, \mathbf{C})$ be a covering approximation space. For any $x \in U$,*
$$N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq N_{\mathbf{C}}(x).$$

Proof. According to Proposition 3 and Theorem 1, $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) = \{x\} \cup \{u \in U : d_{\mathbf{C}}(\{x\}) = d_{\mathbf{C}}(\{u\}) = d_{\mathbf{C}}(\{x,u\}) = 1\}$. We prove $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq N_{\mathbf{C}}(x)$ under two different conditions.
(1) $d_{\mathbf{C}}(\{x\}) = 1$.
According to Definition 11, $|\{K \in \mathbf{C} : x \in K\}| = 1$. Suppose $x \in K_x \in \mathbf{C}$. Therefore $N_{\mathbf{C}}(x) = K_x$. For any $u \in N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x)$, $d_{\mathbf{C}}(\{x,u\}) = 1$, then $u \in K_x$, i.e., $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq K_x$. Hence $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq N_{\mathbf{C}}(x)$.
(2) $d_{\mathbf{C}}(\{x\}) \neq 1$.
We see $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) = \{x\}$. Since $x \in N_{\mathbf{C}}(x)$, then $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq N_{\mathbf{C}}(x)$.

Based on the above proposition, we obtain that the induced space of a covering approximation space generates a bigger covering lower approximation and a smaller covering upper approximation than itself.

Theorem 2. *Let $(U, \mathbf{C})$ be a covering approximation space. For all $X \subseteq U$, $L_{\mathbf{C}}(X) \subseteq L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X)$ and $H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X) \subseteq H_{\mathbf{C}}(X)$.*

Proof. For all $x \in L_{\mathbf{C}}(X)$, according to Definition 4, $N_{\mathbf{C}}(x) \subseteq X$. According to Proposition 8, $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq N_{\mathbf{C}}(x)$. Therefore, $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq X$. Hence

$x \in L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X)$, i.e., $L_{\mathbf{C}}(X) \subseteq L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X)$.
For all $x \in H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X)$, $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \cap X \neq \emptyset$. Since $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq N_{\mathbf{C}}(x)$, then $N_{\mathbf{C}}(x) \cap X \neq \emptyset$. Therefore, $x \in H_{\mathbf{C}}(X)$, i.e., $H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X) \subseteq H_{\mathbf{C}}(X)$.

According to Theorem 2, we see the approximation ability of a covering can be improved by induction. The following example is given to confirm this.

Example 5. (Continued from Example 2) Suppose $X_1 = \{a, b\}$ and $X_2 = \{c, d\}$. Then the covering lower and upper approximations of X_1 and X_2 in $(U, \mathbf{C})$ are as follows.
$L_{\mathbf{C}}(X_1) = \emptyset, H_{\mathbf{C}}(X_1) = \{a, b\}$;
$L_{\mathbf{C}}(X_2) = \{c, d\}, H_{\mathbf{C}}(X_2) = \{a, b, c, d\}$.
According to Definition 6 and Definition 7, $cl_{\mathcal{M}(\mathbf{C})}(\{a\}) = cl_{\mathcal{M}(\mathbf{C})}(\{b\}) = \{a, b\}$, $cl_{\mathcal{M}(\mathbf{C})}(\{c\}) = \{c\}, cl_{\mathcal{M}(\mathbf{C})}(\{d\}) = \{d\}$. Then $\mathbf{C}(\mathcal{M}(\mathbf{C})) = \{\{a, b\}, \{c\}, \{d\}\}$. Then the covering lower and upper approximations of X_1 and X_2 in $(U, \mathbf{C}(\mathcal{M}(\mathbf{C})))$ are as follows.
$L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_1) = H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_1) = \{a, b\}$;
$L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_2) = H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_2) = \{c, d\}$.
Therefore, the covering lower and upper approximations of X_1 and X_2 in $(U, \mathbf{C})$ and $(U, \mathbf{C}(\mathcal{M}(\mathbf{C})))$ have the following relationships.
$L_{\mathbf{C}}(X_1) \subseteq L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_1)$; $H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_1) \subseteq H_{\mathbf{C}}(X_1)$;
$L_{\mathbf{C}}(X_2) \subseteq L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_2)$; $H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_2) \subseteq H_{\mathbf{C}}(X_2)$.

4 A Comparison between Extension of Covering Approximation Space and Its Induction through Transversal Matroid

In this section, we introduce extension of a covering space and compare it with the induction proposed in Section 3. In real problems, if an element of a covering includes all the objects in a universe, it is clear that this kind of element is useless for problem solving. Therefore, Wang and Hu [19] did not discuss coverings with this kind of element. In this paper, we also do not discuss this case.

For any X of a universe U, we denote $\sim X$ as the complement of X in U. At first, we introduce the concepts of the complement of a covering and the extension of the covering.

Definition 12. *([19]) Let $(U, \mathbf{C})$ be a covering approximation space. $\mathbf{C}^{\sim} = \{\sim K : K \in \mathbf{C}\}$ is called the complement of $\mathbf{C}$.*

Example 6. (Continued from Example 2) By the above definition, we have $\mathbf{C}^{\sim} = \{\sim K_1, \sim K_2\}$, where $\sim K_1 = \{d\}, \sim K_2 = \{a, b\}$.

Definition 13. *([19]) Let $(U, \mathbf{C})$ be a covering approximation space. The ordered pair $(U, \mathbf{C}^{\sharp})$ is called the extended space of $(U, \mathbf{C})$, where $\mathbf{C}^{\sharp} = \mathbf{C} \cup \mathbf{C}^{\sim}$ is the extension of $\mathbf{C}$.*

Wang and Hu have proved that the extended space of a covering approximation space generates more exact approximations than itself.

Proposition 6. *([19]) Let $(U, \mathbf{C})$ be a covering approximation space and $X \subseteq U$. Then $L_{\mathbf{C}}(X) \subseteq L_{\mathbf{C}^\sharp}(X)$ and $H_{\mathbf{C}^\sharp}(X) \subseteq H_{\mathbf{C}}(X)$.*

In order to compare the covering lower and upper approximations of the induced space and ones of the extended space, we investigate the relationship between the neighborhoods of the induction and ones of the extension. We first study some properties of the neighborhoods of extension of a covering.

Lemma 1. *Let $(U, \mathbf{C})$ be a covering approximation space. Then $\mathbf{C}^\sharp = (\mathbf{C}^\sharp)^\sim$.*

Proposition 7. *Let $(U, \mathbf{C})$ be a covering approximation space. Then $\{N_{\mathbf{C}^\sharp}(x) : x \in U\}$ is a partition of U.*

Proof. We see $\bigcup_{x \in U} N_{\mathbf{C}^\sharp}(x) = U$. Then we need to prove for any $x, y \in U$, if $N_{\mathbf{C}^\sharp}(x) \neq N_{\mathbf{C}^\sharp}(y)$, then $N_{\mathbf{C}^\sharp}(x) \cap N_{\mathbf{C}^\sharp}(y) = \emptyset$.
We first prove if $z \in N_{\mathbf{C}^\sharp}(x)$, then $x \in N_{\mathbf{C}^\sharp}(z)$. According to Definition 3 and Lemma 1, $z \in N_{\mathbf{C}^\sharp}(x) \Leftrightarrow \forall K \in \mathbf{C}^\sharp(x \in K \rightarrow z \in K) \Leftrightarrow \forall K \in \mathbf{C}^\sharp(z \in\sim K \rightarrow x \in\sim K) \Leftrightarrow \forall \sim K \in \mathbf{C}^\sharp(z \in\sim K \rightarrow x \in\sim K) \Leftrightarrow x \in N_{\mathbf{C}^\sharp}(z)$. That is, if $z \in N_{\mathbf{C}^\sharp}(x)$, then $N_{\mathbf{C}^\sharp}(x) = N_{\mathbf{C}^\sharp}(z)$.
Suppose $N_{\mathbf{C}^\sharp}(x) \cap N_{\mathbf{C}^\sharp}(y) \neq \emptyset$. Then there exists $z \in U$ such that $z \in N_{\mathbf{C}^\sharp}(x)$ and $z \in N_{\mathbf{C}^\sharp}(y)$, i.e., $N_{\mathbf{C}^\sharp}(x) = N_{\mathbf{C}^\sharp}(z)$ and $N_{\mathbf{C}^\sharp}(y) = N_{\mathbf{C}^\sharp}(z)$. In other words, $N_{\mathbf{C}^\sharp}(x) = N_{\mathbf{C}^\sharp}(y)$ which is contradictory with the condition $N_{\mathbf{C}^\sharp}(x) \neq N_{\mathbf{C}^\sharp}(y)$. Therefore, for any $x, y \in U$, if $N_{\mathbf{C}^\sharp}(x) \neq N_{\mathbf{C}^\sharp}(y)$, then $N_{\mathbf{C}^\sharp}(x) \cap N_{\mathbf{C}^\sharp}(y) = \emptyset$.

In order to illustrate the above result, we give an example as follows.

Example 7. (Continued from Examples 2 and 6) We have $\mathbf{C}^\sharp = \{\{a, b, c\}, \{c, d\}, \{d\}, \{a, b\}\}$. Then, $N_{\mathbf{C}^\sharp}(a) = \{a, b\}$, $N_{\mathbf{C}^\sharp}(b) = \{a, b\}$, $N_{\mathbf{C}^\sharp}(c) = \{c\}$ and $N_{\mathbf{C}^\sharp}(d) = \{d\}$. Therefore, $\{N_{\mathbf{C}^\sharp}(x) : x \in U\}$ is a partition of U.

In the following proposition, we will study the relationship between the neighborhoods of the induction and ones of the extension.

Proposition 8. *Let $(U, \mathbf{C})$ be a covering approximation space and $x \in U$.*

$$N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq N_{\mathbf{C}^\sharp}(x).$$

Proof. According to Proposition 3 and Theorem 1, $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) = \{x\} \cup \{u \in U : d_{\mathbf{C}}(\{x\}) = d_{\mathbf{C}}(\{u\}) = d_{\mathbf{C}}(\{x, u\}) = 1\}$. We prove $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq N_{\mathbf{C}^\sharp}(x)$ under two different conditions.
(1) $d_{\mathbf{C}}(\{x\}) = 1$.
Suppose $x \in K_x \in \mathbf{C}$. For any $u \in N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x)$, $d_{\mathbf{C}}(\{u\}) = d_{\mathbf{C}}(\{x, u\}) = 1$. According to Definition 11, we see that $u \in K_x$ and $u \notin K$, i.e., $u \in\sim K$ for all $K \in \mathbf{C} - \{K_x\}$. Therefore, $u \in N_{\mathbf{C}^\sharp}$. Hence $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq N_{\mathbf{C}^\sharp}(x)$.
(2) $d_{\mathbf{C}}(\{x\}) \neq 1$.
We see $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) = \{x\}$. Since $x \in N_{\mathbf{C}^\sharp}(x)$, then $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq N_{\mathbf{C}^\sharp}(x)$.

In the following theorem, we see the induced space of a covering approximation space generates a bigger covering lower approximation and a smaller covering upper approximation than the extend space of the covering approximation space.

Theorem 3. *Let* $(U, \mathbf{C})$ *be a covering approximation space and* $X \subseteq U$. *Then* $L_{\mathbf{C}^\sharp}(X) \subseteq L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X)$ *and* $H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X) \subseteq H_{\mathbf{C}^\sharp}(X)$.

Proof. According to Proposition 8, $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq N_{\mathbf{C}^\sharp}(x)$ for any $x \in U$.
For all $x \in L_{\mathbf{C}^\sharp}(X)$, according to Definition 4, $N_{\mathbf{C}^\sharp}(x) \subseteq X$. Therefore, $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq X$. Hence $x \in L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X)$, i.e., $L_{\mathbf{C}^\sharp}(X) \subseteq L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X)$.
For all $x \in H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X)$, $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \cap X \neq \emptyset$. Since $N_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(x) \subseteq N_{\mathbf{C}^\sharp}(x)$, then $N_{\mathbf{C}^\sharp}(x) \cap X \neq \emptyset$. Therefore, $x \in H_{\mathbf{C}^\sharp}(X)$, i.e., $H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X) \subseteq H_{\mathbf{C}^\sharp}(X)$.

The following example is presented to confirm that the approximation ability of induction of a covering is more stronger that extension of the covering.

Example 8. Let $U = \{a, b, c, d\}$ and $\mathbf{C} = \{K_1, K_2\}$ where $K_1 = \{a, b, c\}, K_2 = \{a, c, d\}$.
Suppose $X_1 = \{a, b\}$ and $X_2 = \{c, d\}$. Then the covering lower and upper approximations of X_1 and X_2 in $(U, \mathbf{C})$ are as follows.
$L_{\mathbf{C}}(X_1) = \emptyset, H_{\mathbf{C}}(X_1) = \{a, b, c, d\}$;
$L_{\mathbf{C}}(X_2) = \emptyset, H_{\mathbf{C}}(X_2) = \{a, b, c, d\}$.
According to the definition of extension of a covering, we have $\mathbf{C}^\sharp = \{K_1, K_2, \sim K_1, \sim K_2\} = \{\{a, b, c\}, \{a, c, d\}, \{d\}, \{b\}\}$. Thus, the covering lower and upper approximations of X_1 and X_2 in $(U, \mathbf{C}^\sharp)$ are as follows.
$L_{\mathbf{C}^\sharp}(X_1) = \{b\}, H_{\mathbf{C}^\sharp}(X_1) = \{a, b, c\}$;
$L_{\mathbf{C}^\sharp}(X_2) = \{d\}, H_{\mathbf{C}^\sharp}(X_2) = \{a, c, d\}$.
According to the definition of induction of a covering through its transversal matroid, we have $\mathbf{C}(\mathcal{M}(\mathbf{C})) = \{cl_{\mathcal{M}(\mathbf{C})}(x) : x \in U\} = \{\{a\}, \{b\}, \{c\}, \{d\}\}$. Hence, the covering lower and upper approximations of X_1 and X_2 in $(U, \mathbf{C}(\mathcal{M}(\mathbf{C})))$ are as follows.
$L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_1) = \{a, b\}, H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_1) = \{a, b\}$;
$L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_2) = \{c, d\}, H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_2) = \{c, d\}$.
The results show that the covering lower and upper approximations of X_1 and X_2 in $(U, \mathbf{C}), (U, \mathbf{C}^\sharp)$ and $(U, \mathbf{C}(\mathcal{M}(\mathbf{C})))$ have the following relationships.
$L_{\mathbf{C}}(X_1) \subseteq L_{\mathbf{C}^\sharp}(X_1) \subseteq L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_1)$; $H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_1) \subseteq H_{\mathbf{C}^\sharp}(X_1) \subseteq H_{\mathbf{C}}(X_1)$;
$L_{\mathbf{C}}(X_2) \subseteq L_{\mathbf{C}^\sharp}(X_2) \subseteq L_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_2)$; $H_{\mathbf{C}(\mathcal{M}(\mathbf{C}))}(X_2) \subseteq H_{\mathbf{C}^\sharp}(X_2) \subseteq H_{\mathbf{C}}(X_2)$.

5 Conclusions

In this paper, we propose induction of a covering approximation space through transversal matroids and study its relationship with extension of the covering approximation space. Generally, the induced space of a covering approximation space generates more exact approximations than the extended space. That is, the approximation ability of induction of a covering is more stronger than extension of the covering. These interesting results demonstrate the potential for studying attribute reduction of covering-based rough sets by matroidal approaches. However, the problem of attribute reduction in rough sets is NP-hard, and the algorithms to solve it are almost greedy ones. While matroids provide well-established platforms for greedy algorithm foundation and implementation. In future works, we will design an algorithm by matroidal approaches to solve some problems in covering-based rough sets, such as attribute reduction.

Acknowledgments. This work is in part supported by the National Science Foundation of China under Grant No. 61170128, the Natural Science Foundation of Fujian Province, China under Grant No. 2012J01294, the Fujian Province Foundation of Higher Education under Grant No. JK2012028, and the Postgraduate Education Innovation Base for Computer Application Technology, Signal and Information Processing of Fujian Province (No. [2008]114, High Education of Fujian).

References

1. Chen, D., Wang, C., Hu, Q.: A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets. Information Sciences 177, 3500–3518 (2007)
2. Huang, A., Zhu, W.: Geometric lattice structure of covering-based rough sets through matroids. Journal of Applied Mathematics 2012, Article ID 236307, 25 pages (2012)
3. Lai, H.: Matroid theory. Higher Education Press, Beijing (2001)
4. Li, F., Yin, Y.: Approaches to knowledge reduction of covering decision systems based on information theory. Information Sciences 179, 1694–1704 (2009)
5. Li, X., Liu, S.: Matroidal approaches to rough set theory via closure operators. International Journal of Approximate Reasoning 53, 513–527 (2012)
6. Lin, T.Y.: Neighborhood systems and relational databases. In: Proceedings of the 1988 ACM Sixteenth Annual Conference on Computer Science, p. 725. ACM (1988)
7. Liu, Y., Zhu, W.: Matroidal structure of rough sets based on serial and transitive relations. Journal of Applied Mathematics 2012, Article ID 429737, 16 pages (2012)
8. Liu, Y., Zhu, W., Zhang, Y.: Relationship between partition matroid and rough set through k-rank matroid. Journal of Information and Computational Science 8, 2151–2163 (2012)
9. Morsi, N., Yakout, M.: Axiomatics for fuzzy rough sets. Fuzzy Sets and Systems 100, 327–342 (1998)
10. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
11. Pomykala, J.A.: Approximation operations in approximation space. Bulletin of the Polish Academy of Sciences 35, 653–662 (1987)
12. Qian, Y., Dang, C., Liang, J., Tang, D.: Set-valued ordered information systems. Information Sciences 179, 2809–2832 (2009)
13. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
14. Slowinski, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Transactions on Knowledge and Data Engineering 12, 331–336 (2000)
15. Tang, J., She, K., Min, F., Zhu, W.: A matroidal approach to rough set theory. Theoretical Computer Science 471, 1–11 (2013)
16. Tang, J., She, K., Zhu, W.: Matroidal structure of rough sets from the viewpoint of graph theory. Journal of Applied Mathematics 2012, Article ID 973920, 27 pages (2012)
17. Tsang, E.C., Chen, D., Yeung, D.S.: Approximations and reducts with covering generalized rough sets. Computers & Mathematics with Applications 56, 279–289 (2008)
18. Wang, C., Chen, D., He, Q., Hu, Q.: A comparative study of ordered and covering information systems. Fundamenta Informaticae 122, 1–13 (2012)
19. Wang, G., Hu, J.: Attribute reduction using extension of covering approximation space. Fundamenta Informaticae 115, 219–232 (2012)
20. Wang, S., Zhu, Q., Zhu, W., Min, F.: Matroidal structure of rough sets and its characterization to attribute reduction. Knowledge-Based Systems 36, 155–161 (2012)

21. Wang, S., Zhu, Q., Zhu, W., Min, F.: Quantitative analysis for covering-based rough sets using the upper approximation number. Information Sciences 220, 483–491 (2013)
22. Wu, W., Zhang, W., Li, H.: Knowledge acquisition in incomplete fuzzy information systems via the rough set approach. Expert Systems 20, 280–286 (2003)
23. Xu, W., Zhang, W.: Measuring roughness of generalized rough sets induced by a covering. Fuzzy Sets and Systems 158, 2443–2455 (2007)
24. Yao, H., Zhu, W.: Conditions for a covering of neighborhoods to be a partition. Submitted to Information Sciences (2013)
25. Yao, Y.Y.: Relational interpretations of neighborhood operators and rough set approximation operators. Information Sciences 111, 239–259 (1998)
26. Zakowski, W.: Approximations in the space (u, π). Demonstratio Mathematica 16, 761–769 (1983)
27. Bonikowski, Z., Bryniarski, E., Skardowska, U.W.: Extensions and intentions in the rough set theory. Information Sciences 107, 149–167 (1998)
28. Zhu, W.: Topological approaches to covering rough sets. Information Sciences 177, 1499–1508 (2007)
29. Zhu, W.: Relationship among basic concepts in covering-based rough sets. Information Sciences 179, 2478–2486 (2009)
30. Zhu, W.: Relationship between generalized rough sets based on binary relation and covering. Information Sciences 179, 210–225 (2009)
31. Zhu, W., Wang, F.: Reduction and axiomization of covering generalized rough sets. Information Sciences 152, 217–230 (2003)
32. Zhu, W., Wang, F.: The fourth type of covering-based rough sets. Information Sciences 201, 80–92 (2012)
33. Zhu, W., Wang, S.: Rough matroids based on relations. Information Sciences 232, 241–252 (2013)

Multi-covering Based Rough Set Model

Lijuan Wang[1,⋆], Xibei Yang[1,2], and Chen Wu[1]

[1] School of Computer Science and Technology
Jiangsu University of Science and Engineering, Zhenjiang, Jiangsu, 212003, P.R. China
zjwanglijuan@sina.com
[2] Key Laboratory of Intelligent Perception and Systems for High–Dimensional Information
(Nanjing University of Science and Technology), Ministry of Education, Nanjing, Jiangsu,
210094, P.R. China

Abstract. In this paper, six types of optimistic multi-covering rough set models and six types of pessimistic multi-covering rough set models are proposed in multi-covering approximation space. From three different points of views, relationships among multi-covering rough set models are deeply investigated. They are relationships among optimistic multi-covering rough set models, relationships among pessimistic multi-covering rough set models, and relationships among optimistic and pessimistic multi-covering rough set models. The obtained results provide a theoretical foundation for the further discussions of multi-covering rough sets.

Keywords: Comparison, Multi-covering approximation space, Multigranulation, Rough set model.

1 Introduction

The covering rough set models are important extensions of the rough set model [1]. In [2], Samanta presented sixteen covering rough set models and studied their implication lattices, six of which were also being systematically studied by W. Zhu in [3,4,5,6,7,8]. The first covering rough set model was proposed by Zakowski [9] through extending Pawlaks rough set theory from partition to covering. Following Zakowski' s work, Pomykala proposed the second covering rough set model in 1987 [10]. His main method was the interior operator which is adopted by the topology theory. The definition of the third type of upper approximation operation [11] is believed to be more reasonable than those of the first and second types, but no properties of this new class of covering generalized rough sets have been discussed. By combining the definitions of three types of covering rough sets, Zhu and Wang proposed the fourth covering rough set model in [6]. From the topological point of view, Zhu presented the fifth covering rough set model in [7], and explored the detailed properties of lower and upper approximations for this new type of rough sets. It is worth noting that the lower approximations of these five models are same. Then, in [8], the sixth covering rough set model was defined by Zhu. This model includes not only the covering upper approximation but also the covering lower approximation.

⋆ Corresponding author.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 236–244, 2013.

From the viewpoint of the multigranulation approach, Qian and Liang et al. [12] proposed the concept of the multigranulation rough set by using a family of the equivalence relations instead of single one. And the multigranulation rough set models (MGRS) are given [12]. Their MGRS can be used to analyze distributed data, and it is more reasonable than Pawlak's rough set in practical applications [12,13].

By considering an evaluation system which involved many experts, the experts in the same field or in different fields may provide the results of the assessment independently. It can form a multi-covering approximation space on the domain. In the multi-covering approximation space, the multi-covering rough set models are proposed by combing covering rough set with multigranulation approach. A comparative analysis is adopted to study the relationships among these models. The research work is carried out from three aspects. They are the relationships among the optimistic multi-covering rough set models, the relationships among the pessimistic multi-covering rough set models, and the relationships among the optimistic and pessimistic multi-covering rough set models.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce the fundamental concepts of covering based rough set models and the multigranulation rough set model. In Section 3, multi-covering based rough set models is presented. Section 4 is focused on the relationships among multi-covering based rough set models. Results are summarized in Section 5.

2 Preliminary Concepts

2.1 Covering Based Rough Set Models

In this section, we will review some basic concepts which are involved in covering based rough set models. According to Zhu's work [3,4,5,6,7,8], six types of covering based rough set models are introduced as follows.

Definition 1. *[8] Let C be a covering of U, $x \in U$, the minimal description of x is defined as*

$$Md_C(x) = \{K \in C | x \in K \wedge (\forall S \in C \wedge x \in S \wedge S \subseteq K \Rightarrow K = S)\}. \quad (1)$$

Definition 2. *[8] Let C be a covering of U, $x \in U$, the neighborhood of x is defined as*

$$Neighbor_C(x) = \cap\{K | x \in K \in C\}. \quad (2)$$

The definitions of the six covering lower and upper approximations are given as follows.

Definition 3. *[8] Let C be a covering of U, $\forall X \subseteq U$. The first covering lower approximation set $CL_C(X)$ and the first covering upper approximation set $FH_C(X)$ with respect to the covering C are defined as follows:*

$$CL_C(X) = \cup\{K \in C | K \subseteq X\}; \quad (3)$$

$$FH_C(X) = CL_C(X) \cup \{Md(x) | x \in X - CL_C(X)\}. \quad (4)$$

Definition 4. *[8] Let C be a covering of U, $\forall X \subseteq U$. The second, the third, the fourth, and the fifth covering upper approximation sets with respect to the covering C are denoted by $SH_C(X)$, $TH_C(X)$, $RH_C(X)$, $IH_C(X)$, where*

$$SH_C(X) = \cup\{K|K \in C, K \cap X \neq \emptyset\}; \tag{5}$$
$$TH_C(X) = \cup\{Md_C(x)|x \in X\}; \tag{6}$$
$$RH_C(X) = CL_C(X) \cup \{K \in C|K \cap (X - CL_C(X)) \neq \emptyset\}; \tag{7}$$
$$IH_C(X) = CL_C(X) \cup \{Neighbor_C(x)|x \in X - CL_C(X)\}. \tag{8}$$

Definition 5. *[8] Let C be a covering of U, $\forall X \subseteq U$. The sixth covering lower approximation set $XL_C(X)$ and the sixth covering upper approximation set $XH_C(X)$ with respect to the covering C are defined as follows:*

$$XL_C(X) = \{x|Neighbor_C(x) \subseteq X\}; \tag{9}$$
$$XH_C(X) = \{x|Neighbor_C(x) \cap X \neq \emptyset\}. \tag{10}$$

Following Wang's work [14], relationships among six types of covering rough set models have been presented. The details are shown in Theorems 1.

Theorem 1. *[14] Let C be a covering of U, $\forall X \subseteq U$. The six types of covering rough set models have some inclusion relations as follows:*

1. $CL_C(X) \subseteq XL_C(X)$;
2. $IH_C(X) \subseteq FH_C(X) \subseteq TH_C(X) \subseteq SH_C(X)$;
3. $IH_C(X) \subseteq FH_C(X) \subseteq RH_C(X) \subseteq SH_C(X)$;
4. $XH_C(X) \subseteq SH_C(X)$.

2.2 Multigranulation Rough Set

The multigranulation rough set (MGRS) [12] is constructed on the basis of a family of indiscernibility relations, and it is different from Pawlak's rough set [1], which is constructed on the basis of a single indiscernibility relation.

In Qian's MGRS, two different models have been defined. The first one is the optimistic MGRS, the second one is the pessimistic MGRS [12].

The target of Qian's optimistic MGRS is approximated through a family of the indiscernibility relations. In lower approximation, the word "optimistic" is used to express the idea that in multi independent granular structures, we need only at least one granular structure to satisfy with the inclusion condition between equivalence class and target. The upper approximation of optimistic multigranulation rough set is defined by the complement of the lower approximation.

Definition 6. *[12] Let I be an information system in which $A_1, A_2, \cdots, A_m \subseteq AT$, then $\forall X \subseteq U$, the optimistic multigranulation lower and upper approximations are*

denoted by $\underline{\sum_{i=1}^{m} A_i}^{O}(X)$ *and* $\overline{\sum_{i=1}^{m} A_i}^{O}(X)$*, respectively,*

$$\underline{\sum_{i=1}^{m} A_i}^{O}(X) = \{x \in U | [x]_{A_1} \subseteq X \vee [x]_{A_2} \subseteq X \vee \cdots \vee [x]_{A_m} \subseteq X\}; \quad (11)$$

$$\overline{\sum_{i=1}^{m} A_i}^{O}(X) = \sim \underline{\sum_{i=1}^{m} A_i}^{O}(\sim X) \quad (12)$$

where $[x]_{A_i}$ $(1 \leq i \leq m)$ *is the equivalence class of* x *in terms of set of attributes* A_i*,* $\sim X$ *is the complement of set* X*.*

Theorem 2. *[12] Let* I *be an information system in which* $A_1, A_2, \cdots, A_m \subseteq AT$*, then* $\forall X \subseteq U$*, we have*

$$\overline{\sum_{i=1}^{m} A_i}^{O}(X) = \{x \in U | [x]_{A_1} \cap X \neq \emptyset \wedge [x]_{A_2} \cap X \neq \emptyset \wedge \cdots \wedge [x]_{A_m} \cap X \neq \emptyset\}.$$

By Theorem 2, we can see that though the optimistic multigranulation upper approximation is defined by the complement of the optimistic multigranulation lower approximation, it can also be considered as a set in which objects have non–empty intersection with the target in terms of each granular structure.

In Qian's pessimistic MGRS [12], the target is still approximated through a family of the indiscernibility relations. However, it is different from the optimistic case. In lower approximation, the word "pessimistic" is used to express the idea that in multi independent granular structures, we need all the granular structures to satisfy with the inclusion condition between equivalence class and target. The upper approximation of pessimistic multigranulation rough set is also defined by the complement of the pessimistic multigranulation lower approximation.

Definition 7. *[12] Let* I *be an information system in which* $A_1, A_2, \cdots, A_m \subseteq AT$*, then* $\forall X \subseteq U$*, the pessimistic multigranulation lower and upper approximations are denoted by* $\underline{\sum_{i=1}^{m} A_i}^{P}(X)$ *and* $\overline{\sum_{i=1}^{m} A_i}^{P}(X)$*, respectively,*

$$\underline{\sum_{i=1}^{m} A_i}^{P}(X) = \{x \in U | [x]_{A_1} \subseteq X \wedge [x]_{A_2} \subseteq X \wedge \cdots \wedge [x]_{A_m} \subseteq X\}; \quad (13)$$

$$\overline{\sum_{i=1}^{m} A_i}^{P}(X) = \sim \underline{\sum_{i=1}^{m} A_i}^{P}(\sim X). \quad (14)$$

Theorem 3. *[12] Let* I *be an information system in which* $A_1, A_2, \cdots, A_m \subseteq AT$*, then* $\forall X \subseteq U$*, we have*

$$\overline{\sum_{i=1}^{m} A_i}^{P}(X) = \{x \in U | [x]_{A_1} \cap X \neq \emptyset \vee [x]_{A_2} \cap X \neq \emptyset \vee \cdots \vee [x]_{A_m} \cap X \neq \emptyset\}.$$

Different from the upper approximation of optimistic multigranulation rough set, the upper approximation of pessimistic multigranulation rough set is represented as a set in which objects have non–empty intersection with the target in terms of at least one granular structure.

3 Multi-covering Based Rough Set Models

Definition 8. *Let U is a nonempty set, $C_1, C_2, \cdots, C_m$ are m coverings of U. If $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, then the ordered pair $(U, \mathbf{C})$ is called as multi-covering approximation space.*

3.1 Optimistic Multi-covering Based Rough Set Models

In the multi-covering approximation space, each covering lower approximation set induced by one single covering approximation space is regarded as the set of certain objects in multi-covering approximation space. They are one of the separate "granular". At least one "granular" in multi granular should meet the requirements. Thus, we call them optimistic models. The lower approximation set in the multi-covering approximation space is the union of covering lower approximation set induced by one single covering approximation space.

Definition 9. *In multi-covering approximation space $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. The first type of optimistic multi-covering lower approximation set $CL^O_{\mathbf{C}}(X)$ based on multi-covering $\mathbf{C}$ is defined as follows:*

$$CL^O_{\mathbf{C}}(X) = \bigcup_{i=1}^{m} CL_{C_i}(X). \tag{15}$$

Similar to Theorem 3, optimistic multi-covering upper approximation set can also obtained by the intersection operation on single covering upper approximation sets.

Definition 10. *In multi-covering approximation space $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. The first type of optimistic multi-covering upper approximation set $FH^O_{\mathbf{C}}(X)$ based on multi-covering $\mathbf{C}$ is defined as follows:*

$$FH^O_{\mathbf{C}}(X) = \bigcap_{i=1}^{m} FH_{C_i}(X). \tag{16}$$

Definition 11. *In multi-covering approximation space $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. The second, the third, the fourth, and the fifth types of optimistic multi-covering upper approximation sets with respect to multi-covering $\mathbf{C}$ are denoted by $SH^O_{\mathbf{C}}(X)$, $TH^O_{\mathbf{C}}(X)$, $RH^O_{\mathbf{C}}(X)$, and $IH^O_{\mathbf{C}}(X)$ respectively, where*

$$SH^O_{\mathbf{C}}(X) = \bigcap_{i=1}^{m} SH_{C_i}(X); \tag{17}$$

$$TH^O_{\mathbf{C}}(X) = \bigcap_{i=1}^{m} TH_{C_i}(X); \tag{18}$$

$$RH^O_{\mathbf{C}}(X) = \bigcap_{i=1}^{m} RH_{C_i}(X); \tag{19}$$

$$IH^O_{\mathbf{C}}(X) = \bigcap_{i=1}^{m} IH_{C_i}(X). \tag{20}$$

Definition 12. *In multi-covering approximation space* $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. *The six type of optimistic multi-covering lower and upper approximation sets with respect to multi-covering* $\mathbf{C}$ *are denoted by* $XL^O_{\mathbf{C}}(X)$ *and* $XH^O_{\mathbf{C}}(X)$ *respectively, where*

$$XL^O_{\mathbf{C}}(X) = \bigcup_{i=1}^{m} XL_{C_i}(X); \tag{21}$$

$$XH^O_{\mathbf{C}}(X) = \bigcap_{i=1}^{m} XH_{C_i}(X). \tag{22}$$

3.2 Pessimistic Multi-covering Based Rough Set Models

In the multi-covering approximation space, each covering lower approximation set induced by one single covering approximation space is regarded as one of the separate "granular". If all "granular" in multi granular must meet the requirements, we call them pessimistic models. The lower approximation set in the multi-covering approximation space is the intersection of covering lower approximation set induced by one single covering approximation space.

Definition 13. *In multi-covering approximation space* $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. *The first type of pessimistic multi-covering lower approximation set* $CL^P_{\mathbf{C}}(X)$ *based on multi-covering* $\mathbf{C}$ *is defined as follows:*

$$CL^P_{\mathbf{C}}(X) = \bigcap_{i=1}^{m} CL_{C_i}(X). \tag{23}$$

Similar to Theorem 3, pessimistic multi-covering upper approximation set can also obtained by the union operation of single covering upper approximation sets.

Definition 14. *In multi-covering approximation space* $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. *The first type of pessimistic multi-covering upper approximation set* $FH^P_{\mathbf{C}}(X)$ *based on multi-covering* $\mathbf{C}$ *is defined as follows:*

$$FH^P_{\mathbf{C}}(X) = \bigcup_{i=1}^{m} FH_{C_i}(X). \tag{24}$$

Definition 15. *In multi-covering approximation space* $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. *The second, the third, the fourth, and the fifth types of pessimistic multi-covering upper approximation sets with respect to multi-covering* $\mathbf{C}$ *are denoted by* $SH^P_{\mathbf{C}}(X)$, $TH^P_{\mathbf{C}}(X)$, $RH^P_{\mathbf{C}}(X)$, *and* $IH^P_{\mathbf{C}}(X)$ *respectively, where*

$$SH^P_{\mathbf{C}}(X) = \bigcup_{i=1}^{m} SH_{C_i}(X); \tag{25}$$

$$TH^P_{\mathbf{C}}(X) = \bigcup_{i=1}^{m} TH_{C_i}(X); \tag{26}$$

$$RH^P_{\mathbf{C}}(X) = \bigcup_{i=1}^{m} RH_{C_i}(X); \tag{27}$$

$$IH^P_{\mathbf{C}}(X) = \bigcup_{i=1}^{m} IH_{C_i}(X). \tag{28}$$

Definition 16. *In multi-covering approximation space* $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. *The six type of pessimistic multi-covering lower and upper approximation sets with respect to multi-covering* $\mathbf{C}$ *are denoted by* $XL^P_{\mathbf{C}}(X)$ *and* $XH^P_{\mathbf{C}}(X)$ *respectively, where*

$$XL^P_{\mathbf{C}}(X) = \bigcap_{i=1}^{m} XL_{C_i}(X); \tag{29}$$

$$XH^P_{\mathbf{C}}(X) = \bigcup_{i=1}^{m} XH_{C_i}(X). \tag{30}$$

4 Relationships among Multi-covering Based Rough Set Models

4.1 Relationships among Optimistic Multi-covering Based Rough Set Models

In this section, we will systematically explore the relationships between two optimistic multi-covering lower approximation sets and the relationships among six optimistic multi-covering upper approximation sets. In multi-covering approximation space $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. By comparing the two more optimistic covering lower approximation sets, we have the following theorem:

Theorem 4. $CL^O_{\mathbf{C}}(X) \subseteq XL^O_{\mathbf{C}}(X)$.

Theorem 4 shows that the first optimistic multi-covering lower approximation set is belong to the sixth optimistic multi-covering lower approximation set.

In multi-covering approximation space $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. By comparing with the six types of optimistic multi-covering upper approximation sets, the following some conclusions can also be found:

Theorem 5. $IH^O_{\mathbf{C}}(X) \subseteq FH^O_{\mathbf{C}}(X) \subseteq TH^O_{\mathbf{C}}(X) \subseteq SH^O_{\mathbf{C}}(X)$.

Theorem 6. $IH^O_{\mathbf{C}}(X) \subseteq FH^O_{\mathbf{C}}(X) \subseteq RH^O_{\mathbf{C}}(X) \subseteq SH^O_{\mathbf{C}}(X)$.

Theorem 7. $XH^O_{\mathbf{C}}(X) \subseteq SH^O_{\mathbf{C}}(X)$.

4.2 Relationships among Pessimistic Multi-covering Based Rough Set Models

In this section, we will systematically explore the relationships between two pessimistic multi-covering lower approximation sets and the relationships among six pessimistic multi-covering upper approximation sets. In multi-covering approximation space $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. By comparing the two more pessimistic covering lower approximation sets, we have the following theorem:

Theorem 8. $CL^P_{\mathbf{C}}(X) \subseteq XL^P_{\mathbf{C}}(X)$.

By comparing with the six types of pessimistic multi-covering upper approximation sets, the following several conclusions can also be found:

Theorem 9. $IH^P_{\mathbf{C}}(X) \subseteq FH^P_{\mathbf{C}}(X) \subseteq TH^P_{\mathbf{C}}(X) \subseteq SH^P_{\mathbf{C}}(X)$.

Theorem 10. $IH^P_{\mathbf{C}}(X) \subseteq FH^P_{\mathbf{C}}(X) \subseteq RH^P_{\mathbf{C}}(X) \subseteq SH^P_{\mathbf{C}}(X)$.

Theorem 11. $XH^P_{\mathbf{C}}(X) \subseteq SH^P_{\mathbf{C}}(X)$.

4.3 Relationships among Optimistic and Pessimistic Multi-covering Based Rough Set Models

In multi-covering approximation space $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. Relationships among optimistic and pessimistic multi-covering lower approximations are concluded as follows:

Theorem 12. $CL^P_{\mathbf{C}}(X) \subseteq CL^O_{\mathbf{C}}(X)$.

Theorem 13. $XL^P_{\mathbf{C}}(X) \subseteq XL^O_{\mathbf{C}}(X)$.

Theorem 14. *There is no inclusion relation between* $XL^P_{\mathbf{C}}(X)$ *and* $CL^O_{\mathbf{C}}(X)$.

Relationships among optimistic and pessimistic multi-covering upper approximations can also be found:

Theorem 15. *In multi-covering approximation space* $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. *We have*

1. $FH^O_{\mathbf{C}}(X) \subseteq FH^P_{\mathbf{C}}(X)$;
2. $SH^O_{\mathbf{C}}(X) \subseteq SH^P_{\mathbf{C}}(X)$;
3. $TH^O_{\mathbf{C}}(X) \subseteq TH^P_{\mathbf{C}}(X)$;
4. $RH^O_{\mathbf{C}}(X) \subseteq RH^P_{\mathbf{C}}(X)$;
5. $IH^O_{\mathbf{C}}(X) \subseteq IH^P_{\mathbf{C}}(X)$;
6. $XH^O_{\mathbf{C}}(X) \subseteq XH^P_{\mathbf{C}}(X)$.

Theorem 16. *In multi-covering approximation space* $(U, \mathbf{C})$, $\mathbf{C} = \{C_1, C_2, \cdots, C_m\}$, $\forall X \subseteq U$. *We have*

1. $FH^O_{\mathbf{C}}(X) \subseteq FH^P_{\mathbf{C}}(X) \subseteq TH^P_{\mathbf{C}}(X) \subseteq SH^P_{\mathbf{C}}(X)$;
2. $FH^O_{\mathbf{C}}(X) \subseteq FH^P_{\mathbf{C}}(X) \subseteq RH^P_{\mathbf{C}}(X) \subseteq SH^P_{\mathbf{C}}(X)$;
3. $TH^O_{\mathbf{C}}(X) \subseteq TH^P_{\mathbf{C}}(X) \subseteq SH^P_{\mathbf{C}}(X)$;
4. $RH^O_{\mathbf{C}}(X) \subseteq RH^P_{\mathbf{C}}(X) \subseteq SH^P_{\mathbf{C}}(X)$;
5. $IH^O_{\mathbf{C}}(X) \subseteq IH^P_{\mathbf{C}}(X) \subseteq FH^P_{\mathbf{C}}(X) \subseteq RH^P_{\mathbf{C}}(X) \subseteq SH^P_{\mathbf{C}}(X)$;
6. $IH^O_{\mathbf{C}}(X) \subseteq IH^P_{\mathbf{C}}(X) \subseteq FH^P_{\mathbf{C}}(X) \subseteq TH^P_{\mathbf{C}}(X) \subseteq SH^P_{\mathbf{C}}(X)$;
7. $XH^O_{\mathbf{C}}(X) \subseteq SH^P_{\mathbf{C}}(X)$.

5 Conclusions

In this paper, we have introduced the multigranulation theory into the multi-covering approximation space, and the optimistic and pessimistic multi-covering rough set models have been presented. Inclusion relations have been found among the optimistic and pessimistic multi-covering lower and upper approximation sets. We will further study on the relationships of approximations accuracy measures among multi-covering rough set models. Furthermore, the knowledge discovery method in the multi-covering approximation space will be carried out in the future.

Acknowledgments. This work is supported by the Natural Science Foundation of China (No. 61170128, 61100116), Natural Science Foundation of Jiangsu Province of China (No. BK2011492), Opening Foundation of Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, the Chinese Academy of Sciences (No. IIP 2012–3).

References

1. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences, 341–356 (1982)
2. Samanta, P., Chakraborty, M.K.: Covering based approaches to rough sets and implication lattices. In: Sakai, H., Chakraborty, M.K., Hassanien, A.E., Ślezak, D., Zhu, W. (eds.) RSFDGrC 2009. LNCS, vol. 5908, pp. 127–134. Springer, Heidelberg (2009)
3. Zhu, W., Wang, F.Y.: Properties of the First Type of Covering-Based Rough Sets. In: Proceedings of DM Workshop 2006, ICDM 2006, Hong Kong, China, pp. 407–411 (2006)
4. Zhu, W.: Properties of the second type of covering-based rough sets. In: Workshop Proceedings of GrC and BI 2006, IEEE WI 2006, Hong Kong, China, pp. 494–497 (2006)
5. Zhu, W., Wang, F.Y.: On three types of covering rough sets. IEEE Transactions on Knowledge and Data Engineering 19(8), 1131–1144 (2007)
6. Zhu, W., Wang, F.Y.: A new type of covering rough sets. In: IEEE IS 2006, London, pp. 444–449 (2006)
7. Zhu, W.: Topological approaches to covering rough sets. Information Science 177(6), 1499–1508 (2007)
8. Zhu, W.: Relationship between generalized rough sets based on binary relation and covering. Information Science 179(1), 210–225 (2009)
9. Zakowski, W.: Approximations in the space. Demonstration Mathematics, 761–769 (1983)
10. Pomykala, J.A.: Approximation operations in approximation space. Bulletin of the Polish Academy of Sciences 35(9-10), 653–662 (1987)
11. Tsang, E., Cheng, D., Lee, J., et al.: On the upper approximations of covering generalized rough sets. In: Proceedings of the 3rd International Conference Machine Learning and Cybernetics, pp. 4200–4203 (2004)
12. Qian, Y.H., Liang, J.Y., Yao, Y.Y., et al.: MGRS: a multi-granulation rough set. Information Science 180(6), 949–970 (2010)
13. Qian, Y.H., Liang, J.Y., Dang, C.Y.: Incomplete multigranulation rough set. IEEE Transactions on Systems, Man and Cybernetics, Part A 40(2), 420–431 (2010)
14. Wang, L.J., Yang, X.B., Yang, J.Y., Wu, C.: Relationships among generalized rough sets in six coverings and pure reflexive neighborhood system. Information Science 207, 66–78 (2012)

Boolean Covering Approximation Space and Its Reduction

Tong-Jun Li and Wei-Zhi Wu

School of Mathematics, Physics and Information Science, Zhejiang Ocean University, Zhoushan, Zhejiang 316000, P.R. China
ltj722@163.com, wuwz@zjou.edu.cn

Abstract. In this paper, Boolean vector algebra theory is introduced into rough set theory. A theoretical framework of Boolean covering approximation space is proposed, and based on the principle of traditional covering rough set theory, a pair of lower and upper approximation operators on a Boolean covering approximation space are defined. Properties of the lower and upper approximation operators are investigated in detail. The duality of the lower and upper approximation operators, and lower and upper definable Boolean vectors are discussed. Finally, reductions of lower and upper approximation operators are explored.

Keywords: Boolean vector, Boolean covering approximation space, Rough sets, Reduction.

1 Introduction

Rough set theory proposed by Pawlak [12] is an important tool to deal with inexact, uncertain and insufficient information in information systems. Lower and upper approximation operators are two basic concepts in rough set models. By using them, knowledge hidden in an information system may be expressed in the form of decision rule. Rough set theory has been successfully applied to many areas, for example, feature selection [3], rule extraction [13], granular computing [9], and so on.

Traditional rough set model is based on equivalence relations on the universe of discourse. However, as pointed out by some scholars, equivalence relation or partition is still restrictive for many applications, i.e. many practical data sets can not be dealt with by traditional rough sets. Thus, some generalized rough set models have been proposed to meet a variety of needs in recent years, for example, binary relation based rough sets [10], covering rough sets [1,5,19], probabilistic rough sets [15,16], etc. As a meaningful extension of traditional rough set model, covering rough sets are taken to deal with more complex practical problems which the traditional one can not handle [11]. Various types of covering rough set models are proposed [8,17,20], and relationships between covering rough sets and other types of rough sets are also discussed [14,21]. Alternatively, on reduction of covering rough sets is paid much attention by some authors [2,11,18,19].

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 245–252, 2013.

Boolean matrix theory [4] is an important mathematical method to deal with many practical problems such as industrial control and electronic circuit design, etc. As for set theory, a subset and a binary relation on a universe can be formulated naturally as a Boolean vector and a Boolean matrix, respectively, and some computations of sets or relations can also be implemented by numerical methods of Boolean matrix theory. Therefore, it is meaningful to use Boolean vector algebra approach to investigate rough sets. However, little work has been done on the study of rough set theory by means of Boolean vector algebra theory. In [7], by means of Boolean matrix, invertible lower and upper rough approximation operators based on binary relation are investigated. In [6], the simultaneous Boolean equation solutions are studied using a rough set method, and the connection between rough set theory and Dempster-Shafer theory of evidence is also discussed. So far, it forcan not be found that Boolean vector algebra theory has been used for the study of covering rough sets in the literature. In this paper, Boolean vector algebra method is introduced to covering rough set theory for the first time. As a result, a theoretical framework of covering rough set theory is established in Boolean vector algebra. By some given Boolean vectors, lower and upper approximations of an arbitrary Boolean vector are defined, and properties of approximation operators are explored. Meanwhile, reduction of lower and upper approximation operators are also discussed.

2 Pawlak Rough Sets and Boolean Vector Space

In this section, we review some basic knowledge about Pawlak rough sets and Boolean vector space.

2.1 Pawlak Rough Sets

Let (U, R) be a Pawlak approximation space, where U is a non-empty and finite set called universe of discourse, and R is a binary equivalence relation on U. For any $X \subseteq U$, the lower and upper rough approximations of X in (U, R) can be defined respectively by:

$$\underline{R}(X) = \cup\{E \in U/R | E \subseteq X\}, \overline{R}(X) = \cup\{E \in U/R | E \cap X \neq \emptyset\},$$

where U/R denotes the set of all equivalence classes of R.

In general, for any $X \subseteq U$, $\underline{R}(X) \subseteq X \subseteq \overline{R}(X)$. And

- X is said to be lower rough definable if $\underline{R}(X) = X$;
- X is said to be upper rough definable if $X = \overline{R}(X)$;
- X is said to be rough definable if $\underline{R}(X) = X = \overline{R}(X)$.

It is evident that three classes of rough definable sets of a Pawlak approximation space are identical.

2.2 Boolean Vector Space

As for basic notions and kowledge of Boolean vector and Boolean matrix, please refer to [4]. Here, the join, meet and negative Boolean operations are expressed by $\vee$, $\wedge$ and $\neg$, respectively, and V_n denotes the sets of all n-dimensional Boolean row vectors. $\mathbf{0}$ and $\mathbf{1}$ denote the Boolean vectors with all 0 entries and 1 entries, respectively.

Definition 1. *A subset $W \subseteq V_n$ is called a Boolean vector subspace, or simply a vector subspace, if $\alpha \vee \beta \in W$ for all $\alpha, \beta \in W$.*

Let $W \subseteq V_n$, denote $[W] = \{\vee D | D \subseteq W\}$, where $\vee D$ represents the join of all Boolean vectors of D. Then it is clear that $[W]$ is a vector subspace of V_n, and $[W]$ is called the vector subspace spanned (or generated) by W.

Definition 2. *A Boolean vector $\alpha \in V_n$ is said to be dependent on $W \subseteq V_n$ if $\alpha \in [W]$. A subset $W \subseteq V_n$ is said to be dependent if at least one of the Boolean vectors is dependent on the rest of the Boolean vectors. If W is not dependent, then it is called independent.*

Therefore, $W \subseteq V_n$ is independent if and only if for any $\alpha \in W$, α is not dependent on $W - \{\alpha\}$.

Definition 3. *Let $W_1, W_2 \subseteq V_n$. W_1 is said to can be represented by W_2 if $W_1 \subseteq [W_2]$. If W_1 can be represented by W_2, and W_2 can be represented by W_1, then we say W_1 and W_2 are equivalent.*

Obviously, W_1 and W_2 are equivalent if and only if $[W_1] = [W_2]$.

Definition 4. *Let $W \subseteq V_n$ and $B \subseteq W$. B is called a basis of W if B is independent, and $[W] = [B]$.*

Proposition 1. *The unique basis exists for any set of Boolean vectors unequal to $\mathbf{0}$.*

If $W \subseteq V_n$ is independent, then the basis of W is itself.

Proposition 2. *Let $W_1, W_2 \subseteq V_n$. Then W_1 is equivalent to W_2 if and only if W_1 and W_2 have the same basis.*

Definition 5. *Let $W \subseteq V_n$. The number of vectors in the basis of W is called the rank of W denoted by $r(W)$.*

Proposition 3. *Let $W_1, W_2 \subseteq V_n$. If W_1 and W_2 are equivalent, then $r(W_1) = r(W_2)$.*

It should be pointed out that if $r(W_1) = r(W_2)$ then W_1 and W_2 may not be equivalent.

3 Boolean Covering Approximation Spaces

$W \subseteq V_n$ with nonzero vectors is called a *covering* of V_n if $\bigvee W = \mathbf{1}$. Further, a covering P of V_n is called a *partition* of V_n if for any $\alpha, \beta \in P$, $\alpha = \beta$ or $\alpha \wedge \beta = \mathbf{0}$.

Definition 6. *A pair (V_n, W) is called a Boolean covering approximation space if W is a covering of V_n. For any $\alpha \in V_n$, a pair of lower and upper approximations, $\underline{W}(\alpha)$ and $\overline{W}(\alpha)$, are defined by*

$$\underline{W}(\alpha) = \bigvee\{\beta \in W | \beta \leq \alpha\}, \overline{W}(\alpha) = \bigvee\{\beta \in W | \beta \wedge \alpha \neq \mathbf{0}\}.$$

The pair $\underline{W}$ and $\overline{W}$ are called lower and upper approximation operators on (V_n, W), respectively.

It is clear that for any $\alpha \in V_n$, $\underline{W}(\alpha)$ and $\overline{W}(\alpha)$ belong to $[W]$.

Example 1. Let $W = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$, where $\alpha_1 = (1,1,0,0,0)$, $\alpha_2 = (0,1,1,1,0)$, $\alpha_3 = (0,1,0,0,1)$, and $\alpha_4 = (0,0,1,1,0)$. Then (V_5, W) is a Boolean covering approximation space. Taking $\alpha = (1,0,1,1,0)$, we can find

$$\{\alpha_i \in W | \alpha_i \leq \alpha\} = \{\alpha_4\}, \{\alpha_i \in W | \alpha_i \wedge \alpha \neq \mathbf{0}\} = \{\alpha_1, \alpha_2, \alpha_4\}.$$

By Definition 6 we have

$$\underline{W}(\alpha) = \alpha_4 = (0,0,1,1,0), \overline{W}(\alpha) = \alpha_1 \vee \alpha_2 \vee \alpha_4 = (1,1,1,1,0).$$

From Definition 6, the following theorem can be easily derived.

Theorem 1. *Let (V_n, W) be a Boolean covering approximation space.*

$(L0)$ $\underline{W}(\mathbf{0}) = \mathbf{0}$,	$(U0)$ $\overline{W}(\mathbf{1}) = \mathbf{1}$;
$(L1')$ $\underline{W}(\mathbf{1}) = \mathbf{1}$,	$(U0')$ $\overline{W}(\mathbf{0}) = \mathbf{0}$;
$(L1)$ $\alpha \leq \beta \Rightarrow \underline{W}(\alpha) \leq \underline{W}(\beta)$,	$(U1)$ $\alpha \leq \beta \Rightarrow \overline{W}(\alpha) \leq \overline{W}(\beta)$;
$(L2)$ $\underline{W}(\alpha \wedge \beta) \leq \underline{W}(\alpha) \wedge \underline{W}(\beta)$,	$(U2)$ $\overline{W}(\alpha \vee \beta) = \overline{W}(\alpha) \vee \overline{W}(\beta)$;
$(L3)$ $\underline{W}(\alpha) \leq \alpha$,	$(U3)$ $\alpha \leq \overline{W}(\alpha)$;
$(L4)$ $\forall \alpha \in [W] \Leftrightarrow \underline{W}(\alpha) = \alpha$;	
$(L5)$ $\underline{W}(\alpha) = \underline{W}(\underline{W}(\alpha))$;	
$(L6)$ $\overline{W}(\alpha) = \underline{W}(\overline{W}(\alpha))$.	

Definition 7. *Let (V_n, W) be a Boolean covering approximation space and $\alpha \in V_n$. α is said to be lower or upper definable if $\underline{W}(\alpha) = \alpha$ or $\overline{W}(\alpha) = \alpha$, respectively.*

Example 2. Let $W = \{\alpha_1, \alpha_2, \alpha_3\}$, where $\alpha_1 = (1,1,0,0,0)$, $\alpha_2 = (0,1,1,0,0)$, and $\alpha_3 = (0,0,0,1,1)$. Then (V_5, W) is a Boolean covering approximation space. Taking $\alpha = (1,1,1,0,0)$ and $\beta = (0,0,0,1,1)$, by Definition 6 we have

$$\underline{W}(\alpha) = (1,1,1,0,0), \overline{W}(\beta) = (0,0,0,1,1).$$

So $\underline{W}(\alpha) = \alpha$, $\overline{W}(\beta) = \beta$. Therefore, α is lower definable, and β is upper definable.

Property (L4) of Theorem 1 shows that a Boolean vector of V_n is lower definable in (V_n, W) if and only if it belongs to $[W]$.

Theorem 2. *Let (V_n, W) be a Boolean covering approximation space and $\alpha \in V_n$. Then α is upper definable if and only if for any $\beta \in W$, $\beta \wedge \alpha \neq \mathbf{0}$ implies $\beta \leq \alpha$.*

Proof. Assume that $\overline{W}(\alpha) = \alpha$. By Definition 6, for any $\beta \in W$, if $\beta \wedge \alpha \neq \mathbf{0}$, then $\beta \leq \overline{W}(\alpha)$. By the assumption we have $\beta \leq \alpha$.

Conversely, if for any $\beta \in W$, $\beta \wedge \alpha \neq \mathbf{0}$ implies $\beta \leq \alpha$, then $\bigvee\{\beta \in W | \beta \wedge \alpha \neq \mathbf{0}\} \leq \alpha$, that is, $\overline{W}(\alpha) \leq \alpha$. Combining $\alpha \leq \overline{W}(\alpha)$ we get $\overline{W}(\alpha) = \alpha$.

Theorem 3. *Let (V_n, W) be a Boolean covering approximation space and $\alpha \in V_n$. Then α is upper definable if and only if $\neg\alpha$ is upper definable.*

Proof. Assume that α is upper definable, that is, $\overline{W}(\alpha) = \alpha$. Then for any $\beta \in W$, if $\beta \wedge \alpha \neq \mathbf{0}$, then $\beta \leq \alpha$. For any $\beta \in W$, if $\beta \wedge \neg\alpha \neq \mathbf{0}$, then $\beta \not\leq \alpha$, thus $\beta \wedge \alpha = \mathbf{0}$, equivalently $\beta \leq \neg\alpha$. According to Theorem 2 we conclude that $\neg\alpha$ is upper definable.

As $\alpha = \neg(\neg\alpha)$, by the above proof we know that if $\neg\alpha$ is upper definable, then α is upper definable.

Example 3. Let W be the covering of V_5 in Example 2 and let $\alpha = (1, 1, 0, 0, 0)$. From Definition 6 we can compute $\overline{W}(\alpha) = (1, 1, 0, 0, 0) = \alpha$. Thus, α is upper definable. Alternatively, $\neg\alpha = (0, 0, 1, 1, 1)$, by Example 2 we know that $\neg\alpha$ is upper definable.

It should be noted that in general, $\underline{W}$ and $\overline{W}$ may not be dual to each other, that is, they may not satisfy the following equations:

$$\overline{W}(\alpha) = \neg\underline{W}(\neg\alpha) \text{ or } \underline{W}(\alpha) = \neg\overline{W}(\neg\alpha), \forall\alpha \in V_n.$$

Theorem 4. *Let (V_n, W) be a Boolean covering approximation space. Then $\underline{W}$ and $\overline{W}$ are dual to each other if and only if W is a partition of V_n.*

Proof. ($\Rightarrow$) For any $\alpha \in W$, clearly $\underline{W}(\alpha) = \alpha$. By the duality of $\underline{W}$ and $\overline{W}$, we have $\neg\alpha = \overline{W}(\neg\alpha)$. So, for any $\beta \in W$, if $\beta \wedge \neg\alpha \neq \mathbf{0}$, then $\beta \leq \neg\alpha$, equivalently $\alpha \wedge \beta = \mathbf{0}$. Thus, for any $\alpha, \beta \in W$, if $\alpha \wedge \beta \neq \mathbf{0}$, then $\alpha \wedge \neg\beta = \mathbf{0}$ and $\beta \wedge \neg\alpha = \mathbf{0}$, that is, $\alpha \leq \beta$ and $\beta \leq \alpha$, thus $\alpha = \beta$. We conclude that W is a partition of V_n.

($\Leftarrow$) Let $\alpha \in V_n$. It is clear that $W = \{\beta \in W | \beta \wedge \neg\alpha = \mathbf{0}\} \cup \{\beta \in W | \beta \wedge \neg\alpha \neq \mathbf{0}\}$. Thus,

$$\mathbf{1} = \vee W = (\vee\{\beta \in W | \beta \wedge \neg\alpha = \mathbf{0}\}) \vee (\vee\{\beta \in W | \beta \wedge \neg\alpha \neq \mathbf{0}\}).$$

Since $\beta \wedge \neg\alpha = \mathbf{0}$ is equivalent to $\beta \leq \alpha$, by Definition 6 we have $\mathbf{1} = \underline{W}(\alpha) \vee \overline{W}(\neg\alpha)$. On the other hand, since $\{\beta \in W | \beta \wedge \neg\alpha = \mathbf{0}\} \cap \{\beta \in W | \beta \wedge \neg\alpha \neq \mathbf{0}\} = \emptyset$, and W is a partition of V_n, it is easy to verify that $\underline{W}(\alpha) \wedge \overline{W}(\neg\alpha) = \mathbf{0}$. Therefore, $\underline{W}(\alpha) = \neg\overline{W}(\neg\alpha)$, or $\overline{W}(\neg\alpha) = \neg\underline{W}(\alpha)$.

The equation in (L2) of Theorem 1 holds only under some conditions.

Theorem 5. *Let (V_n, W) be a Boolean covering approximation space. Then for any $\alpha, \beta \in V_n$, $\underline{W}(\alpha \wedge \beta) = \underline{W}(\alpha) \wedge \underline{W}(\beta)$ if and only if $[W]$ is a σ-algebra.*

Proof. ($\Rightarrow$) To prove that $[W]$ is a σ-algebra, it only need proving that for any $\alpha, \beta \in W$, $\alpha \wedge \beta \in [W]$. For any $\alpha, \beta \in W$, since $\underline{W}(\alpha) = \alpha$ and $\underline{W}(\beta) = \beta$, from $\underline{W}(\alpha \wedge \beta) = \underline{W}(\alpha) \wedge \underline{W}(\beta)$, it follows that $\alpha \wedge \beta = \underline{W}(\alpha \wedge \beta)$. By (L4) of Theorem 1 we have $\alpha \wedge \beta \in [W]$.

($\Leftarrow$) It is clear that $\underline{W}(\alpha \wedge \beta) \leq \underline{W}(\alpha) \wedge \underline{W}(\beta)$. Conversely, for any $\gamma \in W$, if $\gamma \leq \underline{W}(\alpha) \wedge \underline{W}(\beta)$, then there are $\theta, \delta \in W$ such that $\theta \leq \alpha, \delta \leq \beta$, and $\gamma \leq \theta \wedge \delta$. Since $\theta \wedge \delta \in [W]$, there are $\gamma_1, \cdots, \gamma_k \in W$ such that $\theta \wedge \delta = \gamma_1 \vee \cdots \vee \gamma_k$. As $\theta \wedge \delta \leq \alpha \wedge \beta$, we have $\gamma_1 \vee \cdots \vee \gamma_k \leq \alpha \wedge \beta$. Thus, $\gamma_i \leq \alpha \wedge \beta$, $i = 1, \cdots, k$, from $\gamma \leq \gamma_i$, $i = 1, \cdots, k$, it follows that $\gamma \leq \underline{W}(\alpha \wedge \beta)$. Therefore, $\underline{W}(\alpha) \wedge \underline{W}(\beta) \leq \underline{W}(\alpha \wedge \beta)$. We conclude that $\underline{W}(\alpha \wedge \beta) = \underline{W}(\alpha) \wedge \underline{W}(\beta)$ for any $\alpha, \beta \in V_n$.

4 Reduction of Boolean Covering Approximation Space

Theorem 6. *Let (V_n, W_1) and (V_n, W_2) be two Boolean covering approximation spaces. Then $(V_n, W_1 \cup W_2)$ is also a Boolean covering approximation space, and for any $\alpha \in V_n$,*

$$\underline{W_1 \cup W_2}(\alpha) = \underline{W_1}(\alpha) \vee \underline{W_2}(\alpha),\ \overline{W_1 \cup W_2}(\alpha) = \overline{W_1}(\alpha) \vee \overline{W_2}(\alpha).$$

Proof. It directly proved by Definiton 6.

Theorem 7. *Let (V_n, W_1) and (V_n, W_2) be two Boolean covering approximation spaces. Then $\underline{W_1 \cup W_2} = \underline{W_1}$ if and only if W_2 can be represented by W_1.*

Proof. ($\Rightarrow$) From $\underline{W_1 \cup W_2}(\alpha) = \underline{W_1}(\alpha)$ for all $\alpha \in V_n$, we can see that $[W_1 \cup W_2] = [W_1]$. It is clear that $W_2 \subseteq [W_2] \subseteq [W_1 \cup W_2]$. So, $W_2 \subseteq [W_1]$, that is, W_2 can be represented by W_1.

($\Leftarrow$) Assume that $W_2 \subseteq [W_1]$. For any $\beta \in W_2$ and $\alpha \in V_n$, if $\beta \leq \alpha$, then $\beta \leq \underline{W_2}(\alpha)$. From $W_2 \subseteq [W_1]$, we have $\beta \in [W_1]$. There are $\beta_1, \cdots, \beta_k \in W_1$ such that $\beta = \beta_1 \vee \cdots \vee \beta_k$. Clearly, $\beta_i \leq \alpha$, $i = 1, \cdots, k$, from which it follows that $\beta_i \leq \underline{W_1}(\alpha)$, $i = 1, \cdots, k$, which implies that $\beta \leq \underline{W_1}(\alpha)$. Therefore, $\underline{W_2}(\alpha) \leq \underline{W_1}(\alpha)$ for all $\alpha \in V_n$. Accoring to Theorem 6 it is got that $\underline{W_1 \cup W_2}(\alpha) = \underline{W_1}(\alpha)$ for all $\alpha \in V_n$.

Corollary 1. *Let (V_n, W_1) and (V_n, W_2) be two Boolean covering approximation spaces. Then $\underline{W_1} \leq \underline{W_2}$ if and only if W_1 can be represented by W_2.*

Theorem 8. *Let (V_n, W_1) and (V_n, W_2) be two Boolean covering approximation spaces. Then $\underline{W_1} = \underline{W_2}$ if and only if W_1 and W_2 are equivalent.*

Proof. It immediately follows from Corollary 1.

Theorem 9. *Let (V_n, W_1) and (V_n, W_2) be two Boolean covering approximation spaces. Then $\underline{W_1} = \underline{W_2}$ if and only if W_1 and W_2 have the same basis.*

Proof. It follows from Theorem 8 and Proposition 2.

From Theorem 8 and Proposition 3, the below conclusion follows.

Corollary 2. *Let* (V_n, W_1) *and* (V_n, W_2) *be two Boolean covering approximation spaces. If* $\underline{W_1} = \underline{W_2}$*, then* $r(W_1) = r(W_2)$*.*

Theorem 10. *Let* (V_n, W_1) *and* (V_n, W_2) *be two Boolean covering approximation spaces. If for any* $\alpha \in W_1$*, there is* $\beta \in W_2$ *such that* $\alpha \leq \beta$*, then* $\overline{W_1} \leq \overline{W_2}$*.*

Proof. For any $\beta_1 \in W_1$, if $\beta_1 \leq \overline{W_1}(\alpha)$, that is, $\beta_1 \wedge \alpha \neq \mathbf{0}$, as there is $\beta_2 \in W_2$ such that $\beta_1 \leq \beta_2$, we have $\beta_2 \wedge \alpha \geq \beta_1 \wedge \alpha$, then it follows that $\beta_2 \wedge \alpha \neq \mathbf{0}$. Thus, $\beta_1 \leq \beta_2 \leq \overline{W_2}(\alpha)$. We conclude that $\overline{W_1}(\alpha) \leq \overline{W_2}(\alpha)$, $\forall \alpha \in V_n$, that is, $\overline{W_1} \leq \overline{W_2}$.

Theorem 11. *Let* (V_n, W_1) *and* (V_n, W_2) *be two Boolean covering approximation spaces. If for any* $\alpha_1 \in W_1, \alpha_2 \in W_2$*, there are* $\beta_2 \in W_1, \beta_1 \in W_2$ *such that* $\alpha_1 \leq \beta_1$ *and* $\alpha_2 \leq \beta_2$*, then* $\overline{W_1} = \overline{W_2}$*.*

From Theorem 11 we have the following conclusion.

Theorem 12. *Let* (V_n, W) *be a Boolean covering approximation space and* $D \subseteq W$*. If for any* $\alpha \in W - D$*, there is* $\beta \in D$ *such that* $\alpha \leq \beta$*, and for any* $\alpha \in D$*, there is no* $\beta \in D - \{\alpha\}$ *such that* $\alpha \leq \beta$*, then* $\overline{D} = \overline{W}$*, and* $\overline{C} \neq \overline{W}$ *for all* $C \subset D$*.*

5 Conclusions

In this paper, according to covering rough set theory, we establish a theoretical framework of rough set theory in Boolean vector algebra. The notions of Boolean covering approximation space, and lower and upper approximation operators are proposed, some properties of lower and upper approximation operators are investigated. Some sufficient and necessary conditions under which some properties of approximation operators hold are obtained. Lower and upper definable Boolean vectors are discussed, and approaches for judging them are given. Finally, reductions of lower and upper approximation operators are studied. Some sufficient and necessary conditions of lower approximation operator reduction are gained. Especially, a sufficient condition of upper approximation operator reduction is gotten. Introducing Boolean vector algebra to covering rough set theory may help us not only to uncover the mathematical essence of covering rough set theory, but also to develop simple computation methods of covering rough sets.

Acknowledgements. This work was supported by grants from the National Natural Science Foundation of China (Nos. 11071284, 61075120, 61272021, 61202206) and the Zhejiang Provincial Natural Science Foundation of China (Nos. LY12F02021, LZ12F03002).

References

1. Bonikowski, Z., Brynirski, E., Wybraniec, U.: Extensions and intentions in the rough set theory. Information Sciences 107, 149–167 (1998)
2. Chen, D., Wang, C., Hu, Q.: A new approach to attributes reduction of consistent and inconsistent covering decision systems with covering rough sets. Information Sciences 177, 3500–3518 (2007)
3. Hu, Q.H., Pedrgcz, W., Yu, D.R., Lang, J.: Selecting discrete and continuous feasures based on neighborhood decision error minimization. IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics 40, 137–150 (2010)
4. Kim, K.H.: Boolean Matrix Theory and Applications. Marcel Dekber Inc. (1982)
5. Li, T.J., Leung, Y., Zhang, W.X.: Generalized fuzzy rough approximation operators based on fuzzy coverings. International Journal of Approximation Reasoning 48, 836–856 (2008)
6. Liu, G.: Rough set theory based on two universal sets and its applications. Knowledge-Based Systems 23, 110–115 (2010)
7. Liu, G., Sai, Y.: Invertible approximation operators of generalized rough sets and fuzzy rough sets. Information Sciences 180, 2221–2229 (2010)
8. Lin, G., Liang, J., Qian, Y.: Multigranulation rough sets: From partition to covering. Information Sciences 241, 101–118 (2013)
9. Skowron, A., Stepaniuk, J., Swiniarski, R.: Modeling rough granular computing based on approximation spaces. Information Sciences 184(1), 20–43 (2012)
10. Slowinski, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Transactions on Knowledge and Data Engineering 12, 331–336 (2000)
11. Tsang, E.C.C., Chen, D., Yeung, D.S.: Approximations and reducts with covering generalized rough sets. Computers and Mathematics with Applications 56, 279–289 (2008)
12. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
13. Qian, Y.H., Liang, J.Y., Dang, C.Y.: Converse approximation and rule extraction from decision tables in rough set theory. Comput. Math. Appl. 55(8), 1754–1765 (2008)
14. Wang, L., Yang, X., Yang, J., Wu, C.: Relationships among generalized rough sets in six coverings and pure reflexive neighborhood system. Information Sciences 207, 66–78 (2012)
15. Yao, Y.Y.: Probabilistic rough set approximations. International Journal of Approximate Reasoning 49, 255–271 (2008)
16. Yao, Y.Y.: The superiority of three-way decisions in probabilistic rough set models. Information Sciences 181, 1080–1096 (2011)
17. Yao, Y., Yao, B.: Covering based rough set approximations. Information Sciences 200, 91–107 (2012)
18. Yang, T., Li, Q.: Reduction about approximation spaces of covering generalized rough sets. International Journal of Approximate Reasoning 51, 335–345 (2010)
19. Zhu, W., Wang, F.Y.: Reduction and axiomization of covering generalized rough sets. Information Sciences 152, 217–230 (2003)
20. Zhu, W., Wang, F.Y.: The fourth type of covering-based rough sets. Information Sciences 201, 80–92 (2012)
21. Zhu, W.: Relationship between generalized rough sets based on binary relation and covering. Information Sciences 179, 210–225 (2009)

Dynamic Analysis of IVFSs Based on Granularity Computing

Danqing Xu[1], Yanan Fu[1], and Junjun Mao[1,2,3]

[1] School of Mathematical Sciences, Anhui University, Hefei, Anhui 230039, PR China
[2] Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Anhui University, 230039 Hefei, Anhui, PR China
[3] Department of Computer Science, University of Houston, Houston, TX 77204−3010 USA
maojunjunuh@gmail.com

Abstract. Interval-valued fuzzy soft set (IVFSs) is a new and effective mathematical tool used for processing incomplete and uncertain data. In order to describe and measure uncertain information of IVFSs perfectly, dynamic analysis of granular computing based on covering about IVFSs is originally discussed in this paper. Firstly, the α-dominance relation between any two objects in IVFSs is built by constructing the possibility degree or the weighted possibility degree after standardization, then α-dominance class and α-covering approximation space of IVFSs could be generated on this relation. Secondly, knowledge capacity is proposed to measure the granular information through introducing concepts of the description set and the indistinguishability set. Finally, an illustrative example shows dynamic changes of uncertain information under different granular structure.

Keywords: IVFSs, possibility degree, the weighted possibility degree, α-dominance relation, α-covering approximation space, dynamic, knowledge capacity.

1 Introduction

The real world is full of uncertainty, imprecision and vagueness. In fact, most of the concepts we are meeting in everyday life are vague rather than precise. However, most of traditional mathematical tool for formal modeling, reasoning and computing are crisp, deterministic and precise in character, so the classical methods are not always suitable. To solve these problems, some theories such as probability theory, fuzzy set theory[1], intuitionistic fuzzy set theory[2], rough set theory[3], vague set theory[4], and interval mathematic[5] are proposed as efficient tools to deal with different types of uncertainties. However, all these theories have their inherent limitation which is pointed out by Molodtsov[6] because of their inadequacy of the parameterization tools of the theory. In 1999, Molodtsov[6] initiated soft set theory as a new mathematical tool for dealing with uncertainties which is free from the difficulties affecting existing methods. This theory has proven useful in many different fields such as the smoothness of

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 253–260, 2013.

functions, game theory, operations research, Riemann integration, Perron integration, probability theory, and measurement theory[6,7].

Up to the present, research on soft sets has been very active, and there have been many progresses concerning practical applications of soft theory. Maji et al.[8,9] discussed many operations of soft set and firstly applied soft set theory in decision making. Actually, soft set can be combined with other uncertain theories, such as Roy and Maji[10] originally extended the soft set to fuzzy soft set(FSs) and presented a theoretic approach to cope with decision making problems. Then Kong et al.[11] revised the Roy-Maji method by considering "fuzzy choice values". Feng et al.[12] further discussed the application of fuzzy soft sets to decision making in an imprecise environment and firstly proposed an adjustable approach by using level soft set. Similar to[12], Jiang et al.[13] extended this adjustable approach to intuitionistic fuzzy soft set. Majumda and Samanta constructed similarity measures in fuzzy soft set in[14].

In many fuzzy decision making applications the related membership functions are extremely individual (dependent on experts' evaluation of alternatives) and thus cannot be lightly confirmed. It is more reasonable to give an interval-valued data to describe degree of membership; in other words, we can make use of interval-valued fuzzy sets which assign to each element an interval that approximates the "real" (but unknown) membership degree. In respond to this, Yang et al.[15] defined a hybrid model called interval-valued fuzzy soft sets and investigated some of their basic properties. They also presented an algorithm to solve decision making problems based on interval-valued fuzzy soft sets. Feng et al.[16] gave deeper insights into interval-valued fuzzy soft set based decision making discussed in[15].

Nowadays, few people have made granularity analysis for IVFSs, the main reason may be that it's hard to find an equivalence relation or preference relation to generate granular structure. Inspired by[17], we try to investigate the granularity analysis of IVFSs. The remainder of this paper is organized as follows: To facilitate our discussion, we first recall the interval-valued fuzzy soft sets in Section 2. In Section 3, we briefly introduce the possibility degree between any two objects in universe of IVFSs, then construct the α-dominance class of IVFSs, and these classes institute a covering of universe. Section 4 is discussed the dynamic analysis of granular computing based on α-covering approximation space. In Section 5, an illustrative example is given to reflect dynamic changes of uncertain information under different granular structure. Finally, we conclude the paper with a summary in Section 6.

2 Preliminaries

In this section, some basic concepts and notions will be introduced. let U be an non-empty set, called initial universal set of objects and E is a set of parameters in relation to U which is often the set of attributes, characteristics or properties of objects. Let $P(U)$ denote the power set of U. According to[6], the concept of soft sets is defined as follows.

Definition 1.([6]) (Soft set) A pair (F, E) is called a soft set over U, where F is a mapping given by $F : E \to P(U)$.
By definition, a soft set (F, E) over the universe U can be regarded as a parameterized family of subsets of the universe U, which gives an approximation (soft) description of the objects in U. As pointed out by Molodtsov[6], for any parameter $e \in E$, the subset $F(e) \subseteq U$ may be considered as the set of e-approximation elements in the soft set (F, E). It is worth noting that $F(e)$ may be arbitrary: some of them may be empty, and some may have nonempty intersection.

Definition 2.([10]) (FSs) Let $F(U)$ denotes the set of all fuzzy subset of U , a pair (F, E) is called a fuzzy soft set over U, where F is a mapping given by $F : E \to F(U)$.
In this definition, fuzzy subsets are used as substitutes for the crisp subsets. Hence every soft set may be considered as a fuzzy soft set. Also it is obvious that a fuzzy set could be naturally viewed as a fuzzy soft set whose parameter set is a singleton. Generally speaking, $F(e)$ is a subset in U, $\forall e \in E$. Following the standard notations, $F(e)$ can be typically be written as $F(e) = \{(x, F(e)(x)) : x \in U\}$.

Definition 3.([5]) (IVFs) An interval-valued fuzzy set $\hat{X}$ on an universe U is a mapping such that

$$\hat{X} : U \to \mathrm{Int}([0, 1]),$$

where $\mathrm{Int}([0, 1])$ stands for the set of all closed subintervals of $[0, 1]$, the set of all interval-valued fuzzy sets on U is denoted by $\wp(U)$.
Suppose that $\hat{X} \in \wp(U), \forall x \in U, \mu_{\hat{X}} = [\mu^-_{\hat{X}}(x), \mu^+_{\hat{X}}(x)]$ is called the degree of membership an element x to $\hat{X}$. $\mu^-_{\hat{X}}(x)$ and $\mu^+_{\hat{X}}(x)$ are referred to as the lower and upper degrees of membership x to $\hat{X}$ where $0 \le \mu^-_{\hat{X}}(x) \le \mu^+_{\hat{X}}(x) \le 1$.

Definition 4.([15]) (IVFSs) A pair $(\tilde{F}, E)$ is called an interval-valued fuzzy soft set (IVFS) over U, where $\tilde{F}$ is a mapping given by $\tilde{F} : E \to \wp(U)$.
An interval-valued fuzzy soft set is a parameterized family of interval-valued fuzzy subsets of U. $\forall e \in E, \tilde{F}(e)$ is called the interval fuzzy value set of the parameter e. It is easy to see that fuzzy soft sets are special case of IVFSs since interval-valued fuzzy sets are extensions of classical fuzzy sets. According to characters of the IVFSs, every IVFSs could be represented in form of a matrix, so an IVFSs $(\tilde{F}, E)$ can be described as $(\tilde{F}, E) = ([\mu^-_{ij}, \mu^+_{ij}])_{m\times n} = [\tilde{F}(e_j)(x_i)]_{m\times n}$.

3 α-Dominance Class on the Possibility Degree of IVFSs

3.1 Standardized Methods of the IVFSs

In real-life, the attribute always has two types, namely benefit-type and cost-type. In order to eliminate the effect of different physical dimensions to decision making, we need conduct a pretreatment for every IVFSs. Suppose $(\tilde{F}, E) \triangleq ([\mu^-_{ij}, \mu^+_{ij}])_{m\times n}$ is an IVFSs, and $E = \{e_1, e_2, \ldots, e_n\}$ is the set of attributes, $U = \{x_1, x_2, \ldots, x_m\}$ is the object set. Let matrix $R = (\tilde{r}_{ij})_{m\times n} = ([r^-_{ij}, r^+_{ij}])_{m\times n}$

be the standardization of $(\tilde{F}, E)$, where $[r_{ij}^-, r_{ij}^+]$ satisfies $r_{ij}^- = \frac{\mu_{ij}^-}{\sum_{i=1}^m \mu_{ij}^+}, r_{ij}^+ = \frac{\mu_{ij}^+}{\sum_{i=1}^m \mu_{ij}^-}$($e_j$ is benefit-type) or $r_{ij}^- = \frac{(\mu_{ij}^+)^{-1}}{\sum_{i=1}^m (\mu_{ij}^-)^{-1}}, r_{ij}^+ = \frac{(\mu_{ij}^-)^{-1}}{\sum_{i=1}^m (\mu_{ij}^+)^{-1}}$($e_j$ is cost-type).

3.2 The Possibility Degree of IVFSs

Definition 5. (Possibility degree) Suppose $R = (\tilde{r}_{ij})_{m\times n} = ([r_{ij}^-, r_{ij}^+])_{m\times n}$ is the standardization of $(\tilde{F}, E)$, the possibility degree of the object x_i is superior to x_k is defined by

$$P(x_i \geq x_k) = \frac{1}{n}\sum_{j=1}^{n}\{\frac{r_{ij}^+ - r_{kj}^-}{l_{ij}+l_{kj}}[1 - I(r_{ij}^- \geq r_{kj}^+) - I(r_{ij}^+ \leq r_{kj}^-)] + I(r_{ij}^- \geq r_{kj}^+)\}.$$

Where $l_{ij} = r_{ij}^+ - r_{ij}^-, l_{kj} = r_{kj}^+ - r_{kj}^-. i, k = 1, 2, \ldots, m. I(\cdot)$ is an indicator function.

In fact, the coefficient $1/n$ may be viewed as attribute weights of E. Inspired by this notion, the weighted possibility degree can be defined as follow.

Definition 6. (Weighted possibility degree) Suppose $R = (\tilde{r}_{ij})_{m\times n} = ([r_{ij}^-, r_{ij}^+])_{m\times n}$ is the standardization of $(\tilde{F}, E)$, $E = \{e_1, e_2, \ldots, e_n\}$ is attributes set, the attribute weights is $\omega = \{\omega_1, \omega_2, \ldots, \omega_n\}, \sum_{j=1}^n \omega_j = 1, 0 \leq \omega_j \leq 1, j = 1, 2, \ldots, n$. The weighted possibility degree of the object x_i is superior to x_k is defined by

$$P_\omega(x_i \geq x_k) = \sum_{j=1}^{n}\omega_j\{\frac{r_{ij}^+ - r_{kj}^-}{l_{ij}+l_{kj}}[1 - I(r_{ij}^- \geq r_{kj}^+) - I(r_{ij}^+ \leq r_{kj}^-)] + I(r_{ij}^- \geq r_{kj}^+)\}.$$

Where $l_{ij} = r_{ij}^+ - r_{ij}^-, l_{kj} = r_{kj}^+ - r_{kj}^-. i, k = 1, 2, \ldots, m. I(\cdot)$ is an indicator function.

Proposition 1. $P(x_i \geq x_k)$ and $P_\omega(x_i \geq x_k)$ satisfy following properties.
(1)$0 \leq P(x_i \geq x_k) \leq 1, 0 \leq P_\omega(x_i \geq x_k) \leq 1$;
(2)$P(x_i \geq x_k) = \frac{1}{2}, P_\omega(x_i \geq x_k) = \frac{1}{2}$;
(3)$P(x_i \geq x_k) + P(x_i \leq x_k) = 1, P_\omega(x_i \geq x_k) + P_\omega(x_i \leq x_k) = 1$.
Proof. The proof could be obtained easily according to definition 5 and 6.

3.3 α-Dominance Class and α-Covering Approximation Space of IVFSs

According to section 3.2, the (or weighted) possibility degree between any two objects of U could be constructed, then we can obtain a (or weighted)possibility degree matrix $P = [P_{ki}]_{m\times m} \triangleq [P(x_i \geq x_k)]_{m\times m}$ (or $P = [P_{ki}^\omega]_{m\times m} \triangleq [P_\omega(x_i \geq x_k)]_{m\times m}$). For convenience, we take the possibility degree for example in following discussion. In fact, the possibility degree measures the dominance relation between two objects, and every dominance relation could generate dominance class for a given universe.

Definition 7. Suppose P is the possibility degree matrix of $(\tilde{F}, E)$, $\alpha(\in [0,1])$ is a constant, for every $x_k \in U$, the α-dominance class $[x_k]^{\geq\alpha}$ of the object x_k can be defined by $[x_k]^{\geq\alpha} = \{x_i \in U : P(x_i \geq x_k) \geq \alpha\}$.
Obviously, every α-dominance class $[x_k]^{\geq\alpha}(k = 1, 2, \ldots, m)$ is a subset of U, and all α-dominance classes of objects constitute a covering of U, namely $U = \cup_{k=1}^{m}[x_k]^{\geq\alpha}$, we call $C_\alpha = \{[x_k]^{\geq\alpha} : x_k \in U\}_{k=1}^{m}$ is α-covering of U, and (U, C_α) constitute a α-covering approximation space.

4 Dynamic Granularity Analysis Based on α-Covering Approximation Space

In fact, there usually exist some overlaps in the granulation of practical problems, traditional partition model cannot deal with them, while the covering model may play an essential role in some respects. In α-covering approximation space, objects of U construct particles by α-dominance relation. For an element in U, granularity level may change with adjusting α, this suggests uncertainty is decided by the α-dominance relation in covering granular space, and granular structure changes along with α. So we construct some measure methods to describe these dynamic granular structure.

Definition 8. Let (U, C_α) be a α-covering approximation space, $\alpha \in [0,1]$, for $\forall x \in U$, the description set of x in(U, C_α) is defined by $Des_{C_\alpha}(x) = \{K : K \in C_\alpha \wedge x \in K\}$. The indistinguishability set of x in $(U, C_\alpha)$is defined by $Ind_{C_\alpha}(x) = \cap\{K : K \in Des_{C_\alpha}(x)\}$.

Definition 9. Let (U, C_α) be a α-covering approximation space, $\alpha \in [0,1]$,where U is non-empty finite objects set called universe, $C_\alpha = \{[x_k]^{\geq\alpha} : x_k \in U\}_{k=1}^{m}$ is a α-covering of U,the knowledge capacity measurement of C_α is defined as follows:

$$M(C_\alpha) = \begin{cases} 1 & |[x_k]^{\geq\alpha}| = 1, k = 1, 2, \ldots, m \\ 1 - \sum\limits_{x_k \in U} |Ind_{C_\alpha}(x_k)|/|U|^2 & otherwise \end{cases}.$$

where $|\cdot|$ denotes the cardinal number of the set.
In which, when the granular of α-covering C_α is the thickest universe relation, namely $C_\alpha = \{U\}$, it is minimum value 0 of the α-covering's knowledge capacity, this shows that it contains the most uncertain information. When the granular of C_α is the thinnest, namely $|[x_k]^{\geq\alpha}| = 1, k = 1, 2, \ldots, m$, it is maximum value 1 of α-covering's knowledge capacity, and suggests the uncertainty of universe U is the weakest.

Proposition 2. Let (U, C_α) and (U, C_β) be two covering approximation spaces, α, $\beta \in$ [0,1], if for $\forall x \in U, Ind_{C_\alpha}(x) = Ind_{C_\beta}(x)$,then $M(C_\alpha) = M(C_\beta)$.

Definition 10. Let (U, C_α) and (U, C_β)be two covering approximation spaces, α, $\beta \in$ [0,1], if for $\forall K \in C_\alpha$,there exist $S \in C_\beta$ make $K \subset S$ be true, and for $\forall S \in C_\beta$, there exist $K \in C_\alpha$ make $K \subset S$ be true, then we call covering C_α is thinner than C_β, denoted by $C_\alpha \leq C_\beta$.

Proposition 3. Let C_α and C_β be two coverings, if $\alpha > \beta$, then $C_\alpha \leq C_\beta$.

Proposition 4. Let (U, C_α) and (U, C_β) be two covering approximation spaces, $\alpha, \beta \in [0, 1]$, if $\alpha \geq \beta$, then $M(C_\alpha) \geq M(C_\beta)$.

Proof. By $\alpha \geq \beta$, we have $C_\alpha \leq C_\beta$. For $\forall K \in C_\alpha, \exists S \in C_\beta, s.t. K \subset S$, and $\forall S \in C_\beta, \exists K \in C_\alpha, s.t. K \subset S$. Hence $\forall x_k \in U$, we have $Des_{C_\alpha}(x_k) \subset Des_{C_\beta}(x_k)$. So $Ind_{C_\alpha}(x_k) \subset Ind_{C_\beta}(x_k)$, then $|Ind_{C_\alpha}(x_k)| \leq |Ind_{C_\beta}(x_k)|$, namely $1 - \sum_{x_k \in U} |Ind_{C_\alpha}(x_k)|/|U|^2 \geq 1 - \sum_{x_k \in U} |Ind_{C_\beta}(x_k)|/|U|^2$, therefore from definition 9, we have $M(C_\alpha) \geq M(C_\beta)$.

Proposition 4 indicates that by adjusting values of α, we could obtain different granular structures, and make dynamic analysis for uncertain information.

5 Illustrative Examples

A company wants to select a manager, let $U = \{x_1, x_2, \ldots, x_6\}$, be the set of candidates, the attribute set $E = \{e_1, e_2, \ldots, e_6\}$, where e_1=ideological morality, e_2=work attitude, e_3=work style, e_4=knowledge structure, e_5=leadership, e_6=market developing ability. Suppose after statistical process, every candidate's assess information under the various attributes can be expressed as IVFSs over U, denoted by $(\tilde{F}, E)$ (Table 1). Since all attributes in are benefit-type, after standardization, we obtain the matrix $R = (\tilde{r}_{ij})_{m\times n} = ([r_{ij}^-, r_{ij}^+])_{m\times n}$(Table 2). According to definition 5, by computing the possibility degree between any two objects in U, we obtain the possibility degree matrix P.

$$P = \begin{pmatrix} 0.5000 & 0.4288 & 0.4117 & 0.4973 & 0.4166 & 0.4982 \\ 0.5712 & 0.5000 & 0.4908 & 0.5946 & 0.5124 & 0.5934 \\ 0.5883 & 0.5092 & 0.5000 & 0.6228 & 0.5109 & 0.6016 \\ 0.5072 & 0.4054 & 0.3772 & 0.5000 & 0.4080 & 0.4840 \\ 0.5824 & 0.4876 & 0.4891 & 0.5920 & 0.5000 & 0.6057 \\ 0.5018 & 0.4066 & 0.3984 & 0.5160 & 0.3943 & 0.5000 \end{pmatrix}$$

Table 1. The value of $(\tilde{F}, E)$

	e_1	e_2	e_3	e_4	e_5	e_6
x_1	[0.4000,0.7000]	[0.5000,0.6000]	[0.2000,0.5000]	[0.2000,0.8000]	[0.1000,0.7000]	[0.6000,0.9000]
x_2	[0.3000,0.5000]	[0.1000,0.9000]	[0.1000,0.7000]	[0.1000,0.5000]	[0.2000,0.4000]	[0.3000,0.8000]
x_3	[0.4000,0.5000]	[0.2000,0.6000]	[0.3000,0.6000]	[0.4000,0.6000]	[0.0000,0.3000]	[0.4000,0.7000]
x_4	[0.5000,0.8000]	[0.4000,0.9000]	[0.5000,0.7000]	[0.7000,0.9000]	[0.1000,0.3000]	[0.4000,0.6000]
x_5	[0.2000,0.4000]	[0.1000,0.4000]	[0.3000,0.9000]	[0.3000,0.5000]	[0.5000,0.6000]	[0.2000,0.9000]
x_6	[0.4000,0.7000]	[0.3000,0.5000]	[0.4000,0.8000]	[0.6000,0.8000]	[0.4000,0.9000]	[0.2000,0.5000]

When $\alpha = 0.45$, we could calculate 0.45-dominance class and the description set of every object(Table 3).

By definition 8, calculate the indistinguishability set of every object as follows: $Ind_{C_{0.45}}(x_1) = \{x_1, x_4, x_6\}, Ind_{C_{0.45}}(x_2) = \{x_1, x_2, \cdots, x_6\}, Ind_{C_{0.45}}(x_3) = \{x_1, x_2, \cdots, x_6\}$,

Table 2. The value of R

	e_1	e_2	e_3	e_4	e_5	e_6
x_1	[0.1111,0.3182]	[0.1282,0.3750]	[0.0476,0.2778]	[0.0488,0.3478]	[0.0313,0.5385]	[0.1364,0.4286]
x_2	[0.0833,0.2273]	[0.0256,0.5625]	[0.0238,0.3889]	[0.0244,0.2174]	[0.0625,0.3077]	[0.0682,0.3810]
x_3	[0.1111,0.2273]	[0.0513,0.3750]	[0.0714,0.3333]	[0.0976,0.2609]	[0.0000,0.2308]	[0.0909,0.3333]
x_4	[0.1389,0.3636]	[0.1026,0.5625]	[0.1190,0.3889]	[0.1707,0.3913]	[0.0313,0.2308]	[0.0909,0.2857]
x_5	[0.0556,0.1818]	[0.0256,0.2500]	[0.0714,0.5000]	[0.0732,0.2174]	[0.1563,0.4615]	[0.0455,0.4286]
x_6	[0.1111,0.3182]	[0.0769,0.3125]	[0.0952,0.4444]	[0.1463,0.3478]	[0.1250,0.6923]	[0.0455,0.2381]

Table 3. 0.45-dominance class and the description set

0.45-dominance class	the description set
$[x_1]^{\geq 0.45} : \{x_1, x_4, x_6\}$	$Des_{C_{0.45}}(x_1) : [x_k]^{\geq 0.45}, k = 1, 2, \cdots, 6$
$[x_2]^{\geq 0.45} : \{x_1, x_2, \cdots, x_6\}$	$Des_{C_{0.45}}(x_2) : [x_2]^{\geq 0.45}, [x_3]^{\geq 0.45}, [x_5]^{\geq 0.45}$
$[x_3]^{\geq 0.45} : \{x_1, x_2, \cdots, x_6\}$	$Des_{C_{0.45}}(x_3) : [x_2]^{\geq 0.45}, [x_3]^{\geq 0.45}, [x_5]^{\geq 0.45}$
$[x_4]^{\geq 0.45} : \{x_1, x_4, x_6\}$	$Des_{C_{0.45}}(x_4) : [x_k]^{\geq 0.45}, k = 1, 2, \cdots, 6$
$[x_5]^{\geq 0.45} : \{x_1, x_2, \cdots, x_6\}$	$Des_{C_{0.45}}(x_5) : [x_2]^{\geq 0.45}, [x_3]^{\geq 0.45}, [x_5]^{\geq 0.45}$
$[x_6]^{\geq 0.45} : \{x_1, x_4, x_6\}$	$Des_{C_{0.45}}(x_6) : [x_k]^{\geq 0.45}, k = 1, 2, \cdots, 6$

$Ind_{C_{0.45}}(x_4) = \{x_1, x_4, x_6\}, Ind_{C_{0.45}}(x_5) = \{x_1, x_2, \cdots, x_6\}, Ind_{C_{0.45}}(x_6) = \{x_1, x_4, x_6\}$.

According to definition 9, $M(C_{0.45}) = 1 - (3 + 6 + 6 + 3 + 6 + 3)/36 = 9/36$.

Similar to above discussion, when $\alpha = 0.50$, calculate 0.50-dominance class and the description set(Table 4).

Table 4. 0.50-dominance class and the description set

0.50-dominance class	the description set
$[x_1]^{\geq 0.50} : \{x_1\}$	$Des_{C_{0.50}}(x_1) : [x_k]^{\geq 0.50}, k = 1, 2, \cdots, 6$
$[x_2]^{\geq 0.50} : \{x_1, x_2, x_4, x_5, x_6\}$	$Des_{C_{0.50}}(x_2) : [x_2]^{\geq 0.50}, [x_3]^{\geq 0.50}$
$[x_3]^{\geq 0.50} : \{x_1, x_2, \cdots, x_6\}$	$Des_{C_{0.50}}(x_3) : [x_3]^{\geq 0.50}$
$[x_4]^{\geq 0.50} : \{x_1, x_4\}$	$Des_{C_{0.50}}(x_4) : [x_k]^{\geq 0.50}, k = 2, 3, \cdots, 6$
$[x_5]^{\geq 0.50} : \{x_1, x_4, x_5, x_6\}$	$Des_{C_{0.50}}(x_5) : [x_2]^{\geq 0.50}, [x_3]^{\geq 0.50}, [x_5]^{\geq 0.50}$
$[x_6]^{\geq 0.50} : \{x_1, x_4, x_6\}$	$Des_{C_{0.50}}(x_6) : [x_2]^{\geq 0.50}, [x_3]^{\geq 0.50}, [x_5]^{\geq 0.50}, [x_6]^{\geq 0.50}$

By definition 8, calculate the indistinguishability set of every object as follows:
$Ind_{C_{0.50}}(x_1) = \{x_1\}, Ind_{C_{0.50}}(x_2) = \{x_1, x_2, x_4, x_5, x_6\}, Ind_{C_{0.50}}(x_3) = \{x_1, x_2, \cdots, x_6\}$,
$Ind_{C_{0.50}}(x_4) = \{x_1, x_4\}, Ind_{C_{0.50}}(x_5) = \{x_1, x_4, x_5, x_6\}, Ind_{C_{0.50}}(x_6) = \{x_1, x_4, x_6\}$.

Thus $M(C_{0.50}) = 1 - (1 + 5 + 6 + 2 + 4 + 3)/36 = 15/36$.

Since $0.50 > 0.45$, the result is $M(C_{0.50}) > M(C_{0.45})$, the conclusion of proposition 4 has been checked. On the other hand, it suggests that the uncertain information of covering $C_{0.50}$ is less than $C_{0.45}$, that is to say, the granularity of the space $(U, C_{0.50})$ is thinner than $(U, C_{0.45})$.

6 Conclusions

The dynamic analysis problems of IVFSs based on granular computing problems have been investigated in this paper. In order to build α-dominance relation of IVFSs, we present a new measure called the weighted possibility degree among objects in universe, and this relation could generate α-dominance class and α-covering approximation space of IVFSs. Furthermore, for the sake of observing dynamic changes of granularity under different precision, the concept of knowledge capacity is proposed. The example reflects that the dynamic changes of IVFSs' granular space based on the value of α changes. Actually, the knowledge capacity becomes larger along with the granularity of covering becomes thinner.

Acknowledgments. The work is supported by the NNSF of China (NO.61175046), and Provincial Nature Science Research Key Project for Colleges and Universities of Anhui Province(NO.KJ2013A033), and the Academic Innovation Team of Anhui University(NO.KJTD001B) and the Project of Graduate Academic Innovation of Anhui University(NO.10117700013,NO.10117700014).

References

1. Zadeh, L.A.: Fuzzy sets. Inform. Con. 8, 338–353 (1965)
2. Atanassov, K.: Intuitionistic fuzzy sets. Fuzzy Sets and Systems 20, 87–96 (1986)
3. Pawlak, Z.: Rough sets. Int. J. Inform. Comput. Sci. 11, 341–356 (1982)
4. Gau, W.L., Buehrer, D.J.: Vague sets. IEEE Transactions on Systems Man Cybernetics 23, 610–614 (1993)
5. Gorzalzany, M.B.: A method of inference in approaximate reasoning based on interval-valued fuzzy sets. Fuzzy Sets and Systems 21, 1–17 (1987)
6. Molodtsov, D.: Soft set theory-First results. Comput. Math. Appl. 37, 19–31 (1999)
7. Molodtsov, D.: The theory of soft sets. URSS Publishers, Moscow (2004) (in Russian)
8. Maji, P.K., Biswas, R., Roy, A.R.: Soft sets theory. Comput. Math. Appl. 45, 555–562 (2003)
9. Maji, P.K., Roy, A.R.: Application of soft sets in a decision making problem. Comput. Math. Appl. 44, 1077–1083 (2002)
10. Roy, A.R., Maji, P.K.: A fuzzy soft set theoretic approach to decision making problems. J. Comput. Appl. Math. 203, 412–418 (2007)
11. Kong, Z., Gao, L.Q., Wang, L.F.: Comment on "A fuzzy soft set theoretic approach to decision making problems". J. Comput. Appl. Math. 223, 540–542 (2009)
12. Feng, F., Jun, Y.B., et al.: An adjustable approach to fuzzy soft set based decision making. J. Comput. Appl. Math. 234, 10–20 (2010)
13. Jiang, Y.C., Tang, Y., Chen, Q.M.: An adjustable approach to intuitionistic fuzzy soft sets based decision making. Appl. Math. Model. 35, 824–836 (2011)
14. Majumdar, P., Samanta, S.K.: On similarity measures of fuzzy soft sets. Int. J. Adv. Soft. Comput. Appl. 2, 1–8 (2011)
15. Yang, X.B., Lin, T.Y., et al.: Combination of interval-valued fuzzy set and soft set. Comput. Math. Appl. 58, 521–527 (2009)
16. Feng, F., Li, Y.M., et al.: Application of level soft sets in decision making based on interval-valued fuzzy soft sets. Comput. Math. Appl. 60, 1756–1767 (2010)
17. Yao, D., Wang, C., Mao, J., Zhang, Y.: Granularity analysis of fuzzy soft set. In: Li, T., Nguyen, H.S., Wang, G., Grzymala-Busse, J., Janicki, R., Hassanien, A.E., Yu, H. (eds.) RSKT 2012. LNCS, vol. 7414, pp. 409–415. Springer, Heidelberg (2012)

A Novel MGDM Method Based on Information Granularity under Linguistic Setting

Yanan Fu[1], Danqing Xu[1], and Junjun Mao[1,2,3]

[1] School of Mathematical Sciences, Anhui University, Hefei, Anhui 230039, PR China
[2] Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Anhui University, 230039 Hefei, Anhui, PR China
[3] Department of Computer Science, University of Houston, Houston, TX 77204−3010 USA
maojunjunuh@gmail.com

Abstract. The aim of this paper is to investigate the multiple attribute group decision making(MGDM) problems under linguistic information, in which attribute weights and the expert weights are completely unknown, and the attribute values take the form of linguistic variables. Firstly, an objective method based on information granularity and entropy is proposed for acquiring attribute weights. The expert weights by use of attribute weights and the relative entropy are obtained. Secondly, we utilize the numerical weighting linguistic average operator to aggregate the linguistic variables corresponding to each alternative, and rank the alternatives according to the linguistic information. Finally, an illustrative example is given to verify practicality and effectiveness of the developed approach.

Keywords: multiple attribute group decision making, linguistic information, information granularity, entropy, relative entropy.

1 Introduction

Since MGDM problems with linguistic information have a wide application background, both theory and application of linguistic MGDM problems have received extensive attentions[1 − 11]. Slowinski[1] defined four classes of multi-attribute decision problems, depending on the structure of their representation, its interpretation and the kind of questions related. Then, they characterized the rough set methodology for each particular class of decision problems. Kacprzyk[2] proposed two types of measurements: consensus degrees and proximity measures. The first one is used to assess the agreement among all the experts' opinions, while the second one is used to find out how far the individual opinions are form the group opinion. Both types of measurements are computed at three different levels of representation of information: pair of alternatives, alternatives and experts. They have also shown how to make the measurement of consensus possible in multi-granular linguistic GDM problems, it was necessary to unify the different linguistic term sets into a single linguistic term. People usually combine MGDM methods with the four decision models to deal with linguistic

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 261–268, 2013.

MGDM problems: approximate model based on extension principle[3]; ordered language model[4]; 2-tuple model[5,6] and the model that computes with words directly[7]. Wei[6] proposed two extended 2-tuple aggregation operators, based on which a method for linguistic multiple attribute group decision making problems was proposed. Wu[8] put forward a maximizing deviation method based on linguistic weighted arithmetic averaging operator and non-linear optimization. Boran[9] combined $TOPSIS$ method with intuitionistic fuzzy set to deal with linguistic MGDM problems based on intuitionistic fuzzy weighted averaging operator. Xu[10] presented a linguistic hybrid averaging operator and studied some desirable properties of the above operator, then developed a practical approach to MGDM problems in linguistic setting. Xu[11] presented a uniform approach based on linguistic evaluation scale by introducing the concepts of virtual term and virtual term index, and then proposed a method based on the term indices for group decision making problems with multiple attribute linguistic information by defining the additive weighted mean operator and the hybrid aggregation operator. In addition, some researchers[$12-14$] also transformed linguistic information into fuzzy numbers (such as trapezoidal fuzzy numbers or triangular fuzzy numbers), and developed other linguistic MGDM methods by processing the above fuzzy numbers. By transforming multi-granularity uncertain linguistic terms into trapezoidal fuzzy numbers, Fan[14] proposed a group decision making method based on the extended $TOPSIS$ method. In [15] based on the existing MGDM methods with linguistic information, three key evaluation indices are presented to measure the results of MGDM from different aspect by Pang, et al. Different decision makers may provide multi-granular linguistic information in multi-criteria group decision making problems, so Herrera-Viedma et al. defined the measurements of consensus to help gain the more rational decision results[16], and paper[17] provided a way to use multi-granular linguistic model for management decision making in performance appraisal.

To develop a new method to evaluate the results of MGDM problems on linguistic information is aim of this paper. The rest of the paper is organized as follows. Section 2 introduces the operational laws of linguistic variables and briefly reviews the $NWLA$ operator and the relative entropy. In section 3 we present an new methods for MGDM based on the entropy and the relative entropy under linguistic environment. A practical example is given in Section 4. Section 5 concludes the paper.

2 Preliminaries

We consider a finite and totally ordered discrete linguistic label set$S = \{s_l | l = -L, \ldots, -1, 0, 1, \ldots, L\}$, where l is a positive integer, s_l represents a linguistic variable and satisfies $s_q > s_l$ if $q > l$.

Example 1. A set of five labels: $S = \{s_{-2} = very\ thin, s_{-1} = thin, s_0 = fair, s_1 = fat, s_2 = very\ fat\}$. Obviously, the mid linguistic label s_0 represents an assessment of "indifference", and with the rest of the linguistic labels being placed symmetrically around it. To preserve all the given information, the

discrete label set $\bar{S} = \{s_l | l \in [-Q, Q]\}$, where $Q(Q > L)$ is a lager rational number. If $s_l \in S$, then s_l is termed an original linguistic label, otherwise, s_l is termed a virtual linguistic label. For example, consider the discrete linguistic label set S given in Example 1, all the linguistic labels in S are original linguistic labels, the other linguistic labels which do not belong to S, such as $s_{-0.5}$ and $s_{1.5}$, are virtual linguistic labels, here $s_{-0.5}$ denotes a linguistic label located between "thin" and "fair", and $s_{1.5}$ denotes a linguistic label located between "fat" and "very fat".

In general, an expert uses the original linguistic labels to evaluate alternatives, and the virtual linguistic labels can only appear in operation.

Definition 1.[7] Consider any two linguistic labels $s_\alpha, s_\beta \in \bar{S}$, we define their operational laws as follows:
(1) $s_\alpha \oplus s_\beta = s_{\alpha+\beta}$;
(2) $\lambda s_\alpha = s_{\lambda\alpha}$, $\lambda \in [0, 1]$.

Definition 2.[18] Let $NWLA$: $\overline{S^m} \to \bar{S}$, if

$$NWLA_w(s_{\alpha 1}, s_{\alpha 2}, \ldots, s_{\alpha m}) = w_1 s_{\alpha 1} \oplus w_2 s_{\alpha 2} \oplus \cdots \oplus w_m s_{\alpha m} \triangleq \bigoplus_{j=1}^{m} w_j s_{\alpha j}$$

where $w = (w_1, w_2, \ldots, w_m)$ is the weighting vector of the linguistic variable $s_{\alpha j}$, and $w_j \geq 0, j = 1, 2, \ldots, m$, $\sum_{j=1}^{m} w_j = 1$, $s_{\alpha j} \in \bar{S}$, then $NWLA$ is called a numerical weighting linguistic average ($NWLA$) operator.

Definition 3.[19] Let $x_i, y_i \geq 0$, $i = 1, 2, \ldots, n$, and $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i = 1$, we define the relative entropy between discrete probability distributions $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ as follows:

$$h(X, Y) = \sum_{i=1}^{n} x_i \ln(\frac{x_i}{y_i}).$$

3 An Approach to Multiple Attribute Group Decision-Making under Linguistic Setting

Throughout this section, let $D = (d_1, d_2, \ldots, d_t)$ be a set of experts, and $U = (u_1, u_2, \ldots, u_t)$ be a weight vector of experts which is completely unknown, where $u_k \geq 0 (k = 1, 2, \ldots, t)$, $\sum_{k=1}^{t} u_k = 1$. Let $O = (o_1, o_2, \ldots, o_n)$ be a set of alternatives, $A = (a_1, a_2, \ldots, a_m)$ be a finite set of attributes. Suppose $A^{(k)} = (a_{ij}^{(k)})_{m \times n}$ is the decision matrix, where $a_{ij}^{(k)} \in \bar{S}$ takes the form of linguistic value, given by the expert $d_k \in D$, for alternative $o_i \in O$ with respect attribute $a_j \in A$. $w^{(k)} = (w_1^{(k)}, w_2^{(k)}, \ldots, w_m^{(k)})$ be a weight vector of attributes under the expert k which is also completely unknown, where $w_j^{(k)} \geq 0 (j = 1, 2, \ldots, m)$, $\sum_{j=1}^{m} w_j^{(k)} = 1$.

3.1 Attribute Weights Acquisition Based on Information Granularity

Given $a_j \in A$, let $ind(a_j) = (o_i, o_p)|a_{ij}^{(k)} = a_{pj}^{(k)} = s_l\}$, apparently, $ind(a_j)$ is the equivalence relation on O, and $O/ind(a_j)$ is the partition of O, shortly and conveniently denoted by $O/ind(a_j) = \{[o_1]_{a_j}, [o_2]_{a_j}, \ldots, [o_n]_{a_j}\}$.

Definition 4. Let $O/ind(a_j) = \{[o_1]_{a_j}, [o_2]_{a_j}, \ldots, [o_n]_{a_j}\}$, then the entropy $E_{a_j}^{(k)}$ of the attribute a_j given by an expert d_k is defined as:

$$E_{a_j}^{(k)} = -\frac{1}{|O|}\sum_{i=1}^{n}\log\frac{|[o_i]|_{a_j}}{|O|},\ j = 1, 2, \ldots, m,\ k = 1, 2, \ldots, t. \tag{1}$$

where $|[o_i]_{a_j}|$ is the cardinality of the equivalence class of o_i.

Obviously, we have $0 \le E_{a_j}^{(k)} \le ln|O|$. There are two special cases, one is that every alternative has all the same linguistic value s_l under the attribute a_j, then $E_{a_j}^{(k)} = 0$. The other is that each alternative has unique linguistic value under the attribute a_j, $E_{a_j}^{(k)} = ln|O|$. In the former case, the attribute a_j contribute nothing to the decision making process, so we can set less weight even 0 for it. While in the latter case, the a_j can distinguish the alternatives from each other, so we can set more weight for it. Just like shannon entropy, the information entropy $E_{a_j}^{(k)}$ can depict the distinguish ability of the attribute a_j. Therefore we can construct attribute weights based on the following thoughts: For a fixed expert, the more entropy value of one attribute, the bigger weight of this attribute. So we propose the attribute weighting method based on the entropy $E_{a_j}^{(k)}$ in the following, the weight of the attribute a_j under the expert d_k is given by

$$w_j^{(k)} = \frac{E_{a_j}^{(k)}}{\sum_{j=1}^{m} E_{a_j}^{(k)}}. \tag{2}$$

Apparently, $w_j^{(k)} \ge 0$, $\sum_{j=1}^{m} w_j^{(k)} = 1 (j = 1, 2, \ldots, m,\ k = 1, 2, \ldots, t)$.

3.2 Expert Weights Acquisition Based on the Relative Entropy

Owing to relative entropy usually describes discrimination information, we utilize the relative entropy to measure the difference evaluation results between individual expert and others. The discrimination decrease along with the smaller relative entropy value for an expert, then such an evaluation results given by the expert are better. Therefore, the expert should be assigned a bigger weight; Otherwise, such an expert should be evaluated as a very small weight. From Eq. (2) we know all the attribute value $W = (w^{(1)}, w^{(2)}, \ldots, w^{(t)})^T$, where $w^{(k)} = (w_1^{(k)}, w_2^{(k)}, \ldots, w_m^{(k)})$. Combining these two aspects, we have

Definition 5. The expert weight u_k is presented by

$$u_k = \frac{1 - H_k}{\sum_{k=1}^{t} (1 - H_k)}. \tag{3}$$

where $H_k = \sum_{p=1}^{t} \sum_{j=1}^{m} w_j^{(k)} ln \frac{w_j^{(k)}}{w_j^{(p)}}$ is the relative entropy between the expert d_k and others.

3.3 Aggregate Methods and Group Decision-Making Process

In the following we shall utilize the $NWLA$ operators to aggregate the linguistic variables corresponding to each alternative, and rank the alternatives by means of the linguistic information.

Utilize the decision information given in matrix $A^{(k)}$, and the operator:

$$\begin{aligned} z_i &= NWLA_{U,W}^k(a_{i1}^{(k)}, a_{i2}^{(k)}, \ldots, a_{im}^{(k)}) \\ &= \bigoplus_{k=1}^{t} u_k \bigoplus_{j=1}^{m} w_j^{(k)} a_{ij}^{(k)}, \quad k = 1, 2, \ldots, t, \quad j = 1, 2, \ldots, m. \end{aligned} \tag{4}$$

Then we rank o_i and select the best one(s) in accordance with the values of $z_i (i = 1, 2, \ldots, n)$.

Based on the above analysis, we develop a practical method for solving the MGDM problems, in which the information about attribute weights and the expert weights are completely unknown, and the attribute values take the form of linguistic variables.

The method involves the following steps:

Step 1. Let $A^{(k)} = (a_{ij}^{(k)})_{n \times m}$ be a linguistic decision matrix, given by an expert d_k, for the alternative $o_i \in O$ with respect attribute $a_j \in A$. We have $E^{(k)} = (E_{a_1}^{(k)}, E_{a_2}^{(k)}, \ldots, E_{a_m}^{(k)})$ by using Eq. (1).
Step 2. We solve the formula Eq. (2) to determine the attribute weights $w^{(k)} = (w_1^{(k)}, w_2^{(k)}, \ldots, w_m^{(k)})$, $k = 1, 2, \ldots, t$.
Step 3. We get the expert weights $U = (u_1, u_2, \ldots, u_t)$ by use of Eq. (3).
Step 4. Utilize the the attribute weights and expert weights and by Eq. (4), we obtain the aggregated values z_i of the alternatives $o_i (i = 1, 2, \ldots, n)$.
Step 5. Rank all the alternatives o_i by using the overall values $z_i (i = 1, 2, \ldots, n)$ and then get the most desirable one(s).

4 Illustrative Example

Suppose an investment company, which wants to invest a sum of money in the best option. The appropriate invest from among ten alternatives $O = \{o_1, o_2, \ldots, o_{10}\}$. The selection decision is made on the basis of four subjective attributes $A =$

$\{a_1, a_2, \ldots, a_4\}$. There are three expert $D = \{d_1, d_2, d_3\}$. The expert compare these ten companies with respect to the four attributes by using linguistic, and construct the linguistic decision matrix $A^{(k)} = (a_{ij}^{(k)})_{10\times 4}(k = 1, 2, 3)$, as listed in tables 1-3.where $s_{-1} = poor$, $s_0 = fair$, $s_1 = good$. In what follows, we apply the developed procedure to the selection of best investment company from the potential company $o_i(i = 1, 2, \ldots, 10)$.

Table 1. The linguistic decision matrix $A^{(1)}$

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
a_1	s_1	s_{-1}	s_{-1}	s_1	s_{-1}	s_0	s_0	s_1	s_1	s_0
a_2	s_1	s_0	s_{-1}	s_0	s_{-1}	s_0	s_{-1}	s_{-1}	s_{-1}	s_1
a_3	s_0	s_{-1}	s_{-1}	s_{-1}	s_{-1}	s_{-1}	s_0	s_1	s_0	s_0
a_4	s_1	s_0	s_0	s_0	s_0	s_0	s_{-1}	s_0	s_0	s_0

Table 2. The linguistic decision matrix $A^{(2)}$

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
a_1	s_1	s_{-1}	s_1	s_1	s_0	s_0	s_{-1}	s_1	s_1	s_0
a_2	s_0	s_0	s_0	s_0	s_{-1}	s_{-1}	s_{-1}	s_{-1}	s_{-1}	s_1
a_3	s_1	s_{-1}	s_{-1}	s_0	s_{-1}	s_{-1}	s_0	s_1	s_0	s_{-1}
a_4	s_1	s_0	s_0	s_0	s_0	s_0	s_0	s_0	s_0	s_0

Table 3. The linguistic decision matrix $A^{(3)}$

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}
a_1	s_0	s_{-1}	s_{-1}	s_{-1}	s_{-1}	s_{-1}	s_0	s_1	s_1	s_0
a_2	s_1	s_0	s_{-1}	s_{-1}	s_{-1}	s_0	s_1	s_1	s_{-1}	s_0
a_3	s_0	s_{-1}	s_{-1}	s_0	s_{-1}	s_{-1}	s_0	s_0	s_0	s_{-1}
a_4	s_1	s_0	s_1	s_{-1}	s_0	s_0	s_{-1}	s_0	s_0	s_0

We compute the weight of the attribute $w^{(1)} = (w_1^{(1)}, w_2^{(1)}, w_3^{(1)}, w_4^{(1)})$ by the method given in section 3.

$$O/a_1 : \{\{o_1, o_4, o_8, o_9\}, \{o_6, o_7, o_{10}\}, \{o_2, o_3, o_5\}\}$$
$$O/a_2 : \{\{o_1, o_{10}\}, \{o_2, o_4, o_6\}, \{o_3, o_5, o_7, o_8, o_9\}\}$$
$$O/a_3 : \{\{o_8\}, \{o_1, o_7, o_9, o_{10}\}, \{o_2, o_3, o_4, o_5, o_6\}\}$$
$$O/a_4 : \{\{o_1\}, \{o_2, o_3, o_4, o_5, o_6, o_8, o_9, o_{10}\}, \{o_7\}\}$$

Step 1: $E_{a_1}^{(1)} = -\frac{1}{10}(log\frac{4}{10} + log\frac{3}{10} + log\frac{3}{10} + log\frac{4}{10} + log\frac{3}{10} + log\frac{3}{10} + log\frac{3}{10} + log\frac{4}{10} + log\frac{4}{10} + log\frac{3}{10}) = 1.0889$

$E_{a_2}^{(1)} = 1.0297,\ \ E_{a_3}^{(1)} = 0.9433,\ \ E_{a_4}^{(1)} = 0.6390$

Step 2: $w_1^{(1)} = \frac{E_{a_1}^{(1)}}{\sum_{j=1}^{4} E_{a_j}^{(1)}} = \frac{1.0889}{1.0889+1.0297+0.9433+0.6390} = 0.294$

$w_2^{(1)} = 0.278,\ \ w_3^{(1)} = 0.255,\ \ w_4^{(1)} = 0.173.$

Likewise, we can get

$$w_1^{(2)} = 0.309, w_2^{(2)} = 0.284, w_3^{(2)} = 0.309, w_4^{(2)} = 0.098.$$

$$w_1^{(3)} = 0.278, w_2^{(3)} = 0.278, w_3^{(3)} = 0.187, w_4^{(3)} = 0.257.$$

Step 3: Then we obtain that

$$H_1 = \sum_{p=1}^{3}\sum_{j=1}^{4} w_j^{(1)} ln\frac{w_j^{(1)}}{w_j^{(p)}} = w_1^{(1)} ln\frac{w_1^{(1)}}{w_1^{(2)}} + w_2^{(1)} ln\frac{w_2^{(1)}}{w_2^{(2)}} + w_3^{(1)} ln\frac{w_3^{(1)}}{w_3^{(2)}} + w_4^{(1)} ln\frac{w_4^{(1)}}{w_4^{(2)}}$$
$$+ w_1^{(1)} ln\frac{w_1^{(1)}}{w_1^{(3)}} + w_2^{(1)} ln\frac{w_2^{(1)}}{w_2^{(3)}} + w_3^{(1)} ln\frac{w_3^{(1)}}{w_3^{(3)}} + w_4^{(1)} ln\frac{w_4^{(1)}}{w_4^{(3)}} = 0.0558$$

$H_2 = 0.1245$, $H_3 = 0.1467$

Therefore, we get the weight of experts:

$$u_1 = \frac{1 - H_1}{\sum_{k=1}^{3}(1 - H_k)} = 0.353, u_2 = 0.328, u_3 = 0.319.$$

Step 4: By $NWLA$ operators, we obtain the overall values z_i of the alternatives $o_i (i = 1, 2, \ldots, 10)$:

$$z_1 = s_{0.6659}, z_2 = s_{-0.5458}, z_3 = s_{-0.4232}, z_4 = s_{-0.1451}, z_5 = s_{-0.7168}$$

$$z_6 = s_{-0.4432}, z_7 = s_{-0.3463}, z_8 = s_{0.3867}, z_9 = s_{0.0139}, z_{10} = s_{0.0185}$$

Step 5: Rank all the alternatives o_i by using alternatives $z_i (i = 1, 2, \ldots, 10)$:

$$o_1 \succ o_8 \succ o_{10} \succ o_9 \succ o_4 \succ o_7 \succ o_3 \succ o_6 \succ o_2 \succ o_5$$

and thus o_1 is the best choice.

5 Conclusions

In this article, we have investigated the MGDM problems, in which the attribute values take the form of linguistic variables, and the information about attribute weights and the expert weights are completely unknown. In order to determine the attribute weights, a new entropy formula is proposed based on information granularity. Especially, for the situations where the information about the expert weights is completely unknown, we have provided an objective method for obtaining the expert weights. A new method has also been developed for ranking alternatives. A practical application of the developed method to selection of best investment company has also been given.

Acknowledgments. The work is supported by the NNSF of China (NO.61175046), and the Provincial Nature Science Research Key Project for Colleges and Universities of Anhui Province(NO. KJ2013A033) and the Academic Innovation Team of Anhui University(NO.KJTD001B) and the Project of Graduate Academic Innovation of Anhui University(NO. 10117700014, NO. 10117700013).

References

1. Pawlak, Z., Slowinski, R.: Rough set approach to multi-attribute decision analysis. European Journal of Operational Research 72, 443–459 (1994)
2. Kacprzyk, J.: Group decision making with a fuzzy linguistic majority. Fuzzy Sets and Systems 18, 105–118 (1986)
3. Parreiras, R.O., Ekel, P.Y., et al.: A flexible consensus scheme for multicriteria group decision making under linguistic assessments. Information Sciences 180, 1075–1089 (2010)
4. Bordogna, G., Fedrizzi, M., Passi, G.: A linguistic modelling of consensus in group decision making based on OWA operators. IEEE Transactions on Systems, Man and Cybernetics 27, 126–132 (1997)
5. Cabrerizo, F.J., Pérez, I.J., Herrera-Viedma, E.: Managing the consensus in group decision making in an unbalanced fuzzy linguistic context with incomplete information. Knowledge-Based Systems 23, 169–181 (2010)
6. Wei, G.W.: A method for multiple attribute group decision making based on the ET-WG and ET-OWG operators with 2-tuple linguistic information. Expert Systems with Applications 37, 7895–7900 (2010)
7. Xu, Z.S.: Deviation Measures of Linguistic Preference Relations in Group Decision Making. Omega-Int. J. Manage. S. 33, 249–254 (2005)
8. Wu, Z.B., Chen, Y.H.: The maximization deviation method for multiple attribute group decision making under linguistic environment. Fuzzy Sets and Systems 158, 1608–1617 (2007)
9. Boran, F.E., Genc, S., et al.: A multi-criteria intuitionistic fuzzy group decision making for supplier selection with TOPSIS method. Expert Systems with Applications 36, 11363–11368 (2009)
10. Xu, Z.S.: A note on linguistic hybrid arithmetic averaging operator in multiple attribute group decision making with linguistic information. Group Decision and Negotiation 15, 593–604 (2006)
11. Xu, Z.S.: A multi-attribute group decision making method based on term indices in linguistic evaluation scales. Journal of System Engineering 20, 84–88 (2005)
12. Liu, P.D.: A weighted aggregation operators multi-attribute group decision making method based on interval-valued trapezoidal fuzzy numbers. Expert Systems with Applications 38, 1053–1060 (2011)
13. Li, D.F.: Compromise ratio method for fuzzy multi-attribute group decision making. Applied Soft Computing 7, 807–817 (2007)
14. Fan, Z.P., Liu, Y.: A method for group decision-making based on multi-granularity uncertain linguistic information. Expert Systems with Applications 37, 4000–4008 (2010)
15. Pang, J.F., Liang, J.Y.: Evaluation of the results of multi-attribute group decision-making with linguistic information. Omega-Int. J. Science. S. 40, 294–301 (2012)
16. Herrera-Viedma, E., Mata, F.S., Martínez, L., Chiclana, F., Pérez, L.G.: Measurements of Consensus in Multi-granular Linguistic Group Decision-Making. In: Torra, V., Narukawa, Y. (eds.) MDAI 2004. LNCS (LNAI), vol. 3131, pp. 194–204. Springer, Heidelberg (2004)
17. de Andrís, R., García-Lapresta, J.L., Martínez, L.: A Multi-granular Linguistic Model for Management Decision-making in Performance Appraisal. Soft Comput. 14, 21–34 (2010)
18. Xu, Z.S.: A method for multiple attribute decision making with incomplete weight information in linguistic setting. Knowledge-Based Systems 20, 719–725 (2007)
19. Guiasu, S.: Information theory with application. McGraw-Hill, New York (1977)

The Impacting Analysis on Multiple Species Competition

Han-Bing Yan and Xu-Qing Tang*

School of Science, Jiangnan University, Wuxi 214122, China
txq5139@jiangnan.edu.cn

Abstract. Based on the predicted climatic data from 2041 to 2050 and the climatic data and the distribution data of the three tree species that *Larix gmelinii*, *Betula platyphylla Suk* and *Picea koraiensis Nakai* from 1981 to 1990 in Northeast China, 12 climatic factor indicators are extracted by using the theory and methods of hierarchical clustering and fusion technology related fuzzy proximity relations. The prediction mathematical model of tree species distribution is built by using the statistical theory and methods, its algorithm is studied, and the predicted distribution figures produced by single tree species and the ones produced by multiple tree species competition are obtained. By analyzing the prediction, the distributions of three tree species drift to the north. Furthermore, the climate change and competition among species are the main impact factors for predicting distributions of three tree species.

Keywords: Granular computing, tree species, climatic factors, hierarchical clustering, clustering fusion, distribution prediction.

1 Introduction

Global warming has now become the undoubted fact [1]. The climate factors change has profound impact on global ecosystem, especially terrestrial ecosystem, global phenology and species distribution [2]. Therefore the species distribution prediction under climate change is always one of the central issues of the climate change.

Parmesan and Yohe found that migration of species distribution is associated with climate change by analyzing distribution change of more than 1700 species in past 20-140 years [3]. Root et al. found that 80% of species migration is highly correlated to temperature change through the integrated analysis on 1473 species in 143 researches [4]. Some scholars also studied the impact of climate change on a variety of plants and animals, [5-7]. Recently, the impacting analysis on species distribution in different time sequences is increasingly focused on [1]. Erasmus et al. chose annually and monthly mean temperature, annually and monthly maximum temperature, etc [5]. Luoto et al. chose temperature and precipitation in the coldest month, accumulated temperature of greater than 5^oC, etc [6]. Forsman et al. chose annually mean temperature, temperature and precipitation in

* Corresponding author.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 269–276, 2013.

breeding season, etc [7]. These studies all shown that species distribution prediction is closely related to the climatic factors such as temperature, precipitation, evaporation in the distribution. Therefore extracting appropriate climatic factor from the climate data of different time sequences is critical to species distribution prediction.

As the typical tree species in Northeast China, *Larix gmelinii*, *Betula platyphylla Suk* and *Picea koraiensis Nakai* have the very high ecological and economic value [8-10]. On the basis of climate data in the distribution of species, this paper analyzes the impact on their distributions from 2041 to 2050.

2 Data Source

The study area is located in Northeast China, including Liaoning, Jilin, Heilongjiang and eastern Inner Mongolia, belonging to temperate continental monsoon climate. Its area is 147 square kilometer, accountting for 15.3% of the national land area. Based on the survey data provided by Shenyang institute of applied ecology of CAS over the years and the related literatures, the distribution data of three tree species in Northeast China from 1981 to 1990 is obtained by applying GIS. The original distributions (OD) of three tree species are shown in Fig.1. The climate scenario data is obtained by applying MIROC-RegCM based on the data from 1951 to 2000 in China, including 20C3M (1951-2000) and SRESA1B (2001-2100). The average climate data in Northeast China from 1981 to 1990 and from 2041 to 2050 are obtained, and the grid point numbers of *Larix gmelinii*, *Betula platyphylla Suk* and *Picea koraiensis Nakai* denotes $M_1 = 197943$, $M_2 = 155989$, $M_3 = 13085$, respectively. The 5 climatic factors data such as monthly mean evaporation, monthly mean precipitation, monthly mean temperature, monthly mean maximum temperature and monthly mean minimum temperature in the distributions of three tree species are extracted.

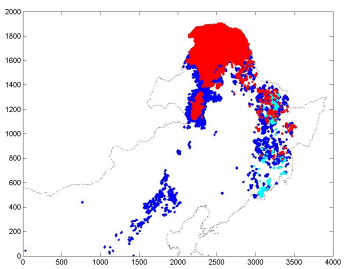


Fig. 1. OD of three tree species in Northeast China, where the red, the blue and the turquoise is *Larix gmelinii*, *Betula platyphylla Suk*, *Picea koraiensis Nakai*, respectively

3 Data Processing

3.1 Extracting Climatic Factor Indicators

On the basis of the above climatic factors data, the research is carried out on extracting the climatic factors impacting tree species in Northeast China by using the hierarchical clustering and fusion technology related fuzzy proximity relations [11-14]. Synthesizing the 5 climatic factors, we obtain that the optimal cluster of the 12 months in the distributions of three tree species is $\{\{1,2,3,4,10,11,12\},\{5,6,7,8,9\}\}$.The result can be interpreted as that the growing period of three tree species is May to September and the non-growing period of them is January to April and October to December every year. These coincide with the conclusions of the related references [15-17]. Therefore, 12 climatic factor indicators impacting three tree species are extracted as follows: E, P, T, T_{max}, T_{min} denotes the annually monthly mean evaporation, precipitation, temperature, maximum temperature and minimum temperature, respectively; E_{5-9}, P_{5-9}, T_{5-9}, T_{max5-9}, T_{min5-9} denotes the monthly mean evaporation, precipitation, temperature, maximum temperature and minimum temperature during May to September, respectively; $T_{max1-4,10-12}$, $T_{min1-4,10-12}$ is the monthly mean maximum temperature and the monthly mean minimum temperature during January to April and October to December.

3.2 Basic Assumptions

The research work is carried out based on the following assumptions.

Assumption 1: The data on the 12 climatic factor indicators in tree species distribution is a random distribution, respectively.

Assumption 2: The dependence of tree species on climatic factors is stable and independent, regardless of the change of the biological and ecological characteristics of tree species, especially their adaptive changes on climatic factors.

Assumption 3: The numerical values of the 12 climatic factor indicators in tree species distribution limit the range of adaptable distribution of tree species.

Form Assumption 1 and 3, the adaptable distribution area of species is determined by the probability related climatic factors. In this paper, the probability of the optimal adaptable distribution (OAD), the one of the medium adaptable distribution (MAD) and the one of the general adaptable distribution (GAD) is token 90% ($\alpha = 0.1$), 95% ($\alpha = 0.05$) and 99% ($\alpha = 0.01$), respectively.

Assumption 4: The distribution of species in the same ecological community meets Gauss competitive exclusion principle (Gauss, 1934).

Assumption 5: All tree species have the same annually diffusion rate under the natural state.

3.3 Confidence Intervals of Climatic Factor Indicators

On the basis of the climatic factors data in OD, the confidence intervals at the level $1-\alpha$ of 12 climatic factor indicators in three tree species distribution are

processed. Let The data of a climatic factor index be $x_1\, x_2, \cdots, x_n$, its order statistics is marked $x_{(1)}, x_{(2)}, \cdots, x_{(n)}$, satisfying $x_{(1)} \leqslant x_{(2)} \leqslant \cdots \leqslant x_{(n)}$. So the steps for calculating the confidence interval $[a, b]$ at the level $1-\alpha$ of the index are as follows: (1) Take $[x_{(k)}, x_{(m)}]$ by using the optimal model $min\{|x_{(m)} - x_{(k)} \mid |(m-k+1)/n \geqslant 1-\alpha)\}$; (2) Calculate $a = (x_{(k)} + x_{(k-1)})/2$, $b = (x_{(m)} + x_{(m+1)})/2$. So the confidence intervals at the level $1-\alpha$ of 12 climatic factor indicators in the three tree species distribution are calculated. For example, the results of *Larix gmelinii* are shown in Table 1. The other tables are omitted, here.

Table 1. Confidence intervals at the level $1-\alpha$ of 12 climatic factor indicators in the original distribution of *Larix gmelinii*

Indicator	E	E_{5-9}	P	P_{5-9}	T	T_{max}
$\alpha = 0.1$	[43.84,51.73]	[82.94,97.66]	[51.80,84.58]	[92.40,154.66]	[-4.09,0.51]	[8.33,13.36]
$\alpha = 0.05$	[43.68,55.39]	[82.98,105.66]	[49.82,87.30]	[90.68,162.14]	[-4.64,0.93]	[7.98,14.17]
$\alpha = 0.01$	[43.68,61.58]	[82.70,115.20]	[47.02,93.18]	[86.68,174.66]	[-5.14,2.74]	[7.28,15.32]
Indicator	T_{min}	T_{5-9}	T_{max5-9}	T_{min5-9}	$T_{max1-4,10-12}$	$T_{min1-4,10-12}$
$\alpha = 0.1$	[-14.54,-9.96]	[10.62,15.68]	[24.12,29.66]	[0.74,5.64]	[-3.04,1.90]	[-25.81,-21.19]
$\alpha = 0.05$	[-15.18,-9.63]	[10.06,15.98]	[23.60,30.12]	[0.36,5.90]	[-3.51,2.61]	[-26.09,-20.26]
$\alpha = 0.01$	[-15.68,-7.95]	[9.46,17.34]	[23.00,31.34]	[-0.26,6.74]	[-3.91, 4.21]	[-26.81,-18.44]

3.4 Predicting Single Tree Species Distribution

According to Assumption 1-3 and Table 1, an algorithm for predicting the distribution of single tree species from 2041 to 2050 is given. For example, the algorithm for predicting distribution of *Larix gmelinii* is given as follows.

Algorithm A:

Step 1: Take out the data y^s_{ijk} of each grid point (i, j) from the predicted data of 2041-2050 on 5 climatic factors, where i denotes column number of each data matrix, j is row number, k is month number, $s = 1, 2, 3, 4, 5$ are the 5 indicators respectively, and $\Omega = \{(i, j) | i = 1, 2, \cdots, 3930, j = 1, 2, \cdots, 1923\}$.

Step 2: Statistically calculate the 12 climatic factor indicators of each point in grid Ω based on Step 1, the results are expressed as Z^t_{ij} respectively, $t = 1, 2, \cdots, 12$ is respectively the 12 climatic factor indicators. Thus matrices Z^t on the 12 climatic factor indicators is obtained.

Step 3: Given α, test the numerical values of the 12 climatic factor indicators of 2041-2050 in each point in OD of *Larix gmelinii* based on Table 1 and Step 2. If their values are in the confidence intervals at the level $1-\alpha$, the point is reserved. So, the reserved area of *Larix gmelinii* from 2041 to 2050 is obtained.

Step 4: Based on the reserved area, select a circle of points (Note: Extending a circle denotes extending 1 km), where the points are extension points. By testing every climatic factor indicator in each extension point, if an extension point passes the test, it is a reserved point and added it into the reserved area.

Step 5: Repeat Step 4 until no extension point passes the test or extension number reaches K.
Step 6: Output the points in the reserved area.

Remark 1: In Step 6, K is computed as follows. If the natural diffusion rate of species is t (km/year), then the diffusion distance is $r \cdot t$ (km) after t years, and $K = [r \cdot t]$, where the value r depends on the biological characteristics of species. Under natural state, the natural diffusion speeds of three tree species are slow, so we take $r = 1$ km/year. For example, there are 60 years from 1981-1990 year to 2041-2050 year, and the diffusion distance is 60 km, so $K = 60$. In a general way, we can adjust the grid interval to determine the value of K. For example, adjusting the grid interval to be $a = r$ (km), 60 years have passed from 1981-1990 year to 2041-2050 year, so $K = 60$.

By Algorithm A, the OAD, MAD and GAD produced from *Larix gmelinii* in Northeast China from 2041 to 2050 are shown in Fig. 2. Similarly, we obtain the predicted figures about OAD, MAD and GAD of *Betula platyphylla Suk* and *Picea koraiensis Nakai*, here we omit them.

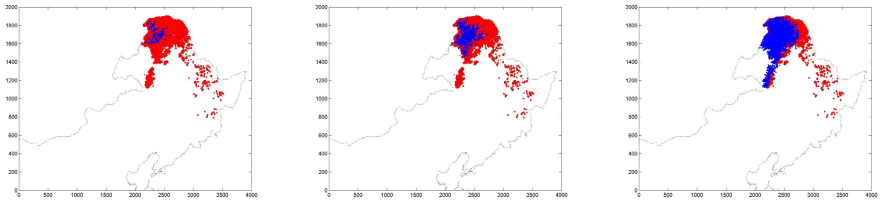


Fig. 2. Prediction of OAD, MAD and GAD of *Larix gmelinii* in Northeast China from 2041 to 2050, where the red denotes the OD and the blue is the prediction distribution

The point number in OAD, MAD, GAD and OD are shown in Table 2.

Table 2. The statistical table of point number in OAD, MAD, GAD and OD

Species	OD	OAD	MAD	GAD
Larix gmelinii	197943	10337	40398	165160
Betula platyphylla Suk	155989	230881	296109	520776
Picea koraiensis Nakai	13085	167	1204	21254

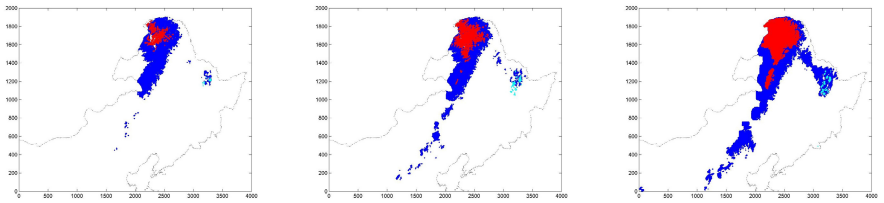
The prediction figures of three tree species, especially *Larix gmelinii* and *Betula platyphylla Suk*, have a lot of overlap. These reflect that three tree species have the similar niche and they usually form mixed forest [18].

3.5 Predicting Multiple Tree Species Competition Distribution

According to Assumption 4 and 5, and the distribution prediction about single tree species in section 3.4, an algorithm for predicting the distribution by multiple species competition from 2041 to 2050 is given as follows.

Algorithm B:
Step 1: Test the values of the 12 climatic factor indicators of 2041-2050 in each point in the OD of 3 tree species based on Z_t in Algorithm A and the corresponding confidence interval tables. If all climatic factor indicator values of a tree species in a point are in the corresponding confidence intervals at the level $1-\alpha$, the point is reserved. Thus the reserved areas of the 3 tree species are obtained, marked the reserved areas of three tree species is respectively S_1, S_2 and S_3, and $S_0 = \cup_{i=1}^{3} S_i$, $\overline{S_0} = \Omega \setminus S_0$.
Step 2: Select a circle of extending points to the outside S_0, and test each new extending point: If the extending point only pass the testing of a tree species, the point is added into the reserved area of the tree species; If the extending point adapts to 2 or more tree species simultaneously, the point belongs to the tree species reserved area that it is the nearest to the point according to Assumption 4 and 5; If no new point is incorporated into the reserved area of a certain tree species in a certain round test, stop the extension of this tree species, and continue the extension of the other tree species.
Step 3: Repeat Step 2 until no extension point passes the test or extension number reaches K.
Step 4: Output the points in the reserved areas.

By Algorithm B, the optimal adaptable competition distribution (OACD) , the medium adaptable competition distribution (MACD) and the general adaptable competition distribution (GACD) produced by three tree species competition from 2041 to 2050 are shown in Fig.3. Furthermore, the point number in OACD, MACD, GACD and OD are shown in Table 3.



OACD MACD GACD

Fig. 3. Prediction of OACD, MACD and GACD of three tree species in Northeast China from 2041 to 2050, where the red, the blue and the turquoise is respectively *Larix gmelinii*, *Betula platyphylla Su* and *Picea koraiensis Nakai*

Table 3. The statistical table of point number in OACD, MACD, GACD and OD

Species	OD	OACD	MACD	GACD
Larix gmelinii	197943	10123	39084	142713
Betula platyphylla Suk	155989	224506	268324	403689
Picea koraiensis Nakai	13085	165	957	1329

4 Discussion and Conclusions

From the predicted figures about single tree species distribution from 2041 to 2050, the corresponding characteristics are obtained as follows: (1) Vast area in OD is no longer suitable for the survival of three tree species; (2) The distributions of three tree species drift to the north under climate change. Furthermore, *Larix gmelinii* and *Betula platyphylla Suk* drift to the northwest, and *Picea koraiensis Nakai* drifts to the northeast. These phenomena are consistent with the conclusions of the related references [3,19,20]; (3) The distribution areas reduce significantly, especially *Larix gmelinii* and *Picea koraiensis Nakai*.

Based on Table 2,3 and Fig.3, the compared table of OAD, MAD, GAD, OACD, MACD and GACD related to OD are shown in Table 4.

Table 4. The percentages of the area of OAD, MAD, GAD, OACD, MACD and GACD related to the area of OD

Species	OAD	MAD	GAD	OACD	MACD	GACD
Larix gmelinii	5.22%	20.41%	83.44%	5.11%	19.75%	72.10%
Betula platyphylla Suk	148.01%	189.83%	333.85%	143.92%	172.01%	258.79%
Picea koraiensis Nakai	1.28%	9.20%	162.43%	1.26%	7.31%	10.16%

From Table 4, the impacts on their distributions by introducing the competition mechanism among three tree species can be interpreted as follows: (1) The competitiveness of *Larix gmelinii* is the strongest in three tree species, i.e., it has the apical dominance, and the climate change is not conducive to its survival. So, the climate change is an important impact factor for predicting its distribution; (2) The competitiveness of *Picea koraiensis Nakai* is the weakest among the three tree species, and the climate change is conducive to the its survival. Thus, the competition among species is an important impact factor for predicting its distribution; (3) The competitiveness of *Betula platyphylla Suk* is medium, and the climate change is conducive to its survival. Thus, the climate change and competition among species are the main impact factors for predicting its distribution.

Acknowledgement. The research is supported in part by the National Natural Science Foundation of China under Grant 11371174 and 11271163, the support of Special Scientific Foundation for Nonprofit Public Industry–Environmental Protection under Grant 200909070 (China) and the Fundamental Research Funds for the Central Universities under Grant JUSRP51317B (China).

References

1. IPCC: Climate change 2007: The physical science basis-Summary for policymakers of the working group I report. Cambridge University Press, Cambridge (2007)
2. Fang, J.Y.: Global ecology. Higher education Press-Springer Verlag, Beijing (2000)
3. Parmesan, C., Yohe, G.: A globally coherent fingerprint of climate change impact across natural systems. Nature 421, 37–42 (2003)
4. Root, T.L., Price, J.T., Hall, K.R., Schneider, S.H., et al.: Fingerprints of global warming on wild animals and plants. Nature 421, 57–60 (2003)
5. Erasmus, B.F.N., Vanjaarsveld, S.A., Chown, L.S., et al.: Vulnerability of South African animal taxa to climate change. Global Change Biology 8, 679–693 (2002)
6. Luoto, M., Poyry, J., Heikkinen, R.K., et al.: Uncertainty of bioclimatic envelope models based on the geographical distribution of species. Global Ecology and Biogeography 14, 575–584 (2005)
7. Forsman, J.T., Monkkoonen, M.: The role of climate in limiting European resident bird popu-lations. Journal of Biogeography 30, 55–70 (2003)
8. Gao, J.W., Ao, W.J., et al.: Origin and biological characteristics of Larix gmelinii in Great Xingan Mountain. Inner Mongolia Science & Economic 10, 99–100 (2003)
9. Sun, Y.H., Yu, H.: A Summary of Research Progress of Picea koraiensis Forest. Forestry Exploration Design 4, 93–94 (2010)
10. Li, J., Luo, Y.Q., Shi, J.: The optimum mixture ratio of larch and birch in terms of biodiversity conservation: a case study in Aershan forest area. Acta Ecologica Sinica 32(16), 4943–4949 (2012)
11. Pedrycz, W.: Collaborative fuzzy clustering. Pattern Recognition Letters 23, 1675–1686 (2002)
12. Zadeh, L.A.: The roles of fuzzy logic and soft computing in the conception, design and deployment of intelligent systems. In: Nwana, H.S., Azarmi, N. (eds.) Software Agents and Soft Computing: Towards Enhancing Machine Intelligence. LNCS, vol. 1198, pp. 181–190. Springer, Heidelberg (1997)
13. Zhang, L., Zhang, B.: Theory of Problem Solving and Its Applications-The Theory and Methods of Quotient Granular Computing. Tsinghua University Press, Beijing (2007)
14. Tang, X.Q., Zhu, P.: Hierarchical clustering problems and analysis of fuzzy proximity relation on granular space. IEEE Transactions on Fuzzy Systems (2012), doi:10.1109/TFUZZ.2012.2230176
15. Wen, X.Q., Gao, Y.G., Wang, Y.G., et al.: Response of Larix gmelinii, spruce and Korean pine to meteorological conditions during phenological phase. Heilongjiang Meteorological 4, 34–36 (2005)
16. Wang, Q.G., Xing, Y.J., Zhou, X.F., Han, S.J.: Ecological and Biological Characteristics of Spruce in Low-lying Land in Eastern Mountain Area of Heilongjiang Province. Journal of Northeast Forestry University 35(3), 4–6 (2007)
17. Chen, S.S., Liu, H.Y., Guo, D.L.: Litter stocks and chemical quality of natural birch forests along temperature and precipitation gradients in eastern Inner Mongolia, China. Chinese Journal of Plant Ecology 34(9), 1007–1015 (2010)
18. Heilongjiang forest editorial board: Heilongjiang forest. Northeast Forestry University Press, Harbin (1993)
19. Cao, F.X., Xu, Q.J., Cao, S.J., Long, J.X., Qi, C.J.: Advances of Global Warming Impact on Species Distribution. Journal of Central South University of Forestry & Technology 28(6), 86–89 (2008)
20. Zhang, L., Liu, S.R., Sun, P.S., Wang, T.L.: Comparative evaluation of multiple models of the effects of climate change on the potential distribution of Pinus massoniana. Chinese Journal of Plant Ecology 35(11), 1091–1105 (2011)

Topological Characterizations for Three Covering Approximation Operators*

Aiping Huang and William Zhu**

Lab of Granular Computing,
Minnan Normal University, Zhangzhou 363000, China
williamfengzhu@gmail.com

Abstract. The topological properties of coverings and their corresponding covering approximation operators have drawn special attention because these topological properties have important applications in rough sets. In this paper, we present some topological characterizations for three covering approximation operators. In the first part, we present certain topological characterizations for the covering lower approximation operator in an infinite universe, while the topological characterizations for the first and the second types of covering upper approximation operators are studied in a finite universe. In the second part, the relationships among three operators and the relationships among three topological spaces are established. In a word, topology theory provides useful tools to study covering-based rough sets.

Keywords: Covering, Approximation operator, Topology, Closure operator.

1 Introduction

Covering is an important data structure and it is widely used to represent data sets in practical applications [1, 3, 10]. As a useful tool to deal with covering data, covering-based rough sets have been attracting more and more research interest [9, 11, 14, 16]. However, diverse problems in covering-based rough sets are NP-hard and the algorithms to solve them are almost greedy ones. In order to establish applicable mathematical structures for these problems, covering-based rough sets are combined with some other theories and methods, especially, topology. The topology provides mathematical tools and interesting topics in studying information systems and rough sets [5–7, 13]. Therefore, the connection between covering-based rough sets and topology has deep theoretical and practical significance beyond doubt.

In this paper, we present topological characterizations for three covering approximation operators. In the first part, some topological characterizations for three covering approximation operators are studied. In an infinite universe, certain necessary and sufficient conditions are given to make a covering lower approximation operator into an

* This work is supported in part by the National Natural Science Foundation of China under Grant No. 61170128, the Natural Science Foundation of Fujian Province, China, under Grant No. 2012J01294, and the Science and Technology Key Project of Fujian Province, China, under Grant No. 2012H0043.

** Corresponding author.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 277–284, 2013.

interior operator. Similarly, we also discuss the other two covering upper approximation operators in a finite universe. In the second part, the three operators and the three topological spaces corresponding to them are compared.

The paper is organized as follows. In Section 2, we present some fundamental concepts of covering-based rough sets and topology. Section 3 presents some topological characterizations for three covering approximation operators. In Section 4, we compare three covering approximation operators and the topological spaces corresponding to them. Section 5 conclusions this paper.

2 Basic Definitions

In this section, we present some fundamental concepts relative to Pawlak's rough sets, covering-based rough sets and topology.

2.1 Pawlak's Rough Sets and Covering-Based Rough Sets

In Pawlak's rough set theory, the lower and upper approximation operations are two key concepts. Let U be a finite set and R an equivalence relation of U. $\forall X \subseteq U$, the lower and upper approximations of X are defined as follows, respectively:
$R_*(X) = \bigcup\{P_i \in U/R : P_i \subseteq X\}$,
$R^*(X) = \bigcup\{P_i \in U/R : P_i \bigcap X \neq \emptyset\}$.

Proposition 1. *[8] Let $\emptyset$ be the empty set and $-X$ the complement of X in U. Pawlak's rough sets have the following properties:*

(1L) $R_*(U) = U$	*(1H)* $R^*(U) = U$
(2L) $R_*(\emptyset) = \emptyset$	*(2H)* $R^*(\emptyset) = \emptyset$
(3L) $R_*(X) \subseteq X$	*(3H)* $X \subseteq R^*(X)$
(4L) $R_*(X \bigcap Y) = R_*(X) \bigcap R_*(Y)$	*(4H)* $R^*(X \bigcup Y) = R^*(X) \bigcup R^*(Y)$
(5L) $R_*(R_*(X)) = R_*(X)$	*(5H)* $R^*(R^*(X)) = R^*(X)$
(6L) $R_*(-X) = -R^*(X)$	*(6H)*$R^*(-X) = -R_*(X)$
(7L) $X \subseteq Y \Rightarrow R_*(X) \subseteq R_*(X)$	*(7H)* $X \subseteq Y \Rightarrow R^*(X) \subseteq R^*(X)$
(8L) $R_*(R_*(X)) = R_*(X)$	*(8H)* $R^*(R^*(X)) = R^*(X)$
(9L) $\forall K \in U/R, R_*(K) = K$	*(9H)* $\forall K \in U/R, R^*(K) = K$

Next, we review some concepts of covering-based rough sets and some types of covering approximation operators. If $\mathcal{C}$ is a family of nonempty subsets of U and $\bigcup \mathcal{C} = U$, then $\mathcal{C}$ is called a covering of U. Let $\mathcal{C}$ be a covering of U and $x \in U$. Denote $Md(x) = \{K \in \mathcal{C} : x \in K \; and \; \forall S \in \mathcal{C}(x \in S \; and \; S \subseteq K \Rightarrow K = S)\}$, $I(x) = \bigcup_{x \in K} K$ and $N(x) = \bigcap_{x \in K} K$. $Md(x)$, $I(x)$ and $N(x)$, which are called the minimal description of x, the indiscernible neighborhood of x and the neighborhood of x, were first proposed in [2], [12] and [13], respectively. For all $x \in U$, if $|Md(x)| = 1$ then $\mathcal{C}$ is called a unary covering. This concept was first proposed in [12]. The following definition reviews some types of covering approximation operators.

Definition 1. *[15] Let $\mathcal{C}$ be a covering of U and $X \subseteq U$. One can define the operators as follows:*

$CL(X) = \bigcup\{K \in \mathcal{C} : K \subseteq X\}$,
$SL(X) = \{x \in U : \forall K \in \mathcal{C}(x \in K \Rightarrow K \subseteq X)\} = \{x \in X : I(x) \subseteq X\}$,
$FH(X) = CL(X) \bigcup(\bigcup\{\bigcup Md(x) : x \in (X \setminus CL(X))\})$,
$SH(X) = \bigcup\{K : K \bigcap X \neq \emptyset\} = \bigcup\{I(x) : x \in X\}$,
$IH(X) = CL(X) \bigcup\{N(x) : x \in X \setminus CL(X)\} = \bigcup_{x \in X} N(x)$,

CL, FH, SH and IH are called the covering lower approximation operator, the first, the second and the fifth type of covering upper approximation operators, respectively. Note that the operators SH and SL are dual. With respect to the properties of Pawlak's rough sets listed in Proposition 1, the following holds.

Proposition 2. *[15] SL has properties (1L), (2L), (3L), (4L) and (7L) of Proposition 1, and SH has the properties (1H), (2H), (3H), (4H) and (7H) of Proposition 1.*

Proposition 3. *[13] IH has properties (1H), (2H), (3H), (4H), (5H), (7H) and (9H) of Proposition 1.*

2.2 Topology

In this subsection, we remind some concepts of topology which can be found in [4].

Definition 2. *A topological space is a pair $(U, \mathcal{T})$ consisting of a set U and a family $\mathcal{T}$ of subsets of U satisfying the following conditions:*
(O1) $\emptyset, U \in \mathcal{T}$.
(O2) $\mathcal{T}$ is closed under arbitrary unions.
(O3) $\mathcal{T}$ is closed under finite intersections.

The subsets of U belonging to the topology $\mathcal{T}$ are called the open sets of the space, and their complements are called closed sets. A family of sets $\mathcal{B} \subseteq \mathcal{T}$ is called a base for a topology $\mathcal{T}$ if any open set can be expressed as a union of some elements of $\mathcal{B}$. A topology in which any open set is simultaneously closed is called a clopen topology.

Proposition 4. *Let U be a set. The operator $cl : P(U) \to P(U)$ (resp. $i : P(U) \to P(U)$) is a closure (resp. an interior) operator of a topological space if and only if it satisfies the following axioms.*
(I): $\forall X, Y \subseteq U$, $cl(X \bigcup Y) = cl(X) \bigcup cl(Y)$ (resp. $i(X \bigcap Y) = i(X) \bigcap i(Y)$),
(II): $\forall X \subseteq U, X \subseteq cl(X)$ (resp. $i(X) \subseteq X$),
(III): $cl(\emptyset) = \emptyset$ (resp. $i(U) = U$),
(IV): $\forall X \subseteq U, cl(cl(X)) = cl(X)$ (resp. $i(i(X)) = i(X)$).

3 Topological Characterization for Three Covering Approximation Operators

In covering-based rough sets, any set is approximated by basic knowledge, and accuracy is heavily determined by the knowledge, as well. For the reasons, it is necessary for us to study the introduced operators in details.

3.1 Topological Characterization for the Operator CL

In this subsection, we present some topological characterizations for the covering lower approximation operator in an infinite universe.

Lemma 1. *CL has properties (1L), (2L), (3L), (5L), (7L) and (9L) of Proposition 1 in an infinite universe.*

The following proposition presents a necessary and sufficient condition for operator CL to satisfy property $(4L)$ of Proposition 1.

Proposition 5. *Let $\mathcal{C}$ be a covering of U. $\forall X, Y \subseteq U$, $CL(X \bigcap Y) = CL(X) \bigcap CL(Y)$ iff $\forall K_1, K_2 \in \mathcal{C}$ and $x \in K_1 \bigcap K_2$, there exists $K \in \mathcal{C}$ such that $x \in K \subseteq K_1 \bigcap K_2$.*

Proof. ("$\Rightarrow$":) As we know, for all $K \in \mathcal{C}$, $CL(K) = K$. According to the assumption, we know for all $K_1, K_2 \in C$, $CL(K_1 \bigcap K_2) = CL(K_1) \bigcap CL(K_2)$. Hence $CL(K_1 \bigcap K_2) = K_1 \bigcap K_2$. If $x \in K_1 \bigcap K_2$, then $x \in CL(K_1 \bigcap K_2)$, that is, there exists $K \in \mathcal{C}$ such that $x \in K \subseteq K_1 \bigcap K_2$. ("$\Leftarrow$":) According to Lemma 1, $CL(X \bigcap Y) \subseteq CL(X) \bigcap CL(Y)$. Now we need to prove $CL(X) \bigcap CL(Y) \subseteq CL(X \bigcap Y)$. For all $x \in CL(X) \bigcap CL(Y)$, there exist $K_1, K_2 \in \mathcal{C}$ such that $x \in K_1 \subseteq X$ and $x \in K_2 \subseteq Y$, that is, $x \in K_1 \bigcap K_2 \subseteq X \bigcap Y$. According to the assumption, there exists $K \in \mathcal{C}$ such that $x \in K \subseteq K_1 \bigcap K_2$, thus $x \in CL(X \bigcap Y)$. Hence $CL(X) \bigcap CL(Y) \subseteq CL(X \bigcap Y)$.

Based on the result, a necessary and sufficient condition for operator CL to be an interior operator is presented.

Corollary 1. *Let $\mathcal{C}$ be a covering of U. CL is an interior operator iff $\forall K_1, K_2 \in \mathcal{C}$ and $x \in K_1 \bigcap K_2$, there exists $K \in \mathcal{C}$ such that $x \in K \subseteq K_1 \bigcap K_2$.*

In fact, we can show the other necessary and sufficient condition for operator CL to be an interior operator from the viewpoint of unary covering.

Lemma 2. *[5] Let $\mathcal{C}$ be a covering of U. $\mathcal{C}$ is unary iff $\forall K_1, K_2 \in \mathcal{C}$ and $x \in K_1 \bigcap K_2$, there exists $K \in \mathcal{C}$ such that $x \in K \subseteq K_1 \bigcap K_2$.*

Corollary 2. *CL is an interior operator iff covering $\mathcal{C}$ is unary.*

The following proposition provides another necessary and sufficient condition for operator CL to be an interior operator in terms of topological bases.

Lemma 3. *[4] A family $\mathcal{A}$ of U is a base for a topology $\mathcal{T}$ on U iff $\forall K_1, K_2 \in \mathcal{A}$ and $x \in K_1 \bigcap K_2$, there exists $K \in \mathcal{A}$ such that $x \in K \subseteq K_1 \bigcap K_2$.*

Proposition 6. *CL is an interior operator iff covering $\mathcal{C}$ is a base for a topology $\mathcal{T}_{CL} = \{X \subseteq U : CL(X) = X\}$.*

Proof. ("$\Leftarrow$":) It is obvious. ("$\Rightarrow$":) We need to prove $\mathcal{C}$ is a base for a topology $\mathcal{T}_{CL}$. For all $K \in \mathcal{C}$, $CL(K) = K$. Then $\mathcal{C} \subseteq \mathcal{T}_{CL}$. For all $X \in \mathcal{T}_{CL}$, $X = CL(X) = \bigcup\{K \subseteq U : K \subseteq X\}$. Hence, $\mathcal{C}$ is a base of $\mathcal{T}_{CL}$.

3.2 Topological Characterization for the Operator FH

In this subsection, we provide some topological characterizations for the first type of covering upper approximation operator in a finite universe. First, we introduce a result.

Lemma 4. *[16] $\mathcal{C}$ is a unary covering iff FH is a closure operator.*

We find that the condition for FH to be a closure operator is the same as the one which makes CL an interior operator. Therefore, some necessary and sufficient conditions for FH to be a closure operator are provided by the operator CL.

Proposition 7. *Let $\mathcal{C}$ be a covering of U. The following statements are equivalent.*
(1) FH is a closure operator.
(2) $\mathcal{T}_{CL}$ is a topology on U.
(3) C is a base of $\mathcal{T}_{CL}$.
(4) $Md(x) = \{N(x)\}$.

Proof. We just prove (4). By definition, if C is a unary covering of U, then for any x, $|Md(x)| = 1$. Since $N(x) = \bigcap Md(x)$, $N(x) \in Md(x)$. Conversely, if $Md(x) = \{N(x)\}$, then $N(x) \in \mathcal{C}$. Thus $|Md(x)| = 1$, that is, $\mathcal{C}$ is a unary covering.

Based on the above results, one may consider that the operator CL and the operator FH induce the same topology. In other words, they are dual. Indeed, that is not so. An example is provided to illustrate the problem.

Example 1. Let $\mathcal{C} = \{\{1,5\},\{1,2,5\},\{3,4\}\}$ be a covering of $U = \{1,2,3,4,5\}$. Then $Md(1) = Md(5) = \{\{1,5\}\}$, $Md(2) = \{\{1,2,5\}\}$ and $Md(3) = Md(4) = \{\{3,4\}\}$. Thus $\mathcal{C}$ is a unary covering. Let $X = \{2,3,4\}$. $CL(X) = CL(\{2,3,4\}) = \{3,4\}$ and the dual of $CL(X)$, that is, $-CL(-X)$ is $-CL(\{1,5\}) = \{2,3,4\}$. $FH(\{2,3,4\}) = CL(\{2,3,4\}) \bigcup (\bigcup \{\bigcup Md(x) : x \in X \backslash CL(X)\}) = \{3,4\} \bigcup \{\bigcup Md(2)\} = \{3,4\} \bigcup \{\bigcup \{\{1,2,5\}\}\} = \{3,4\} \bigcup \{1,2,5\} = U$, it follows that $FH(X) = U \neq -CL(-X) = \{3,4\}$. Thus CL and FH are not dual operators.

Now that the topologies induced by CL and FH are not the same. One may think that which topology is the one induced by the operator FH.

Proposition 8. *If $\mathcal{C}$ is a unary covering of U, then $FH = IH$ and $\mathcal{T}_{FH} = \mathcal{T}_{IH}$.*

Proof. According to (4) of Proposition 7 and the definition of FH and IH, $FH = IH$. Based on Proposition 3 and 4, IH is a closure operator and $\mathcal{T}_{IH} = \{-X : IH(X) = X\}$ is a topology. Therefore $\mathcal{T}_{FH} = \mathcal{T}_{IH}$.

3.3 Topological Characterization for the Operator SH

In this subsection, we characterize the second type of covering upper approximation operator from the viewpoint of topology. Now, we remind the following result.

Lemma 5. *[5] Let $\mathcal{C}$ be a covering of U. SH is a closure operator iff $\{I(x) : x \in U\}$ is a partition of U.*

The following proposition presents the other equivalence characterization for operator SH to be a closure operator from the viewpoint of covering itself. First, we can obtain the following result.

Proposition 9. *Let $\mathcal{C}$ be a covering of U. $\{I(x) : x \in U\}$ is a partition of U iff $\mathcal{C}$ satisfies (TRA) condition: $\forall x, y, z \in U$, $x, z \in K_1 \in \mathcal{C}$, $y, z \in K_2 \in \mathcal{C}$, there exists $K_3 \in \mathcal{C}$ such that $x, y \in K_3$.*

Proof. ("$\Leftarrow$"): $\forall x, y \in U$, $I(x) \bigcap I(y) = \emptyset$ or $I(x) \bigcap I(y) \neq \emptyset$. If $I(x) \bigcap I(y) \neq \emptyset$, then there exists $z \in I(x)$ and $z \in I(y)$. According to the definition of $I(x)$ and $I(y)$, there exist K_1, K_2 such that $x, z \in K_1$ and $y, z \in K_2$. According to the hypothesis, we know there exists $K_3 \in \mathcal{C}$ such that $x, y \in K_3$. Now we need to prove only $I(x) = I(y)$. $\forall u \in I(x)$, there exists $K \in \mathcal{C}$ such that $u, x \in K$. Since $x, y \in K_3$, there exists $K^{'} \in \mathcal{C}$ such that $u, y \in K^{'}$, that is, $u \in I(y)$, thus $I(x) \subseteq I(y)$. Similarly, we can prove $I(y) \subseteq I(x)$. Hence, $I(x) = I(y)$, that is, $\{I(x) : x \in U\}$ forms a partition of U. ("$\Rightarrow$"): $\forall x, y, z \in U$, $x, z \in K_1 \in \mathcal{C}$ and $y, z \in K_2 \in \mathcal{C}$, we can obtain $z \in I(x)$ and $z \in I(y)$. That implies $I(x) \bigcap I(y) \neq \emptyset$. Since $\{I(x) : x \in U\}$ forms a partition of U, $I(x) = I(y)$. Thus there exists $K_3 \in \mathcal{C}$ such that $x, y \in K_3$.

Corollary 3. *$\mathcal{C}$ is a covering satisfying the (TRA) condition iff SH is a closure operator.*

Therefore, we can present some fundamental properties of the topology induced by operator SH.

Proposition 10. *Let $\mathcal{C}$ be a covering of U. If $\mathcal{C}$ satisfies the (TRA) condition, then $\mathcal{T}_{SH} = \{-X : SH(X) = X\}$ is a clopen topology on U and $\{I(x) : x \in U\}$ is a base of $\mathcal{T}_{SH}$.*

Proof. According to Corollary 3, we know that if $\{I(x) : x \in U\}$ is a partition of U then SH coincides with Pawlaks upper approximation operator, which is well-known to be a closure operator of a clopen topology. Since $SH(I(x)) = I(x)$ for all $x \in U$, $\{I(x) : x \in U\} \subseteq \mathcal{T}_{SH}$. Since for all $X \in \mathcal{T}_{SH}$, $X = SH(X) = SL(X) = \bigcup_{x \in X} I(x)$. Therefore $\{I(x) : x \in U\}$ is a base of $\mathcal{T}_{SH}$.

4 Relationships among Above Three Topological Spaces

In this section, we study the relationships among above three operators and the relationship among the three topological spaces induced by them in a finite universe.

Proposition 11. *If $\mathcal{C}$ is a covering of U satisfying the (TRA) condition, then $CL(SL(X)) = SL(X)$ for all $X \subseteq U$.*

Proof. For all $X \subseteq U$, $CL(SL(X)) = \bigcup\{K \in \mathcal{C}, K \subseteq SL(X)\}$. For all $x \in CL(SL(X))$, there exists $K \in \mathcal{C}$ such that $x \in K \subseteq SL(X)$, $x \in SL(X)$. Conversely, for all $x \in SL(X)$, then $I(x) \subseteq X$. Thus there exists $K \in \mathcal{C}$ such that $x \in K$ and $I(x) \bigcap I(y) \neq \emptyset$ for all $y \in K$. From Proposition 9, $\{I(x) : x \in U\}$ is a partition. Then for all $y \in K$, $I(y) = I(x) \subseteq X$, that is, $K \subseteq SL(X)$. Hence, $SL(X) \subseteq CL(SL(X))$.

Combining Proposition 10 with Proposition 11, we obtain the following result.

Proposition 12. *If CL and SH are closure operators, then $\mathcal{T}_{SH} \subseteq \mathcal{T}_{CL}$.*

The above proposition points out that $\mathcal{T}_{CL}$ is finer than $\mathcal{T}_{SH}$, but the converse dose not hold. The following proposition presents a necessary and sufficient condition for these two topological spaces to be equal.

Proposition 13. *The family C is a partition of U iff $\mathcal{T}_{SH} = \mathcal{T}_{CL}$.*

Proof. ("$\Rightarrow$"): Since $\mathcal{C}$ is a partition, we know $CL = R_*$ and $SH = R^*$. Because R^* and R_* are dual operators, thus $\mathcal{T}_{SH} = \mathcal{T}_{CL}$. ("$\Leftarrow$"): If $\mathcal{C}$ is not a partition, then there exist $K_1, K_2 \in \mathcal{C}$ such that $K_1 \bigcap K_2 \neq \emptyset$. Thus $K_1 - K_2 \neq \emptyset$ or $K_2 - K_1 \neq \emptyset$. We might as well suppose $K_1 - K_2 \neq \emptyset$, then $K_2 \subset K_1 \bigcup K_2 \subseteq I(x)$ for all $x \in K_1 \bigcap K_2$. Hence, $I(x) \nsubseteq K_2$. According to the definition of SL, we know $x \notin SL(K_2)$, that is, $SL(K_2) \neq K_2$. Thus $K_2 \notin \mathcal{T}_{SL} = \mathcal{T}_{SH}$, i.e. $\mathcal{T}_{SH} \neq \mathcal{T}_{CL}$ which contradicts the assumption that $\mathcal{T}_{SH} = \mathcal{T}_{CL}$. Therefore, C is a partition.

Now, we study the relation between the topologies induced by FH and SH, respectively. Similarly, we present the relation between FH and SH firstly.

Proposition 14. *If $\mathcal{C}$ is a unary covering of U, then $FH(SH(X)) = SH(X)$ for all $X \subseteq U$.*

Proof. According to Proposition 3 and 8, $FH = IH$ and $SH(X) \subseteq IH(SH(X))$ for all $X \subseteq U$. Now we need to prove $IH(SH(X)) \subseteq SH(X)$. $\forall x \in IH(SH(X))$, there exists $y \in SH(X)$ such that $x \in N(y)$. Since $y \in SH(X)$, there exists $z \in X$ such that $y \in I(z)$, that is, there exists $K \in \mathcal{C}$ such that $y, z \in K$, thus $x \in N(y) \subseteq K$. So $x, y, z \in K$ which implies $x \in I(z)$, that is, $x \in SH(X)$. Therefore, $FH(SH(X)) = SH(X)$.

The following proposition establishes the relationship between the topologies induced by FH and SH, respectively.

Proposition 15. *If FH and SH are closure operators, then $\mathcal{T}_{SH} \subseteq \mathcal{T}_{FH}$.*

Proof. If FH is a closure operator, then $\mathcal{C}$ is unary. From Proposition 14, we have $X = SH(X) = FH(SH(X)) = FH(X)$ for all $X \in \mathcal{F}_{SH}$. This implies $X \in \mathcal{F}_{FH}$, thus $\mathcal{F}_{SH} \subseteq \mathcal{F}_{FH}$. For all $X \in \mathcal{T}_{SH}$, $-X \in \mathcal{F}_{FH}$ because $\mathcal{F}_{SH} \subseteq \mathcal{F}_{FH}$. Then $X \in \mathcal{T}_{FH}$. Therefore $\mathcal{T}_{SH} \subseteq \mathcal{T}_{FH}$.

The above proposition shows that $\mathcal{T}_{FH}$ is finer than $\mathcal{T}_{SH}$, but the converse dose not hold, as one can verify the following example.

Example 2. Assume the same covering as in Example 1. Then $\mathcal{C}$ is a unary covering. We can also obtain $I(1) = I(2) = I(5) = \{1, 2, 5\}$ and $I(3) = I(4) = \{3, 4\}$, then $\{I(x) : x \in U\}$ is a partition. Therefore SH and FH induced by $\mathcal{C}$ are closure operators and $\mathcal{F}_{SH} = \{\emptyset, \{1, 2, 5\}, \{3, 4\}, U\}$ and $\mathcal{F}_{FH} = \{\emptyset, \{1, 5\}, \{1, 2, 5\}, \{3, 4\}, U\}$. Since $\mathcal{T}_{SH}$ is a clopen topology, we know $\mathcal{T}_{SH} = \mathcal{F}_{SH}$ and $\mathcal{T}_{FH} = \{\emptyset, \{2, 3, 4\}, \{1, 2, 5\}, \{3, 4\}\}$. It is obvious $\mathcal{T}_{SH} \subseteq \mathcal{T}_{FH}$ but $\mathcal{T}_{FH} \nsubseteq \mathcal{T}_{SH}$.

When $\mathcal{C}$ is a partition, these three topologies are the same.

Proposition 16. *$\mathcal{C}$ is a partition iff $\mathcal{T}_{SH} = \mathcal{T}_{CL} = \mathcal{T}_{FH}$.*

5 Conclusion

This paper has presented the topological characterizations for three covering approximation operators and established the relationships among the topological spaces corresponding to them. However, there are still many problems to be solved, for example, we can apply the topological spaces to the issues of covering reductions.

References

1. Bianucci, D., Cattaneo, G., Ciucci, D.: Entropies and co-entropies of coverings with application to incomplete information systems. Fundamenta Informaticae 75, 77–105 (2007)
2. Bonikowski, Z., Bryniarski, E., Wybraniec-Skardowska, U.: Extensions and intentions in the rough set theory. Information Sciences 107, 149–167 (1998)
3. Chen, D., Wang, C., Hu, Q.: A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets. Information Sciences 177, 3500–3518 (2007)
4. Engelking, R.: General Topology. Polish Scientific Publishers, Warszawa (1977)
5. Ge, X., Bai, X., Yun, Z.: Topological characterizations of covering for special covering-based upper approximation operators. Information Sciences 204, 70–81 (2012)
6. Lin, T.Y., Liu, G., Chakraborty, M.K., Slezak, D.: From Topology to Anti-reflexive Topology. In: Proc. of FUZZ IEEE 2013, Hyderabad, India, July 7-10 (2013)
7. Kondo, M.: On the structure of generalized rough sets. Information Sciences 176, 589–600 (2005)
8. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
9. Qin, K., Gao, Y., Pei, Z.: On covering rough sets. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 34–41. Springer, Heidelberg (2007)
10. Ślęzak, D., Wasilewski, P.: Granular Sets - Foundations and Case Study of Tolerance Spaces. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS (LNAI), vol. 4482, pp. 435–442. Springer, Heidelberg (2007)
11. Yao, Y., Yao, B.: Covering based rough set approximations. Information Sciences 200, 91–107 (2012)
12. Zhu, W., Wang, F.: Relationships among three types of covering rough sets. In: Granular Computing, pp. 43–48 (2006)
13. Zhu, W.: Topological approaches to covering rough sets. Information Sciences 177, 1499–1508 (2007)
14. Zhu, W., Wang, F.: Reduction and axiomization of covering generalized rough sets. Information Sciences 152, 217–230 (2003)
15. Zhu, W., Wang, F.: On three types of covering rough sets. IEEE Transactions on Knowledge and Data Engineering 19, 1131–1144 (2007)
16. Zhu, W.: Relationship among basic concepts in covering-based rough sets. Information Sciences 179, 2478–2486 (2009)

Rough Set Granularity: Scott Systems Approach

Marcin Wolski[1] and Anna Gomolińska[2]

[1] Department of Logic and Philosophy of Science,
Maria Curie-Skłodowska University, Lublin, Poland
marcin.wolski@umcs.lublin.pl
[2] Institute of Computer Science,
University of Białystok, Poland
anna.gom@math.uwb.edu.pl

Abstract. The paper addresses the problem of concept formation (in other words, knowledge granulation) in the framework of rough set theory. The proper treatment of this problem requires taking into account both the dynamics of the universe and different scales at which concepts may be formed. These both aspects have been already separately discussed in rough set theory, with special emphasis put upon the Granular Computing paradigm as a suitable framework to deal with different scales of description. Following the example of the game *Life*, construed by Hawking as a simple means of explaining the process of concept formation in science, we shall describe a corresponding dynamics in Pawlak information systems.

Keywords: concept, rough set, granular computing, Scott system.

1 Introduction

The main methodological assumption of rough set theory [9] is that knowledge about a universe of objects is given it terms of concepts, that is, definable, with respect to gathered pieces of information, subsets of the universe – the most explicit expression of this assumption may be found in [10]. Undefinable sets are then approximated by a pair of definable sets (concepts). As is well-known, this (original) framework has two important features: it is static, that is, the universe does not change; it is flat, that is, there is a single scale from which the universe is described. However, in the real science (a) the universe is dynamic and (b) there are different scales used in descriptions of the universe; e.g., atomic scale, molecular scale, there are also different macroscopic scales, e.g., galaxies. Therefore, the full treatment of the problem of concept formation (knowledge granulation) requires addressing both (a) and (b).

Regarding (a), the problem of the universe extension by new objects has been already discussed in the framework of rough sets in, e.g., [13]; more general discussion of dynamics in Pawlak information systems may be found in, e.g., [2]. Regarding (b), this problem to a large extent is addressed by the Granular Computing (GrC) methodology, e.g. [11], which has been extensively developed

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 285–292, 2013.

since its introduction by Zadeh [15]. The GrC methodology assumes computing with *words* rather than *letters* (or *symbols*). In other words, sometimes we may compute with particles, but other time we may need to compute with molecules or even galaxies. Thus, we may compute at different scales. The GrC paradigm may be considered as a systematic study of this phenomenon in the context of computer science.

Hawking in [6] uses the game *Life* [4] to explain to the reader how new concepts emerge on different scales while observing a simple and dynamic configuration of cells. Following the very same example we would like to describe the process of concept formation (knowledge granulation) – addressing both (a) and (b) – in the framework of rough set theory. To this end, we start with Pawlak information systems, allow dynamics as considered in [2] (so objects may both enter and exit the system), and then provide a formal description of a "zoom-out" scale. Of course, these scales could be described in terms of different theories. In the paper we decided to choose Scott information systems [12]. The motivation is purely theoretical and may be shortly described as follows. Both rough sets and generalised rough sets [1,7,8] can be represented in the increasing form or disjoint form. Each form requires its own partial order and the corresponding infimum and supremum operations. The problem of which role may be played in rough set theory by both orders taken together was suggested by Marek and Truszczyński in [8]. An answer for the disjoint representation of generalised roughs sets was given in, e.g., [14]. Here we would like to complete this research and to show how both orders can be applied to the increasing representation. In consequence, we shall obtain Scott information systems regarded as coarser scales which are compatible with Pawlak information systems.

2 Rough Sets and Game of *Life*

2.1 Conways's *Life* Game

Game of life, also known as *Life*, was invented by Conway in 1970 [4] and immediately became a topic of interest for biologists, physicists, mathematicians and computer scientists. *Life* is played on an array of squares, where each square can be "alive" (black) or "dead" (white). The rules (representing interactions with an environment) according to which squares change their states are very simple:

1. A live square with two or three live neighbours stay alive.
2. A dead square with exactly three live neighbours change its state and becomes a live cell.
3. In all other cases a cell dies or, if it is dead, remains dead.

Although *Life* is a very simple game, it can generate conglomerations of squares which are equivalent to universal Turing machines. Thus, the final "product" of this game can be really complex. As Hawking put it [6], *Life* demonstrates that although the "physics" (i.e. basic granules) is very simple, the "chemistry" (i.e. higher order granules) may be very rich and complicated. At the micro scale there are only dead or live cells, but under macro scales there are different concepts of pulsars, gliders or even glider guns, where the latter ones give birth to new gliders.

2.2 Rough Sets

In the present section we briefly recall basic concepts from rough set theory which are relevant to our study.

Definition 1 (Information System). *A quadruple* $\mathcal{I} = \langle U, Att, Val, f \rangle$ *is called an* information system*, where:*

- U *is a non–empty finite set of objects;*
- Att *is a non–empty finite set of attributes;*
- $Val = \bigcup_{A \in Att} Val_A$*, where* Val_A *is the (non-empty) value–domain of the attribute* A*;*
- $f : U \times Att \mapsto Val$ *is a partial information function, such that for all* $A \in Att$ *and* $a \in U$*, when* $f(a, A)$ *is defined, then* $f(a, A) \in Val_A$*.*

If f *is a total function, i.e.* $f(a, A)$ *is defined for all* $a \in U$ *and* $A \in Att$*, then the information system* $\mathcal{I}$ *is called* complete*; otherwise, it is called* incomplete*.*

When f is generalised to a function from $U \times Att$ to $\mathcal{P}(Val)$, where $\mathcal{P}(Val)$ is the powerset of Val, then the information system is *nondeterministic*. In what follows we focus our attention on complete and deterministic systems.

If we distinguish in an information system two disjoint classes of attributes Att_C and Att_D, called condition and decision attributes, respectively, then the system will be called a *decision table*.

An information system $\mathcal{I}$ gives rise to an equivalence relation E, called an *indiscernibility relation*, defined as:

$$E = \{(a, b) : \forall A \in C \subseteq Att.\ \forall X \in Val\ (f(a, A) = X \Leftrightarrow f(b, A) = X)\}.$$

Customarily, E is often written as $IND(Att)$, the partition induced by the relation $IND(Att)$ is denoted by $U/IND(Att)$, and $[a]_{IND(Att)}$ denotes the equivalence class of $IND(Att)$ defined by $a \in U$. A simple generalisation of $(U, IND(Att))$ is given by the concept of an approximation space:

Definition 2 (Approximation Space). *A pair* (U, E)*, where* U *is a non-empty set and* E *is an equivalence relation on* U*, is called an* approximation space*. A subset* $X \subseteq U$ *is called* definable*, if* $X = \bigcup \mathcal{Y}$ *for some* $\mathcal{Y} \subseteq U/E$*, where* U/E *is the family of equivalence classes of* E *(the quotient set of* E*).*

Definition 3 (Approximation Operators). *Let* (U, E) *be an approximation space. For every concept* $X \subseteq U$*, its* E*-lower and* E*-upper* approximations *are defined as follows, respectively:*

$$\underline{X} = \{a \in U : [a]_E \subseteq X\}, \quad \overline{X} = \{a \in U : [a]_E \cap X \neq \emptyset\}.$$

By the usual abuse of language and notation, the operator $\underline{\ \ } : \mathcal{P}(U) \to \mathcal{P}(U)$ sending X to $\underline{X}$ will be called the *lower approximation operator*, whereas the operator $\overline{\ \ } : \mathcal{P}(U) \to \mathcal{P}(U)$ sending X to $\overline{X}$ will be called the *upper approximation operator*.

Definition 4 (Increasing Representation of Rough Sets). *For an approximation space* (U, E) *and* $X \subseteq U$*, a pair* $(\underline{X}, \overline{X})$ *is called an* increasing representation of X.

The set $U \setminus \overline{X}$ is often called an *exterior of* X and denoted by $Ext(X)$, whereas $Bnd(X) = \overline{X} \setminus \underline{X}$ is the boundary region of X.

It is a standard methodological assumption that a given information system represents merely a sample or fragment of a bigger universe. In the course of extending the universe of objects, we can change the underlying cells by addition of a new object or we can regard this object as a rule how to change the states of cells. Following GrC methodology [11,15] and *Life* [4] we would like to regard the expansion of the universe as a method of changing the states of cells. So, as Zadeh would put it [15], the idea is to compute with a *word*, not with its letters. Following *Life* [4] we would like to assign to each cell a state. Suppose now that we are given a decision table and we would like to approximate the extension X of $D \in Att_D$ (where D is Boolean, that is, a property). So we compute $\underline{X}$, $Bnd(X)$ and $Ext(X)$. In this way we have assigned cells their initial states: the cells belonging to the lower approximation $\underline{X}$ are *alive* or *black*; cells from the boundary region $Bnd(X)$ or *grey*; cells form the exterior of X are *white*. The set of grey and black cells associated with a single attribute D would be regarded a single conglomeration (similarly like, e.g., a glider in *Life*). What will happen when a new object a is added?

2.3 Rules of Game

The rules of the game are straightforward and very simple. They have a descriptive character rather than normative. Firstly, let us fix X as the extension of $D \in Att_D$ in the initial state of the game. Secondly, the new object a may satisfy D or may not. Suppose the former case; then a may be similar to some $b \in X$ or may not. If it similar to b whose equivalence class $[b]_E$ is black, then $[b]_E$ remains black. If $[b]_E$ is grey then it remains grey. If a is not similar to any object in X, but it similar to some b in $Ext(X)$ then $[b]_E$ is changed from white to grey. If a is not similar to any object except itself then nothing is changed. Now suppose that a does not satisfy D, then, as previously, it may be similar to some objects from X. If a is similar to b whose equivalence class $[b]_E$ is black, then $[b]_E$ is changed to grey. If $[b]_E$ is grey, then it remains grey. As earlier, if a is not similar to any object except itself then nothing is changed. Of course, white cells remain white.

Let us once again emphasise that the underlying partition of the universe is unchanged; what changes are states of cells (equivalence classes). Thus, we can regard a partition like an array of cells which can be in one of the three states: "alive" (black), "possibly alive" (grey) and "dead" (white). A sequence of new objects defines the rules of according to which cells change their states.

According to the rules of our game the upper approximation $\overline{X}$ increases, the lower approximation $\underline{X}$ decreases and the boundary region $Bnd(X)$ increases. No cell dies. In order to make the game more interesting and similar to *Life*, let us allow cells to die. It can be done in a number of ways. In this paper we follow suggestions given by Ciucci in [2], who assumes that an object may exit the information system. A natural example is given by an object which was confirmed to satisfy some D and later it turned out that this confirmation was (methodologically) invalid. So, apart from the list of new objects, let be given a list of "old" objects which are supposed to leave the system – these objects will be marked by "exit". If all objects which satisfy D and belong to a given grey or black cell are marked by "exit", then the cell becomes dead (white). If all objects from a grey cell which do not satisfy D are marked "exit" then the cell becomes black. In this way during the game the approximations of a given attribute D are really dynamic.

2.4 Scott Systems and Higher-Order Granules/Concepts

Let be given an information system $\mathcal{I} = \langle U, Att, Val, f\rangle$, $D = \{D_1, D_2, \ldots, D_m\}$ a set of decision Boolean attributes, and a sequence $(\stackrel{+}{-}a_n)_{n\geq 1}$ of objects which we will add or erase from the system: new (added) objects will be denoted by $+a$, whereas objects which are supposed to leave the system (i.e. "exit") by $-a$. The extension of each D_i before the game is denoted by $|D_i|$. Thus the initial state of the game is given by $\mathcal{U}_0 = \{(\underline{X^i}, \overline{X^i}) : X^i = |D_i|\}$. The state $\mathcal{U}_j = \{(\underline{X^i}_j, \overline{X^i}_j) : X^i = |D_i|\}$ is defined in the obvious way on the basis of: (a) the previous state of the game $\mathcal{U}_{j-1} = \{(\underline{X^i}_{j-1}, \overline{X^i}_{j-1})\}$, (b) the object $\stackrel{+}{-}a_j$ to be added or removed, and (c) the rules of our game. It is worth emphasising that for any n it holds that $\underline{X^i}_n \subseteq \overline{X^i}_n$. In other words, the initial state of the game consists of rough sets and during the game we obtain the generalised rough sets (X, Y), where both sets $X \subseteq Y$ are definable (exact) [1,7,8].

As written above, each D_i can be construed as a conglomeration of cells. Now we would like to consider the task of grouping these conglomerations into higher order granules (i.e. granules which would represent concepts formed at coarser scales).

From purely mathematical point of view, each conglomeration is represented as a pair of sets (X_1, X_2). Of course, sets can be partially ordered by the standard set inclusion $\subseteq$ and generate a lattice. Let us recall now a pre-bilattice product from bilattice theory [3,5].

Definition 5 (Pre-Bilattice Product). *Let L_1 and L_2 be two complete lattices. Define a pre-bilattice product $L_1 \circ L_2$ by:*

- *the carrier set is $L_1 \times L_2$;*
- *$(a_1, a_2) \leq_k (b_1, b_2)$ iff $a_1 \leq_{L_1} b_1$ and $a_2 \leq_{L_2} b_2$;*
- *$(a_1, a_2) \leq_t (b_1, b_2)$ iff $a_1 \leq_{L_1} b_1$ and $b_2 \leq_{L_2} a_2$;*

for all a_1, $b_1 \in L_1$ and a_2, $b_2 \in L_2$.

The case where these two complete lattices are actually families of sets ordered by $\subseteq$ is the easiest and most relevant to our game. In what follows, the meet and join operators corresponding to $\leq_k$ will be denoted by $\wedge_k$ and $\vee_k$, whereas $\wedge_t$ and $\vee_t$ will denote these operators for $\leq_t$ [5,3]:

$$(X_1, X_2) \wedge_k (Y_1, Y_2) = (X_1 \cap Y_1, X_2 \cap Y_2), \;\; (X_1, X_2) \vee_k (Y_1, Y_2) = (X_1 \cup Y_1, X_2 \cup Y_2),$$

$$(X_1, X_2) \wedge_t (Y_1, Y_2) = (X_1 \cap Y_1, X_2 \cup Y_2), \;\; (X_1, X_2) \vee_t (Y_1, Y_2) = (X_1 \cup Y_1, X_2 \cap Y_2).$$

Thus, starting from $\mathcal{I} = \langle U, Att, Val, f \rangle$, $D = \{D_1, D_2, \ldots, D_m\}$, and the list $(\overset{+}{-} a_n)$ of object to be added or removed, for every stage n of the game we can build two lattices:

$$\mathcal{U}_n^k = \langle U_n^k, \wedge_k, \vee_k \rangle, \;\; \mathcal{U}_n^t = \langle U_n^t, \wedge_t, \vee_t \rangle,$$

both having $\mathcal{U}_n = \{(\underline{X^i}_n, \overline{X^i}_n) : X^i = |D_i|\}$ as the generator.

In the paper we focus our attention upon a lattice $\mathcal{U}_n^k$; this lattice would represent possible interactions between D_i regarded as conglomerations of cells. The lattice $\mathcal{U}_n^t$ is less interesting from the perspective of rough set theory since its operations may produce objects which are not generalised rough sets; however, the partial order $\leq_t$ underlying $\mathcal{U}_n^t$ will be of great importance.

The appeal of *Life* comes mainly from the fact that although the basic rules are very simple, new complex concepts emerge quite naturally. For example, in the rules of game there is no concept such as "glider", "move" or "collide", but on a macro scale we can deduce such laws as: gliders move diagonally. Similarly, we would like to introduce into this game some more complex concepts saying that some number conglomerations of cells form a new structure (of higher order than its components).

Let us recall that objects are taken from $\mathcal{U}_n^k$ and our aim is to find a recipe for gathering them into collections. We shall define this recipe by means of $\vee_t$ from $\mathcal{U}_n^t$. A collection $\mathcal{X}$ of objects $(X_1, X_2) \in \mathcal{U}_n^k$ will be called *compatible* only if $\bigvee_t \mathcal{X}$ is a generalised rough set [1,7,8]. Now we need a mathematical structure within which compatible sets arise naturally. Interestingly, compatibility may be regarded as *consistency* in Scott information systems:

Definition 6 (Scott Information System). *A Scott information system is an ordered quadruple $\langle T, Con, a_0, \vdash \rangle$, where T is a set of tokens (information atoms), a_0 the least informative atom, Con a family of finite subsets of T, and $\vdash \subseteq Con \times T$, which satisfies the following conditions:*

- *if $a \in X \in Con$, then $X \vdash a$;*
- *if $X \vdash Y$ and $Y \vdash a$, then $X \vdash a$;*
- *if $X \vdash a$, then $X \cup \{a\} \in Con$;*
- *for all $a \in T$ it holds that $\{a\} \in Con$;*
- *for all $X \in Con$, $X \vdash a_0$;*
- *if $X \in Con$ and $Y \subseteq X$, then $Y \in Con$;*

where $X \vdash Y$ means $X \vdash a$ for all $a \in Y$.

As a set of tokens we take the elements of $\mathcal{U}_n^k$ and call a set $\mathcal{X} \subseteq U_n^k$ consistent iff it is empty or form a compatible set. Now, our aim now is to define $\vdash$ in terms of $\leq_k$ and $\leq_t$. The main constraint is, of course, the condition that if $a \in X \in Con$, then $X \vdash a$. It seems that we may define four entailment relations:

$$\mathcal{X}_n \vdash_k^{\wedge} (X_1, X_2) \text{ iff } \bigwedge\nolimits_k \mathcal{X}_n \leq_k (X_1, X_2);$$

$$\mathcal{X}_n \vdash_k^{\vee} (X_1, X_2) \text{ iff } (X_1, X_2) \leq_k \bigvee\nolimits_k \mathcal{X}_n;$$

$$\mathcal{X}_n \vdash_t^{\wedge} (X_1, X_2) \text{ iff } \bigwedge\nolimits_t \mathcal{X}_n \leq_t (X_1, X_2);$$

$$\mathcal{X}_n \vdash_t^{\vee} (X_1, X_2) \text{ iff } (X_1, X_2) \leq_t \bigvee\nolimits_t \mathcal{X}_n.$$

However, $\vdash_t^{\wedge}$ has not the least informative atom; the only candidate is $(U, \emptyset)$, which is not a generalised rough set.

Proposition 1. *The following quadruples form Scott information systems:*

- $\langle U_n^k, Con, (\emptyset, \emptyset), \vdash_k^{\vee} \rangle$;
- $\langle U_n^k, Con, (U, U), \vdash_k^{\wedge} \rangle$;
- $\langle U_n^k, Con, (\emptyset, U), \vdash_t^{\vee} \rangle$.

However, when we consider further notions from Scott information systems, only one on them will remain intuitive.

Definition 7 (Ideal and Total Elements). *The* ideal elements *of an information system are subsets X of T such that:*

- *X is consistent: every finite subset of X belongs to Con;*
- *closed under entailment: if $Y \subseteq X$ and $Y \vdash a$, then $a \in X$.*

An ideal element X is called total *iff it is maximal with respect to the inclusion.*

From the perspective of ideal and total elements, the only intuitive Scott information system among the three described above is $\langle \mathcal{U}_n^k, Con, (\emptyset, U), \vdash_t^{\vee} \rangle$. Total elements would represent (Boolean) attributes D which are definable in terms of Att_C (that is, Att_C includes all pieces of knowledge needed to define D). Indeed, starting from $\mathcal{X} = \{(X, X)\}$ and closing it under the entailment $\vdash_t^{\vee}$ we shall obtain an ideal element I. This element is also total. When we start from $\mathcal{X} = \{(X, Y)\}$ and close it under the entailment we shall obtain an ideal element J which is not total since $J \subseteq I$. The other cases $\vdash_k^{\vee}$ and $\vdash_k^{\wedge}$ are much less intuitive; the natural candidates which would generate total elements are (U, U) and $(\emptyset, \emptyset)$, respectively. But, both elements have been considered as providing no information.

In consequence, starting from an information system $\mathcal{I} = \langle U, Att, Val, f \rangle$, a set $D = \{D_1, D_2, \ldots, D_m\}$ of Boolean attributes, and a list of object $(^{+}_{-} a_j)$ (to be added or removed) we may obtain – for every phase n of this game

– an intuitive Scott information system $\langle \mathcal{U}_n^k, Con, (\emptyset, U), \vdash_t^\vee \rangle$ which provides a natural granulation of objects from $\mathcal{U}_n^k$ after considering the object $\overset{+}{-}a_n$. The list $(\overset{+}{-}a_j)$ may be regarded as a process of interactions of an information system with an environment and the sequence of Scott systems $\langle \mathcal{U}_n^k, Con, (\emptyset, U), \vdash_t^\vee \rangle$ would represent an evolution of this information system. Then two processes represented by $\overset{+}{-}a_n$ and $\overset{+}{-}b_n$, respectively, may be defined equivalent provided that the respective Scott systems are equivalent.

Summing up, in the paper we have described dynamics in Pawlak information systems [9] resulting from expansion/contraction of the universe of objects. We have followed the GrC methodology [11,15], and taking the inspirations from Conway's game *Life* [4], decided to change merely the states of equivalence classes in the course of expansion/contraction process. In result, rough sets have been converted into generalised rough sets [1,7,8]. In order to define a "zoom-out" scale for concept formation, these new sets have been grouped into families by means of bilattice orderings [3,5]. Finally, we have described these families in terms of Scott information systems and consistent sets [12].

References

1. Banerjee, M., Chakraborty, M.K.: Rough sets through algebraic logic. Fundamenta Informaticae 28(3-4), 211–221 (1996)
2. Ciucci, D.: Temporal dynamics in information tables. Fundamenta Informaticae 115(1), 57–74 (2012)
3. Fitting, M.C.: Bilattices in logic programming. In: Epstein, G. (ed.) The Twentieth International Symposium on Multiple-Valued Logic, pp. 238–246 (1990)
4. Gardner, M.: The fantastic combinations of John Conway's new solitaire game "life". Scientific American 223, 120–123 (1970)
5. Ginsberg, M.L.: Multivalued logics: a uniform approach to reasoning in artificial intelligence. Computational Intelligence 4, 256–316 (1988)
6. Hawking, S., Mlodinow, L.: The Grand Design. Bantam Press (2010)
7. Iwiński, T.B.: Algebraic approach to rough sets. Bull. Polish Acad. Sc. (Math.) 35(9-10), 673–683 (1987)
8. Marek, V.W., Truszczyński, M.: Contributions to the theory of rough sets. Fundamenta Informaticae 39(4), 389–409 (1999)
9. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers (1991)
10. Pawlak, Z.: Wiedza z perspektywy zbiorów przybliżonych. Institute of Computer Science Report 23 (1992)
11. Pedrycz, W. (ed.): Granular Computing: An Emerging Paradigm. STUDFUZZ. Physica-Verlag (2001)
12. Scott, D.: Domains for denotational semantics. In: Nielsen, M., Schmidt, E.M. (eds.) ICALP 1982. LNCS, vol. 140, pp. 577–613. Springer, Heidelberg (1982)
13. Suraj, Z., Pancerz, K.: Some remarks on computing consistent extensions of dynamic information systems. In: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA 2005), pp. 420–425 (2005)
14. Wolski, M.: Complete Orders, Categories and Lattices of Approximations. Fundamenta Informaticae 72, 421–435 (2006)
15. Zadeh, L.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems 90(2), 111–127 (1997)

Incremental Possibilistic K-Modes

Asma Ammar[1], Zied Elouedi[1], and Pawan Lingras[2]

[1] LARODEC, Institut Supérieur de Gestion de Tunis, Université de Tunis
41 Avenue de la Liberté, 2000 Le Bardo, Tunisie
asma.ammar@voila.fr, zied.elouedi@gmx.fr
[2] Department of Mathematics and Computing Science, Saint Marys University
Halifax, Nova Scotia, B3H 3C3, Canada
pawan@cs.smu.ca

Abstract. This paper proposes an incremental version of a soft clustering approach under uncertainty. The possibility theory and the k-modes algorithm are combined together in an incremental way to deal with two aspects of uncertainty. On one hand, the possibility theory deals with uncertain values of attributes of instances using possibility distributions and handles the belonging of objects to different clusters based on possibilistic membership degrees. On the other hand, the incremental aspect is studied in this new method by adding clusters without re-clustering initial instances. Experimental results clearly demonstrate the advantages of our proposal in a variety of databases using different evaluation criteria.

Keywords: Incremental clustering, possibility theory, k-modes method, possibilistic membership, possibility degree.

1 Introduction

Incremental clustering has been widely applied in various fields [5], [6], [7], [8]. It can be obtained by adding instances, attributes, and clusters over time. The main advantage of incremental clustering methods is the possibility to minimize the use of the main memory, to save time and to adapt the dynamic changes in real-word (e.g. detection of new groups of customers when selling goods).

As uncertainty appears frequently in real-world problems (e.g. when measuring blood pressure, temperature or humidity levels), many uncertainty theories have been proposed in order to deal with uncertain framework. Possibility theory is a well-known uncertainty theory for handling uncertainty and leading to a better decision making. This theory has been successfully combined with hard and soft clustering methods such as [1], [2], [3].

Combining uncertain soft clustering methods with incremental learning is of great interest because they complement each other. Uncertain soft clustering approaches describe with more precision the similarities between instances of databases and clusters. They take into consideration the degree of uncertainty of knowledge. As a result, we obtain objects that belong to different clusters based on membership degrees. However, incremental clustering offers multiple

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 293–303, 2013.

advantages by improving the final partitions without re-clustering initial objects, and hence saving time, through the use of results.

In this paper, we propose an incremental possibilistic k-modes method (IPKM) which combines possibility theory and incremental learning with k-modes algorithm. First, the proposed approach deals with uncertain values of attributes through possibilistic degrees then, it indicates the possibilistic membership of each object to different clusters. Finally, we study an incremental aspect of this soft clustering approach by adding new clusters over time and updating the partitions without re-clustering initial instances.

The rest of the paper is structured as follows: Section 2 and section 3 provide an overview of the possibility theory and the k-modes method. Section 4 presents details of our proposal i.e. the incremental possibilistic k-modes. Section 5 analyses experimental results.

2 Possibility Theory

Possibility theory is a well-known uncertainty theory proposed by Zadeh in [10] then, improved through various works (e.g. Dubois and Prade [11]).

2.1 Possibility Distribution

Given the universe of discourse $\Omega = \{\omega_1, \omega_2, ..., \omega_n\}$, the possibility distribution function denoted by π can be defined in either numerical or qualitative setting. π associates to each element (or state) ω_i from Ω [10] a possibility degree taking values from the scale L. In our case, we deal with the quantitative setting of the possibility theory where we have numerical possibility degrees and $L = [0, 1]$. Based on π, some concepts can be defined such that:

- The normalization described by $max_i \{\pi(\omega_i)\} = 1$.
- The extreme cases of knowledge namely: the complete knowledge presented by $\exists \omega_0$, $\pi(\omega_0) = 1$ and $\pi(\omega) = 0$ otherwise and the total ignorance defined by $\forall \omega \in \Omega, \pi(\omega) = 1$.

2.2 Possibilistic Similarity Measure

The information affinity [9] (detailed in Equation (1)) is a well-known possibilistic similarity measure. It is applied on two normalized possibility distributions π_1, π_2 in order to measure their similarity.

$$InfoAff(\pi_1, \pi_2) = 1 - 0.5\,[D(\pi_1, \pi_2) + Inc(\pi_1, \pi_2)]. \tag{1}$$

with $\mathrm{D}(\pi_1, \pi_2) = \frac{1}{n}\sum_{i=1}^{n} |\pi_1(\varpi_i) - \pi_2(\varpi_i)|$, $Inc(\pi_1, \pi_2) = 1 - \max(\pi_1(\varpi)\, Conj\, \pi_2(\varpi))$ and $\forall \omega \in \Omega, \Pi_{Conj}(\omega) = \min(\Pi_1(\omega), \Pi_2(\omega))$.

Note that if $IA(\pi_1, \pi_2)$ is smaller than $IA(\pi_1, \pi_3)$, π_1 is considered more similar with π_2 than π_3.

3 The K-Modes Method and Its Extension

3.1 The SKM

The standard k-modes method denoted in this paper by SKM [14] [15] is a clustering method dealing with large categorical data sets. It is a modified version of the k-means algorithm [12] that uses a simple matching dissimilarity measure and a frequency-based function in order to cluster objects into k clusters.

Given two objects X_1=(x_{11}, x_{12}, ..., x_{1m}) and X_2=(x_{21}, x_{22}, ..., x_{2m}) with m categorical attributes. The simple matching method ($d \in [0, 1]$) is defined in Equation (2):

$$d\left(X_1, X_2\right) = \sum_{t=1}^{m} \delta\left(x_{1t}, x_{2t}\right) \quad . \tag{2}$$

In fact, $\delta\left(x_{1t}, x_{2t}\right) = 0$ when $x_{1t} = x_{2t}$ and it is equal to 1 otherwise. As a consequence, d takes the value of 0 when all the values of attributes relative to X_1 and X_2 are similar and takes the value of m otherwise. Generally, given a set of n objects $S = \{X_1, X_2, ..., X_n\}$ with its k-modes $Q = \{Q_1, Q_2, ..., Q_k\}$ and k clusters $C = \{C_1, C_2, ..., C_k\}$, it is possible to aggregate it into $k \leq n$ clusters. The minimization of the clustering cost function is given by:

$$\min \ D(W, Q) = \sum_{j=1}^{k} \sum_{i=1}^{n} \omega_{i,j} d(X_i, Q_j) \tag{3}$$

where W is an $n \times k$ partition matrix and $\omega_{i,j} \in \{0, 1\}$ is the membership degree of X_i in C_j.

Although it is successful when clustering large categorical databases, the SKM has a several issues while clustering objects in an uncertain framework. Data sets can contain uncertainty at different levels (e.g. in attributes values of instances and/or in the belonging of objects to different clusters). To overcome this drawback, many researchers have reported and analyzed this issue and introduced modifications to the SKM parameters such as [1], [2], [3], and [4]. The next subsection, presents the KM-PF [4] which uses the possibility theory to handle uncertainty when clustering instances to k clusters.

3.2 The KM-PF

The k-modes under possibilistic framework denoted by KM-PF [4] is an uncertain clustering approach based on the k-modes and the possibility theory. It is an improved version of two possibilistic approaches proposed in [1] and [2].

On the one hand, the KM-PF uses the possibility theory to deal with uncertainty in the attributes' values of instances and assigns possibilistic degrees of membership describing the similarity between the instances and clusters. On the other hand, the k-modes algorithm is used to cluster each uncertain object to several clusters.

The KM-PF uses an uncertain training set created artificially by defining a possibility distribution for each attribute. The possibilistic distribution expresses the extent to which the attribute value is true. Moreover, the KM-PF adapts a possibilistic measure that computes the similarity between each uncertain object and modes. It also defines possibilistic membership degrees for updating the modes.

4 Incremental Possibilistic K-Modes

4.1 Parameters

Our proposal uses the following parameters:

1. An uncertain training set: We artificially create an uncertain training set. It contains both certain and uncertain values of attributes. Each value of the attributes relative to different instances are replaced by a possibility degree with respect to the real attributes' values. Thus all objects and modes have possibility values. These values describe the degree of uncertainty. By presenting the values of modes through possibility degrees, the final results will depend less on the initial partition.
2. The possibilistic similarity measure: Each attribute is presented through a possibility distribution relative to a particular object. As a result, to compute the possibilistic similarity between objects and modes we have to sum the information affinity [9] applied on the possibility distributions of the modes and objects. We apply the $IA(X_1, X_2)$ which is given by:

$$IA(X_1, X_2) = \frac{\sum_{j=1}^{m} InfoAff(\pi_{1j}, \pi_{2j})}{m}. \quad (4)$$

   where m is the total number of attributes.
3. The possibilistic membership degrees: They present values from $[0, 1]$ describing the similarities between each instance of the training set and all clusters.We use ω_{ij} to denote the degree of belonging of the object i to the cluster j. Note that there are two extreme cases, the first one is when $\omega_{ij} = 1$. In this case, the object i is considered as very similar to the mode of the cluster j. The second one is obtained for $\omega_{ij} = 0$ where the object i does not belong to the cluster j because there is no similarity between i and the mode of j. The possibilistic membership degree is obtained by computing the possibilistic similarity measure (Equation (4)).
4. The update of clusters' modes: We randomly choose k initial modes taken from the objects of the training set. The update of these modes depends on the degrees of possibility assigned to each object and the membership values ω_{ij}. The following steps describes how to update the k modes:

- For each cluster j, we compute the number of instances that have the highest value of ω_{ij} such that: $NO_j = count_j(\max_i \omega_{ij})$.

- We define a new parameter W which expresses the weight assigned for the initial (k') and added (n) clusters. It is given by:

$$W_j = \begin{cases} \frac{NO_j}{total\ number\ of\ objects} & \text{if } NO_j \neq 0, \\ \frac{1}{total\ number\ of\ objects\ +1} & \text{otherwise.} \end{cases} \quad (5)$$

- We compute the values of the new mode M'_j as follows:

$$\forall j \in k, M'_j = W_j \times Mode_j. \quad (6)$$

  Using this formula, the new mode will depend on the number of the most similar objects to it.

5. The addition of n new clusters after getting the final partition from k' initial clusters: It consists of improving the obtained partition by adding n clusters without re-clustering the initial instances. We assign to the added clusters n modes from the set of objects after removing the k' objects chosen as k' first modes. Then, we have to compute the ω_{ij} of the added clusters and the initial instances and to update our method.

4.2 Algorithm

Begin

1. *Randomly select ($k' = k - n$) initial modes, one mode for each cluster.*
2. *Compute the possibilistic similarity measure IA between instances and modes using Equation (4) then determine the membership degree ω_{ij} of each object to the k' clusters.*
3. *Allocate an object to the k' clusters using the possibilistic membership.*
4. *Compute the weight W_j for each cluster j using Equation (5) then, update the cluster mode using Equation (6).*
5. *Retest the similarity between objects and modes. Reallocate objects to clusters using possibilistic membership degrees then update the modes.*
6. *Repeat (4) until all clusters are stable.*
7. *Add n new clusters and compute the possibilistic similarity measure IA between the objects and new clusters. Determine the ω_{ij} of the objects relative to the added clusters.*
8. *Re-compute the weight W_j for each cluster j using Equation (5) then, update the cluster mode using Equation (6).*
9. *Repeat (4) and (5) until all clusters are stable.*

End

---

5 Experiments

5.1 The Framework

We used several real-world data sets taken from UCI machine learning repository [13]. They consist of Balloons (Bal), Soybean (S) Post-Operative Patient (POP), Balance Scale (BS), Solar-Flare (SF) and Car Evaluation (CE) data sets. Note that the number of classes of these databases represents the k clusters to form. Table 1 describes these data sets.

Table 1. Description of the data sets

Databases	**#Instances**	**#Attributes**	**#Classes**
Balloons (Bal)	20	4	2
Soybean (S)	47	35	4
Post-Operative Patient (POP)	90	8	3
Balance Scale (BS)	625	4	3
Solar-Flare (SF)	1389	10	3
Car Evaluation (CE)	1728	6	4

We have artificially introduced uncertainty in the values of attributes of instances of UCI data sets. Each categorical value has been presented by a possibility degree taken from $[0, 1]$. We have two extreme cases detailed with examples (see Table 2, Table 3 and Table 4) as follows:

Table 2. Example of four instances of Balloons data set

Attribute information				**Color**	**Size**	***Act***	***Age***
Color	yellow,	purple	X_1	yellow	small	stretch	adult
size	large,	small	X_2	yellow	small	stretch	child
act	stretch,	dip	X_3	yellow	small	dip	adult
age	adult,	child	X_4	yellow	small	dip	child
Classes inflated			True,	False			

1. Certain case (certain attributes' values): where the new values of attributes are set with respect to the case of complete knowledge in possibility theory. In other words, each true value of the training set is replaced by the possibility degree 1. The remaining values take the possibility degree 0.
2. Uncertain case (uncertain values of attributes): Only true values take the possibility degree 1, all remaining values can have degrees from $]0, 1[$.
   Note that Table 3 and Table 4 contain values created artificially, which are randomly generated by our program.

Table 3. Pre-treatment of the values of objects from Balloons data set in certain case

	yellow	**purple**	**small**	**large**	**stretch**	**dip**	**adult**	**child**
X_1	1	0	1	0	1	0	1	0
X_2	1	0	1	0	1	0	0	1
X_3	1	0	1	0	0	1	1	0
X_4	1	0	1	0	0	1	0	1

Table 4. Pre-treatment of the values of objects from Balloons data set under uncertainty

	yellow	**purple**	**small**	**large**	**stretch**	**dip**	**adult**	**child**
X_1	1	0.127	1	0.445	1	0.075	1	0.259
X_2	1	0.097	1	0.276	1	0.054	0.239	1
X_3	1	0.278	1	0.162	0.45	1	1	0.431
X_4	1	0.157	1	0.119	0.083	1	0.123	1

5.2 Evaluation Criteria

The evaluation criteria consist of the accuracy [14] $AC = \frac{\sum_{j=1}^{k} a_j}{T}$ (where a_j is the correctly classified objects from the total number of object n), the error rate $ER = 1 - AC$, the iteration number (IN) and the execution time (ET). The IN denotes the number of iterations needed to classify the objects after adding new clusters. The ET is the time taken to get the final partition. Note that high value of AC implies better clustering results.

5.3 Experimental Results

This section details the results obtained from experimentation using the artificial databases and the evaluation criteria. We compare the incremental method to the SKM and KM-PF then, we analyze all results. Our study is divided into two parts relative to the certain case corresponding to the complete knowledge in possibility theory and uncertain case where attributes' values take random possibility degrees. Note that we cross validate by dividing observations into training and test sets. Besides, the experiments are carried out for six different modes and the average of accuracy is calculated.

Certain Case. We define a possibility distribution for each attribute of object expressing the extent to which the actual value is true. In other words, each true value takes the degree 1 and the remaining values take 0. This case describe the complete knowledge in possibility theory where only the value known by certainty is allowed to take 1 and all other uncertain values take 0.

Table 5 shows the results based on the error rate, the iteration number and the execution time.

Table 5. The incremental possibilistic k-modes IPKM vs. SKM and KM-PF

		Bal	***S***	***POP***	***BS***	***SF***	***CE***
k'		1	3	2	2	2	3
	ER	0.57	0.54	0.42	0.36	0.27	0.32
SKM	IN	5	7	8	7	10	9
	ET/s	10.35	11.9	13.73	29.41	1794.35	2580.03
	ER	0.37	0.38	0.3	0.29	0.16	0.28
KM-PF	IN	2	4	4	2	3	2
	ET/s	0.27	1.07	1.16	7.3	42.27	76.53
	ER	0.37	0.38	0.3	0.29	0.16	0.28
IPKM	IN	2	4	4	2	3	2
	ET/s	0.27	1.07	1.16	7.3	42.27	76.53
k		2	4	3	3	3	4
	ER	0.48	0.4	0.32	0.22	0.13	0.2
SKM	IN	9	10	11	13	14	11
	ET/s	14.55	16.08	17.23	37.81	2661.634	3248.613
	ER	0.26	0.22	0.25	0.17	0.07	0.11
KM-PF	IN	3	6	6	2	6	4
	ET/s	0.9	1.34	1.4	8.51	55.39	89.63
k= k'+n				**n=1**			
	ER	0.25	0.2	0.2	0.15	0.07	0.08
IPKM	IN	3	4	5	2	4	2
	ET/s	0.7	1.02	0.82	5.32	33.7	69.5

Looking at Table 5 and using k' clusters, we remark that the incremental method provides the same results as the KM-PF. Both of them generate better results (i.e. less error rate, IN and ET) than the SKM.

After increasing k', the IPKM improves its results by adding a new cluster ($n = 1$) without re-clustering initial objects but by using the k' partitions. Our proposal reduces the error rate, the number of iteration and uses less time than the SKM and the KM-PF. The use of stable partitions considerably improves the accuracy of the IPKM compared to the other methods. In fact, SKM and KM-PF need more time to cluster the instances to k new clusters.

Uncertain Case. In this case, we handle uncertain databases where the value of each categorical object is replaced by a possibility degree that belongs to $]0, 1]$. We have also introduced two new parameters in order to better analyze the results. They consist of the parameters A and d which define respectively the percentage of uncertain attributes in the training set and the degree of possibility of each replaced attribute value. Table 6 presents the average of the error rate generated by our proposal and the KM-PF.

Table 6. The average of error rate of the IPKM vs KM-PF

		Bal	***S***	***PO***	***BS***	***SF***	***CE***
KM-PF and IPKM	**k'**	1	3	2	2	2	3
A < 50% and 0<d<0.5		0.43	0.31	0.35	0.3	0.28	0.19
A < 50% 1 and 0.5≤d≤ 1		0.42	0.35	0.39	0.29	0.17	0.23
A ≥ 50% and 0<d<0.5		0.27	0.29	0.31	0.19	0.11	0.18
A ≥ 50% and 0.5≤d≤ 1		0.34	0.3	0.38	0.3	0.2	0.19
n=1	**k=k'+n**	2	4	3	3	3	4
A < 50% and 0<d<0.5	KM-PF	0.36	0.23	0.27	0.21	0.14	0.13
	IPKM	0.35	0.2	0.25	0.2	0.13	0.1
A < 50% 1 and 0.5≤d≤	KM-PF	0.35	0.27	0.29	0.2	0.11	0.17
	IPKM	0.33	0.25	0.26	0.2	0.1	0.15
A ≥ 50% and 0<d<0.5	KM-PF	0.19	0.2	0.22	0.13	0.09	0.1
	IPKM	0.15	0.17	0.18	0.11	0.09	0.09
A ≥ 50% and 0.5≤d≤ 1	KM-PF	0.27	0.21	0.28	0.2	0.13	0.12
	IPKM	0.26	0.2	0.25	0.2	0.11	0.1

From Table 6, we notice that the IPKM provides the same error rate as the KM-PF when we use k' initial modes. By adding a new cluster i.e. $k = k' + 1$ the IPKM uses the latest partition to continue the clustering task. Thus, the IPKM computes the possibilistic membership degrees of all instances to the new cluster then, updates the clusters' modes until we get a stable partition. However, the KM-PF re-clusters all objects from beginning by taking a new number of clusters which is k.

Table 7 details the IN and ET of the proposed approach and the KM-PF, which shows that the IPKM needs the same number of iteration and time as the KM-PF to get the final partition when we deal with k' clusters.

Furthermore, when we increase the number of clusters by 1 ($k = k' + 1$), the KM-PF is forced to re-cluster all instances using this new number of clusters which makes the clustering process longer and wastes time. In contrast to the

Table 7. The IN and ET of the IPKM vs KM-PF

		Bal	***S***	***PO***	***BS***	***SF***	***CE***
KM-PF and IPKM	**k'**	1	3	2	2	2	3
The IN of the main program		3	4	8	2	8	4
The elapsed time in seconds		0.67	0.76	0.95	9.65	56.78	90.3
n=1	**k=k'+n**	2	4	3	3	3	4
The IN of the main program	KM-PF	3	4	8	2	8	4
	IPKM	2	2	4	2	4	3
The elapsed time in seconds	KM-PF	0.67	0.76	0.95	9.65	56.78	90.3
	IPKM	0.42	0.53	0.71	7.87	35.67	82.54

KM-PF, the IPKM introduces the new cluster in the existing partitions by using results given in the step before.

The ability of the IPKM to add clusters and avoiding re-clustering initial instances are its main advantages. Generally, the incremental aspect on which the possibilistic k-modes is based has obviously improved the results of both the SKM and KM-PF by providing lower error rate and IN and especially by saving much execution time.

6 Conclusion

In this paper we have proposed a new clustering approach that deals with the k-modes method in an uncertain framework and using incremental learning. We used the possibility theory to deal with two levels of uncertainty in the k-modes method, namely in the attribute values and in the belonging of objects to several clusters. Then, we studied the incremental aspect of the possibilistic k-modes by adding n new clusters and without re-clustering initial instances.

After testing our proposal using different evaluation criteria and comparing it to the SKM and KM-PF, our incremental possibilistic k-modes shows its performance by providing less error rate than the SKM and the KM-PF with less execution time.

References

1. Ammar, A., Elouedi, Z.: A New Possibilistic Clustering Method: The Possibilistic K-Modes. In: Pirrone, R., Sorbello, F. (eds.) AI*IA 2011. LNCS (LNAI), vol. 6934, pp. 413–419. Springer, Heidelberg (2011)
2. Ammar, A., Elouedi, Z., Lingras, P.: K-Modes Clustering Using Possibilistic Membership. In: Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B., Yager, R.R. (eds.) IPMU 2012, Part III. CCIS, vol. 299, pp. 596–605. Springer, Heidelberg (2012)

3. Ammar, A., Elouedi, Z., Lingras, P.: RPKM: The Rough Possibilistic K-Modes. In: Chen, L., Felfernig, A., Liu, J., Raś, Z.W. (eds.) ISMIS 2012. LNCS, vol. 7661, pp. 81–86. Springer, Heidelberg (2012)
4. Ammar, A., Elouedi, Z., Lingras, P.: The K-Modes Method under Possibilistic Framework. In: Zaïane, O.R., Zilles, S. (eds.) Canadian AI 2013. LNCS, vol. 7884, pp. 211–217. Springer, Heidelberg (2013)
5. Langford, T., Giraud-Carrier, C., Magee, J.J.: Detection of infectious outbreaks in hospitals through incremental clustering. In: Quaglini, S., Barahona, P., Andreassen, S. (eds.) AIME 2001. LNCS (LNAI), vol. 2101, pp. 30–39. Springer, Heidelberg (2001)
6. Lin, J., Vlachos, M., Keogh, E.J., Gunopulos, D.: Iterative Incremental clustering of time series. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 106–122. Springer, Heidelberg (2004)
7. Charikar, M., Chekuri, C., Feder, T., Motwani, R.: Incremental clustering and dynamic information retrieval. In: Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, pp. 626–635 (1997)
8. Ester, M., Kriegel, H.P., Sander, J., Wimmer, M., Xu, X.: Incremental clustering for mining in a data warehousing environment. In: Proceedings of the 24rd International Conference on Very Large Data Bases, pp. 323–333. Morgan Kaufmann (1998)
9. Jenhani, I., Ben Amor, N., Elouedi, Z., Benferhat, S., Mellouli, K.: Information Affinity: a new similarity measure for possibilistic uncertain information. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS (LNAI), vol. 4724, pp. 840–852. Springer, Heidelberg (2007)
10. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems 1, 3–28 (1978)
11. Dubois, D., Prade, H.: Possibility theory: An approach to computerized processing of uncertainty. Plenum Press (1988)
12. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceeding of the 5th Berkeley Symposium on Math., Stat. and Prob., pp. 281–296 (1967)
13. Murphy, M.P., Aha, D.W.: Uci repository databases (1996), `http://www.ics.uci.edu/mlearn`
14. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery 2, 283–304 (1998)
15. Huang, Z., Ng, M.K.: A note on k-modes clustering. Journal of Classification 20, 257–261 (2003)

Improving Semantic Clustering of EWID Reports by Using Heterogeneous Data Types*

Andrzej Janusz[1], Adam Krasuski[2], and Marcin Szczuka[1]

[1] Institute of Mathematics, The University of Warsaw,
Banacha 2, 02-097, Warsaw, Poland
[2] Chair of Computer Science, The Main School of Fire Service
Słowackiego 52/54, 01-629 Warsaw, Poland
krasus@inf.sgsp.edu.pl, {janusza,szczuka}@mimuw.edu.pl

Abstract. In this article we investigate an impact of inclusion of different data types into a clustering process. As a case-study we use reports from the EWID database which is a system used by Polish State Fire Service for documenting incidents. Each incident reported in that database is characterized by a set of quantitative attributes and by natural language descriptions of the cause, scene and the course of actions undergone by firefighters. We show that the utilization of both of those data types for a clustering purpose can be beneficial in terms of semantic homogeneity of the resulting groups. We argue that such clusters might serve as a useful tool in the firefighters' training process.

Keywords: Semantic clustering, interactive learning, heterogeneous data.

1 Introduction

The national fire & rescue services are typically equipped with incident data reporting systems (IDRS) which gather information about the conducted actions. Implementations of IDRS usually remain at the level of simple reporting, with no attempt to model domain knowledge related to the risks and logistics of rescue actions, and with no possibility to organize and analyze the data in a truly meaningful way. Semantically driven data organization is important in order to reason about events that are both well-supported in available data sets and easy to understand by the users. It is important to operate with bigger clusters or granules of objects in order to assign them with statistics that reflect the model's types and dynamics. The usage of domain knowledge in order to build granules with both semantically and statistically meaningful descriptions is the key to create models of complex real-world phenomena, in a process that one may call as a *granular knowledge discovery*.

* This work was partially supported by the Polish National Science Centre grants 2011/01/B/ST6/03867 and 2012/05/B/ST6/03215, and by the Polish National Centre for Research and Development (NCBiR) - grant O ROB/0010/03/001 under Defence and Security Programmes and Projects: "Modern engineering tools for decision support for commanders of the State Fire Service of Poland during Fire&Rescue operations in buildings".

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 304–314, 2013.

One possible way to build a system of granules, i.e., discover the knowledge in a form of groups (collections, granules) of data entities (basic data objects), is by clustering them. In applications, such us the one presented in this article, where both data and the knowledge is inherently imprecise, vague and incomplete, the discovery (identification) of clusters naturally leads us to usage of tools from the inventory of *soft computing*. In the particular application to our data, these tools are related to semantic knowledge processing and soft clustering.

Typically, *soft clustering* is understood as a technique that makes use of clusters with "soft" boundaries, e.g., rough or fuzzy. In our case the clusters themselves are crisply defined and once clustering is done the element belongs to only one of the clusters (granules). The "softness" in the approach described in this paper is associated with the definition of data objects. They are characterized using imprecise attributes and the similarity between them is a mere reflection of their semantic relatedness that we are trying to model. The main challenge is, in fact, to provide a relatively simple clustering algorithm with good quality input, so that the resulting clusters are meaningful and useful. The requirement for the resulting clusters (granules) to be meaningful is very important, as the evaluation of data is done by hand and is limited to a relatively small sample (see Section 3). It should be mentioned, that the approach presented in the paper may also be extended by using a soft clustering algorithm (e.g. fuzzy C-means instead of k-means) but, since the evaluation and interpretation of the results in this case is still ahead of us, we only address this case as a direction for further investigations (see Section 6).

The data objects that we are dealing with are the records from EWID system – the IDRS of Polish State Fire Service (see Section 2). The records in EWID contain heterogeneous information, a mix of binary/numerical attributes with descriptions in natural language. From prior knowledge and following the advice of domain experts we have established a general rule that drives our investigation. Namely: *Emergences with similar combinations of threats should be considered similar*. Therefore we construct our method for evaluation of cluster coherency based on the Threats Matrix (see Sections 3,5), a special representation of threats present in a given F&R operation. The Threats Matrix is created by hand and used as a tool to evaluate the automatic grouping (semantic clustering) of records in EWID.

One of the most important problem during the granule creation is definition of coherency of the granules. The artificial numerical measures like Silhouette width [1] or Calinski-Harabasz index [2], could not reflected the complex phenomena like an emergency scene. In order to define a more proper measure we introduced an abstract layer which helps to compare the similarity of incidents.

The goal of creating proper groups of incidents using semantic clustering is motivated by the need for identification and classification of general types of operations with respect to a presence of particular types of threats and particular types of threatened objects. Such a classification would be extremely helpful in evaluating the actions of commanders at the scene. It will also serve as an excellent tool for preparing the material used in training of firefighters. The main

principle is that no threat at the fire ground should be left without a proper reaction of the Fire Service. Also, specific threats generate specific actions, so it is of paramount importance to give the firefighters the ability and tools to recognize the threat scenario and act accordingly.

The paper is organized as follows. First we introduce EWID and describe the kinds of data that we are able to extract from it (Section 2), then we describe the process of data preparation and incident labeling (evaluation) by domain experts (Section 3). Next we present the techniques and experimental framework used to perform the clustering (Section 4) and present the evaluation of the obtained results (Section 5). We conclude with discussion and presentation of possible directions of further work in Section 6.

2 The EWID Reporting System

Each of approximately 500 Fire and Rescue Units (JRG) of the State Fire Service of Poland (PSP) on average conducts around 3 fire and rescue actions daily. After every single action there is a report created in an internal computer system of PSP called EWID. The data collected in the EWID database is divided into two sections – structured (database fields) and unstructured (a description in natural language (NL)). Every day around 1 500 reports are uploaded to the Headquarter of the State Fire Service of Poland. Commanders are oblige to manually fill the structure part of the report, which contain over 500 positions. Then, they have to describe in their own words a cause, conditions at the scene and their actions. Due to the amount of data that needs to be provided during a submission, many reports contain wrong or incomplete information. These errors distort statistics and impede analysis of the data.

The structural part consists of attributes describing all types of incidents. Depending on a category of an incident, the number of attributes that take values different than zero varies between 120 and 180 for a report. The most of the attributes are boolean (True/False) type but there are also numerical values (e.g. fire area, amount of water used). The natural language description part is an extension to the attribute part. It was designed to store information, which can not be represented in a form of a set of attributes. Unfortunately there is no clear regulation what should be written in the NL part. Therefore, in this part a full spectrum of data, from detailed information such as time coordinates to very general and brief descriptions can be found. The simple statistics reveal that the NL part contains approximately three sentences which describe the situation at the fire ground, actions undertaken by a commander and weather conditions.

To this day, the EWID has stored reports on approximately 7 million incidents. Undoubtedly, the EWID is a rich source of information about threats arising at the incident scene and about appropriate but sometimes also flawed counter-measures. However, this database is difficult to process and to analyze. One of the reasons for this is the curse of dimensionality and the necessity of processing the natural language descriptions. The simple methods tend to not reflect the phenomena behind the EWID data, therefore more sophisticated methods are

needed. Recently, a few papers were published which present more advanced approaches to analyzing such data. They utilize the methods from the data mining domain [3–5], text mining [6] or even the granular computing approach [7]. However, in our opinion even the most promising algorithms for knowledge discovery should interact with domain experts.

When analyzing data such as Fire Service reports, an expert can interpret their semantics, find interesting patterns or cases and can guide the direction of the research. The works of Poelmans et. al (see [8–10]) show that domain experts supported by tools for pre-processing the information and presenting it in a way convenient for the experts, may help in discovering important knowledge from structured and unstructured data (e.g. police reports). Therefore, we are interested in developing tools that could facilitate work of experts by organizing the available data into meaningful categories. One way to approach this task is to devise reliable and scalable algorithms for partitioning the data into semantically homogeneous clusters [11].

3 Data Collection and Processing

In our experiments we used a subset of all EWID data corresponding to incidents reported in Warsaw, in the years between 1992 and 2011. We selected only the reports representing the category of fires in residential buildings. This data set consisted of 31 556 reports.

In the labeling process, the experts were requested to use a pre-defined framework, so called Threats Matrix. The task of recognition and categorization of threats and threatened object is formalized in the handbook of tactics of German Fire Service [12]. After arriving at a fire ground or an emergency scene, German commanders have to evaluate and recognize the appearing threats. In order to do this systematically and not to miss any of the threats they have to fill the Threats Matrix (in German – Gefahrenmatrix) [12]. The Threats Matrix helps to identify the threats emerging at the scene and the objects to which those threats apply. The columns of the matrix represent the possible threats, and the rows correspond to objects which could be threatened. Table 1 depicts the Threats Matrix.

In German language, column names of the Threats Matrix were chosen so that they could be easily remembered. In order to help fire brigade commanders in memorizing all the threats, the German names were chosen so that they form the pattern AAAA-C-EEEE which corresponds to *Angstreaktion, Atemgifte, Atomare Strahlung, Ausbreitung, Chemische Stoffe, Einsturz, ElektrizitÃđt, Erkrankung, Explosion.* The sign '–' in the table indicates that the threat does not apply to the object. German commanders use the Threats Matrices to better organize their actions at the fire ground.

We created a special methodology for labeling the EWID reports [13] with the risks defined in the Threats Matrix. The labeling process consists of two main phases: the *tutorial phase* and the *labeling phase*. The tutorial phase is focused on introducing the Threats Matrix and the layout of the EWID incident

Table 1. The Threats Matrix used by German commanders. Legend: A1 – Fear, A2 – Toxic smoke, A3 – Radiation, A4 – Fire spreading, C – Chemical substances, E1 – Collapse, E2 – Electricity, E3 – Disease or injury, E4 – Explosion.

Threat/object	A1	A2	A3	A4	C	E1	E2	E3	E4
People (ME)									
Animals (T)									
Environment (U)	–					–	–	–	
Property (S)	–	–						–	
Rescuers (MA)									
Equipment (G)	–	–						–	

reports to experts. It is divided into three consecutive parts. In the first part, experts were introduced to the format and the purpose of the Threats Matrix. In the second part, some examples of filled Threats Matrices are presented and discussed with the experts. In the third part, experts receive an exemplary EWID report together with a corresponding filled Threats Matrix.

The labeling phase consists of many evaluation steps. In every step the experts were provided with a single EWID report. On the ground of the information about the incident described in the report, they are asked to evaluate threats which appeared during the incident and to fill in the Threats Matrix. Every report description used in our experiment was labeled by only one expert. In total, we collected 406 labeled incident descriptions. Since it is reasonable to assume that similar incidents are characterized by similar threats, we use the labeled documents to evaluate semantic homogeneity of different clusterings of our data.

4 Experimental Settings

In our experiments we wanted to find out what impact on a granulation of data has a choice of EWID reports' representation. Since the reports have two parts that convey different types of information (the structured and natural language parts) and require different preprocessing, we were interested which of them is more important. We also wanted to check whether a concatenation of those two seemingly different representations can be beneficial for identification of semantically homogeneous clusters in the data.

We performed the experiment in R System [14]. The two parts of the incident reports were subjected to different preprocessing steps. Attributes from the structured part, which took only a single value (zero) on the preselected set of 31 556 reports were removed from the data set. From the remaining 353 attributes, those with numerical values were linearly scaled into the $[0, 1]$ interval. Next, the vectors representing each report were normalized (value of each of the attributes was divided by the Euclidean norm of the vector).

After preprocessing, an information system was constructed and represented by a sparse matrix implemented list of entity-attribute-value (EAV) triples in the *slam* library[1].

The natural language parts of the reports underwent a typical preprocessing for textual data. First, the strings representing descriptions of causes, the site and the course of the F&R services intervention were tokenized and stop-words that corresponded to irrelevant phrases were removed. A dictionary of stop-words was built by a domain expert. The remaining words were lemmatized using the Morfologik[2] software. Next, bag-of-words representations of the reports were created and each incident description was associated with a numerical vector of TF-IDF weights of the terms. If the term w_i, that corresponds to the i-th position in a vector representation of a report $d \in D$, appeared n_i times in this report, then TF-IDF weight of w_i in d is defined as:

$$d_i = tf_i \times idf_i = \frac{n_i}{\sum_{j=1} n_j} \times \log \frac{|D|}{|\{r \in D : w_i \in r\}|}.$$

All the TF-IDF vectors were normalized. Due to their sparsity, they were also stored in a from of a sparse matrix with 9 632 columns corresponding to different terms.

Having the numerical representations of the two parts of the EWID reports we were able to easily combine them by concatenating the corresponding vectors. In this way we obtained three information systems representing our data. The first one was based on information from the structural part, the second represented natural language description of incidents and the third one was a hybrid of the first two. Our main goal in the experiment was to find out which of those representations is more suitable for extracting data granules consisting of semantically similar incidents.

5 Evaluation of Semantic Homogeneity

To perform clustering of data we utilized the *spherical k-means* algorithm implemented in R library *skmeans*. The algorithm is very similar to the classical *k-means*, but instead of the Euclidean distance it uses the *cosine distance*[3] which makes it more suitable for dealing with high-dimensional data. In order to thoroughly evaluate impact of the different data representations on semantic homogeneity of clusters we compare the results obtained for 20 different values of the parameter k in a range between 2 and 100. We used the *pclust* as a clustering method with the number of repetitions set to 25.

We began by comparing values of an internal clustering quality measure obtained for each representation of data and different values of k. For this purpose,

[1] The documentation of the package can be found in the CRAN repository: `http://cran.r-project.org/web/packages/slam/index.html`

[2] Morfologik project page `http://morfologik.blogspot.com`

[3] In practice, the cosine distance which is equal to *arc cosine* of two vectors, is approximated by a function $dist(x, y) = 1 - cosine(x, y)$ that does not satisfy the triangle inequality.

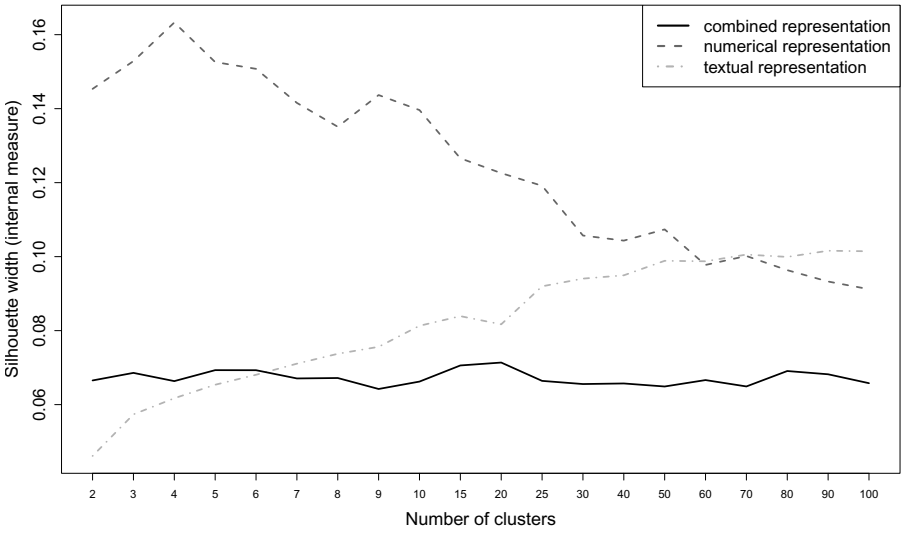


Fig. 1. A comparison of internal average silhouette widths of clusterings into an increasing number of groups and using different representations of the incidents. The figure shows average values from 5 repetitions of the experiment.

we decided to use the *average silhouette width* measure which is described in details in [15]. The results of this comparison are shown in Figure 1.

Since values of the silhouette width are always in the $[-1, 1]$ interval and the representations of the data were normalized before the experiment, it is meaningful to compare different clusterings using this particular measure. From the figure it seems that the structural part conveys information that allows to identify more distinct granules of the data. The silhouettes of the clusterings on the structural and the textual data for different values of k are strongly negatively correlated. The separability of clusters computed using the textual part of the data increases with the increasing number of groups. At the same time the silhouettes of clusterings performed on the structured part systematically diminishes. Interestingly, separability of clusters obtained from the hybrid representation is usually much lower than in the case of the other two and seems to be independent of the number of considered groups.

Although the internal measure seems to favour the structural information, we still wanted to verify whether the separability with regard to the cosine distance corresponds to semantic homogeneity of the clustering. As it was already mentioned in Section 3, the semantic similarity of incidents requiring involvement of F&R services can be considered in terms of the associated risks and threats. Because of this fact, for the evaluation of semantic coherency we could utilize the reports that were labeled with pairs of threats and threatened objects (i.e. the risks – see Section 3) by experienced commanders.

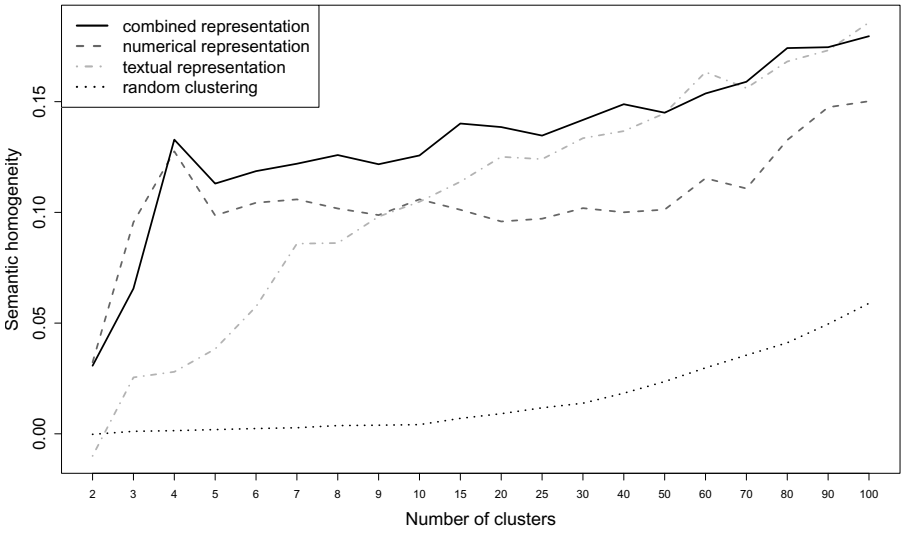


Fig. 2. A comparison of semantic homogeneity of clusterings into an increasing number of groups and using different representations of the incidents. The figure shows average values from 5 repetitions of the experiment.

Following the research described in [16] we defined an analogical measure of the semantic homogeneity. If T_i and T_j are the sets of labels (threats) associated with i-th and j-th incidents from our validation set, then their dissimilarity can be defined as:

$$F_1 dissim(T_i, T_j) = 1 - 2 \cdot \frac{precision(T_i, T_j) \cdot recall(T_i, T_j)}{precision(T_i, T_j) + recall(T_i, T_j)},$$

where

$$precision(T_i, T_j) = \frac{|T_i \cap T_j|}{|T_i|} \quad \text{and} \quad recall(T_i, T_j) = \frac{|T_i \cap T_j|}{|T_j|}.$$

A *semantic dissimilarity* between two groups of labeled incidents G_1 and G_2 can be defined as an average $F_1 dissim$ between pairs of incidents from different groups:

$$semDissim(G_1, G_2) = \frac{\sum_{T_i \in G_1, T_j \in G_2} F_1 dissim(T_i, T_j)}{|G_1| \cdot |G_2|}.$$

If for a division of data into groups we denote by $G(T_i)$ the labeled incidents that belong to the same group as T_i, and by V we denote the validation set of all incidents labeled by the commanders, then we can define a semantic homogeneity of T_i as:

$$homogeneity(T_i) = \frac{B(T_i) - A(T_i)}{\max\left(A(T_i), B(T_i)\right)}, \quad \text{where}$$
$$A(T_i) = semDissim(T_i, G(T_i) \setminus T_i) \quad \text{and}$$
$$B(T_i) = semDissim(T_i, V \setminus G(T_i)).$$

As the semantic homogeneity of a clustering we will take the average semantic homogeneity of incidents from our validation set. Values of this measure computed for clusterings into different number of groups and using different representations of data are shown in Figure 2.

The evaluation results clearly show advantages of using the hybrid representation of the incidents from the EWID system. The clusterings obtained from the combined structural and textual parts of the reports achieved the highest scores for nearly all values of k. Moreover, even when results of one of the other representations were significantly lower for a partitioning into a particular number of clusters, the semantic homogeneity of the clustering in the combined attribute space remained robust. It is also worth mentioning that comparing to random divisions into groups, the semantic homogeneities of all investigated clusterings were significantly higher.

6 Conclusions and Future Work

The experimental evaluation of the proposed method for incident clustering based on heterogeneous (structural+text) data representation shows improvement in terms of semantic coherence and usability of resulting clusters. Moreover, we have confirmed our previous claim that the internal measures of cluster quality are inappropriate for our task. The lack of semantic context makes these measures hardly relevant as they are unable to capture the kind of objects' relatedness we are looking for. The clustering supported by the evaluation based on expert knowledge collected with a use of Threat Matrices led us also to a conclusion, that in the existing framework of the EWID system the NL description part of an incident report is absolutely crucial and that we cannot expect meaningful results from approaches that ignore this component. At the same time, this description is in the most cases insufficient to obtain a complete understanding of the incidents. All in all, the heterogeneous approach seems to be the most logical and promising.

The clustering of incidents and the resulting clusters have already proven to be a helpful addition to evolution of models and processes used in the training and evaluation of Fire Service operations. In particular, they have a positive influence on robustness of decision making process of commanders at the scene. We are fully aware that our method of semantic measurements of cluster homogeneity is tied to the specific data set. However, we strongly believe that the expertise gained through our experiments would be also useful in other domains, such as police reports or medical reords, where data is stored in similar manner as fire&rescue reports.

In the immediate future we plan to investigate the scenario in which not only the objects (cluster elements) are "soft" but also the clusters are constructed in a "soft" way. That would entail using clustering techniques that create *soft granules*, i.e., clusters that may overlap. Such an effect may be achieved using fuzzy clustering algorithms such as the *fuzzy c-means* or *fuzzy-spherical-k-means*. The clustering model that makes it possible for an incident to belong to many clusters in various degrees holds a promise for better detection of incident reports that are either erroneous (due to incomplete or careless input) or atypical (rare and important). Detection of both kinds of unusual situations may significantly improve the decision making processes that are based on the EWID data and might be helpful in a training process of Fire Service commanders.

References

1. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53–65 (1987)
2. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics-Theory and Methods 3(1), 1–27 (1974)
3. Holmes, M., Wang, Y., Ziedins, I.: The application of data mining tools and statistical techniques to identify patterns and changes in fire events. Technical Report 95, New Zealand Fire Service, Auckland, NZ (May 2009)
4. Xiangxin, L.: Rational judging method of fire station layout based on data mining. In: 2nd IEEE International Conference on Emergency Management and Management Sciences (ICEMMS), pp. 455–458. IEEE (2011)
5. Krasuski, A., Kreński, K., Łazowy, S.: A method for estimating the efficiency of commanding in the State Fire Service of Poland. Fire Technology 48(4), 795–805 (2012)
6. Kreński, K., Krasuski, A., Łazowy, S.: Data mining and shallow text analysis for the data of State Fire Service. In: Proceedings of Concurrency, Specification and Programming - XXth International Workshop, CS&P 2011, Białystok University of Technology, pp. 313–321 (2011)
7. Krasuski, A., Ślęzak, D., Kreński, K., Łazowy, S.: Granular knowledge discovery framework. In: Pechenizkiy, M., Wojciechowski, M. (eds.) New Trends in Databases & Inform. AISC, vol. 185, pp. 109–118. Springer, Heidelberg (2012)
8. Elzinga, P., Poelmans, J., Viaene, S., Dedene, G., Morsing, S.: Terrorist threat assessment with formal concept analysis. In: Proceedings of the 2010 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 77–82. IEEE (2010)
9. Poelmans, J., Elzinga, P., Viaene, S., Hulle, M.M.V., Dedene, G.: Gaining insight in domestic violence with emergent self organizing maps. Expert Syst. Appl. 36, 11864–11874 (2009)
10. Poelmans, J., Elzinga, P., Dedene, G., Viaene, S., Kuznetsov, S.: A concept discovery approach for fighting human trafficking and forced prostitution. In: Andrews, S., Polovina, S., Hill, R., Akhgar, B. (eds.) ICCS-ConceptStruct 2011. LNCS, vol. 6828, pp. 201–214. Springer, Heidelberg (2011)
11. Szczuka, M.S., Janusz, A.: Semantic clustering of scientific articles using explicit semantic analysis. T. Rough Sets 16, 83–102 (2013)

12. Graeger, A., Cimolino, U., de Vries, H., Sümersen, J.: Einsatz-und Abschnittsleitung: Das Einsatz-Führungs-System (EFS). Ecomed Sicherheit (2009)
13. Krasuski, A., Wasilewski, P.: The Detection of Outlying Fire Service's Reports. The FCA Driven Analytics. In: Cellier, P., Distel, F. (eds.) Contributions to the 11th International Conference on Formal Concept Analysis, TU Dresden, pp. 35–50 (2013)
14. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008)
15. Kaufman, L., Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley Interscience, New York (1990)
16. Janusz, A., Ślęzak, D., Nguyen, H.S.: Unsupervised similarity learning from textual data. Fundamenta Informaticae 119(3), 319–336 (2012)

Rough Clustering Generated by Correlation Clustering

László Aszalós and Tamás Mihálydeák

Department of Computer Science, Faculty of Informatics,
University of Debrecen, Debrecen, Hungary
{aszalos.laszlo,mihalydeak.tamas}@inf.unideb.hu

Abstract. Correlation clustering relies on a relation of similarity (and the generated cost function). If the similarity relation is a tolerance relation, then not only one optimal partition may exist: an object can be approximated (from lower and upper side) with the help of clusters containing the given object and belonging to different partitions. In practical cases there is no way to take into consideration all optimal partitions. The authors give an algorithm which produces near optimal partitions and can be used in practical cases (to avoid the combinatorial explosion). From the practical point of view it is very important, that the system of sets appearing as lower or upper approximations of objects can be taken as a system of base sets of general (partial) approximation spaces.

Keywords: Rough clustering, correlation clustering, set approximation.

1 Introduction

Clustering is a task of grouping a set of objects in such a way, that the objects in the same cluster (group) are more similar to each other, than to the objects in the other clusters. This clustering gives a partition (disjoint set of sets) of the whole set of the objects. A member of a partition is called *cluster* in the following. There are many algorithms to construct this partition (see [1] for a recent survey). Some of them assign a distance to each pair of objects, and are based on the distance or on the density of objects to connect or separate them. It is common to use these algorithms for discrete and even categorical data, where the concept of distance is unnatural. Algorithms finally produce a partition or a tree of subsets [2]. As in most cases the number of groups is given [3], clusters can be generated from the tree in the latter case easily.

There is an exceptional clustering method: the correlation clustering [4–6]. In this case the number of clusters is not given in advance, but there is a cost function which has to be minimized. In correlation clustering no attribute is taken into account, but there is a relation of similarity. At the beginning this relation was total, but here it is assumed that it is partial, and moreover is symmetric, and reflexive naturally; so it is a partial tolerance relation [7, 8]. This means that two object can be similar, dissimilar, or it is possible, that

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 315–324, 2013.

there is no information about their similarity: namely no one compared them yet, or they are incomparable. There is no degree of similarity or dissimilarity as given in [9], the correlation clustering can manage alone the inconsistencies of the tolerance relation.

Let matrix $M = (m_{ij})$ be the matrix of the partial relation R_M of similarity: $m_{ij} = 1$ whenever objects i and j are similar, $m_{ij} = -1$ whenever objects i and j are dissimilar, and $m_{ij} = 0$ otherwise. A partition of a set S is a function $p : S \to \mathbb{N}$. Objects x and y $(x, y \in S)$ are in the same cluster at partitioning p, if $p(x) = p(y)$.

The cost function counts the negative cases i.e. it gives the number of cases whenever two dissimilar objects are in the same cluster, or two similar objects are in different clusters. The cost function of a partition p and a relation R_M with matrix M is

$$f(p, M) = \frac{1}{2} \sum_{i<j} (m_{ij} + |m_{ij}|) - \sum_{i<j} \delta_{p(i)p(j)} m_{ij},$$

where δ is the Knockecker delta symbol [10]. For a fixed relation the partition with the minimal cost function value is called *optimal*. Solving a correlation clustering problem is equivalent to minimizing its cost function, for the fixed relation. If the value of this optimal cost function is 0, the partition is called *perfect*.

It is easy to check that the solution cannot be generally perfect for a similarity relation. Take the relation on the left of Fig. 1. The dashed line denotes dissimilarity and the normal line similarity. E.g. a man (a) similar to a Paris doll (b) by its shape, and to a mouse (c) by its organs, but the doll and the mouse are dissimilar. On the right, Fig. 1 shows all the partition of these objects, where rectangles indicates the clusters. The thick lines denote the pairs which are counted in the cost function. In the upper row the value of the cost function is 1 (in each case), while in the two other cases it is 2 and 3, respectively.

The number of partition can be given by the Bell number [11], which grows exponentially. Hence, in general — even in the case of some dozens of objects — the optimal partition cannot be determined in reasonable time, thus a search

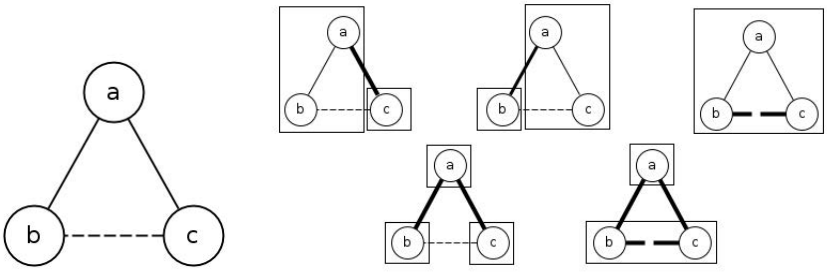


Fig. 1. Minimal frustrated similarity graph and its partitions

algorithm which produces a near optimal partition would be more useful in practical cases. The authors examined the effectiveness of the most common methods for correlation clustering, and constructed some new effective ones [12]. The latter methods allow us to get a near optimal solution even for relations on thousand objects in reasonable time.

A typical application of clustering is to add new objects to the clusters determined before based on a sample set of objects. At our methods the complexity of determining the cluster of a new object is $O(n^2)$ in general, but when restructuring is unavoidable it is $O(n^3)$. There are some methods where the complexity of adding a new object to the clusters is $O(n)$, but in this case the starting clusters cannot be modified.

The structure of the papers is as follows. After reviewing some important properties of correlation clustering, the authors show how it can be used to construct lower and upper approximations of an object based on a given similarity (tolerance) relation. In many practical cases there is no way (without combinatorial explosion) to give the exact approximations relying on all optimal clustering processes and so an approximation algorithm is applied to get the approximations based on near optimal ones. The results are visualized in different examples. Finally the problem of handling noise is discussed.

2 Rough Correlation Clustering

The traditional correlation clustering produces a partition, where an object belongs to exactly one cluster, but in many cases an object could be added to several clusters. Clustering algorithms usually use a random number generator, so the cluster containing a given object is determined by a random factor. In medical database, for example, it can cause some unacceptable results.

For us the similarity is a rough relation in the sense, that it is not an equivalence relation, and so there is no well–defined partition behind it, but there are some different partitions produced by optimal clustering processes: an object may belong to several clusters (belonging to different partitions) at the same time. If we take into consideration all optimal partitions, three different cases appear. Two objects

- always are in the same cluster,
- always are in different clusters, or
- sometimes are in the same and sometimes are in different clusters.

At first take the relation in Fig. 1. There are three optimal partitions (in the upper row). There is no other object which is always in the same partition as the object a in these cases. The same is true for objects b and c.

Let the lower approximation of the object x be the set of objects which are always in the same partition with x. Hence the lower approximation of a is a singleton, containing a (in the following $l(a) = \{a\}$). The lower approximations of b and c are similar. These lower approximations are denoted by grey rectangles in Fig. 2.

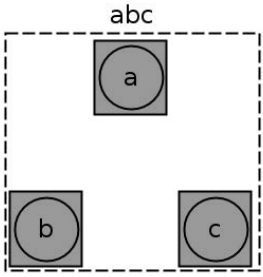


Fig. 2. Lower and upper approximations of the relation of Fig.1

Let the upper approximation of the object x be the set of objects which are at least once in the same partition with x. The third partition in Fig. 1 contains all objects, so the upper approximation is a set containing all of them ($u(a) = u(b) = u(c) = \{a, b, c\}$).

Let us define the concepts more formally. If $p_1, \ldots, p_n$ are the optimal partitions of S for relation R_M, then the lower approximation of $x \in S$ is $l(x) = \{y | p_i(x) = p_i(y),$ for all $1 \leq i \leq n\}$, and upper approximation of x is $u(x) = \{y | p_i(x) = p_i(y),$ for some $1 \leq i \leq n\}$. By definition $l(x) \subseteq u(x)$, for any object x. Moreover this definition satisfies Lingras and Peters' criteria [13] on rough clustering.

Remark that functions l and u assign sets to objects. They approximate objects by sets. The approximation l is strict in the sense, that if one could fill some information gap at the original relation R_M, or, by other words, narrow the partiality of R_M, then for the improved lower approximation l' the following holds: $l(x) \subseteq l'(x)$ for any object x. Similarly the approximation u is tolerant: for the improved upper approximation u', for any object x, $u'(x) \subseteq u(x)$ holds. Hence the truth — real similarity relation — is between the two approximations.

Note, that this definition does not use the tolerance relation R_M in a such direct way, as Kryszkiewicz [14] did for set approximations. Here the optimal partitions are determined by the tolerance relation, and the partitions generate the lower and the upper approximations. An optimal partition can overwrite the tolerance relation: it can put two objects into the same cluster, while they are dissimilar by R_M, and can put two objects into different clusters while they are similar by R_M. Therefore this method could produce a more precise clustering by reconditioning the tolerance relation.

By using a different notation, if $P(x)$ denotes the cluster of x in partition p, i.e. $P(x) = \{y | p(x) = p(y)\}$, then $l(x) = \cap_i P_i(x)$ and $u(x) = \cup_i P_i(x)$.

On the base of these approximations two relation can be defined: $xR_l y$ iff $x \in l(y)$ and $xR_u y$ iff $x \in u(y)$. It is left to the reader, to check that the relation of R_l is an equivalence relation, and the members of $l(x)$ for some object x are indiscernible with respect to R_l. Moreover the S/R_l gives the R_l-elementary sets

in the Pawlak sense. The set of lower approximations $\mathcal{B} = \{l(x)|x \in S\}$ can be treated as a one-layered base system. The relation R_u is not transitive, so it cannot be an equivalence relation in general. But the reflexivity and symmetry holds, as they can be proved easily by their definition.

Let us see a more complicated case! We generated 13 points randomly in the unit square, as Fig. 3 shows. Two points are similar, if their Euclidean distance is less than or equal to 0.2, and they are dissimilar, if their distance is greater or equal to 0.8. If their distance greater than 0.2 but less than 0.8 we didn't want to decide about their similarity, in these cases the relation is not defined. Dissimilarity holds only between object m and object b, d, g, h and l. Object b, d, g, h and l are similar to each other, except g and l; object e is similar to c and j, but c and j are not similar. Similarly object k is similar to i and f, but i and f are not similar.

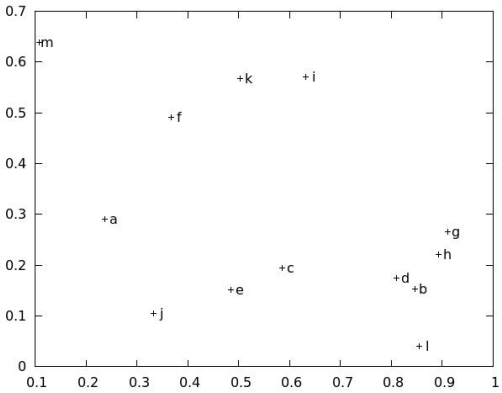


Fig. 3. Randomly generated points on the unit square

Our software[1] checked all the $27,644,437$ different partitions, and found 37 optimal ones. In these case the value of the cost function is 0, so these partitions are perfect. The generated lower and upper approximations of the points of similarity is presented in Fig. 4. The figure on the left express the lower and upper approximations. The figure on the right shows the same without names. Here the i^{th} column and i^{th} row of the picture correspond to one object. Different rows denote different objects. The rows are grouped in a such way, that the objects with the same lower approximations follow each others. Hence it does not present the exact lower and upper approximation for a selected object, but it demonstrates the structure of all approximations. It can be useful at large set of objects.

[1] It is available at `http://www.inf.unideb.hu/~aszalos/roughclusters`

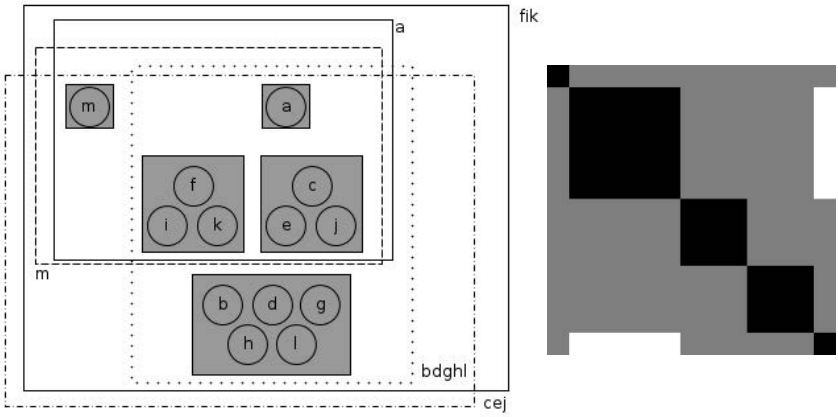


Fig. 4. Rough clusters of the points of Fig. 3 with dissimilarity ≥ 0.8

The lower and upper approximations correspond to the expectations in Fig. 4. E.g. $b \notin u(m)$, as object b and m are dissimilar, and in perfect clustering dissimilar elements are in different clusters. Moreover $f \in l(i)$, because both of them are similar to object k, and in perfect clustering the similar objects are in the same cluster. Hence object f and k are in the same cluster, like k and i. Therefore f and i need to be in the same cluster.

If we take the same points in Fig. 3, but the dissimilarity begins at distance 0.5, then m differs from each object, but a, f and k. There is no relation between a and m, so they can be in the same or in different clusters, so $a \in u(m)$ and $a \notin l(m)$. There are 4 perfect clusterings, so $k \notin u(m)$, because object k is similar to object i which differs from m. Object f is similar to k, so $f \notin u(m)$. The same pairs are similar as before, so the lower approximations are the same, but there is a new limit for dissimilarity, so the upper approximation changed dramatically as Fig. 5 shows on the left.

Fig. 5. Rough clusters of the points of Fig. 3 with dissimilarity ≥ 0.5 (on the left) and the same when singletons are handled specially as introduced in Section 5 (on the right)

3 Approximation by Means of Approximations

As it was mentioned before, in the case of large sets there is no effective way to find all optimal partitions. Hence it is not possible to get the exact (based on all optimal partitions) lower and upper approximations. We tested the partitions of the first natural numbers, with respect to the similarity based on the number of common divisors. But even in the case of 50 numbers there are millions of optimal partitions, thus an exhaustive search is not admissible here.

Let's approximate the optimal partitions. Use a search method, and generate the lower and upper approximations from the best (maybe not optimal) partitions.

To demonstrate this kind of approximation, 500 points are generated on the unit square. The points and the result of the clustering are in Fig. 6. In general it is a dirty test to use any clustering method on a random data to get something, because the concept of distance is missing, e.g. the scale factor of the different attributes is not known. At our test the relation R_M of the similarity is given. It is the same as before, two objects are similar, if their Euclidean distance is less than or equal to 0.2; and they are dissimilar, if their distance is greater or equal to 0.5.

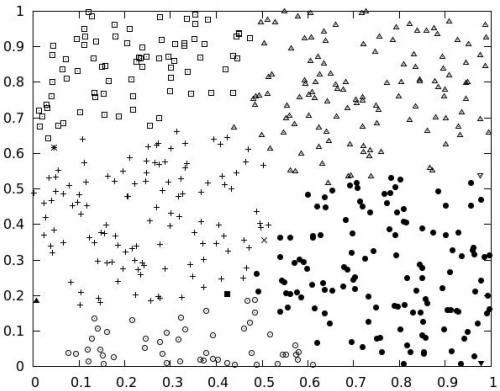


Fig. 6. Rough clustering of 500 random points on the unit square

Our algorithm was run 1, 10, 100 and 1000 times, and their best results were similar, as Table 1 shows. The generated lower and upper approximations were almost the same in the last two cases. Relying on these results it is very probable that there is no need to use many iterations to get different near optimal results. We note that a similar result appeared when the dissimilarity started at 0.8 and this gave thousands of best solutions which generated many small sets as lower approximations.

Table 1. Trials to find the optimal partitions

Trials	No. best partitions	Value of the cost function
1	1	2613
10	1	2613
100	3	2613
1000	4	2613

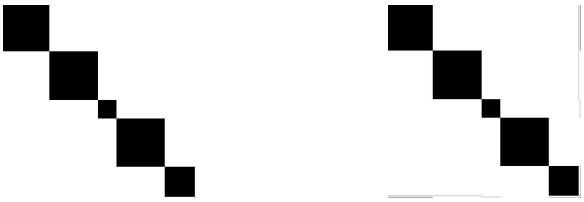


Fig. 7. Lower and upper approximations based on 1, and 4 partitions

As Fig. 6 shows, the approximations differ only at some points, hence the lower and upper approximations are very close to the real clusters.

If there is only one best partition, then the lower and upper approximations are the same: $l(x) = u(x)$ for all objects x. Hence the figure on the left in Fig. 7 contains 5 big and 3 very small squares. (The small squares are at the right bottom of the picture.) The structure on the right is more complicated: it contains 6 singleton clusters. These singletons are similar to some big clusters, as the grey lines show on the right and on the bottom. This means, that e.g. the signs $\times$ and the star on the left near to 0.6 can belong to plus signs in Fig. 6. This star sign can belong to dotted squares and filled circles, too.

4 Handling of Singletons

Usually the objects which do not belong to any bigger cluster are treated as noise, and not taken into account. This can be done in this case, too. The lower approximations in Fig. 7 has 3 and 6 singletons, respectively; but it is hard to notice them. Hence they can be left out. There are two possibilities to do this:

1. The process is the same: take all best partitions P_i, and construct $l(x) = \cap_i P_i(x)$, but the base system does not contains singletons:

$$\overline{\mathcal{B}} = \{l(x) | l(x) \neq \{x\},\ x \in S\}\,.$$

The base system remains one-layered, but became partial [15].

2. The singletons are left out from the best partitions. This complicates the former results. For example take the simplest frustrated similarity relation in Fig. 1 In this case $P_1(a) = \{a, b\}$, $P_2(a) = \{a, c\}$ and $P_3(a) = \{a, b, c\}$, so $\hat{l}(a) = \{a\}$, as before. $P_1(b) = \{a, b\}$, $P_3(b) = \{a, b, c\}$, so $\hat{l}(b) = \{a, b\}$. $P_2(b)$ is singleton, so it is not counted. Similarly $\hat{l}(c) = \{a, c\}$. Before the relation R_l was an equivalence relation, but here it no longer holds. E.g. $a \in \hat{l}(c)$, but $c \notin \hat{l}(a)$. Moreover if base system is defined as

$$\widehat{\mathcal{B}} = \left\{ \hat{l}(x) | x \in S \right\},$$

then $a \in \hat{l}(b)$ and $a \in \hat{l}(c)$, so the base system is not one-layered.

If object x differs from many ones, and hence in the best partitions it is always a singleton, then the best way to define $\hat{l}(x)$ is as an empty set (empty intersection of empty sets). This means that the property $x \in \hat{l}(x)$ will not hold in this case. Hence the third property (in Mitra at all [16]) — if x is not a member of any lower approximation, then it belongs to two or more upper approximations — does not hold, because x cannot be a member of any upper approximation.

We repeated the calculations with this approximation variant. The figures in Fig. 4 remain the same, and Fig. 5 on the left changed only at one point (on the bottom left): as h is singleton in each optimal partition, but it is in the same cluster than a, therefore $a \in \hat{l}(h)$. In reverse a is not singleton in three cases, and it is in the same cluster as h only once. Hence $h \in \widehat{u}(a)$, but $h \notin \hat{l}(a)$, as Fig. 5 on the right demonstrates. Moreover repeating the calculations of clustering 500 points, the result is almost the same as in Fig. 7, just the 3 clusters on the bottom right are deleted, because they were singletons in all optimal partition. This result is the same as the result of former variant.

5 Conclusion and Further Work

In this paper the authors showed that if a relation R_M is given, then the base system of an approximation space could be generated. The next step is to create this relation for some common task in data mining, and compare results with [17].

As the original relation changes, the optimal partitions could be different [18]. Our software could easily follow this modification, and find the best partitions, and regenerate the base system. Hence we are interested in the complexity issues of this kind of dynamism in real world applications.

Acknowledgement. The publication was supported by the TÁMOP-4.2.2.C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund.

References

1. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recognition Letters 31, 651–666 (2010)
2. Kumar, P., Krishna, P.R., Bapi, R.S., De, S.K.: Rough clustering of sequential data. Data & Knowledge Engineering 63, 183–199 (2007)
3. Yang, L., Yang, L.: Study of a cluster algorithm based on rough sets theory. In: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, ISDA 2006, vol. 1, pp. 492–496. IEEE Computer Society, Washington, DC (2006)
4. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning 56, 89–113 (2004)
5. Becker, H.: A survey of correlation clustering. Advanced Topics in Computational Learning Theory, 1–10 (2005)
6. Zimek, A.: Correlation clustering. ACM SIGKDD Explorations Newsletter 11, 53–54 (2009)
7. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
8. Mani, A.: Choice inclusive general rough semantics. Information Sciences 181, 1097–1115 (2011)
9. Dakuan, W., Xianzhong, Z., Xin, D., Chen, Z.: Variable rough set model and its knowledge reduction for incomplete and fuzzy decision information systems. International Journal of Information Technology 3, 140–144 (2006)
10. Néda, Z., Sumi, R., Ercsey-Ravasz, M., Varga, M., Molnár, B., Cseh, G.: Correlation clustering on networks. Journal of Physics A: Mathematical and Theoretical 42, 345003 (2009)
11. Aigner, M.: Enumeration via ballot numbers. Discrete Mathematics 308, 2544–2563 (2008)
12. Aszalós, L., Mária, B.: Advanced Search Methods. Educatio Társadalmi Szolgáltató Nonprofit Kft (2012) (in Hungarian)
13. Lingras, P., Peters, G.: Applying rough set concepts to clustering. In: Peters, G., Lingras, P., Lzak, D., Yao, Y. (eds.) Rough Sets: Selected Methods and Applications in Management and Engineering, pp. 23–37. Springer, London (2012)
14. Kryszkiewicz, M.: Rough set approach to incomplete information systems. Information sciences 112, 39–49 (1998)
15. Csajbók, Z., Mihálydeák, T.: A general set theoretic approximation framework. In: Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B., Yager, R.R. (eds.) IPMU 2012, Part I. CCIS, vol. 297, pp. 604–612. Springer, Heidelberg (2012)
16. Mitra, S., Pedrycz, W., Barman, B.: Shadowed c-means: Integrating fuzzy and rough clustering. Pattern Recognition 43, 1282–1291 (2010)
17. Parmar, D., Wu, T., Blackhurst, J.: MMR: an algorithm for clustering categorical data using rough set theory. Data & Knowledge Engineering 63, 879–893 (2007)
18. Peters, G., Weber, R., Nowatzke, R.: Dynamic rough clustering and its applications. Applied Soft Computing 12, 3193–3207 (2012)

Recursive Profiles of Businesses and Reviewers on Yelp.com*

Matt Triff[1] and Pawan Lingras[2]

[1] Department of Computer Science
University of Saskatchewan, Saskatoon, Saskatchewan, Canada
[2] Department of Mathematics and Computing Science, Saint Mary's University
Halifax, Nova Scotia, B3H 3C3, Canada
matt.triff@gmail.com, pawan@cs.smu.ca

Abstract. This paper uses a novel recursive meta-profiling technique where profiles from one set of objects dynamically change the representation of another set of objects. Two profiling schemes evolve in parallel influencing each other through indirect recursion. This is demonstrated with the help of a yelp.com dataset consisting of businesses and reviewers. A business is represented by static information obtained from the database and dynamic information obtained from clustering of reviewers who reviewed the business. Similarly, the reviewer representation augments the static representation from the database with profiles of businesses who are reviewed by these reviewers. The resulting service provides a facility for users to find similar businesses/reviewers based on the grading of the business, easy/hard grading, and types of businesses. It also provides a succinct profile of business/reviewer based on these factors, so users can put the reviews in context.

1 Introduction

At yelp.com, a business can be represented by an information granule consisting of the number of five-star, four-star, ..., one-star reviews received. A reviewer can be similarly represented by an information granule consisting of votes, five-star, four-star, ..., one-star reviews submitted. Traditionally, data mining process begins with representation of objects based on raw data from the dataset. The number of different star reviews for the businesses and by the reviewers can be easily retrieved from the yelp.com database. These can be used in data mining activities such as clustering to create the profiles of businesses and reviewers. Clustering is an unsupervised learning process that groups similar objects. It can be used to create profiles of businesses based on type of reviews received or profiles of reviewers based on types of reviews submitted. However, this only grades the businesses based on the reviews. It does not tell us how easy or hard the reviewers were. This project addresses that issue by evolving business profiles in parallel with profiles of reviewers, These profiles recursively enhance the static information obtained from the database. For example, we add the profiles of businesses that were reviewed by a reviewer in the reviewer representation. Similarly, we add the profiles

* Video link: http://www.mtriff.com/yelp/video.php

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 325–336, 2013.

of reviewers who reviewed a business in the representation of the business. This creates an indirectly recursive definition of businesses and reviewers. The meta-profiling algorithm proposed in this paper iterates through bi-directional connections between businesses and reviewers to resolve the indirect recursive representations. The resulting meta-profiles of businesses will also describe the profiles of reviewers (created as part of the integrated meta-clustering) who reviewed the businesses. On the other hand, the meta-clusters of reviewers will also contain information about the profiles of businesses who were reviewed by the reviewers from a given cluster. That means that the readers not only know how the businesses are graded, but how easy or hard the reviewers are. Similarly, if a reader chooses to follow a reviewer, she can find out how easy or hard the reviewers are as well as the popularity of the businesses graded by the reviewer. The profiles from the the clustering are further refined by additional filters that will help users focus on types of businesses. The resulting service provides a facility for users to find similar businesses/reviewers based on the grading of the business, easy/hard grading, and types of businesses. It also provides a succinct profile of a business/reviewer based on these factors, so users can put the reviews in context.

1.1 Crisp Clustering Using *k*-means

k-means clustering is one of the most popular statistical clustering techniques [3]. The objective is to assign n objects to k clusters. The process begins by randomly choosing k objects as the centroids of the k clusters. The objects are assigned to one of the k clusters based on the minimum value of the distance $d(\boldsymbol{x}_l, \boldsymbol{c}_i)$ between the object vector $\boldsymbol{x}_l$ and the cluster vector $\boldsymbol{c}_i$. The distance $d(\boldsymbol{x}_l, \boldsymbol{c}_i)$ can be the standard Euclidean distance.

After the assignment of all the objects to various clusters, the new centroid vectors of the clusters are calculated as:

$$\boldsymbol{c}_i = \frac{\sum_{\boldsymbol{x}_l \in \boldsymbol{c}_i} \boldsymbol{x}_l}{\mid \boldsymbol{c}_i \mid}, \; where \; 1 \leq i \leq k.$$

Here $\mid \boldsymbol{c}_i \mid$ is cardinality of cluster $\boldsymbol{c}_i$. The process stops when the centroids of clusters stabilize, i.e. the centroid vectors from the previous iteration are identical to those generated in the current iteration.

Quality of clustering is an important issue in application of clustering techniques to real world data. A good measure of cluster quality will help in deciding various parameters used in clustering algorithms. One such parameter that is common to most clustering algorithms is the number of clusters. Several cluster validity indices have been proposed to evaluate cluster quality obtained by different clustering algorithms[1,2]. Many of the cluster validity measures are functions of the sum of within-cluster scatter to between-cluster separation. The scatter within the ith cluster, denoted by S_i, and the distance between cluster $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$, denoted by d_{ij}, are defined as follows:

$$S_i = \left(\frac{1}{\mid \boldsymbol{c}_i \mid} \sum_{\boldsymbol{x} \in \boldsymbol{c}_i} distance(\boldsymbol{x}, \boldsymbol{c}_i) \right)^{1/q} \tag{1}$$

$$d_{ij} = distance(\boldsymbol{c}_i, \boldsymbol{c}_j) \tag{2}$$

where $\boldsymbol{c}_i$ is the center of the ith cluster. $|\boldsymbol{c}_i|$ is the number of objects in $\boldsymbol{c}_i$. $distance(\boldsymbol{x}, \boldsymbol{y})$ is the distance between two vectors. Depending upon the application, we can choose any distance function. Two popular distance functions are Euclidean distance and inverse of cosine similarity function. This study uses Euclidean distance.

We can sum up the scatter within cluster for all the clusters in a clustering scheme C as:

$$S(C) = \sum_{i=1}^{k} S_i \tag{3}$$

Similarly, between-cluster distance for a clustering scheme can be summed as:

$$D(C) = \sum_{i=1}^{k} \sum_{j=1}^{k} d_{ij} \tag{4}$$

It is advisable to plot both of these measures for the datasets under study. Usually, the scatter within cluster starts rising rapidly, while distance between cluster starts falling rapidly when the number of clusters falls below a certain value. The knee of the curves can be used as the range for determining an appropriate number of clusters. We will demonstrate this process for all the datasets used in this study.

Table 1. Static and dynamic parts of reviewer data

Reviewer ID	Static representation ($\boldsymbol{sr}$)							Dynamic representation ($\boldsymbol{dr}$)						
	Total	*	**	***	****	*****	votes	$\boldsymbol{bc_1}$	$\boldsymbol{bc_2}$	$\boldsymbol{bc_3}$	$\boldsymbol{bc_4}$	$\boldsymbol{bc_5}$	$\boldsymbol{bc_6}$	$\boldsymbol{bc_7}$
	$\boldsymbol{sr_1}$	$\boldsymbol{sr_2}$	$\boldsymbol{sr_3}$	$\boldsymbol{sr_4}$	$\boldsymbol{sr_5}$	$\boldsymbol{sr_6}$	$\boldsymbol{sr_7}$	$\boldsymbol{dr_1}$	$\boldsymbol{dr_2}$	$\boldsymbol{dr_3}$	$\boldsymbol{dr_4}$	$\boldsymbol{dr_5}$	$\boldsymbol{dr_6}$	$\boldsymbol{dr_7}$
$\boldsymbol{r_1}$	6	0	0	17	33	50	11	0	17	17	17	17	0	33
$\boldsymbol{r_3}$	9	0	22	0	33	44	6	22	0	67	11	0	0	0
$\boldsymbol{r_i}$	...	...	...	...	...	...	...	...	...	...	...	...	...	...
$\boldsymbol{r_{nr}}$	18	6	0	17	67	11	32	11	6	39	28	0	6	11

2 Recursive Profiling Algorithm

Since K-means algorithm depends on randomly selected initial centroids of the clusters, we apply the algorithm multiple times and choose a clustering scheme that has the most compact clusters. Cluster compactness and manual inspection of cluster centroids was used to determine the optimal number of clusters.

On yelp.com, a business is reviewed by many reviewers and a reviewer reviews many businesses creating a bi-directional graph. Our clustering of reviewers is going to use profiles derived from the clustering of businesses, and vice versa. Let $R = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{nr}\}$ be the set of reviewers and $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{nb}\}$ be the set of businesses. Here, $nr = 43,873$ is the number of reviewers and $nb = 11,537$ is the number of businesses in yelp.com dataset. Furthermore, let $RC = \{\boldsymbol{rc}_1, \boldsymbol{rc}_2, \ldots, \boldsymbol{rc}_{kr}\}$ be the clustering scheme of reviewers and $BC = \{\boldsymbol{bc}_1, \boldsymbol{bc}_2, \ldots, \boldsymbol{bc}_{kb}\}$ be the clustering

Table 2. Static and dynamic parts of business data

Business ID	Static representation ($\boldsymbol{sb}$)						Dynamic representation ($\boldsymbol{db}$)						
	Total	*	**	***	****	*****	$\boldsymbol{rc_1}$	$\boldsymbol{rc_2}$	$\boldsymbol{rc_3}$	$\boldsymbol{rc_4}$	$\boldsymbol{rc_5}$	$\boldsymbol{rc_6}$	$\boldsymbol{rc_7}$
	$\boldsymbol{sb_1}$	$\boldsymbol{sb_2}$	$\boldsymbol{sb_3}$	$\boldsymbol{sb_4}$	$\boldsymbol{sb_5}$	$\boldsymbol{sb_6}$	$\boldsymbol{db_1}$	$\boldsymbol{db_2}$	$\boldsymbol{db_3}$	$\boldsymbol{db_4}$	$\boldsymbol{db_5}$	$\boldsymbol{db_6}$	$\boldsymbol{db_7}$
$\boldsymbol{b}_1$	67	7	10	12	58	12	3	2	18	5	59	6	8
$\boldsymbol{b}_2$	4	0	0	25	25	50	50	0	0	0	25	0	25
$\boldsymbol{b}_i$	...	...	...	...	...	...	...	...	...	...	...	...	...
$\boldsymbol{b}_{nb}$	18	0	11	11	61	17	13	0	0	0	73	0	13

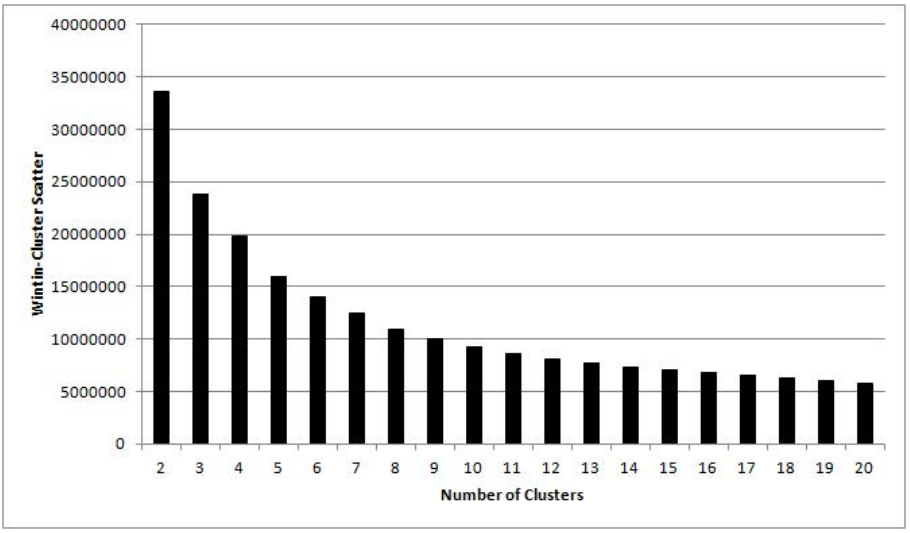


Fig. 1. Plot of within cluster scatter for businesses

scheme of businesses. After studying the compactness of clusters shown in Fig. 2 and resulting centroids, it was decided that the number of reviewer clusters $kr = 7$. The knee of the curve for the scatter within cluster shows that the scatter starts rising rapidly after $kr = 11$. The increase in the scatter intensifies after $kr = 7$. Similarly, the number of business clusters based on the knee of the curve for scatter within clusters shown in Fig. 1 was decided to be 7, i.e. $kb = 7$, as 7 is just about the middle of the knee of the curve.

The reviewer $\boldsymbol{r}_j$ is represented by a static data part $\boldsymbol{sr}_j$ and dynamic data part $\boldsymbol{dr}_j$, i.e. $\boldsymbol{r}_j = (\boldsymbol{sr}_j, \boldsymbol{dr}_j)$ as shown in Table 1. Here, $\boldsymbol{sr}_j$ is the data that are extracted from the raw dataset such as types of reviews (total, *,**,***,****,*****,votes). The dynamic part $\boldsymbol{dr}_j$ will be derived from the clustering of businesses. We will represent $\boldsymbol{dr}_j = (m_{j1}, m_{j2}, \cdots, m_{jkb})$, where m_{ji} is the normalized count of businesses that the reviewer $\boldsymbol{r}_j$ reviewed that falls in $\boldsymbol{bc_i}$ cluster of businesses.

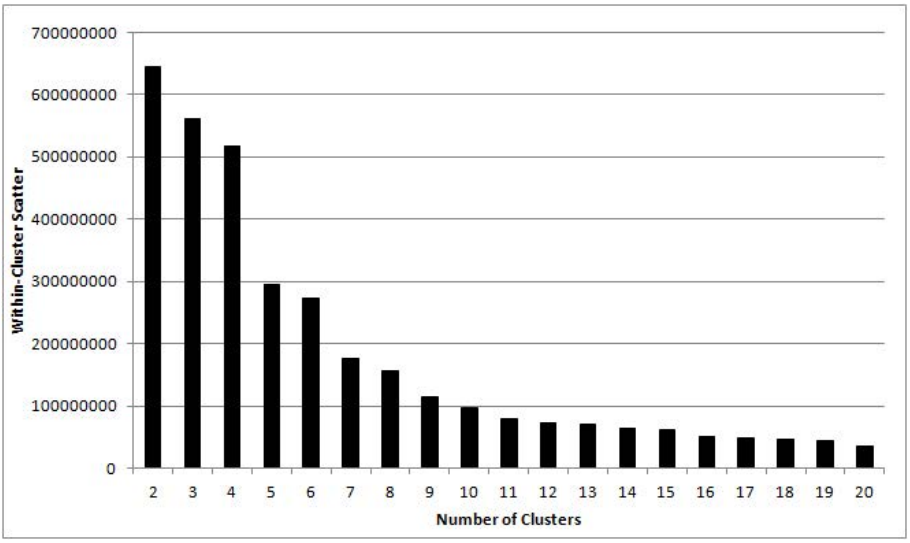


Fig. 2. Plot of within cluster scatter for reviewers

1. Cluster the businesses using their static representations, i.e. a business $\boldsymbol{b}_j = \boldsymbol{sb}_j$, where $\boldsymbol{sb}_j$ is a vector of attribute values retrieved from the transaction data set.
2. Calculate the dynamic representation $\boldsymbol{dr}_j$ of a reviewer $\boldsymbol{r_j}$:

$$\boldsymbol{dr}_j = (m_{j1}, m_{j2}, ..., m_{jkb}), \tag{5}$$

   where m_{ji} is the count of businesses from cluster $\boldsymbol{bc_i}$ who were reviewed by reviewer$\boldsymbol{r_j}$ in the previous clustering of businesses.
3. Cluster the reviewers with concatenation of static and dynamic representations, i.e. a reviewer $\boldsymbol{r}_j = (\boldsymbol{sr}_j, \boldsymbol{dr}_j)$,
4. Calculate the dynamic representation $\boldsymbol{db}_j$ of a business c_j :

$$\boldsymbol{db}_j = (m_{j1}, m_{j2}, ..., m_{jk}), \tag{6}$$

   where m_{ji} is the count of reviewers from cluster $\boldsymbol{rc_i}$ who reviewed business $\boldsymbol{b_j}$ in the previous clustering of reviewers.
5. Cluster the businesses with concatenation of static and dynamic representations, i.e. a reviewer $\boldsymbol{b}_j = (\boldsymbol{sb}_j, \boldsymbol{db}_j)$,
6. If the values of $\boldsymbol{db}_j$ for a business $\boldsymbol{b}_j$ or the values of $\boldsymbol{dr}_i$ for a reviewer $\boldsymbol{r}_i$ have changed, go back to step 2.

Fig. 3. Proposed indirectly recursive meta-clustering algorithm

Similiarly, the business $\boldsymbol{b}_i$ will be represented by a static data part $\boldsymbol{sb}_i$ and dynamic data part $\boldsymbol{db}_i$, i.e. $\boldsymbol{b}_i = (\boldsymbol{sb}_i, \boldsymbol{db}_i)$ as shown in Table 2. The static part $\boldsymbol{sb}_i$ will represent the number of reviews (total, *,**,***,****,*****) received for the business from the raw dataset, while the dynamic part $\boldsymbol{db}_i$ will be derived from the clustering of reviewers. That is, $\boldsymbol{db}_i = (m_{i1}, m_{i2}, \cdots, m_{ikr})$, where m_{ij} is the normalized count of reviewers that reviewed the business $\boldsymbol{b}_i$ that falls in $\boldsymbol{rc_j}$ cluster from the clustering of reviewers. Since we do not have any clustering results at the beginning of the iterative process, we first cluster businesses based solely on the static part. The subsequent clustering of both businesses and reviewers will use the static and dynamic representations. The iterations will stop when values of the dynamic parts $\boldsymbol{dr}_j$ and $\boldsymbol{db}_i$ for all objects stabilize. Fig. 3 provides the formal description of the proposed iterative meta-clustering algorithm.

3 Experiments with the Yelp.com Dataset

The dataset from the Yelp dataset challenge consisted of 11,537 businesses, 43,837 users, and 229,907 reviews for Phoenix, Arizona. It was decided to represent each business based on the total number of reviews, and percentage of reviews from each rating category going from one-star to five-star. Similarly, a reviewer was also represented by total number of reviews, and percentage of reviews from each rating category (one-star to five-star) were submitted by the reviewer as well as the number of votes received by the reviewer. The above-mentioned static information was retrieved directly from the Yelp dataset. As mentioned earlier, this static information was supplemented by the connections between the reviewers and the businesses. For each business, we created a list of reviewers who reviewed the business. The categorization of these reviewers made up the dynamic representation of the business as shown in Table 2. Similarly, we created a list of businesses reviewed by each reviewer. This list was used to create categorization of businesses that went into the dynamic representation of the reviewer as shown in Table 1. A statistical summary of static data obtained from the yelp.com dataset is shown in Table 3 for businesses and in Table 4 for reviewers. The statistical summary suggests a skewed distribution. Based on this, as well as Fig. 1 and Fig. 2, we chose to increase the number of clusters from five to seven. This allowed for separation of some of the higher value objects into small clusters of their own.

The parallel meta-clustering, described in Fig. 3, created the meta-centroids for various categories of businesses and reviewers. The data extraction and preparation was

Table 3. Summary of static business data

Measure	Total	*	**	***	****	*****
Min.	3	0	0	0	0	0
1st Qu.	4	0	0	0	6.25	12.2
Median	6	3.509	0	10.43	30	32.97
Mean	19.93	12.718	9.123	13.91	28.32	35.93
3rd Qu.	16	20	15.094	25	41.28	55.26
Max.	844	100	100	100	100	100

Table 4. Summary of static reviewer data

Measure	Total	*	**	***	****	*****	votes
Min.	1	0	0	0	0	0	0
1st Qu.	1	0	0	0	0	0	0
Median	2	0	0	0	0	33.33	2
Mean	4.921	12.76	8.598	9.523	26.56	42.57	14.56
3rd Qu.	4	0	0	0	50	100	5
Max.	588	100	100	100	100	100	14933

Table 5. Centroids from iterative meta-clustering of business data

Cluster ID	Static representation (sb)						Dynamic representation (db)							Size
	Total sb_1	* sb_2	** sb_3	*** sb_4	**** sb_5	***** sb_6	rc_1 db_1	rc_2 db_2	rc_3 db_3	rc_4 db_4	rc_5 db_5	rc_6 db_6	rc_7 db_7	
bc_1	12	16	19	31	22	12	6	1	3	1	69	10	10	2510
bc_2	11	6	5	7	22	61	21	1	5	1	60	6	6	2644
bc_3	101	6	9	16	37	31	3	0	20	2	65	4	6	852
bc_4	13	6	7	12	55	20	8	1	3	1	75	5	8	2764
bc_5	5	6	3	2	6	83	63	0	6	1	20	7	3	1449
bc_6	5	51	11	9	14	15	11	1	5	2	34	42	6	1214
bc_7	335	3	6	14	39	38	4	0	7	22	60	2	5	104

performed using a number of Python scripts. The meta-clustering algorithm was implemented using a UNIX bash script that iteratively called an R program for clustering and a Python program for creating dynamic representations of clustering. K-means clustering used 1000 iterations and was applied 10 times to get compact clusters. The dynamic representation seemed to stabilize after 21 iterations of indirectly recursive meta-clustering. On a high performance computing cluster provided by ace-net.ca the meta-clustering took only one minute.

The business clusters are shown in Table 5 and the meta-centroids of the reviewer clusters are shown in Table 6. The last column in each table shows the size of each cluster. The resulting business profiles are more refined than conventional clustering process as they use associations with the profiles of the reviewers that reviewed the business. We can describe these enhanced profiles as follows:

- bc_1 **Ambivalently rated even by softies** - Modest number of evenly spread reviews, most coming from rc_5, which tends to give mostly four stars reviews.
- bc_2 **Well rated by softies** - Modest number of reviews mostly five and four stars, most coming from rc_5 and rc_1, which tend to give mostly four and five stars reviews.
- bc_3 **Well rated by balanced reviewers** - Large number of reviews mostly four and five stars with noticeable three stars, most coming from rc_5 (gives mostly four stars) and rc_3 (capable of giving two and three stars).
- bc_4 **Reasonably rated by mostly softies** - Modest number of reviews mostly four and five stars, most coming from rc_5, which tends to give mostly four stars reviews.

Table 6. Centroids from iterative meta-clustering of reviewer data

Cluster ID	Static representation (sr)						
	Total sr_1	* sr_2	** sr_3	*** sr_4	**** sr_5	***** sr_6	votes sr_7
rc_1	2	2	2	2	4	91	2
rc_2	443	2	6	26	46	20	13074
rc_3	3	4	13	12	6	65	4
rc_4	2	6	8	10	29	47	3
rc_5	10	3	6	12	64	15	25
rc_6	2	63	18	12	3	5	2
rc_7	180	5	9	23	39	24	2235

Cluster ID	Dynamic representation (dr)							Size
	bc_1 dr_1	bc_2 dr_2	bc_3 dr_3	bc_4 dr_4	bc_5 dr_5	bc_6 dr_6	bc_7 dr_7	
rc_1	9	34	6	16	29	3	3	8922
rc_2	19	17	32	16	2	4	10	2
rc_3	2	3	84	3	1	1	5	8805
rc_4	2	2	7	2	1	1	84	5064
rc_5	12	11	42	21	1	2	10	14094
rc_6	26	12	21	12	4	23	2	6908
rc_7	17	12	36	18	1	2	13	78

bc_5 **Sparsely but very well rated by softies** - Fewest number of reviews mostly five stars, most coming from rc_1 and rc_5, which tend to give mostly four and five stars reviews.

bc_6 **Sparsely and lowly rated by both hard and soft groups** - Fewest number of reviews mostly one and two stars, most coming from rc_6 (gives one stars reviews) and rc_5 (gives four stars reviews).

bc_7 **Reasonably rated by many softies** - Largest number of reviews mostly four and five stars with noticeable three stars, most coming from rc_5 (gives mostly four stars) and rc_4 (gives mostly five and four stars).

The association of reviewer information with business cluster is inversely applicable to the reviewer profiles, which are refined using the profiles of the businesses who are reviewed by these reviewers. These augmented reviewer profiles can be described as:

rc_1 **Infrequent and very soft, cover the spectrum** - Very few and almost exclusively five star reviews, spread evenly across most business clusters.

rc_2 **Extremely prolific and balanced, do not cover most and least popular** This group of two is essentially an outlier with a large number of reviews and votes, and these users should be treated separately as prolific reviewers. Their reviews are mostly four, three, and five stars. They do not have too many reviews for bc_5 and bc_6 (which do not receive too many reviews), and bc_7 which receive most reviews.

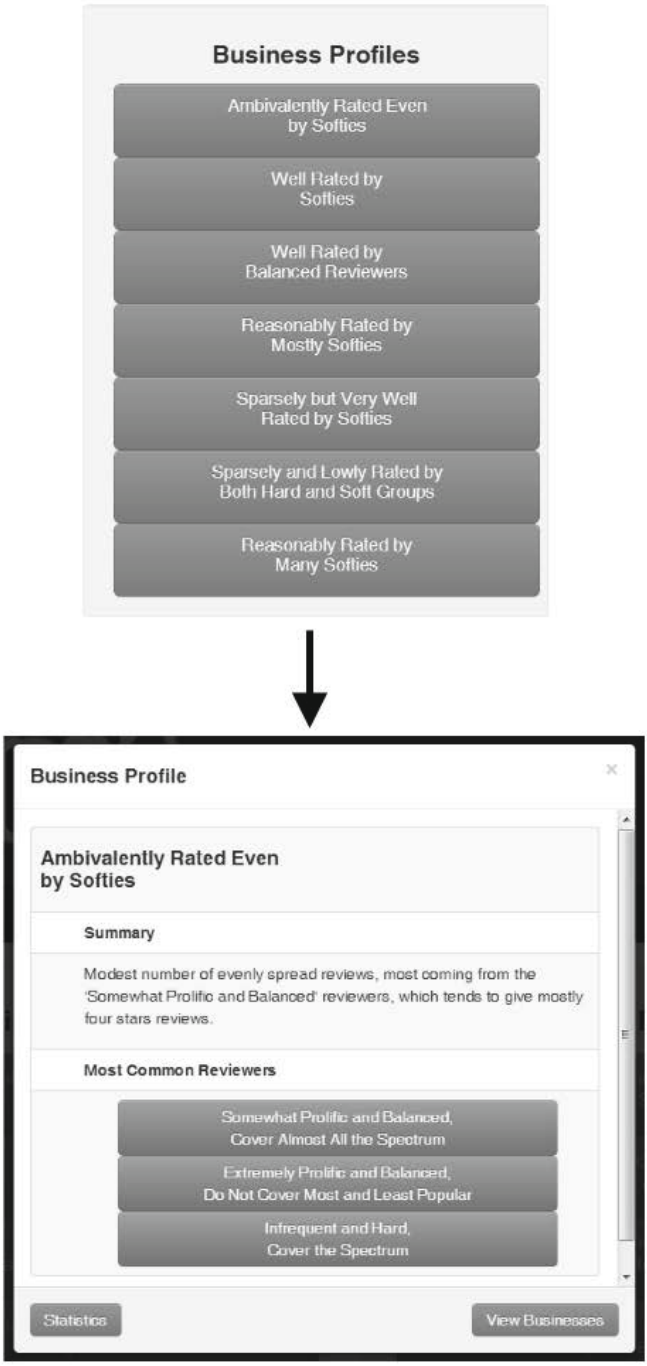


Fig. 4. Clicking on a business profile label brings up the profile

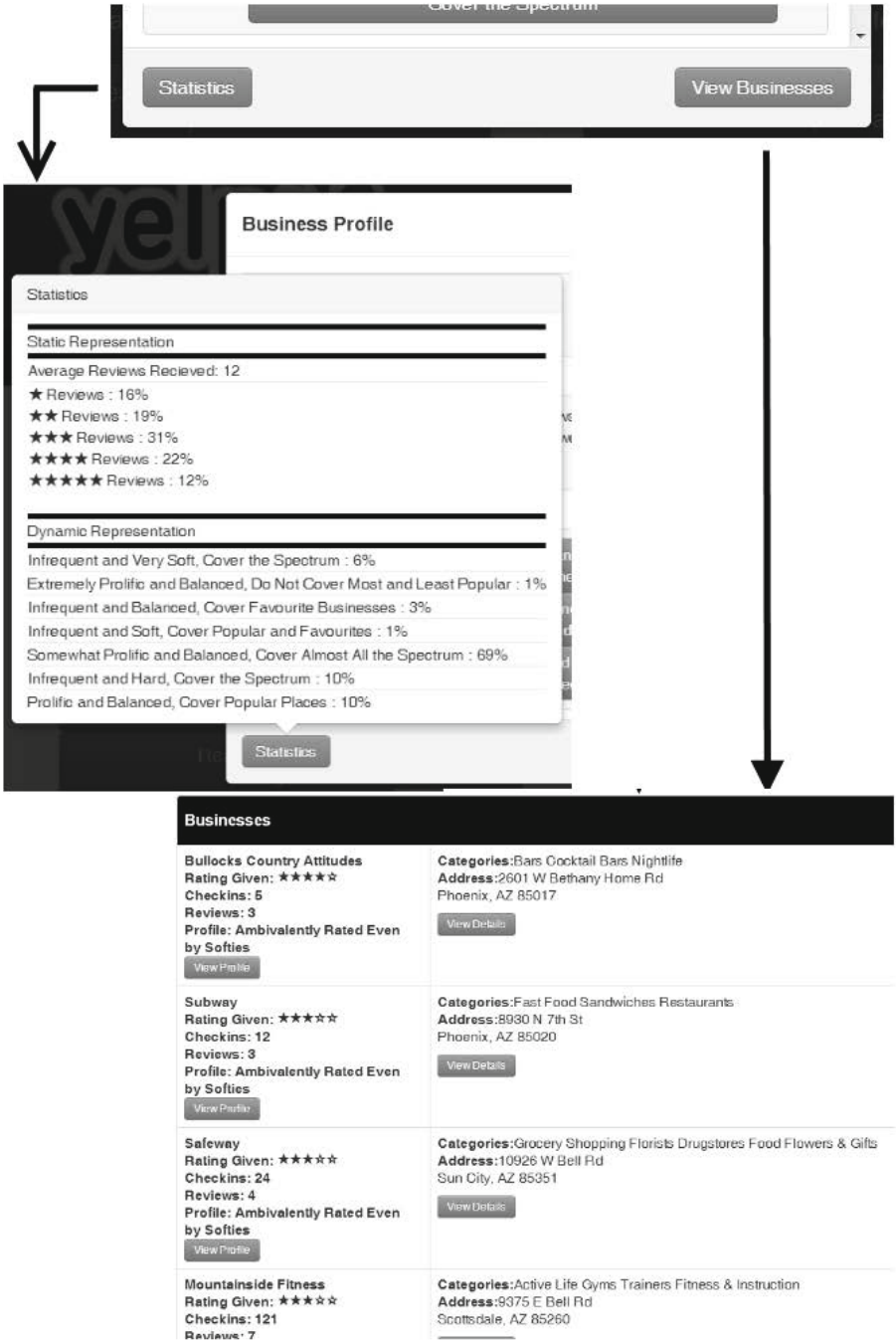


Fig. 5. Displaying statistics and list for a business profile

rc_3 **Infrequent and balanced, cover favourite businesses** - Very few and mostly five star reviews, but fair amount of two and three stars as well. Most reviews are for bc_3, which has large number of four and five stars.

rc_4 **Infrequent and soft, cover popular and favourites** - Very few and mostly five and four star reviews. Almost all reviews are for bc_7, which has very large number of four and five stars.

rc_5 **Somewhat prolific and balanced, cover almost all the spectrum** - Modest number of reviews and votes: mostly four, five, and three stars. They do not have too many reviews for bc_5 and bc_6 (which do not receive too many reviews).

rc_6 **Infrequent and hard cover the spectrum** -Very few and mostly one and two star reviews. The reviews are spread evenly across most business clusters.

rc_7 **Prolific and balanced, cover popular places** - Large number of reviews and votes are mostly four, three, and five stars. They do not have too many reviews for bc_5 and bc_6 (which do not receive too many reviews).

4 How Yelp.com Can Use These Results

We have created a website that provides a facility for users to find businesses/reviewers based on the grading of the business, easy/hard grading, and types of businesses. It also provides a succinct profile of a business/reviewer based on these factors, so users can put the reviews in context. Fig. 3 shows a collage of screenshots of business profile labels and a business profile that will appear by clicking on a label. Fig. 3 shows the statistics for the business profile and the list of business corresponding to a given profile. The interface for browsing reviewer profile is similar.

A reader can also choose to follow a group of reviewers which have the same profile or look at reviews of all the businesses with similar profile. In such a case, the profiles described in the previous section are further filtered using the types of businesses. The meta-profiles of businesses describe the profiles of reviewers who reviewed the businesses. Similarly, the meta-profiles of reviewers contain the profiles of businesses who were reviewed by the reviewers. That means that the readers not only know how the businesses are graded, but how easy or hard the reviewers are.

A video: **http://www.mtriff.com/yelp/video.php** demonstrates the service. The website lists the profile labels of businesses and reviewers. Users can click on the profiles that interest them, which brings down sliding windows that show more information about the profile and gives a list of business categories for each profile. Depending on the business category chosen, a preview of reviews from the list appears for browsing. A similar facility is provided for the reviewer profiles.

5 Summary and Conclusions

This paper describes a novel meta-clustering algorithm that operates in a granular network consisting of businesses and reviewers on a site such as yelp.com. The clustering of businesses and reviewer evolves in parallel.

The profiles of businesses that are evolved in the process are used in the representation of reviewers. Similarly, the profiles of reviewers are used in the representation

of business. This leads to an indirect recursion in the object representation. The recursive profiles created through meta-clustering not only describe the characteristics of a business but those of the reviewers who reviewed the business. Similarly, the reviewer profiles combine characteristics of the reviewer and the businesses they review.

The proposed meta-clustering allows the readers to put the reviews in context. The readers can tell how strict or easy and prolific the reviewers who reviewed a business are. The readers can also choose to follow the group of reviewers that follow certain types of businesses.

The paper is supplemented by a website **http://www.mtriff.com/yelp/** and video **http://www.mtriff.com/yelp/video.php** to demonstrate the effectiveness of our approach.

References

1. Davies, D.L., Bouldin, D.W.: A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1, 224–227 (1979)
2. Lingras, P., Chen, M., Miao, D.: Rough Cluster Quality Index Based on Decision Theory. IEEE Transactions on Knowledge and Data Engineering 21(7), 1014–1026 (2009)
3. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceeding of the Fifth Berkeley Symposium on Math., Stat. and Prob., pp. 281–296 (1967)

An Illustrative Comparison of Rough k-Means to Classical Clustering Approaches

Georg Peters[1] and Fernando Crespo[2]

[1] Munich University of Applied Sciences, Munich, Germany &
Australian Catholic University, North Sydney, Australia
georg.peters@cs.hm.edu
[2] Universidad de Valparaíso, Santiago, Chile
fernando.crespo@uv.cl

Abstract. Rough clustering has gained increasing attention in the last decade with applications in such diverse areas like bioinformatics, traffic control and retail. The relationship between rough clustering and, in particular, fuzzy and possibilistic concepts is still a topic that is raised first and foremost by practitioners who are looking for an adequate clustering algorithm. Therefore, we compare rough k-means to fuzzy c-means, possibilistic c-means and to classical k-means in our paper. We show that rough k-means is closer related to classical k-means than to fuzzy and possibilistic c-means. Besides brief theoretical evaluations we perform illustrative experiments on artificial data and the IRIS data.

Keywords: k-Means, Rough k-Means, Soft Clustering.

1 Introduction

Clustering algorithms are widely used in data mining. Probably the most well-known approach is the k-means algorithm [4,8] which assigns objects unambiguously to clusters. However, many real life applications are characterized by ambiguous situations and vagueness. To address such situations soft clustering algorithms have been proposed. Prominent examples are Bezdek's fuzzy c-means (FCM) [1,2] and Krishnapuram and Keller's possibilistic c-means (PCM) [5]. In 2004, Lingras and West suggested rough k-means (RKM) [7] as a further soft clustering approach. For a survey on rough clustering the reader is referred to Lingras and Peters [6]. Although its relationship to established hard and soft clustering approaches has been discussed rough clustering is still challenged by the following questions: (1) what are its differences to hard, fuzzy or possibilistic clustering and (2) when should rough clustering been used. Peters et al. [11] provided an overview on soft clustering. In contrast to this we provide a perspective from rough clustering and treat possibilistic clustering pari passu to hard and fuzzy clustering. The paper is organized as follows. In Section 2 we discuss some fundamental differences between these hard and soft clustering approaches. In the subsequent section we present illustrative examples comparing RKM to hard, fuzzy and possibilistic clustering approaches. A discussion and summary section concludes the paper.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 337–344, 2013.

2 Soft Clustering

Soft clustering algorithms like FCM, PCM and RKM can be considered as generalizations of the classical k-means (see Fig. 1). Fuzzy and possibilistic clustering can be regarded as one sub-family of hard k-means generalizing its binomial degrees of similarity $\{0, 1\}$ to continuous values in $[0, 1]$ intervals.

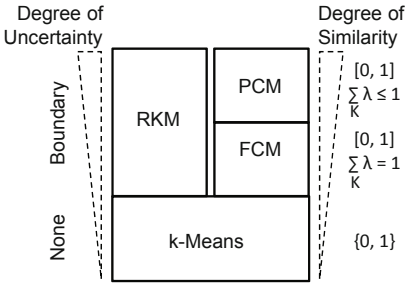


Fig. 1. Generalizations of the k-Means

Rough clustering forms another sub-family by adding a uncertainty dimension to classical k-means. As in classical k-means objects surely belong to a cluster (positive region of a cluster) or they surely do not belong to the cluster (its negative region). Beyond these two regions the boundary in rough clustering is characterized by objects that only may belong to a cluster. Hence, the binomial membership degrees of hard k-means are generalized towards trinomial memberships (see Tab. 1).

Table 1. Range of the Membership Degrees

Algorithm	Membership		
	Valence	Grades $\lambda_{i,k}$	Restrictions for an Object i
k-Means	Bivalent	$\lambda_{i,k} \in \{0, 1\}$	$\sum_{k=1}^{K} \lambda_{i,k} = 1$
RKM	Trivalent	$\underline{\lambda}_{i,k}, \widehat{\lambda}_{i,k} \in \{0, 1\}$	$\left(\sum_{k=1}^{K} \underline{\lambda}_{i,k} = 1 \wedge \sum_{k=1}^{K} \widehat{\lambda}_{i,k} = 0\right) \vee \left(\sum_{k=1}^{K} \underline{\lambda}_{i,k} = 0 \wedge \sum_{k=1}^{K} \widehat{\lambda}_{i,k} \geq 2\right)$
FCM	Continuous	$\lambda_{i,k} \in [0, 1]$	$\sum_{k=1}^{K} \lambda_{i,k} = 1$
PCM	Continuous	$\lambda_{i,k} \in [0, 1]$	$\sum_{k=1}^{K} \lambda_{i,k} \leq 1$

3 Experimental Evaluations

3.1 Experiments on Artificially Generated Data

Description of the Artificially Generated Data. In our comparative study we use four artificially generated two-dimensional data sets which can be depicted in figures and descriptively interpreted. The four data sets consist of 30 objects each (see Fig. 2). Each of them has been designed to supposedly address the specific strengths of each of the four algorithms. They have the following characteristics:

- *Data Set 1 (ADS1)* - Two separated clusters (Fig. 2:UpperLeft). The data set is designed for the k-means which supposedly performs well when clusters are clearly separated.
- *Data Set 2 (ADS2)* - Two separated clusters with two objects "in-between" (Fig. 2:UpperRight). The data set is designed for RKM which provides a buffer zone for objects between clusters.
- *Data Set 3 (ADS3)* - Two overlapping clusters (Fig. 2:LowerLeft). The data set is designed for the FCM which supposedly performs well when clusters are overlapping.
- *Data Set 4 (ADS4)* - Two separated clusters and additionally one outlier (Fig. 2:LowerRight). The data set is designed for PCM which supposedly performs well in the presence of outliers.

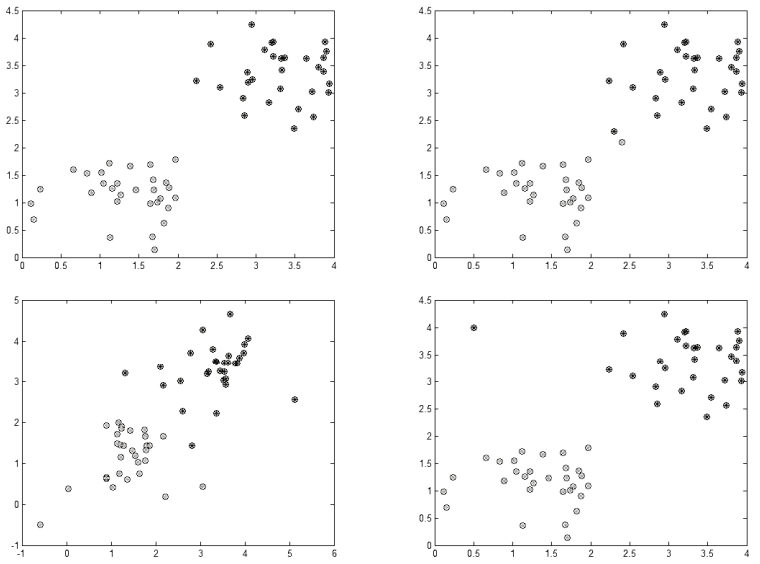


Fig. 2. Data Sets

Note, the difference between clusters with objects in-between (like ADS2) and overlapping clusters (ADS3) is difficult to define precisely since the transition between these data constellations is blurry. As will be seen, our experiments provide similar results for these data sets.

Results of the Experiments. In our experiments, we apply the refined version of RKM as suggested in [10]. We obtain the following results in our experiments.

ADS1 - Separated Clusters. This data set was designed to supposedly suite the k-means best.

- *k-Means.* As assumed k-means performs well on data set ADS1. Not surprisingly it assigns all objects to the correct clusters.
- *Rough k-Means.* In the case of RKM we obtain an empty boundary. All objects are assigned to the lower approximations of the two clusters. Hence, the results are identical to those obtained by k-means.
- *Fuzzy c-Means.* All objects have membership degrees higher than 0.8 to their most similar clusters. This indicates well separated clusters. Defuzzification leads to identical results as obtained by k-means.
- *Possibilistic c-Means.* Due the relaxed constraint regarding the memberships of PCM the obtained membership degrees are generally lower than in fuzzy clustering. The sum of the memberships degrees goes even down to 0.2 for some objects. However, when we analyze the relative memberships of the objects ($\lambda_{i,1}/\lambda_{i,2}$) similar results as obtained by k-means can be derived by defuzzification.

All clustering algorithms perform well on ADS1. FCM and PCM provide continuous membership degrees; for a hard decision defuzzification is required. k-means and RKM deliver identical hard classifications of the objects; these two algorithms are the preferred choice for the analysis of non-overlapping clusters.

ADS2 - Clusters with Objects In-Between. This data set was designed to supposedly suite RKM best.

- *k-Means.* The k-means algorithm separates the clusters as depicted in Figure 3:UpperLeft.[1] However, classifying the objects this way seems to be not optimal for the objects close to the separating line since these objects are similar to both clusters, a fact that cannot be disclosed by the k-means results.
- *Rough k-Means.* The approximations obtained by RKM are shown in Figure 3:UpperRight. Two objects are in the boundary region of both sets indicating that they cannot be assigned clearly to one or the other cluster. If required, the boundary could be expanded and would contain a higher number of objects then.

[1] Note, the separating lines in all figures do not represent actual boarder lines. The displayed lines are for demonstration purposes only.

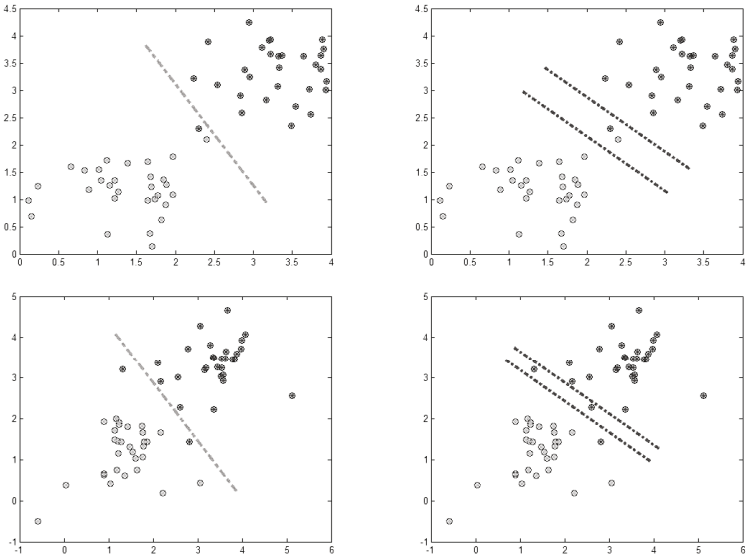


Fig. 3. Some Clustering Results

- *Fuzzy c-Means.* An identical result to the one by k-means is obtained when the fuzzy partition is defuzzified. However, the membership degrees clearly show how representative the objects are for a particular cluster. While the objects close the separating line have similar membership degrees to both clusters, objects in the center region of a cluster have high membership degrees to this cluster and low degrees to the other.
- *Possibilistic c-Means.* Due the relaxed constraint regarding the memberships of PCM the obtained membership degrees are lower than in fuzzy clustering. However, results are comparable to those provided by FCM.

The results confirm that FCM as well as PCM are adequate methods for clusters with objects in-between. The k-means algorithm struggles to deliver intuitive results. RKM appears to be a good compromise balancing the needs for hard decisions (k-means) and continuous membership degrees (FCM and PCM). It distinguishes between objects that surely belong to a cluster and objects that are similar to more than one cluster.

ADS3 - Overlapping Clusters. This data set was designed to supposedly suite FCM best. The results equal the results obtained for the ADS2 data set (objects in-between). The main difference is that the number of objects in-between has grown in comparison to ADS2 so that the clusters clearer overlap. Hence, we discuss the results only briefly here. In k-means the objects in the overlapping region of the clusters are assigned to the lower left cluster (Fig. 3:LowerLeft). For RKM the results are depicted in Figure 3:LowerRight. Two of the objects in the

overlapping region of the clusters are assigned to the boundary region of both sets. Depending on the selection of the initial parameters the number of objects in the boundary may vary. The results of rough clustering show the unclear membership status of these objects. Using FCM, objects in the center of the overlapping region get similar membership degrees to both clusters indicating their status between the clusters. After defuzzification these objects belong to the lower left cluster like in k-means. Basically the same applies to the PCM.

ADS4 - Clusters with an Outlier. This data set was designed to supposedly suite PCM best.

- *k-Means.* In k-means the outlier object is assigned to the upper right cluster practically ignoring it as an outlier.
- *Rough k-Means.* In RKM the outlier is the only object assigned to the boundary region of both clusters showing its status as an object with unclear memberships. See Peters [9] for more on outliers in RKM.
- *Fuzzy c-Means.* After defuzzification the results of FCM are similar to those provided by k-means. Now, the outlier belongs to the lower left cluster. However, analyzing the results in detail shows that the outlier has almost identical membership degrees to both clusters. This is an indicator that the object lies in-between the clusters or is an outlier.
- *Possibilistic c-Means.* The sum of the outlier's membership degrees to both clusters is significantly below 1 indicating it as an outliers. Hence, PCM performs well in the presence of outliers.

Possibilistic c-means performs well for data that contain outliers. FCM also indicates outliers by assigning them similar membership degrees. However, it does not have the capability to distinguish between outliers and objects in overlapping regions like PCM has. The result provided by k-means is not convincing since it does not identify the outlier. RKM assigns the outlier - and only the outlier - to the boundary region indicating its unclear membership. Hence, RKM is a good enhancement of k-means in the presence of outliers. However, a drawback still remains: like FCM RKM does not distinguish between outliers and objects in overlapping regions.

3.2 Experiments on the IRIS Data

Description of the IRIS Data. Now we analyze the relationship of k-means and RKM in more details. Therefore, we exclude FCM and PCM from the experiments. We apply k-means and RKM to the IRIS data [3]. This data set is suitable for our evaluation since one class (IRIS1) is separated from the remaining two (IRIS2, IRIS3) and these two classes overlap. Therefore, the IRIS data provide similar characteristics like our artificially generated data sets ADS1 (separated clusters) and ADS2 (objects in-between) and ADS3 (overlapping clusters). Only ADS4 (data with an outlier) cannot be represented by the IRIS data. In all experiments the number of clusters is set to $K = 3$ (clusters CL1, CL2, and CL3).

Results of the Experiments. As reference we perform k-means first (experiment E0). Then we perform 11 experiments E1, ..., E11 on RKM with increasing thresholds from $\zeta = 1.0$ to 2.0 in 0.1 steps. The weight of the lower approximation is set to $\underline{w} = 0.9$. The results of E0, E1 ($\zeta = 1.0$), E3 ($\zeta = 1.2$), and E11 ($\zeta = 2.0$) are presented in more detail in the following paragraphs.

Experiment E0: k-Means. Regarding the separated class IRIS1 no classification error is observed. For the overlapping classes IRIS2 and IRIS3 we observe 16 classification errors. Two objects of IRIS2 are wrongly assigned to cluster CL3 and 14 objects of class IRIS3 are assigned to CL2.

Experiment E1: Rough k-Means ($\zeta = 1.0$). In experiment E1, an empty boundary region is obtained, i.e., RKM melts down to k-means.

Experiment E3: Rough k-Means ($\zeta = 1.2$). The results for experiment E3 are summarized in Table 2. In comparison to experiment E1 the initial settings have been relaxed so that the boundary region is larger than in E1. Hence, the boundary region functions as a buffer zone for the objects between the clusters CL2 and CL3. As a trade-off a lower number of objects has been assigned to the correct clusters. The clustering results are summarized in Table 2 (with ✓ (↯) objects assigned to correct (wrong) clusters/lower approximations and ∼ objects assigned to boundary regions).

Experiment E11: Rough k-Means ($\zeta = 2.0$). RKM converges towards similar solutions for $\zeta \geq 2.0$. Hence, we will briefly discuss the results for $\zeta = 2.0$. Only 4 objects are correctly classified, while 144 objects are in the boundary region.

Table 2. Summarized Clustering Results for E3

k-Means	Rough k-Means	# Objects
✓	✓	131
✓	∼	3
↯	∼	3
↯	↯	13

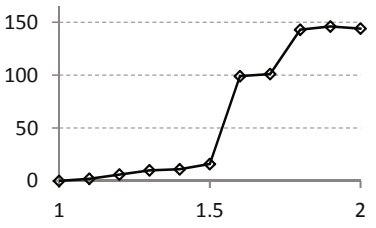


Fig. 4. Number of Objects in Boundary Regions (IRIS)

Obviously, the given parameters lead to an oversized buffer region. Figure 4 displays the number of objects in boundary regions for the eleven experiments.

4 Discussion and Conclusion

The following considerations play an important role to understand the differences between RKM and alternative clustering algorithms analyzed in this paper. RKM is regarded as a soft computing clustering algorithm like fuzzy and possibilistic c-means. However, our experiments show that the results of k-means and RKM are identical for separated clusters. For data sets with overlapping classes RKM provides a useful buffer zone between the clusters. Outliers are assigned to the clusters' boundaries indicating that they are not belonging to a certain cluster. We regard these characteristics as useful enrichments of classical k-means. While fuzzy and possibilistic clustering form one sub-family (adding *ambiguity based on similarity*) of hard k-means rough clustering establishes a second sub-family (adding *uncertainty due to missing or wrong information*). So, RKM is closely related to k-means and only linked via k-means to fuzzy and possibilistic c-means.

References

1. Bezdek, J., Harris, J.: Fuzzy partitions and relations. Fuzzy Sets and Systems 1(2), 111–127 (1978)
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Algorithms. Plenum Press, New York (1981)
3. Frank, A., Asuncion, A.: UCI Machine Learning Repository (2010)
4. Jain, A.: Data clustering: 50 years beyond k-means. Pattern Recognition Letters 31, 651–666 (2010)
5. Krishnapuram, R., Keller, J.M.: A possibilistic approach to clustering. IEEE Transactions on Fuzzy Systems 1(2), 98–110 (1993)
6. Lingras, P., Peters, G.: Rough clustering. WIREs Data Mining and Knowledge Discovery 1, 64–72 (2011)
7. Lingras, P., West, C.: Interval set clustering of web users with rough k-means. Journal of Intelligent Information Systems 23, 5–16 (2004)
8. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley & Los Angeles, CA, vol. I, pp. 281–297. Univ. of California Press (1967)
9. Peters, G.: Outliers in rough k-means clustering. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 702–707. Springer, Heidelberg (2005)
10. Peters, G.: Some refinements of rough k-means. Pattern Recognition 39, 1481–1491 (2006)
11. Peters, G., Crespo, F., Lingras, P., Weber, R.: Soft clustering - fuzzy and rough approaches and their extensions and derivatives. International Journal of Approximate Reasoning 54(2), 307–322 (2013)

A Kidnapping Detection Scheme Using Frame-Based Classification for Intelligent Video Surveillance

Ryu-Hyeok Gwon, Kyoung-Yeon Kim, Jin-Tak Park,
Hakill Kim, and Yoo-Sung Kim

Department of Information & Communication Engineering, Inha University, Korea
hotklh@hanmail.net, yskim@inha.ac.kr

Abstract. The purpose of this study is to develop a kidnapping event detection scheme for intelligent video surveillance by frame-based classification which is able to assort each frame into a kidnapping or normally accompanying situation. In this study, for generating training data from videos, a semi-automatic video annotation tool named INHA-VAT is used. Also, we developed a frame-based event classifier using Bayesian network model to distinguish the frame of kidnapping situations from one of accompanying ones. When a video has more frames of kidnapping situation than the threshold ratio after two people meet in the video, the proposed scheme detects and notifies the occurrence of kidnapping event. To check the feasibility of the proposed scheme, we also performed the accuracy evaluation against test videos. According to the experiment results, the proposed scheme could detect kidnapping situations appropriately according to the threshold ratio.

Keywords: Kidnapping detection, intelligent video surveillance, frame-based event classification, Bayesian network, discriminative features.

1 Introduction

Recently, CCTVs(closed-circuit televisions) are widely used for security purposes, especially for obtaining criminal evidence. Although there have been some controversies about the privacy problem concerned with CCTVs, the usefulness of CCTVs is above suspicion and proved by actual cases where CCTVs were successfully used for solving criminal events. Therefore it seems that the demand of CCTV is continuously increased since these positive cases highlight the need of CCTV [1].

Nowadays, in most cases, one human operator monitors several CCTVs alone [2]. But, according to previous studies, one operator could effectively monitor only up to 16 cameras at once and this number varied depending on the complexity of screen layout [3]. Also, according to the experimental results of previous studies, operator who monitors two or more CCTVs at the same time might miss 45% of dangerous situations after 12 minutes and 95% after 22 minutes, respectively [4]. It shows the critical limitation of real-time manual monitoring of CCTVs by human operators. Furthermore, current CCTV systems are mainly used only for obtaining evidences since they simply record videos to storage devices, and monitoring or searching

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 345–354, 2013.

functions are done manually. Another drawback of the current CCTV systems is that they are unable to recognize unusual events automatically in real-time from videos.

To overcome these limitations of real-time CCTV systems, previous studies have suggested several real-time automatic recognition methods of human activities. However, they are only available to detect simple activities such as passing through a door or roaming around a shop, not to detect complex human interactions like kidnapping, accompanying, or fighting.

In this study, to develop a kidnapping detection scheme for intelligent video surveillance system, we created several scenes of kidnapping situations and accompanying ones, respectively and record them as sample video data. Also, as the ground truth data for training and testing of frame-based event classification scheme, we manually extracted Region of Interest (ROI) for human object from the sample videos by using INHA-VAT [5]. We also developed a frame-based event classifier which can assort each frame into kidnapping or accompanying situation with respect to the discriminative features related to the variation patterns of ROIs and speed of ROIs in the consecutive frames in a video. Then, we suggested a kidnapping detection scheme for intelligent video surveillance system based on the developed frame-based classifier. Using the developed kidnapping event detection scheme within intelligent CCTV system, it is possible that the system automatically analyzes video and recognizes kidnapping events in real-time, and then directly notifies them to CCTV operators to cope with the situations.

The remainder of this paper is organized as follows. In section 2, we discuss shortly the related works on human activity recognitions. Section 3 explains how we defined and extracted the discriminative features from videos to develop the frame-based event classifier. Then, we also describe how this classifier can be used in the kidnapping detection scheme. Section 4 explains the result of the performance evaluations of the developed kidnapping detection scheme against various situations. Finally, we describe the conclusion and future work in section 5.

2 Related Work

In [6], human activities from video recorded in a shopping center are classified into the predefined types of human activities. This work is to develop automatic monitoring system which assists CCTV operators. In this classification, the system considered all trajectories concerning common activities which are related with typical human moving routes at the shopping center such as a person entering a shop, leaving a shop or just passing in front of a shop. But, in all frames of each video, regions of the doors or of the glass walls in a shop are previously identified. And only horizontal(x-axis) directions of human trajectories are considered. That means it recognizes only simple human activities within the given constraints. Suppose that a man appears on the right side of screen and that side indicates the region of the door of a shop. Here, when the man comes through the door, the rate of change of the x-coordinate is getting smaller while x-coordinate of man's ROI moves from the right side to the center of the screen. Therefore, x-coordinate corresponding to the time is used to classify and to recognize activities.

In [7], system automatically recognizes human activities from consecutive video frames recorded in a room. Here, significant types of human activities are: leaving out, sitting down or standing up, using computer, picking up or putting down objects, opening or closing cabinet and so on. In this system, human activities are recognized by referring pre-knowledge about the structures of the room. So, human activities are modeled as a state transition diagram. The system recognizes a state change from one state to others according to the changes of the screen, the location of human, and the location of the tracked object. For instance, if the state of cabinet was "closed" at the previous frame, it would be changed to "opened" at the present frame after recognizing the change of the cabinet region image. But, this changes the state of the object only by recognizing the change of image at the region of the object. Hence, if the amount of pre-knowledge is not enough, there might be some constraints with the recognition of human activities since this system is too much dependent on pre-knowledge.

In [6] and [7] both two systems can recognize specified regions or human's simple activities, but could not recognize events occurred between two or more people. Therefore, the systems are not able to recognize human's complex activities as unusual events. And also, since activities are defined only based on the specified regions, the change occurred in other regions cannot be recognized.

Hence, to solve these problems of previous studies, we have defined and extracted various features which can be used as clues to distinguish between kidnapping and accompanying for each frame from sample videos. Using these features we develop a frame-based event classifier and then we also suggest kidnapping detection scheme for intelligent video surveillance based on the training results about kidnapping (or accompanying) by using data mining techniques.

3 Kidnapping Event Detection Scheme

3.1 Frame-Based Event Classifier

The videos used in this study are about the situations that two people run into each other at parking lot. Like images shown in Figure 1, four videos were recorded for general accompanying situations while nine videos for kidnapping situations, totally thirteen sample videos were directed and recorded by ourselves. The average length of video is 285 frames for about 20 seconds.

To generate training and testing data, we created several scenes of kidnapping situations and accompanying ones, respectively and recorded them as sample videos. Then, we generated ground truth data from these sample videos by using INHA-VAT which is developed by ourselves as a video annotation tool [5]. Simply, INHA-VAT is a semi-automatic video annotation system which is helpful to effectively generate annotation data including basically location and size of object's ROI for each frame. With this system, we can generate annotation data by drawing rectangles each of which indicates ROI for each object. This system also can save the generated ground truth data such as x and y coordinates of the ROI's center point, width, height, and so on in XML format to be used as training data for building a frame-based event classifier.

Fig. 1. A general accompanying situation (top) and a kidnapping situation (bottom)

From the annotation data of videos, we generated the training data for all frames after two people meet in each video. That is, from the annotation data, we generated 14 attributes as the candidate of discriminative features as shown in Table 1 which will be used to distinguish between kidnapping situations and accompanying ones.

From the annotation data of consecutive frames in a video, we could calculate the difference and the change rate of feature values between the former frame and the latter one. In this way, we generated 14 candidate attributes given in Table 1. Also, we considered Maxwidth_rate attribute and Maxheight_rate attribute which are obtained at the rate of width (and height) of ROI after encountering / maximum sum of humans' widths (and heights) before encountering, respectively.

Table 1. 14 candidates of discriminative features for the frame-based event classifier

Attribute Name	*Description*
Width_rate	Change rate of ROI widths to the previous frame
Height_rate	Change rate of ROI heights to the previous frame
Width_rate + Height_rate	Sum of Width_rate and Height_rate
X_distance	Difference of ROI X-axis value from the previous frame
Y_distance	Difference of ROI Y-axis value from the previous frame
All_distance	Difference of ROI values from the previous frame
X_rate	Change rate of ROI X-axis value to the previous frame
Y_rate	Change rate of ROI Y-axis value to the previous frame
X_rate + Y_rate	Sum of X_rate and Y_rate
Width_sub	Difference of ROI width from the previous frame
Height_sub	Difference of ROI height from the previous frame
Width_sub + Height_sub	Sum of Width_sub and Height_sub
Maxwidth_rate	(Width of ROI after encountering) / (Maximum sum of humans' widths before encountering)
Maxheight_rate	(Height of ROI after encountering) / (Maximum value of humans' heights before encountering)

However, in general, to build a classifier we use only the effectively discriminative features among all considerable candidates in data mining field. This step is referred to as feature selection (or attributes selection) [8]. We do feature selection step against 14 candidate features by WEKA's attribute selection function to know which attributes have a decisive effect on the classification of situations. WEKA[8], an open data mining tool offers feature selection functions which select more discriminative features than others from the candidates. That means, feature selection schemes choose features that have more discriminative power for the classifications than others. There are various options which WEKA supports for feature selection and here we choose the following options since these are the default setting:

Attribute Evaluator: CfsSubsetEval
Search Method: BestFirst

By using the feature selection of WEKA with the above option, 4 attributes, X_rate, X_rate + Y_rate, Maxwidth_rate, and Maxheight_rate are selected from 14 candidate attributes. Here, with 10-folds cross validation, we measured the accuracy of classification of video frames into kidnapping situations or accompanying situations for three classification model; a Bayesian network model(named BayesNet in WEKA), a decision tree model(named J48 in WEKA), and a support vector machine(named SMO in WEKA) with only selected-attributes. 10-folds cross validation is an accuracy testing method which divides data sets into 90% for training and the remaining 10% for testing and repeatedly tests with different data subsets. Measured result is given in Table 2.

Table 2. The classification accuracy for various classification models (with 10-folds CV)

Type of Classifier	*Accuracy (%)*
BayesNet	94.862
J48	90.568
SMO	86.933

In the result, BayesNet shows the highest classification accuracy as 94.862%, which is over 4 percent higher than the accuracy of J48 and that of SMO, respectively. According to the result, we design a frame-based event classifier of Bayesian network model with selected 4 features.

3.2 Kidnapping Detection with Frame-Based Event Classifier

The intelligent video surveillance system analyzes video frames to recognize the occurrence of kidnapping event. For it, we propose a frame-based event classifier which is able to distinguish frames of kidnapping situations from ones of accompanying situations. Of course, the frame-based event classifier proposed in this paper uses 4 selected feature values extracted from each frame. For it, from the input frame of a video, the system extracts human information. Against all frames after the first meeting frame in which two people meet first in a video, the frame-based event classifier assorts each frame into a kidnapping situation or into an accompanying situation with respect to 4 feature values of the frame. If the ratio of the number of kidnapping frames to that of accompanying frames after the first frame where two people meet is higher than the threshold, the system detect and notify the occurrence of kidnapping situation eventually. Figure 2 shows the process of kidnapping event detection scheme which consists of 3 steps; initial human detection phase, identifying the first meeting frame where two people meet first, and detection of kidnapping by the frame-based event classifier against frames after the first meeting frame in a video.

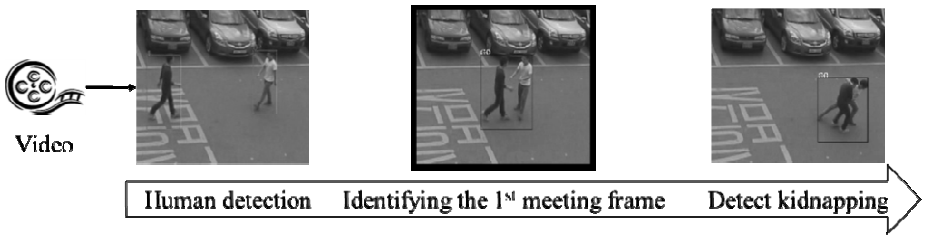


Fig. 2. Three Phases of Kidnapping Detection Scheme

To automatically extract human information from videos, first we developed an automated human recognition module which is mainly based on Histograms of Oriented Gradients (HOG) algorithm [9]. Figure 3 shows the procedure of the human recognition module developed to generate test data for kidnapping event detection scheme. In order to reduce hole-type noises which are generated after removing background from input image, this algorithm uses a median filter since it is commonly

used to remove noises in signal processing field by filling hole-type noises with pixel values of center of mask. Next, HOG algorithm is used to recognize humans after reducing noises. It is also commonly used in image processing field for detecting objects. Next, since there is too much computational complexity when executing optical flow for all pixels of each of detected human objects, we have used optical flow algorithm with extracted features that resulted from the second derivative of the object. Same as training data, this human recognition module draws rectangle boxes which indicate ROI for each object. And also, it extracts features about human's ROI such as center coordinates of video frame, width, height, and encounter status from videos.

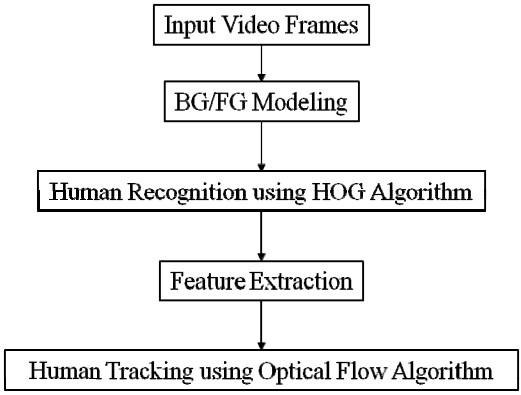


Fig. 3. Human Detection and Tracking

By using the extracted human information, the system identifies the first meeting frame in which two people meet. Identifying the first meeting frame in a video is very important to successfully detect events concerned with two or more people. Hence, against all frames that come after the first meeting frame in video, we should check whether each frame has any clues to detect kidnapping events. To check it, we develop a frame-based event classifier which is able to distinguish a scene of kidnapping situation from that of accompanying one. According to the experiment result described in section 3.1, the frame-based event classifier is developed based on Bayesian network model and uses 4 feature values selected as the more discriminative attributes than others.

To detect occurring of a kidnapping event, we used the following formula. If the ratio of the number of frames classified into kidnapping frame to the number of all frames after the first meeting frame in video is higher than the given threshold value, the system detects and notifies a kidnapping event from the input video.

$$(K / N) \times 100 \geq T \tag{1}$$

Here, N is the number of all the frames after the first meeting frame in a video and K stands for the number of frames classified as kidnapping situation after the first meeting frame. And T is the threshold, the minimum percentage of kidnapping frames among the all frames after the first meeting frame in a kidnapping video. Of course,

choosing the appropriate threshold value is very important to detect kidnapping event without false alarms. It can be defined according to the types and environments of intelligent video surveillance applications.

4 Experiments

For performance evaluation of the proposed kidnapping event detection scheme, we measured the accuracy of testing thirteen videos; nine kidnapping videos and four general accompanying videos. Here, we checked the number of the misclassified frames after the first meeting frame of the each test video. The measurement results for two kinds of situations are given in Table 3 and Table 4, respectively.

In Table 3, the number of misclassified frames which are classified into accompanying situation from kidnapping videos is up to 4 and in average about 1 while, in Table 4, the number of misclassified frames from accompanying videos is up to 24 and in average about 14. In kidnapping situations, misclassified frames constitute up to 6.7% of whole video frames. On the other hand, in general accompanying situation, misclassified frames constitute up to 42.86% of whole video frames.

Table 3. Number of misclassified frames for kidnapping videos

Video	*Number of Misclassified Frames*
kidnap1.avi	2
kidnap2.avi	0
kidnap3.avi	0
kidnap4.avi	1
kidnap5.avi	0
kidnap6.avi	0
kidnap7.avi	2
kidnap8.avi	0
kidnap9.avi	4

Table 4. Number of misclassified frames for accompanying videos

Video	*Number of Misclassified Frames*
accompany1.avi	9
accompany2.avi	8
accompany3.avi	24
accompany4.avi	15

Here, we checked that accompanying situations are likely to have more misclassified frames than kidnapping situations. One possible reason for this result is that we used more videos for kidnapping situation than for accompanying situation to get better detection accuracy against kidnapping situations while generating classifier by using data mining techniques. Another possible reason is that these misclassifications may be caused from the detection error of human object's ROI recognized by our human detection module. But although low performance of human recognition

algorithm makes some misclassifications, it is unlikely to make misjudgment about kidnapping situation since this study focused on the recognition of kidnapping situation and false negative (recognizing kidnapping as accompanying) ratio is higher than false positive (recognizing accompanying as kidnapping) ratio.

The most misclassified video about accompanying is "accompany3.avi" and its cumulative result of misclassifications is given in Figure 4 as an example. Here, the frame numbers of x-axis stand for the sequence number after the first meeting frame in that video. In Figure 4, the number of misclassified frames has been linearly increased after frame number 19. It shows that the number of misclassified frames has not been increased from the beginning of this video. Hence, the kidnapping detection scheme is able to recognize whether kidnapping situation or accompanying situation by setting T of formula (1) with the maximum number of misclassified frames in general accompanying situation, 43%.

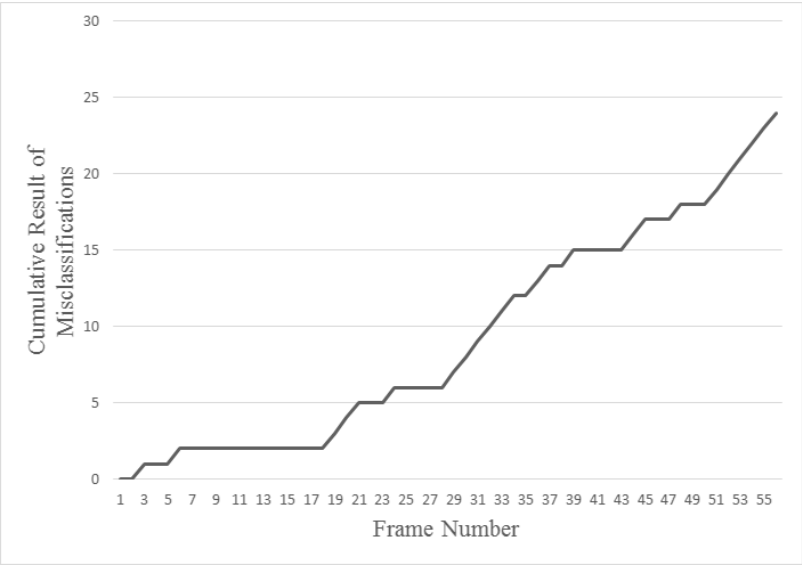


Fig. 4. Cumulative result of misclassifications corresponding to frame number

5 Conclusion

In this study, a kidnapping event detection scheme is proposed in which a frame-based event classifier is used to distinguish kidnapping situation and general accompanying ones. To classify kidnapping situations and general accompanying situation for each frame in video, we generated human's ROI information using INHA-VAT and self-developed human recognition algorithm. Based on the basic attributes of humans' ROI, we selected 4 discriminative features which have more discriminative power than others. Using only these 4 attributes, we developed a frame-based event classifier of Bayesian network model which assorts each frame into kidnapping situation or accompanying ones.

The kidnapping event detection scheme consists of three phases; initial human detection phase, identifying phase of the first meeting frame where two people meet first in the input video, and kidnapping detection phase. Based on the human detection information, the system identifies the first meet frame in a video. Then, against all frames after the first meeting frame in the input video, the frame-based event classifier classifies into kidnapping situations or accompanying ones. When the ratio of the number of kidnapping situations to the number of accompanying one after the first meeting frame is higher than the threshold defined based on the surveillance purposes and environments, the kidnapping event detection scheme detects and notifies the occurrence of kidnapping event in the input video. Also, from the experiment results, we recognized the developed kidnapping event detection scheme can distinguish between kidnapping situations and accompanying ones and finally detect kidnapping events well with the appropriate threshold value.

As future works, we will try to develop other event detection scheme which is possible to recognize not only kidnapping and accompanying situations, but also other situations that may arise in human society.

Acknowledgement. This research was funded by the MSIP(Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013.

References

1. Kang, S.J., Park, J.E., Lee, K.H.: An Analysis for Effect of Crime Preventive CCTV in Residential Areas through Public Opinion Survey. Journal of the Architectural Institute of Korea (Planning & Design) 25(4), 235–244 (2009)
2. Duque, D., Santos, H., Cortez, P.: Prediction of Abnormal Behaviors for Intelligent Video Surveillance Systems. In: Computational Intelligence and Data Mining (2007)
3. Tickner, A.H., Poulton, E.C.: Monitoring up to 16 synthetic television pictures showing a great deal of movement. Ergonomics 16, 381–401 (1973)
4. Ainsworth, T.: Buyer Beware. Security Oz 19, 18–26 (2002)
5. Kim, J.S., Kim, H.I., Kim, Y.S.: A Video Annotation System with Automatic Human Detection from Video Surveillance Data. In: Korea Computer Congress, vol. 39(1) (2012)
6. Nascimento, J.C., Figueiredo, M.A.T., Marques, J.S.: Segmentation and Classification of Human Activities. In: International Workshop on Human Activity Recognition and Modeling, Oxford, UK (2005)
7. Ayers, D., Shah, M.: Monitoring Human Behavior from Video Taken in and Office Environment. Image and Vision Computing 19(12-1), 833–846 (2001)
8. Witten, I.H., Frank, E.: Data Mining: Practical Machine Leaning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
9. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego (2005)
10. Elarbi- Boudihir, M., Al-Shalfan, K.A.: Intelligent Video Surveillance System Architecture for Abnormal Activity Detection. In: The International Conference on Informatics and Applications, Kuala Terengganu, Malaysia (2012)
11. Hansen, D.M., Mortensen, B.K., Duizer, P.T.: Automatic Annotation of Humans in Surveillance Video. In: Fourth Canadian Conference on Computer and Robot Vision, Montreal, Canada (2007)

Global Decisions Taking on the Basis of Dispersed Medical Data

Małgorzata Przybyła-Kasperek and Alicja Wakulicz-Deja

University of Silesia, Institute of Computer Science,
Będzińska 39, 41-200 Sosnowiec, Poland
{malgorzata.przybyla-kasperek,alicja.wakulicz-deja}@us.edu.pl
http://www.us.edu.pl

Abstract. The main aim of the article is to present a decision-making system using dispersed knowledge. The article introduces the system with dynamically generated coalitions. The local knowledge bases, on the basis of which a similar classification for the test object is made, are combined into a coalition. In the proposed system, the classification process can be divided into several steps. In the first step we describe the classification of a test object made on the basis of local knowledge base, by probability vectors over decision classes. We cluster local knowledge bases with respect to similarities of probability vectors. For every cluster, we find a kind of combined information. Finally, we classify the test object using the method for the conflict analysis. The main aim of the paper is to present the results of experiments on medical data. In experiments the situation is considered in which medical data from one domain are collected in many medical centers. We want to use all of the collected data at the same time in order to make a global decisions.

Keywords: decision-making system, global decision, coalition, conflict analysis.

1 Introduction

In recent years, distributed decision making has become of increasing importance and awareness in decisions making and decisions analysis. Modern society, with its overwhelming diversity of interests and developments and its ever growing complexity, can no longer be understood and governed by the paradigm of centralized decision making. Nowadays, the amounts of information stored in repositories are still increasing so centralized processing and analyzing this information is difficult. Furthermore, information are frequently collected in many separate units. For example, in the medical field often in different medical centers, information from one domain, are collected. The problem, which is considered in this article, concerns the use of many local knowledge bases at the same time in the process of global decision-making.

In this paper a new approach to the organization of the system's structure that uses dispersed knowledge is proposed. In earlier papers [9,13,14], a system

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 355–365, 2013.

in which local knowledge bases having common conditional attributes form a group was considered. Additionally, in the paper [9] the use of rough set theory in the process of global decision-making was considered. The paper describes the application of the conditional attributes reduction technique to local knowledge bases. In general it was found that in most cases the use of attribute reduction in a decision-making system using dispersed knowledge reduces the error rate of classification.

The new approach is based on the assumption that one group should contain the knowledge bases on the basis of which a similar classification for the test object is made. In this paper, a system, in which knowledge bases will be combined into groups (coalitions) in a dynamic way, is proposed. For every group, a kind of combined information is determined. Since the sets of attributes, conditions on the basis of which agents classify the test object do not have to be disjoint, an inconsistency in knowledge can occur. Therefore, a method for the elimination of inconsistencies in the knowledge is discussed here. Finally, the test object is classified by voting among clusters, using the combined information from each of clusters. The problem of conflict analysis arises because the inference is being conducted in groups of knowledge bases. By a conflict, we mean a situation in which conflicting decisions are taken for the specified set of conditions on the basis of knowledge stored in different groups of knowledge bases. The main aim of this article is to verify the effectiveness of the system in case of using dispersed medical data.

The concept of distributed decision making is widely discussed in the paper [10]. The concept of taking a global decision on the basis of local decisions is also used in issues concerning the multiple model approach. Examples of the application of this approach can be found in the literature [1,12]. Also in many other papers [3,11], the problem of using distributed knowledge is considered. This paper describes a different approach to the global decision-making process. We assume that the set of local knowledge bases that contain information from one domain is pre-specified. The only condition which must be satisfied by the local knowledge bases is to have common decision attributes.

An important issue which is discussed in this article is the coalition formulation. In the papers of Z. Pawlak [7,8], a model is considered, which describes a conflict situation in which the agents have decided to analyze the conflict by using a peaceful method. In such a situation the relations of conflict, friendship and neutrality were defined. In this paper, some issues of conflict analysis and coalition formation that were given in Pawlak's model are used. There are many different approaches to the analysis of medical data, including the nondeterministic decision rules [6] and cluster analysis and decision units conception [5].

The paper is organized as follows. The second section introduces the definitions and describes the organization of a decision-making system. This section is divided into three parts. The first part of this section explains how coalitions are created. The second part describes the structure of a decision-making system. The last part of the second section presents the methods of elimination of inconsistencies in the knowledge and conflict analysis. The third section shows a

description and the results of experiments carried out using some medical data sets. The article concludes with a short summary in the fourth section.

2 Notations and Definitions of Decision-Making System Using Dispersed Knowledge

We assume that the set of local knowledge bases that contain dispersed medical data from one domain is pre-specified. The only condition which must be satisfied by the local knowledge bases is to have common decision attributes. We assume that each local knowledge base is managed by one agent, which is called a resource agent.

Definition 1. *We call* ag *in* $Ag = \{ag_1, \ldots, ag_n\}$ *a resource agent if he has access to resources represented by a decision table* $D_{ag} := (U_{ag}, A_{ag}, d_{ag})$*, where* U_{ag} *is a finite nonempty set called the universe;* A_{ag} *is a finite nonempty set of conditional attributes,* V^a_{ag} *is a set of attribute a values;* d_{ag} *is referred to as a decision attribute,* V^d_{ag} *is called the value set of* d_{ag}*.*

We want to designate homogeneous groups of resource agents. The agents who agree on the classification for a test object into the decision classes will be combined in the group. This is a new approach to the structure of a decision-making system using dispersed knowledge.

2.1 Forming Coalitions

In this section the relation of friendship and the relation of the conflict between agents will be defined, and the process of combining resource agents into clusters, which are groups of agents remaining in the relation of friendship, will be described. Definitions of the relations of friendship and conflict as well as the method for determining the intensity of conflicts were taken from the papers of Z. Pawlak [7,8].

Let there be given a test object $\bar{x}$ for which we want to generate a global decision. Let for the object $\bar{x}$ the values of conditional attributes belonging to the set $\bigcup_{i=1}^{n} A_{ag_i}$ be defined. In order to determine groups of agents, from each decision table of a resource agent $D_{ag_i}, i \in \{1, \ldots, n\}$ and from each decision class $X^{ag_i}_v, v \in V^{d_{ag_i}}$, the smallest set containing at least m_1 objects is chosen, for which the values of conditional attributes bear the greatest similarity to the test object. The value of the parameter m_1 is selected experimentally. The subset of relevant objects is the union of the sets of objects selected from all decision classes. In order to determine the subset of relevant objects, the measure of similarity is used. In the proposed system any similarity measures could be applied. Since the data sets, which are examined in experiments, have qualitative, quantitative and binary attributes, the Gower similarity measure [14] is used.

The next stage in the process of generating groups of agents is to determine the vectors of values specifying the classification of the test object made by the agents. So, for each resource agent, the vector that indicates the level of certainty

with which the decisions are taken by the agent for the test object is generated. Each coordinate of the vector is determined on the basis of relevant objects that were previously selected from the decision table of the resource agent. Thus, for each resource agent $i \in \{1, \ldots, n\}$, a c-dimensional vector $[\bar{\mu}_{i,1}(\bar{x}), \ldots, \bar{\mu}_{i,c}(\bar{x})]$ is generated, where the value $\bar{\mu}_{i,j}(\bar{x})$ means the certainty with which the decision $v_j \in V^d, j \in \{1, \ldots, c\}$, $c = card\{V^d\}$ is made about the object $\bar{x}$ by the resource agent ag_i. The value $\bar{\mu}_{i,j}(\bar{x})$ is defined as follows:

$$\bar{\mu}_{i,j}(\bar{x}) = \frac{\sum\limits_{y \in U^{rel}_{ag_i} \cap X^{ag_i}_{v_j}} s(\bar{x}, y)}{card\{U^{rel}_{ag_i} \cap X^{ag_i}_{v_j}\}}, i \in \{1, \ldots, n\}, j \in \{1, \ldots, c\}, \tag{1}$$

where $c = card\{V^d\}$, $U^{rel}_{ag_i}$ is the subset of relevant objects selected from the decision table D_{ag_i} of a resource agent ag_i and $X^{ag_i}_{v_j}$ is the decision class of the decision table of resource agent ag_i; $s(x, y)$ is the measure of similarity between objects x and y.

On the basis of the vector of values defined above a vector of rank assigned to the values of the decision attribute is specified. The vector of rank is defined as follows: rank 1 is assigned to the values of the decision attribute which are taken with the maximum level of certainty. Rank 2 is assigned to the values of the decision attribute that have the maximum level of certainty in the set of decisions that have not received the rank 1, etc. Proceeding in this way for each resource agent $ag_i, i \in \{1, \ldots, n\}$, the vector of rank $[r_{i,1}(\bar{x}), \ldots, r_{i,c}(\bar{x})]$ will be defined.

Relations between agents are defined by their views on the classification of the test object $\bar{x}$ to the decision class. We define the function $\phi^{\bar{x}}_{v_j}$ for the test object $\bar{x}$ and each value of the decision attribute $v_j \in V^d$; $\phi^{\bar{x}}_{v_j} : Ag \times Ag \to \{0, 1\}$

$$\phi^{\bar{x}}_{v_j}(ag_i, ag_k) = \begin{cases} 0 & \text{if } r_{i,j}(\bar{x}) = r_{k,j}(\bar{x}) \\ 1 & \text{if } r_{i,j}(\bar{x}) \neq r_{k,j}(\bar{x}) \end{cases} \quad \text{where } ag_i, ag_k \in Ag. \tag{2}$$

Definition 2. *Agents $ag_i, ag_k \in Ag$ are in a friendship relation due to the object $\bar{x}$ and decision class $v_j \in V^d$, which is written $R^+_{v_j}(ag_i, ag_k)$, if and only if $\phi^{\bar{x}}_{v_j}(ag_i, ag_k) = 0$. Agents $ag_i, ag_k \in Ag$ are in a conflict relation due to the object $\bar{x}$ and decision class $v_j \in V^d$, which is written $R^-_{v_j}(ag_i, ag_k)$, if and only if $\phi^{\bar{x}}_{v_j}(ag_i, ag_k) = 1$.*

We also define the intensity of conflict between agents using a function of distance between agents. We define the distance between agents $\rho^{\bar{x}}$ for the test object $\bar{x}$: $\rho^{\bar{x}} : Ag \times Ag \to [0, 1]$

$$\rho^{\bar{x}}(ag_i, ag_k) = \frac{\sum_{v_j \in V^d} \phi^{\bar{x}}_{v_j}(ag_i, ag_k)}{card\{V^d\}}, \text{ where } ag_i, ag_k \in Ag. \tag{3}$$

Definition 3. *We say that agents $ag_i, ag_k \in Ag$ are in a friendship relation due to the object $\bar{x}$, which is written $R^+(ag_i, ag_k)$, if and only if $\rho^{\bar{x}}(ag_i, ag_k) < 0.5$.*

Agents $ag_i, ag_k \in Ag$ *are in a conflict relation due to the object* $\bar{x}$*, which is written* $R^-(ag_i, ag_k)$*, if and only if* $\rho^{\bar{x}}(ag_i, ag_k) \geq 0.5$*.*

The two ways to create dynamic groups of knowledge bases are considered.

Dynamically Generated Disjoint Clusters
The dynamically generated disjoint clusters containing agents which are in the friendship relation are defined as follows. Using the definitions of the function of distance between agents, we determine the distance between each pair of resource agents. Then the cluster generation process is initiated as follows. Initially, each resource agent is treated as a separate cluster. These two steps are performed until the stop condition (which is given in the first step) is met.

1. One pair of different clusters is selected (in the very first step a pair of different resource agents) for which the distance reaches a minimum value. If the selected value of the distance is less than 0.5, then agents from the selected pair of clusters are combined into one new cluster. Otherwise, the clustering process is terminated.
2. After defining a new cluster, the values of the distances between the clusters are recalculated. The following method for recalculating the value of the distance is used. Let $\rho^x : 2^{Ag} \times 2^{Ag} \to [0, 1]$, let D_i be a cluster formed from the merger of two clusters $D_i = D_{i,1} \cup D_{i,2}$ and let it be given a cluster D_j then

$$\rho^x(D_i, D_j) = \begin{cases} \frac{\rho^x(D_{i,1}, D_j) + \rho^x(D_{i,2}, D_j)}{2} & \text{if } \rho^x(D_{i,1}, D_j) < 0.5 \text{ and} \\ & \rho^x(D_{i,2}, D_j) < 0.5 \\ \max\{\rho^x(D_{i,1}, D_j), \rho^x(D_{i,2}, D_j)\} & \text{if } \rho^x(D_{i,1}, D_j) \geq 0.5 \text{ or} \\ & \rho^x(D_{i,2}, D_j) \geq 0.5 \end{cases}$$

The proposed clustering process is similar to the hierarchical agglomerative clustering method. However, the proposed method has a clearly defined stop condition. The stop condition is based on the assumption that one cluster should not contain two resource agents that are in a conflict relation due to the test object.

Dynamically Generated Clusters with a Non-Empty Intersection
The dynamically generated clusters with non-empty intersection containing agents which are in the friendship relation are defined as follows.

Definition 4. *Let* Ag *be the set of resource agents. A cluster due to classification of the object* x *is the maximum, due to the inclusion relation, subset of resource agents* $X \subseteq Ag$ *such that*

$$\forall_{ag_i, ag_k \in X} \quad R^+(ag_i, ag_k). \tag{4}$$

Thus, the cluster is the maximum, due to the inclusion relation, set of resource agents remaining in the friendship relation due to the object x*.*

An algorithm for clusters generation due to classification of the object x is as follows. We assume that a set of resource agents Ag is given, and the value $\rho^x(ag_i, ag_k)$ is determined for each $ag_i, ag_k \in Ag$. Some initial values $X_1 = Ag, i = 1, j = 1$ are established.

1. While $i \leq j$, the following is executed:
   (a) Values of the distance function are checked $\rho^x(ag_l, ag_k)$ for each pair of agents $ag_l, ag_k \in X_i$. If there is a pair of agents $ag_l, ag_k \in X_i$ such that $\rho^x(ag_l, ag_k) \geq 0.5$ then $j = j + 1$ and the two following sets are defined: $X_j = X_i \setminus \{ag_l\}$, $X_i = X_i \setminus \{ag_k\}$.
   (b) The above step is executed until the set X_i will satisfy the condition: $\rho^x(ag_l, ag_k) < 0.5$ for each pair of agents $ag_l, ag_k \in X_i$. If this condition is met, then $i = i + 1$.
2. From the sets $X_i, i = 1, \ldots, j$ the largest sets, due to the inclusion relation are selected. Selected sets are the clusters due to the classification of the object x.

2.2 Structure of a Decision-Making System Using Dispersed Knowledge

After the completion of the clustering process, for both systems (with disjoint clusters or clusters with a non-empty intersection), a synthesis agent *as* is defined for each cluster that contains at least two resource agents. If a single resource agent forms a cluster, it becomes the synthesis agent. In this way, a hierarchical structure of the system is created. At the lowest level of the hierarchy there are resource agents, and at a higher level there are synthesis agents.

Definition 5. *By the multi-agent decision-making system with dynamically generated clusters we mean* $WSD_{Ag}^{dyn} = \left\langle Ag, \{D_{ag} : ag \in Ag\}, \{As_x : x \text{ is a classified object}\}, \{\delta_x : x \text{ is a classified object}\} \right\rangle$ *where* Ag *is a finite set of resource agents;* $\{D_{ag} : ag \in Ag\}$ *is a set of decision tables of resource agents;* As_x *is a finite set of synthesis agents defined for clusters dynamically generated for the test object* x, $\delta_x : As_x \to 2^{Ag}$ *is a injective function which each synthesis agent assigns a cluster generated due to classification of the object* x.

2.3 Elimination of Inconsistencies in the Knowledge and Conflict Analysis

On the basis of the knowledge of agents from one cluster, local decisions are taken. An important problem that occurs when taking a global decision is to eliminate inconsistencies in the knowledge stored in different knowledge bases. This problem stems from the fact that the system has the general assumptions and we do not require that the sets of conditional attributes of decision tables are disjoint. We understand inconsistency of knowledge to be situations in which, on the basis of two different knowledge bases that have common attributes and for the same values for common attributes using logical implications, conflicting decisions are made.

In previous papers some methods of elimination inconsistencies in the knowledge have been proposed [9,13,14]. In this paper, one of these methods - the approximated method of the aggregation of decision tables, will be used.

This method for the elimination of any inconsistencies in the knowledge will be implemented for resource agents belonging to one cluster. The essence of the method is to create objects of the aggregated decision table from the relevant objects selected from the decision table of a resource agent. The new objects are constructed by combining objects from the decision tables of the resource agents that belong to one cluster, but only those objects are combined for which the values of the decision attribute and common conditional attributes are equal. The approximated method of aggregation of decision tables was proposed and described in detail in the paper [13].

Conflict analysis is implemented after completion of the process of inconsistencies elimination in knowledge, because then the synthesis agents have access to the knowledge on the basis of which they can independently establish the value of a local decision to just one cluster. Two methods to resolve the conflict analysis will be used in this paper: the method of weighted voting and the method of a densitybased algorithm. These methods allow the analysis of conflicts and enable to generate a set of global decisions. In the case of the density-based method the generated set will contain not only the value of the decisions that have the greatest support of knowledge stored in local knowledge bases, but also those for which the support is relatively high. These methods were discussed in detail in the paper [13].

3 Experiments

The aim of the experiments is to examine the quality of the classification made on the basis of dispersed medical data by the decision-making system with dynamically generated clusters. An additional objective is to compare the effectiveness of this system with the results obtained in the papers of other authors [2,4] who have used the medical data in non-dispersible form. For each knowledge bases from medical domain the results of using classical classification algorithms and decision-making system using dispersed knowledge are presented. For the experiments the following data, which are in the UCI repository, were used: Lymphography data set, Primary Tumor data set. Both sets of data was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia (M. Zwitter and M. Soklic provided this data). Lymphography is a medical imaging technique in which a radiocontrast agent is injected, and then an X-ray picture is taken to visualize structures of the lymphatic system. This test method gives great service especially in the evaluation of cancer stage of the lymphatic system. In the Primary Tumor data set, on the basis of values of attributes such as histologic-type, supraclavicular etc. a decision is taken where (of 22 organs) the cancer cells are located. In order to determine the efficiency of inference of the multi-agent decision-making system with respect to the analyzed data, each data set was divided into two disjoint subsets: a training set and a test set. Table 1 gives a numerical summary of the data sets.

We will consider a situation in which medical data from one domain are collected in different medical centers. We want to use all of the collected data at

Table 1. Data set summary

Data set	# The training set	# The test set	# Conditional attributes	# Decision classes
Lymphography	104	44	18	4
Primary Tumor	237	102	17	22

the same time in order to make a global decisions. This approach not only allows the use of all available knowledge, but also should improve the efficiency of inference. In order to consider the discussed situation it is necessary to provide the knowledge stored in the form of a set of decision tables. Therefore, the training set was divided into a set of decision tables. Divisions with a different number of decision tables were considered. For each of the data sets used, the decision-making system with five different versions (with 3, 5, 7, 9 and 11 resource agents) were considered. For these systems, we use the following designations: WSD^{dyn}_{Ag1} - 3 resource agents; WSD^{dyn}_{Ag2} - 5 resource agents; WSD^{dyn}_{Ag3} - 7 resource agents; WSD^{dyn}_{Ag4} - 9 resource agents; WSD^{dyn}_{Ag5} - 11 resource agents. Note that the division of the data set was not made in order to improve the quality of the decisions taken by the decision-making system, but in order to store the knowledge in a distributed form. We consider the situation, that is very common in life, in which data are collected in different medical centers as separate knowledge bases. The division of the data set into the decision tables of resource agents was carried out as follows. In the first step, the cardinality of the set of conditional attributes in each decision table of a resource agent was determined and the number of common conditional attributes of the decision tables was defined. These values were defined by the authors. Then, the conditional attributes were randomly assigned to the decision tables so that the conditions which were defined earlier were met and each conditional attribute that appears in the data set is included in at least one set of the conditional attributes of the decision tables. The decision attribute in the decision tables is the same as the decision attribute in the data set. Each universe of the decision tables includes all of the objects from the data set.

The measures of determining the quality of the classification are:

- estimator of classification error e in which an object is considered to be properly classified if the decision class used for the object belonged to the set of global decisions generated by the system;
- estimator of classification ambiguity error e_{ONE} in which object is considered to be properly classified if only one, correct value of the decision was generated to this object;
- the average size of the global decisions sets $\overline{d}_{WSD^{dyn}_{Ag}}$ generated for a test set.

In the description of the results of experiments for clarity some designations for algorithms have been adopted: $A(m_2)$ - the approximated method of the aggregation of decision tables; W - the method of weighted voting; $G(\varepsilon, MinPts)$ - the method of a density-based algorithm.

The results of the experiments with the Lymphography data set are presented in Table 2. In the table the following information is given: the name of multi-agent decision-making system (System); the algorithm's symbol (Algorithm); the three measures discussed earlier $e, e_{ONE}, \overline{d}_{WSD_{Ag}^{dyn}}$; the time t needed to analyse a test set expressed in minutes. Based on the results of the experiments given in Table 2, the following conclusions can be drawn. In most cases better results for the system with dynamically generated clusters with non-empty intersection were obtained comparring to the decision-making system with dynamically generated disjoint clusters. The results of the experiments with the Primary Tumor data set are presented in Table 3. Again, better results were obtained for the system with dynamically generated clusters with non-empty intersection.

The papers [2,4] also shows the experiments with the Lymphography and the Primary Tumor data set. Data in the non-dispersible form were examined. Table 4 presents the results given in this papers. Presented, in this paper, results

Table 2. Summary of experiments results with the Lymphography data set

Dynamically generated clusters with non-empty intersection

System	Algorithm	e	e_{ONE}	$\overline{d}_{WSD_{Ag}^{dyn}}$	t
WSD_{Ag1}	$A(1)G(0.0625;2)$	0.091	0.591	1.5	0.01
$m_1 = 2$	$A(1)G(0.0105;2)$	0.159	0.273	1.114	0.01
WSD_{Ag2}	$A(1)G(0.062;2)$	0.068	0.523	1.455	0.01
$m_1 = 1$	$A(1)G(0.0245;2)$	0.182	0.386	1.205	0.01
WSD_{Ag3}	$A(1)G(0.0515;2)$	0.114	0.545	1.432	0.01
$m_1 = 1$	$A(1)G(0.0005;2)$	0.159	0.273	1.114	0.01
WSD_{Ag4}	$A(1)G(0.0625;2)$	0.114	0.568	1.455	0.01
$m_1 = 1$	$A(1)G(0.052;2)$	0.136	0.455	1.318	0.01
WSD_{Ag5}	$A(1)G(0.069;2)$	0.114	0.568	1.455	0.07
$m_1 = 2$	$A(1)G(0.054;2)$	0.182	0.545	1.364	0.07

Dynamically generated disjoint clusters

System	Algorithm	e	e_{ONE}	$\overline{d}_{WSD_{Ag}^{dyn}}$	t
WSD_{Ag1}	$A(1)G(0.0624;2)$	0.091	0.591	1.545	0.01
$m_1 = 2$	$A(1)G(0.0092;2)$	0.182	0.295	1.159	0.01
WSD_{Ag2}	$A(1)G(0.0775;2)$	0.136	0.636	1.500	0.01
$m_1 = 2$	$A(1)G(0.029;2)$	0.159	0.364	1.205	0.01
WSD_{Ag3}	$A(1)G(0.0858;2)$	0.136	0.591	1.455	0.01
$m_1 = 2$	$A(1)G(0.0006;2)$	0.159	0.273	1.114	0.01
WSD_{Ag4}, $m_1 = 2$	$A(1)G(0.0702;2)$	0.136	0.455	1.318	0.01
WSD_{Ag5}	$A(1)G(0.084;2)$	0.159	0.614	1.477	0.07
$m_1 = 1$	$A(1)G(0.0672;2)$	0.182	0.545	1.364	0.07

Table 3. Summary of experiments results with the Primary Tumor data set

Dynamically generated clusters with non-empty intersection

System	Algorithm	e	e_{ONE}	$\overline{d}_{WSD_{Ag}^{dyn}}$	t
WSD_{Ag1}, $m_1 = 5$	$A(2)G(0.00549; 2)$	0.373	0.814	3.020	0.01
WSD_{Ag2}, $m_1 = 17$	$A(3)G(0.0009; 2)$	0.343	0.814	2.990	0.02
WSD_{Ag3}, $m_1 = 2$	$A(1)G(0.0006; 2)$	0.353	0.892	3.784	0.02
WSD_{Ag4}, $m_1 = 3$	$A(3)G(0.00556; 2)$	0.353	0.912	3.765	0.05
WSD_{Ag5}, $m_1 = 1$	$A(2)G(0.0001; 2)$	0.314	0.922	4.294	0.33

Dynamically generated disjoint clusters

System	Algorithm	e	e_{ONE}	$\overline{d}_{WSD_{Ag}^{dyn}}$	t
WSD_{Ag1}, $m_1 = 5$	$A(2)G(0.00549; 2)$	0.373	0.814	3.020	0.01
WSD_{Ag2}, $m_1 = 17$	$A(3)G(0.0003; 2)$	0.353	0.814	2.990	0.02
WSD_{Ag3}, $m_1 = 5$	$A(5)G(0.00573; 2)$	0.373	0.912	3.755	0.02
WSD_{Ag4}, $m_1 = 4$	$A(3)G(0.0063; 2)$	0.343	0.902	3.667	0.05
WSD_{Ag5}, $m_1 = 6$	$A(1)G(0.0003; 2)$	0.333	0.941	4.294	0.33

can not be compared uniquely with the results shown in Table 4. Because the decision-making system, described in the paper, generates a set of decisions, while Table 4 shows the results of algorithms that generate one decision. It should be noted that for the Lymphography data set the average size of the global decisions sets is small, since it is close to the value 1. In the case of the Primary Tumor data set the average size of the global decisions sets is between 3 and 4, note that there are 22 decision classes. This means that this result may be considered as a quite good result. However, the quality of classification has significantly improved in comparison with the results shown in Table 4. Moreover, very important advantage of the proposed decision-making system is the possibility of using dispersed knowledge, which are collected in different medical centers.

Table 4. Results of experiments from other papers

Lymphography		Primary Tumor	
Algorithm	Error rate	Algorithm	Error rate
Bayes	0.17	Bayes	0.61
AQR	0.24	AQR	0.65
CN2	0.22	CN2	0.63
AQ15	0.18	AQ15	0.59
Human Experts	0.15	Human Experts	0.58
Random Choice	0.75	Random Choice	0.95

4 Conclusions

In this paper a new approach to structure creation of decision-making system using dispersed knowledge was proposed. In this approach dynamically generated disjoint clusters and dynamically generated clusters with non-empty intersection are used. In the experiments, which are presented in the article, dispersed medical data have been used: Lymphography data set, Primary Tumor data set. The usage of dispersed medical data is very important, because in many medical centers, information from one domain, are collected. Thus, these data are in the dispersed form. Based on the presented results of experiments it can be concluded that the proposed decision-making system achieve good results for dispersed medical data.

References

1. Bazan, J., Peters, J., Skowron, A., Nguyen, H., Szczuka, M.: Rough set approach to pattern extraction from classifiers, In: Electronic Notes in Theoretical Computer Science 82, Elsevier Science Publishers (2003)
2. Clark, P., Niblett,T.: Induction in Noisy Domains. In Bratko I., Lavrac N. (Eds.) Progress in Machine Learning, 11–30 (1987)
3. Delimata, P., Suraj, Z.: Feature Selection Algorithm for Multiple Classifier Systems: A Hybrid Approach. Fundamenta Informaticae 85 (1-4), IOS Press, Amsterdam, 97–110 (2008)
4. Michalski, R., Mozetic, I. Hong, J., Lavrac, N.: The Multi-Purpose Incremental Learning System AQ15 and its Testing Applications to Three Medical Domains. In Proceedings of the Fifth National Conference on Artificial Intelligence, 1041–1045 (1986)
5. Nowak-Brzezińska, A., Simiński, R.: Knowledge mining approach for optimization of inference processes in medical rule knowledge bases. J. of Medical Infor. & Tech., 20, 19–27 (2012)
6. Marszał-Paszek, B., Paszek, P.: Nondeterministic decision rules in classification process for medical data. J. of Medical Infor. & Tech., 17, 59–64 (2011)
7. Pawlak, Z.: On conflicts. Int. J. of Man-Machine Studies 21, 127–134 (1984)
8. Pawlak, Z.: An Inquiry Anatomy of Conflicts. Journal of Information Sciences 109, 65–78 (1998)
9. Przybyła-Kasperek, M., Wakulicz-Deja, A.: Application of reduction of the set of conditional attributes in the process of global decision-making, Fundamenta Informaticae 122 (4), 327–355 (2013)
10. Schneeweiss, C.: Distributed decision making. Springer, Berlin (2003)
11. Skowron, A., Wang, H., Wojna, A., Bazan, J.: Multimodal Classification: Case Studies. T. Rough Sets, 224–239 (2006)
12. Ślęzak, D., Wróblewski, J., Szczuka, M.: Neural network architecture for synthesis of the probabilistic rule based classifiers, In: Electronic Notes in Theoretical Computer Science 82, Elsevier, (2003)
13. Wakulicz-Deja, A., Przybyła-Kasperek, M.: Multi-Agent Decision Taking System, Fundamenta Informaticae 101(1-2), 125–141 (2010)
14. Wakulicz-Deja, A., Przybyła-Kasperek, M.: Application of the method of editing and condensing in the process of global decision-making, Fundamenta Informaticae 106 (1), 93–117 (2011)

Soft Clustering to Determine Ambiguous Regions during Medical Images Segmentation

Manish Joshi[1] and Monica Mundada[2]

[1] School of Computer Sciences, North Maharashtra University
Jalgaon, Maharashtra, India
joshmanish@gmail.com
[2] Department of Computer Science, Pratap College, Amalner, MS, India
monicamundada5@gmail.com

Abstract. Image segmentation is an essential step in almost all image processing applications and very critical particularly for medical images. Image segmentation procedure segments an image into appropriate number of regions. Several techniques have been proposed and experimented to obtain effective image segmentation. Clustering is one of the commonly used image segmentation techniques.

There exist ambiguous regions in an image and segmenting these regions correctly is a challenging task. Different clustering approaches are explored by researchers to deal these ambiguous regions in order to obtain better image segmentation. We propose rough clustering approach to explicitly determine ambiguous regions from an image. Once ambiguous regions are identified segmentation would be easier.

In this paper we present our experiments of image segmentation using crisp K-means clustering algorithms and rough K-means (RKM) clustering algorithms. With the help of various images we demonstrate that RKM algorithm is able to determine ambiguous regions distinctly whereas K-means forced pixels of ambiguous regions to either region. Furthermore, we analyze how other soft clustering techniques deals with ambiguous regions.

Keywords: clustering, K means, Rough k means, rough sets, Fuzzy sets, Image segmentation.

1 Introduction

Image analysis process initiates with the task of segmentation [29]. The effectiveness of later steps of image analysis rely on the quality of a segmentation process. Hence, considerable efforts are taken to improve the probability of successful segmentation. Image segmentation has wide spread applications in many fields including multimedia databases, color image and video transmission over internet, digital broadcasting, interactive TV, video-on-demand, computer-based training, distance education, video-conferencing [8]. The focus areas of research fields where image segmentation used are computer science, geography, medical imaging, criminal justice, and remote sensing. Medical Image Processing (MIP) in particular deals with sensitive and demands very high precision of segmentation.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 366–373, 2013.

The process of clustering is differentiated as Key word based clustering and Content based clustering [24]. In key word based clustering - the identical features of clusters are recognized by the keyword specified from the user. Content based clustering [26] works on features of image such as shapes, texture, color etc. Clustering algorithms are differentiated based on their cluster model, as hierarchical clustering, centroid-based clustering, distribution-based clustering, Density - based clustering etc.

K- means clustering method is very popular for pattern recognition [6,15]. In K-means clustering a centroid vector is computed for every cluster. The centroid is chosen randomly with an aim to minimize the overall distance within the clusters. Both supervised and unsupervised clustering techniques are used in image segmentation. In supervised clustering method [3], grouping is done according to user feedback. In unsupervised clustering, the images with high features similarities to the query may be very different in terms of semantics. The K-means clustering algorithm needs some initial cluster set and if these are chosen incorrectly, the K-means algorithm fails to produce good segmentation.

A conventional clustering algorithm such as K-means categorizes an image pixel into precisely one cluster. This is well and good when all pixels in an image are clearly separable and can be segmented in different segments without any ambiguity. But for an image having ambiguous regions we need different approach of clustering. Ambiguous region of a medical image might get segmented forcefully as a wrong region, which ultimately may cause harm to a patients. Fuzzy clustering [16,23,17], Evidential Clustering and rough set clustering [22,4,20] provide an ability to specify the membership of an image pixel to multiple clusters, which can be useful in real world applications of image segmentation. We propose to test RKM algorithm to determine whether it can find out ambiguous regions from an image.

The remaining paper is organized as follows. Section 2 presents related work in this field of image segmentation using various clustering approaches. Section 3 describes in short about rough clustering. We present our experimental details and discuss the results obtained in section 4 followed by conclusions in section 5.

2 Related Work

Several techniques are used for medical image segmentation. We review a few approaches that use some sort of clustering as a basis of image segmentation. A new marker controlled watershed algorithm along with k means algorithm for medical image analysis. In this watershed transform is used to segment gray matter, white matter and cerebrospinal fluid from magnetic resonance (MR) brain image [18].

A new Genetic Approach on Medical Image Segmentation by Generalized Spatial Fuzzy C-Means Algorithm (GSFCA). The algorithm improved the level of accuracy and efficiency of image segmentation [1]. A new framework of content based image retrieval was put forward with integration of semantic cluster classifier with K - Means algorithm. Subrajeet Mohapatra et al. [16] proposed

new hybrid algorithm fuzzy algorithm with the segmentation of leukocytes and their components. The work incorporates the combination of both rough sets and fuzzy sets in clustering framework performance. The improved Hybrid Clustering Algorithm is presented for fast, accurate and noise adaptive clinical analysis of brain MRI [23]. The concept of lower and upper approximations of rough sets is incorporated to make the segmentation robust to noise. The images are pre-processed with a neighborhood averaging spatial filter. Rough set approximations are also proposed for image segmentation by [7]. FCM based algorithms in terms of segmentation accuracy for both noise-free and noise-inserted MR images [9,30]. The comparative study proposed by Li Hao [5] shows that all segmentation algorithms especially the thresholding algorithms are sensitive to noise and face difficulty in segmenting images with low contrast and inhomogeneous regions. For complex medical images, these general algorithms can only be used as parts of a more sophisticated algorithm. A modified, FCM based on two stages was proposed [2]. In this rough set theory was incorporated with FCM by reduction theory the initial clusters centers at first cluster set is eliminated.

Either crisp K-means or flexible clustering algorithms (FCM, ECM or RKM) are used for image segmentation in above mentioned research proposals. However, we propose to use RKM to determine ambiguous regions from an image.

The short description of Rough K-means algorithms is presented in the next section.

3 Rough Clustering Approach

In addition to clearly identifiable groups of objects, it is possible that a data set may consist of several objects that lie on the fringes [27]. The conventional clustering techniques mandate that such objects belong to precisely one cluster. Such a requirement is found to be too restrictive in many data mining applications [9]. In practice, an object may display characteristics of different clusters. In such cases, an object should belong to more than one cluster, and as a result, cluster boundaries necessarily overlap. Fuzzy set representation of clusters, using algorithms such as fuzzy C-means, makes it possible for an object to belong to multiple clusters with a degree of membership between 0 and 1 [10]. In some cases, the fuzzy degree of membership may be too descriptive for interpreting clustering results. Rough set based clustering provides a solution that is less restrictive than conventional clustering and less descriptive than fuzzy clustering.

Lingras and West [12] provided an efficient alternative based on an extension of the K-means algorithm [15]. Incorporating rough sets into K-means clustering requires the addition of the concept of lower and upper bounds. The incorporation required redefinition of the calculation of the centroids to include the effects of lower and upper bounds. The next step was to design criteria to determine whether an object belongs to the lower and upper bounds of a cluster.

The rough K-means approach has been a subject of further research. Peters [19] discussed various refinements of Lingras and West's original proposal [12]. These included calculation of rough centroids and the use of ratios of distances as opposed to differences between distances similar to those used in the rough set based Kohonen algorithm described in [13]. The rough K-means [14] and its various extensions [28] have been found to be effective in distance based clustering. However, there is no theoretical work that proves that rough K-means explicitly finds an optimal clustering scheme. Moreover, the quality of clustering that is maximized by the rough clustering is not precisely defined [25]. We compare crisp and rough clustering algorithm results and present our observations in section 4.

Rough K-Means Algorithm. We represents each cluster $c_i, 1 \leq i \leq k$, using its lower $\underline{A}(c_i)$ and upper $\overline{A}(c_i)$ bounds. All objects that are clustered using the algorithm follow basic properties of rough set theory such as:

(P1) An object $\boldsymbol{x}$ can be part of at most one lower bound
(P2) $\boldsymbol{x} \in \underline{A}(\boldsymbol{c}_i) \Longrightarrow \boldsymbol{x} \in \overline{A}(\boldsymbol{c}_i)$
(P3) An object $\boldsymbol{x}$ is not part of any lower bound

$$\Updownarrow$$

$\boldsymbol{x}$ belongs to two or more upper bounds.

Algorithm 1 depicts the general idea of the algorithm. The values of p, w_{lower}, w_{upper} are finalized based on the experiments described in [14]. A new set of centroids emerges at the end of individual iteration and objects are reassigned to the lower/upper bound of appropriate clusters.

Like crisp clustering, the cluster membership of an object in case of rough clustering is also determined by the distance of an object from the cluster centroids. Additionally, in order to determine whether an object belongs to a lower bound of a single cluster or to an upper bound of two or more clusters the roughness parameter 'threshold' (p) is introduced. The procedure to determine the cluster membership for rough clustering is given below. For each object vector, $\boldsymbol{v}$, let $d(\boldsymbol{v}, \boldsymbol{c}_j)$ be the distance between itself and the centroid of a cluster $\boldsymbol{c}_j$. Let $d(\boldsymbol{v}, \boldsymbol{c}_i) = \min_{1 \leq j \leq k} d(\boldsymbol{v}, \boldsymbol{c}_j)$. The ratios $d(\boldsymbol{v}, \boldsymbol{c}_j)/d(\boldsymbol{v}, \boldsymbol{c}_i)$, $1 \leq i, j \leq k$, are compared with a cut-off value to determine the membership of an object $\boldsymbol{v}$. This parameter is called as a *threshold*. Let $T = \{j : d(\boldsymbol{v}, \boldsymbol{c}_j)/d(\boldsymbol{v}, \boldsymbol{c}_i) \leq threshold \text{ and } i \neq j\}$.

1. If $T \neq \emptyset$, $\boldsymbol{v} \in \overline{A}(\boldsymbol{c}_i)$ and $\boldsymbol{v} \in \overline{A}(\boldsymbol{c}_j), \forall j \in T$. Furthermore, $\boldsymbol{v}$ is not part of any lower bound. The above criterion guarantees that property (P3) is satisfied.
2. Otherwise, if $T = \emptyset$, $\boldsymbol{v} \in \underline{A}(\boldsymbol{c}_i)$. In addition, by property (P2), $\boldsymbol{v} \in \overline{A}(\boldsymbol{c}_i)$.

In next section we discuss details of our experiments with image segmentation using RKM and KM.

Data:
k: the number of clusters,
$D(n, m)$: a data set containing n objects where each object has m dimensions,
Result: A set of clusters. Each cluster is represented by the objects in the lower region and in boundary region (upper bound),

p: a roughness threshold value (1.4);
w_lower: relative importance assigned to lower bound (0.75);
w_upper: relative importance assigned to upper bound (0.25);
arbitrarily choose k objects from D as the initial cluster centers (centroids);
repeat
(re)assign each object to lower/upper bounds of appropriate clusters by determining its distance from each cluster centroid;
update the cluster centroids using number of objects assigned to the cluster;
until *no change* ;

Algorithm 1. The rough K-means clustering algorithm

4 Experimental Setup and Results

In order to test the usefulness of RKM algorithm in determination of ambiguous regions from images, we decided to implement the RKM and K-means algorithms on a few images and observed the resulting segmentation.

The step by step procedure implemented to obtain image segmentation using K means and RKM clustering is as given below.

1. We decided to focus on gray level medical images.
2. The two dimensional intensity matrix of an image is clustered using K-means function of Matlab.
3. According to the K-means clustering an image is reconstructed. Now image shows separate segments.
4. For rough K-means clustering we used a RKM algorithm that is implemented in Java.
5. The same two dimensional intensity matrix of an image is inputted to obtain rough clusters. Pixels that belong to upper approximation regions of multiple clusters are ambiguous and presented distinctly in resulting image segmentation.
6. Based on RKM result the image is segmented with an additional region that corresponds to ambiguous region of an image.

We experimented with various images and most of these images are medical images. With the help of following three distinct images we would like to put forth our observations.

A brain image has three distinct regions that correspond to three important constituents of brain namely normal brain tissues, ventricles and cerebrospinal fluid (CSF), and pathology of brain. Figure 2(a) shows an original brain image. Figure 2(b) is a K-means clustered image segmentation in which three regions

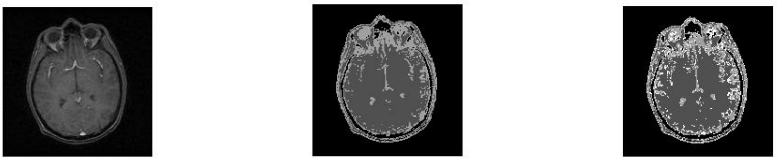


Fig. 1. Brain Image Segmentation

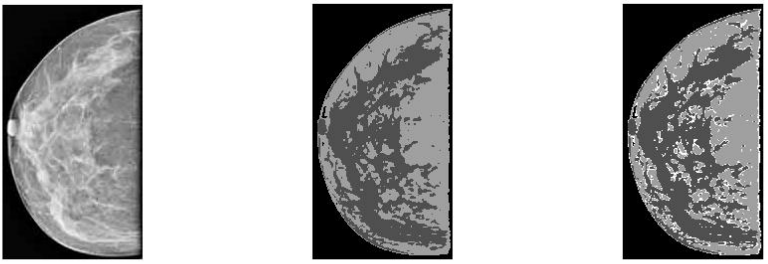


Fig. 2. Mammogram Image Segmentation

Fig. 3. Peacock Image Segmentation

are displayed but some pixels are misplaced into other region. Figure 2(c) is a RKM clustering based image segmentation in which ambiguous region is clearly indicated by white pixels.

Mammogram images are segmented to determine if there is a presence of tumor in breast. Incorrect segmentation may lead to wrong diagnosis. Hence, in case of lack of detail information a pixel should not be assigned to any region as done in K-means clustering based image segmentation in Figure 3(b). RKM clustering based image segmentation is shown in Figure 3(c). Here ambiguous region is marked with white pixels.

In original peacock image as shown in Figure 4(a) we can see overlapping of peacock tail and branch. Proper distinction is necessary. Figure 4(b) and 4(c) shows K-means and RKM clustering based image segmentation. Ambiguous pixels are marked with white color in rough clustering result image (Figure 4(c)).

We can observe that in all images an ambiguous region is clearly identified by RKM clustering. It is presented with white pixels. Whereas images segmented using K-means could not identify ambiguous regions as the ambiguous pixels are pushed to one of the existing regions.

5 Conclusions

In this paper, rough set based k-means algorithm was proposed to visualize the ambiguous area distinctly. We experimented with number of images and observed that RKM clustering algorithm is able to determine ambiguous regions present in medical images successfully. More work shall be performed in future to determine if RKM can be used to determine ambiguous regions from colored images.

References

1. Chintalapalli, M.: Image segmentation by clustering (using Mahalanobis distance)
2. Jobin Christ, M.C., Parvathi, R.M.S.: Magnetic Resonance Brain Image Segmentation. International Journal of VLSI design & Communication Systems (VLSICS) 3(4), 121–133 (2012)
3. Grira, N., Crucianu, M., Boujemma, N.: Unsupervised and semi-supervised clustering: a brief survey, France (2005)
4. Haldar, A., Dasgupta, A.: Colour image segmentation using rough set K-means algorithm. Int. Jor. of Computer Applications 57(12) (2012)
5. Hao, L.: Registration-Based Segmentation of Medical Images. School of Computing National University of Singapore (2006)
6. Hartigan, J.A., Wong, M.A.: Algorithm AS136: A K-Means Clustering Algorithm. Applied Statistics 28, 100–108 (1979)
7. Hirano, S., Tsumoto, S.: Rough representation of a region of interest in medical images. Int. J. Approx. Reasoning 40(1-2), 23–34 (2005)
8. Ikonnomakis, N., Plataniotis, K.N., Venetsanopoulos, A.N.: Colour image segmenatiaon for multimedia applications. Jor. of Intelligent and Robotics Systems 28, 5–20 (2000)
9. Jain, A.K.: Data Clustering: 50 Years Beyond K-Means. Department of Computer Science & Engineering, Michigan State University, East Lansing, Michigan 48824 USA
10. Jain, A.K.: Fundamentals of Digital Image Processing, 1st edn. Pearson Education, India (2003)
11. Ji, Z., Sun, Q., Xia, Y., Chen, Q., Xia, D., Feng, D.: Generalized rough fuzzy c-means algorithm for brain MR image segmentation. Computer Methods and Programs in Biomedicine 108(2), 644–655 (2012)
12. Lingras, P., West, C.: Interval Set Clustering of Web Users with Rough K-Means. Journal of Intelligent Information Systems 23, 5–16 (2004)
13. Lingras, P.: Applications of rough set based K-means, kohonen SOM, GA clustering. In: Peters, J.F., Skowron, A., Marek, V.W., Orłowska, E., Słowiński, R., Ziarko, W.P. (eds.) Transactions on Rough Sets VII. LNCS, vol. 4400, pp. 120–139. Springer, Heidelberg (2007)

14. Lingras, P., Chen, M., Miao, D.: Precision of rough set clustering. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 369–378. Springer, Heidelberg (2008)
15. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)
16. Mohapatra, S., Patra, D., Kumar, K.: Unsupervised Leukocyte Image Segmentation Using Rough Fuzzy Clustering. In: International Scholarly Research Network, ISRN Artificial Intelligence (2012)
17. Nyma, A., Kang, M., Kwon, Y.-K., Kim, C.-H., Kim, J.-M.: A Hybrid Technique for Medical Image Segmentation. Journal of Biomedicine and Biotechnology, Article ID 830252 (2012), doi:10.1155/2012/830252
18. Miki, K., Patel, M.H.: Survey on Image Segmentation Using Different K-Mean Algorithms (2013) ISSN: 2277 - 8179
19. Peters, G.: Some Refinements of Rough k-Means. Pattern Recognition 39, 1481–1491 (2006)
20. Pawlak, Z.: Rough Set. Int. Jor. of Computer and Info. Sci. 11, 341–356 (1982)
21. Sharma, N., Bajpai, A., Litoriya, R.: Comparison the various clustering algorithms of weka tools. International Journal of Emerging Technology and Advanced Engineering 2(5) (2012)
22. Shi, Z., Chao, Y., He, L., Nakamaru, T., Itoh, H.: Rough set based FCM algorithm for image segmentation. Int. Jor. of Computational Sci. 1 (2007)
23. Srivastava, A., Asati, A., Bhattacharya, M.: A Fast and Noise-Adaptive Rough-Fuzzy Hybrid Algorithm for Medical Image Segmentation. In: 2010 IEEE International Conference on Bioinformatics and Biomedicine (2010)
24. Thilagamani, S., Shanthi, N.: A survey on image segmentation through clustering. Int. Jor. of Research and Reviews in Info. Sci. 1(1) (2011)
25. Tripathy, B.K., Ghosh, A.: Data Clustering Algorithms Using Rough Sets (2013)
26. Veltkamp, R.C., Tanase, M.: Content based image retrieval systems: a survey, Utrecht University (2000)
27. Venkateswaran, R., Muthukumar, S.: Genetic Approach on Medical Image Segmentation by Generalized Spatial Fuzzy C- Means Algorithm. In: IEEE International Conference on Computational Intelligence and Computing Research (2010)
28. Weifeng, D., Haiming, L., Yan, G., Dan, M.: Another kind of fuzzy rough set. IEEE International Conference Granular Computing 1, 145–148 (2005)
29. Zhang, Y.J.: A survey on evaluation methods for image segmentation. In: Int. Symposium on Signal Processing and Its Applications, Malaysia, August 13-16, pp. 13–16 (2001)
30. Zhong, N., Skowron, A.: A rough set-based knowledge discovery process. Comput. Sci. 11(3), 603–619 (2001)

Domain Adaptation for Pathologic Oscillations

Rory Lewis[1,2], Chad A. Mello[2], James Ellenberger[2], and Andrew M. White[1]

[1] Departments of Pediatrics & Neurology, University of Colorado Denver, Anschutz Medical Campus, Aurora, CO, 80045, USA
[2] Department of Computer Science, University of Colorado at Colorado Springs, Colorado Springs, CO USA 80918, USA

Abstract. This paper presents a platform to bridge datamining techniques and concepts in the field of neurosciences with state-of-the-art data mining, in particular domain adaptation. In non-clinical environs, once an exhaustive search for a particular item of knowledge seems to be impractical, there is the natural tendency to switch to heuristic methods to expedite the search. Conversely, when neuroscientists are in the same situation, they will trust exhaustive searches rather than heuristics such as clinical decision-support systems (CDSS). This is particularly when electroencephalography (EEG) sequences are used to search for pathologic oscillations in the brain. The purpose of this paper is to promising results illustrating how an intelligent agent can data mine explicit types of pathologic oscillations in the human brain.

1 Introduction

In previous using work the authors have shown that in a domain of time versus amplitudinal strength in EEGs of a person during seizure, the neural oscillations of artifact remain stationary in continuous clustered segments while the neural oscillations seizure activity move (*see* Figure 1) [10]. To validate this the authors used their rough set-based discretization and clustering tool called neuroClustering^{TM}. The reason why the clustering tool could create these three distinct clusters is because when the human brain is working efficiently there is a precise interaction of neural activities that renders oscillatory synchronization [13], [8], [3], [5], which creates a harmonic ebb and flow of electricity observable on an electroencephalogram (EEG). Conversely, when one develops a neurological pathology this synchronization breaks down. Discovering classifiers for these pathologic oscillations is crucial in finding their cures. This will only change when a CDSS incorporates intelligence that can learn in one domain and retrain itself across another domain. In the continuing goal to establish an autonomous machine-learning CDSS for detecting pathologic oscillations from a plurality of EEG domains for neuroscientists, the authors present a state-of-the-art rough set theory methodology that embraces domain adaptation. The paper will now illustrate how discretizing and clustering EEG signals in this manner is also conducive for domain adaptation. First we review neuroClustering^{TM}then we address the motivation for the experiments, domain adaptation, along with the experiment process and results.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 374–379, 2013.

neuroClustering^{TM}: The ability discretize large portions of signal based EEGs allows a machine to convert large portions of complex fourier transforms to small 2-dimensional arrays of x, y members. Now if this format is conducive for domain adaptation we can train off of neurodiagnostic domain consisting of many "patients" and quickly match the closest patients to a new patient. This would yield a new resource for studying the abnormal synchronization processes found in the pathologic oscillations associated with neuropsychiatric disorders. It is known in the field of neurodiagnostic studies, that proper overall interpretation of EEG findings rely on valid correlations of associated clinical semiology identifying specific oscillatory states [7]. Problems such as the subjective nature of what constitutes a seizure [16] and the insurmountable resources required to have machines learn and identify pathologic oscillations is unacceptable. To do this let $\mathbf{X} = x_1, x_2, ..., x_N$ represent the finite set and $2 \leq c \leq N$ is an integer. The objective was to partition data set $\mathbf{X}$ into c clusters where one assumes that c is known [15]. With classical sets one defines a hard partition as a family of subsets $\{A_i | 1 \leq i \leq c \subset P(X)\}$ where A_i jointly contain all the data in $\mathbf{X}$, which must be not empty, pairwise disjoint and $\bigcup_{i=1}^{c} A_i$ should reconstruct $\mathbf{X}$ [14]. We need μ_{ik} to attain real values in [0,1] therefore fuzzy partitioning is used as a generalization of hard partitioning where we let $\mathbf{X} = [x_1, x_1, ..., x_N]$ represent the finite set where $2 \leq c \leq N$ be an integer where the fuzzy partitioning space for $\mathbf{X}$ is the set $M_{fc} = \mathbf{u}_{ij} \in \mathbf{R}^{N \times c} | \mu_{ik} \in [0, 1], \forall i, k$; where the i-th column of $\mathbf{U}$ contains values of the membership function of the i-th fuzzy subset of $\mathbf{X}$ and constrains the sum of each column to 1. This means that total membership of each x_k in $\mathbf{X}$ equals 1 thus making the distribution of the memberships (artifact or seizure) amongst the c fuzzy subsets flexible in nature. For discretization, the authors used a quantitative indice to evaluate the efficiency of this rough clustering algorithm, incorporating the concepts of rough sets capturing the average degree of completeness of knowledge of all the clusters [11].

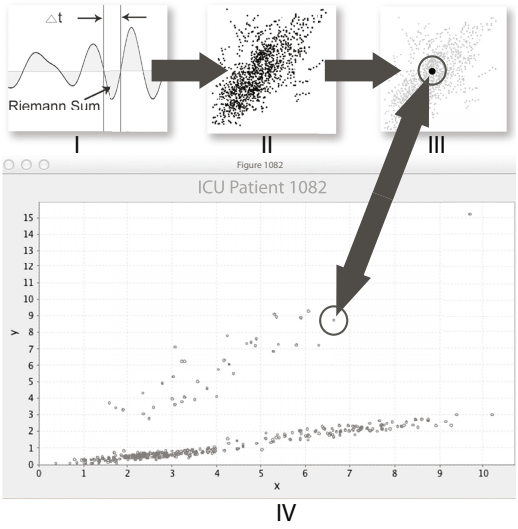


Fig. 1. neuroClustering^{TM}: (**I**) Original EEG is split with a spline to extract change of time and Riemann sum values. (**II**) Each point represents one instance of change of time versus Riemann during 0.333 seconds. (**III**) FCM centroid of the cluster in *II* is instantiated onto (**IV**) which has three distinct areas. Bottom right cluster rdefines artifact. Bottom left left is "normal" activity Upper cluster is seizure.

2 Domain Adaption Experiments

As seen in Figure 2(a), the classic setting for machine learning is single domain learning where the goal is to have an input x that predicts a corresponding output y, $x \rightarrow y$, where we assume that x and y are drawn against some probability distribution of the joint pair x and y, $(x, y) \sim Pr[x, y]$. For example, our x's may be a post neurosurgery ICU patient's (patient) EEG and our goal is to have a machine predict whether that patient is incurring a fatal pathologic oscillation. As the author has seen over the past few years applying rough set theory to patients in the neuro ICU ward where the goal is to learn what one, if any, of the hundreds of seizures a post brain surgery will have will be a fatal one - the type that shuts off bodily functions and only alert the neurosurgeon of this event and not the others, simply takes too long. By the time the machine has learned and trained off of the same patient the patient has left the hospital, the neuro surgeon has had 60 false positives a day or in one cade, the patient died.

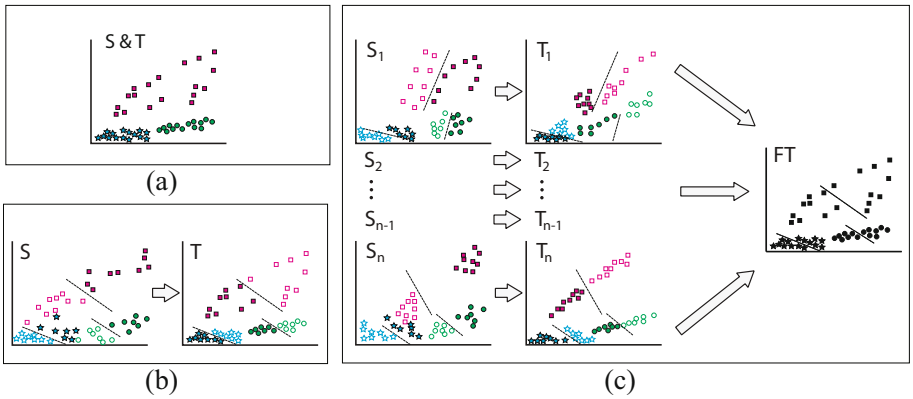
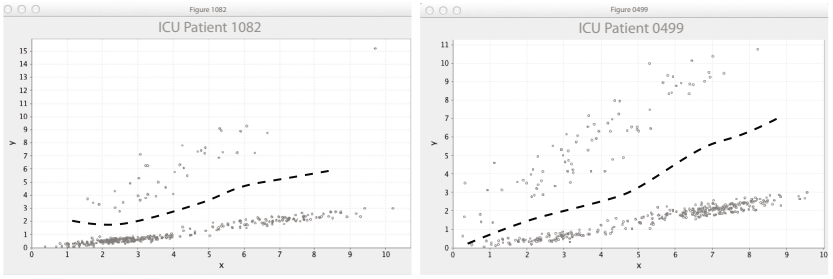


Fig. 2. *Domain Adaptation*™: (a) Classical, $D : \mathbb{R}^k \rightarrow \mathbb{R}$. (b) This paper, where domain adaptation where $D_1^{\mathbb{R}^K \rightarrow \mathbb{R}}$, and (c) Future work 4 $D_{w,x,y,z}^{\mathbb{R}^K \rightarrow \mathbb{R}}$.

Enter domain adaptation which is based off of its critical "Single Good Hypothesis"; $\exists h^*, \epsilon_S(h^*), \epsilon_T(h^*) small.$ which is in essence saying, for us in the neuro ICU ward that there exists, somewhere out there, a classification rule that is based off of previous patients (S) who were also in this neuro ICU ward, that correctly predicts a particular type of pathologic oscillation in both the training P_n's EEG data and the target P_{n+1}'s EEG data where the error is small. Note that in *Shared Support* the two distributions are similar while in *Shared Representation Learning* the two distributions are completely different.As shown in Figure 2(b) we now we have two distributions, a *source* (S) distribution and a *target* (T) distribution. The *source* distribution, $(x, y) \sim Pr_S[x, y]$, is the *training* distribution where classification rules are derived from many previous *patients* while the *target* (T) distribution, $(x, y) \sim Pr_T[x, y]$, is the *test* distribution which may elicit rules derived from the closest match of the previous

patients that were in the neuro ICU ward. Many methodologies can be used to do this including, Covariate Shift [6], Representation Learning [1], Feature Based Supervised Adaptation [4]and Parameter Based Supervised Adaptation [17], to name a few. This is represented in Figure 2 by the swapping of the colors representing the stars for normal, the circles for artifact and the squares for seizure, the colors of interior white and full colors in the three aforementioned shapes are swapped out across the cross/ cut lines between the S and T's. in Note that in Figure 2(a) the new patient when using classical KDD and applying domain adaptation terminology upon it, is actually is actually both the *source*(S) and the *target* (T) distribution.



(a) (b)

Fig. 3. EEG Clustering: 2 of 2000 ICU "patients": Dotted line bisects seizures above normal & artifact clusters below

The goal of the experiments was to recognize the confidence of classification rules of a plurality of synthesized patients according to how well their values distinguish between the instances of the same and different classes close to one another. Using Scala we created 2,000 patient's EEGs with some having no seizures, some have two seizures, some having no artifact and some have two sets of artifact (*see* Figure 3). We selected a cohort of 670 patients, P_{001}, P_{002} ... P_{670} to only be trained by using the classical iterations for learning and training. We set a predefined threshold from an expert epileptologist and our optimal subset was randomly chosen by selecting a random 33% of the values. We performed the iterations 6 times. Next, using a J48 Generator in Weka we submitted the training to a tester and then trained off of another randomly chosen 33% of the synthesized data [9] which is based off of Quinlan's original bagging and boosting methodology [12]. In the classic supervised learning context we determine how effective a system performs on test data by bounding the empirical error of the training data with the expected error on the test data $\epsilon_{\text{test}} \leq \hat{\epsilon}_{\text{train}} + \sqrt{\frac{\text{complexity}}{\text{n}}}$ where the complexity may be Girosi, Rademacher, etc., and as the number of training entities increases so do the test and training errors converge. In Domain Adaption however, the test error $\varepsilon_{\text{test}}$ is measured on a new distribution [2].

To test the hypothesis the authors ran J48 algorithms twice on two randomly selected sets of 670 synthesized patients. In $cohort_1$ six iterations of training and learning on 66 randomly selected patients off the same distribution were run using Naive Bayes (classical)(*see* dotted line in Figure 4). In $cohort_2$ six iterations of training off the source distribution were trained on 66 randomly selected patient from $cohort_2$ (*see* solid line in Figure 4) and learned on the 66 "new" patients.

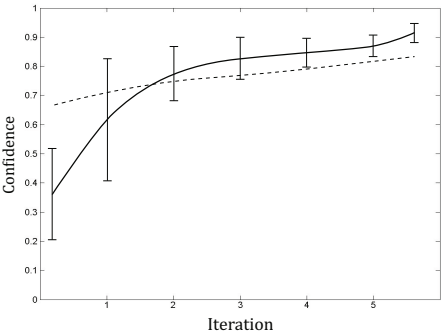


Fig. 4. *Initial Results*: Domain Adaptation (solid line) started off consistently below the classical KDD (dotted line's) confidence levels

3 Conclusions and Future Work

As seen in Figure 2(c), the future work is the motivation behind this paper. In (c) where iwe move towards a new class of machine learning that can **adapt**, and **select** all the closest S1's to T1s, **adapt**, and **select** all the closest S2's to T2's which could be animal databases, and then **adapt**, and **select** all the closest S3's to T3's which could be conversations between doctors. To make this model move closer towards how humans think, it must also apply many Sn's to the closest Tn's which could be YouTube, textbooks, strings, arrays, audio and ... essentially all inhomogenous databases that right now cannot possibly be **fused** to the Final Target FT. As seen in these experiment results, in all the domain adaptation cases, the results started lower than the classical KDD results varying from 21% to 53% on the first iteration but soon surpassed the classically trained patients' confidence levels. We are concerned that we do not understand exactly why the domain adaptation starts off lower. The focus in these experiments was to see if we could make domain adaptation work and compare it to a classical KDD approach - hence the focus was not the drilling down into the resultants each step of the way. The authors are already performing work to compare the same type of randomly selected patients to many more types and see what the results tell us. The end result is that domain adaptation is showing strong results and opening up many doors allowing machine learning to learn off of one distribution and train on another. More so, with more testing we may soon qualify to test these results on real human in the post neurosurgery ICU wards at Anschutz Medical School in early 2014.

References

1. Blitzer, J., McDonald, R., Pereira, F.: Domain adaptation with structural correspondence learning. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 120–128. Association for Computational Linguistics (2006)

2. Daumé III, H., Marcu, D.: Domain adaptation for statistical classifiers. Journal of Artificial Intelligence Research 26(1), 101–126 (2006)
3. Dumas, G., Nadel, J., Soussignan, R., Martinerie, J., Garnero, L.: Inter-brain synchronization during social interaction. PLoS One 5(8), e12166 (2010)
4. Evgeniou, T., Pontil, M.: Regularized multi–task learning. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 109–117. ACM (2004)
5. Ferrarelli, F., Sarasso, S., Guller, Y., Riedner, B.A., Peterson, M.J., Bellesi, M., Massimini, M., Postle, B.R., Tononi, G.: Reduced natural oscillatory frequency of frontal thalamocortical circuits in schizophrenia. Archives of General Psychiatry, pages, archgenpsychiatry–2012 (2012)
6. Heckman, J.J.: Sample selection bias as a specification error. Econometrica: Journal of the Econometric Society, 153–161 (1979)
7. Hogan, R.: Automated eeg detection algorithms and clinical semiology in epilepsy: Importance of correlations. Epilepsy & Behavior 22, S4–S6 (2011)
8. John, E., Prichep, L., Fridman, J., Easton, P.: Neurometrics: Computer-assisted differential diagnosis of brain dysfunctions. Science (1988)
9. Kohavi, R., Sommerfield, D., Dougherty, J.: Data mining using 𝓂 𝓁 𝒸++ a machine learning library in c++. In: Proceedings Eighth IEEE International Conference on Tools with Artificial Intelligence, pp. 234–245. IEEE (1996)
10. Lewis, R., Mello, C.A., Carlsen, J., Grabenstatter, H., Brooks-Kayal, A., White, A.M.: Autonomous neuroclustering of pathologic oscillations using discretized centroids. In: 8th International Conference on Mass Data Analysis of Images and Signals with Applications in Medicine, New York, USA, July 13-16 (2013)
11. Lingras, P., West, C.: Interval set clustering of web users with rough k-means. Journal of Intelligent Information Systems 23(1), 5–16 (2004)
12. Quinlan, J.R.: Bagging, boosting, and c4. 5. In: Proceedings of the National Conference on Artificial Intelligence, pp. 725–730 (1996)
13. Schnitzler, A., Gross, J.: Normal and pathological oscillatory communication in the brain. Nature Reviews Neuroscience 6(4), 285–296 (2005)
14. Setnes, M., Babuska, R.: Fuzzy relational classifier trained by fuzzy clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 29(5), 619–625 (1999)
15. Trinidad, J.F., Shulcloper, J.R., Corts, M.S.: Structuralization of universes. Fuzzy Sets and Systems 112(3), 485–500 (2000)
16. Williams, P.A., Hellier, J.L., White, A.M., Staley, K.J., Dudek, F.E.: Development of spontaneous seizures after experimental status epilepticus: Implications for understanding epileptogenesis. Epilepsia (Series 4) 48, 157–163 (2007)
17. Yu, K., Tresp, V., Schwaighofer, A.: Learning gaussian processes from multiple tasks. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 1012–1019. ACM (2005)

Discernibility in the Analysis of Binary Card Sort Data

Daryl H. Hepting* and Emad H. Almestadi

Department of Computer Science, University of Regina
3737 Wascana Parkway, Regina, SK, S4S 0A2 Canada
{hepting,almestae}@cs.uregina.ca

Abstract. In an open card sorting study of 356 facial photographs, each of 25 participants created an unconstrained number of piles. We consider all 63,190 possible pairs of photos: if both photos are in the same pile for a participant, we consider them as rated similar; otherwise we consider them as rated dissimilar. Each pair of photos is an attribute in an information system where the participants are the objects. We consider whether the attribute values permit accurate classification of the objects according to binary decision classes, without loss of generality. We propose a discernibility coefficient to measure the support of an attribute for classification according to a given decision class pair. We hypothesize that decision class pairs with the support of many attributes are more representative of the data than those with the support of few attributes. We present some computational experiments and discuss opportunities for future work.

1 Introduction

Card sorting [7] is an accessible technique to elicit data about participant impressions of various stimuli. We consider the analysis of data from a card sorting study of 356 facial photographs (178 Caucasian and 178 First Nations). The photographs were laminated on 5 by 4 inch cards. Participants were asked to view photos one at a time and place each photo on a pile with photos which they judged to be similar, without disturbing existing piles. The number of piles was not constrained. Within the 25 participants, the number of piles made ranged between 4 and 38. For each participant, photos in the same pile were considered to be rated as similar (distance of 0) and photos in different piles were considered to be rated as dissimilar (distance of 1). In this way, we attached a rating to each of the 63,190 pairs that can be made from 356 photos. Participants rated the similarity of each photo in relation to other photos. The smallest unit of this similarity judgement is the photo pair, so therefore the photo pairs are the attributes in this information system. Only a small fraction of these comparisons were made directly, specifically amongst the photo being placed and whichever photos were visible at the tops of existing piles. The study and a preliminary

* This paper benefitted from discussions with Dominik Ślęzak.

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 380–387, 2013.

analysis has been described elsewhere [4]. From that preliminary analysis, it was hypothesized that different strategies for sorting the photos may be used amongst the participants studied.

We continue to work at identifying and understanding the different strategies that may be at work. The earlier work looked for meaningful ways to distinguish between 2 groups. In particular, we looked at various qualities inherent in or identified about the photos as the basis for constructing decision class pairs. In this paper, we continue the search for identifiable strategies from a quantitative perspective. Although we still consider binary decision classes, each of these decision classes may be later further subdivided as required.

Gathering the ratings for each pair of photos (attribute) from each participant led to a binary vector of length 25 that became associated with the attribute. Some photos were not recorded during data entry, so the distance for pairs formed with these photos was -1. Our approach reported here replaced each -1 within these binary vectors with 0 and 1 in turn to generate all possible alternative patterns in new binary vectors. In cases when an attribute had an incomplete original binary vector (containing -1 values), the attribute became associated with all newly generated binary vectors. Any duplicate binary vectors were removed with the associated attributes moved to the single remaining instance of the vector. The result of this process was a list 28,379 unique binary vectors. Following Table 1, each of the vectors was assigned an ID. None of the unique vectors was the inverse of another vector in the list.

Table 1. Sample binary vectors. Each bit position represents a participant (object). The table shows the IDs associated with binary vectors: interpreted as integers, vectors are valued from 0 to $2^{n-1}-1$ on the left and from 2^n-1 to 2^{n-1} on the right. Interpreted as a decision class specification, all objects with the same value are assigned to the same decision class. Therefore, a vector and its inverse have the same ID. The first row does not have an ID because the vector and its inverse do not contain both 0 and 1.

ID	Binary Vector	Inverse Vector
-	000	111
1	001	110
2	010	101
3	011	100

Each bit position in the binary vector represents a participant (object). These binary vectors have 2 possible interpretations. On the one hand, each vector represents the values for a particular attribute. A zero (0) indicates that the particular participant judged the photo pair to be similar (distance = 0). A one (1) indicates that the particular participant judged the photo pair to be dissimilar (distance = 1). On the other hand, each vector represents a possible way to assign the objects into 2 decision classes. Participants with the same value are assigned to the same decision class. (See Table 1 for more detail.)

The unique vectors distilled from participant data represent only a very small fraction of the total possible ways to divide 25 participants into 2 groups, yet

they are an appealing starting point because they record real participant behaviour. If we look amongst them for evidence of differing strategies employed by participants in the judgement of facial similiarity, we may be encouraged to find attributes that allow a highly accurate classifier to be built for a particular binary decision class specification. We suggest that this is a necessary but not a sufficient criterion for identification of "good" decision class pairs. We suggest that a better indication of "good"-ness for a decision class pair is the number of attributes from which a highly accurate classifier can be built.

Our approach, reported here, has been to develop a measure of discernibility that can be easily computed and used to quantitatively assess how well a particular attribute can be used to discern objects according to a given decision class pair. These results were calibrated in a small test with the Rough Set Exploration System [2].

The rest of the paper is organized in the following way. Section 2 discusses discernibility and develops new measures related to discernibility. Section 3 details some computational experiments, including the use of the Rough Set Exploration System [2]. Section 4 presents some conclusions based on the obtained results and discusses some opportunities for future work.

2 Discernibility

Discernibility is a key idea in rough set theory [6,8], and it can be applied here to understand participant judgements in 2 ways:

- by examining all judgements made by pairs of participants (objects): It is possible for a pair of participants to disagree about every attribute, in which case the participants would be readily discernible. It is also possible for a pair of participants to agree about every attribute, in which case the participants would be indiscernible.
- by examining all judgements made about each attribute: It is possible for all participants to agree with each other about an attribute, in which case the attribute would not contribute to the discernibility of the participants. It is not possible for all participants to disagree with each other about an attribute, because each participant rates an attribute as either "Similar" (0) or "Dissimilar" (1). For a given vector, the product of the number of 0's and the number of 1's indicates the amount of "disagreement" (discernibility). Equation 1 defines the maximum discernibility possible within a binary vector of length n.

We focus our attention here on those vectors with maximum discernibility (which contain either 12 zeroes and 13 ones or 13 zeroes and 12 ones). In this way, we hope to focus on the most informative attributes [1]. By doing so, we are left with 1705 vectors out of the total 28,379 with which we began.

In these vectors, 156 out of the 300 possible pairs of participants are different (either 01 or 10) and only 300 - 156 = 144 of the possible pairs of participants are the same (either 00 or 11). Beginning with a vector that specifies the binary

decision classes, we wish to compare it with attribute vectors to see how well the decision class pair represents the observed data.

Choi *et al.* [3] present 75 different ways to assess the similarity between 2 binary vectors. The task of assessing the discernibility of a binary vector with respect to another is somewhat different. As outlined in Table 1, we consider that a vector and its inverse represent the same assignment of objects to decision classes. This interpretation is different than Janusz and Ślęzak [5], for example, who regarded inverse vectors as complementary rather than similar. In our case, we are concerned with values on diagonals of the contingency table (see Table 2).

Table 2. Contingency table consistent with Choi *et al.* [3]. Rows labelled as x_0 and x_1 indicate respectively 0's and 1's in vector x. Columns labelled as y_0 and y_1 indicate respectively 0's and 1's in vector y. $D_{\text{coeff}}(x, y) = 1$ if $a + d = n$ or $b + c = n$.

	y_0	y_1	sum
x_0	a	c	$a + c$
x_1	b	d	$b + d$
sum	$a + b$	$c + d$	$a + b + c + d = n$

$$D_{\max} = \begin{cases} (\frac{n}{2})^2 & \text{when } n \text{ is even} \\ (\frac{n}{2}) \times ((\frac{n}{2}) + 1) & \text{when } n \text{ is odd} \end{cases} \tag{1}$$

$$D_{\text{coeff}}(x, y) = \frac{ad + bc}{D_{\max}} \tag{2}$$

$$D_{\text{dist}}(x, y) = 1 - D_{\text{coeff}}(x, y) \tag{3}$$

Given n objects, there will be $\binom{n}{2}$ pairs of objects. Consider that each of these objects is assigned to 1 of 2 decision classes. Pairs of objects from different decision classes will be discernible with respect to an attribute if the values for that attribute are different for these pairs of objects. Equation 1 defines the maximum number of object pairs with objects from different decision classes. $D_{\text{coeff}}(x, y)$, defined in Equation 2, compares 2 binary vectors (one a decision class specification and the other containing attribute values) and computes the number object pairs from different decision classes that have different attribute values over the maximum number of such pairs. The range for the $D_{\text{coeff}}(x, y)$ is $[0, 1]$. Notice that $D_{\text{coeff}}(x, y) = D_{\text{coeff}}(y, x)$ for any pair of vectors, x and y. The coefficient is meant to answer the question "Does the attribute with values given by x help to discern objects in decision classes specified by y?" If $D_{\text{coeff}} = 1$ (or close to it), the answer is "Yes". Either ad or $bc = D_{\max}$, which means that the attribute values match the decision class specification (or its inverse) exactly (what was earlier called a "splitting pair" [4]). If $D_{\text{coeff}} = 0$ (or close to it), the answer is "No". Either ac or $bd = D_{\max}$, and the attribute contributes nothing to the discernibility of the decision classes. This value will only occur if all of the attribute values are the same. Equation 3 defines a distance in terms of D_{coeff}.

Table 3. Three sample binary vectors, labelled as A, B, and C for brevity, are compared. (Following the convention outlined in Table 1, their numerical IDs are as follows: A = 350655, B = 350639, and C = 11184810.) To the right of each pair of vectors is the contingency table for the comparison. $D_{\text{coeff}}(A, B) = 144/156 = 0.923$ and $D_{\text{coeff}}(A, C) = (56 + 25)/156 = 0.519$.

ID	Values																							
A	0	0	0	0	0	0	1	0	1	0	1	0	1	1	0	0	1	1	0	1	1	1	1	1
B	1	1	1	1	1	1	0	1	0	1	0	1	0	0	1	1	0	0	1	0	1	0	0	0

	B_0	B_1
A_0	0	12
A_1	12	1

ID	Values																							
A	0	0	0	0	0	0	1	0	1	0	1	0	1	1	0	0	1	1	0	1	1	1	1	1
C	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

	C_0	C_1
A_0	7	5
A_1	5	8

3 Experimentation

Each of the 1705 vectors, in turn, was interpreted as the decision class specification, in preparing to apply the rough set attribute reduction methodology [6]. D_{coeff} was computed for all attribute vectors with respect to the given decision class specification, and the average coefficient was computed for each candidate decision class specification. We then chose the vectors with the maximum and minimum average, represented in Table 4. In addition to the interpretation of support for a decision class specification, the average coefficient can also be interpreted as a measure of the importance of the attribute(s) associated with each vector. In this case, the coefficient answers the question "Is this attribute important in discerning objects?" If $D_{\text{coeff}} = 1$ (or close to it), the answer is "Yes". If $D_{\text{coeff}} = 0$ (or close to it), the answer is "No".

Table 4. Max. (D_{coeff} average = 0.590) and Min. (D_{coeff} average = 0.515) vectors, compared. $D_{\text{coeff}}(max, min) = 0.5$.

ID	Values																							
Max.	1	1	1	1	1	1	0	1	0	1	0	1	0	0	1	1	0	0	1	0	0	0	0	0
Min.	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	0	1	1	0	1	0	0	1

	Max_0	Max_1
Min_0	7	6
Min_1	6	6

Hepting *et al.* [4] focused on reducing the number of attributes required as input to RSES [2] in order to accurately classify participants according to a decision class pair. Instead of looking only for the existence of an accurate classification via RSES, this work is concerned with exploring the limits of an accurate classification: how many different attributes support accurate classification according to a specified decision class pair? We hypothesize that candidate decision class pairs with the support of many attributes are more representative of the data than those with the support of few attributes.

Table 5. Summary of results from 2 sets (Max., left and Min., right) of runs of RSES. Data from each bin was run 10 times and averages are reported. Dashes indicate that the bin had no data.

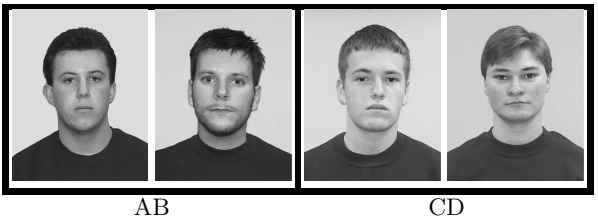
Bin	Nbr. Attr.	Avg. Coeff.	Avg. Acc.	Std. Dev.	Avg. Red.	Avg. Rule
0.95	0	-	-	-	-	-
0.90	5	0.923	1	0	1.32	12
0.85	15	0.853	0.977	0.037	1.88	33.8
0.80	25	0.845	0.992	0.024	1.61	29.4
0.75	25	0.788	0.969	0.040	1.85	35.8
0.70	25	0.731	0.915	0.067	2.46	51
0.65	25	0.692	0.862	0.087	1.92	37.6
0.60	25	0.636	0.846	0.103	2.9	65.7
0.55	25	0.596	0.808	0.075	2.6	54.8
0.50	25	0.545	0.746	0.115	3.53	85

Max.

Bin	Nbr. Attr.	Avg. Coeff.	Avg. Acc.	Std. Dev.	Avg. Red.	Avg. Rule
0.95	0	-	-	-	-	-
0.90	0	-	-	-	-	-
0.85	0	-	-	-	-	-
0.80	0	-	-	-	-	-
0.75	0	-	-	-	-	-
0.70	0	-	-	-	-	-
0.65	5	0.673	0.823	0.089	2.36	19.4
0.60	25	0.634	0.838	0.133	2.67	57.1
0.55	25	0.596	0.769	0.103	2.84	62.8
0.50	25	0.545	0.662	0.121	3.64	82.2

Min.

For both of the Min. and Max. vectors, we created bins for D_{coeff} between 0.5 and 1.0 in increments of 0.05. RSES input files were generated to test the classification accuracy using up to 25 attributes from only 1 specified bin. The bins, the number of attributes in each bin, and the average coefficient value for the bin are indicated in Table 5.



AB CD

EF AG

Rule			Dec. Class	Nbr. Classified
EF = Similar	&	AG = Similar	0	11
EF = Similar	&	AG = Dissimilar	0	1
EF = Dissimilar	&	AG= Similar	0	1
EF = Dissimilar	&	AG = Dissimilar	1	12

Fig. 1. Photo pairs AB and CD are associated with vector "Max". Pairs EF and AG are the attributes in one of the reducts, followed by corresponding rules for classification.

Rule						Dec. Class	Nbr. Classified
JI = Similar	&	KL = Similar	&	MN = Dissimilar		0	2
JI = Similar	&	KL = Dissimilar	&	MN = Dissimilar		0	8
JI = Dissimilar	&	KL = Dissimilar	&	MN = Dissimilar		0	3
JI = Similar	&	KL = Dissimilar	&	MN = Similar		1	2
JI = Dissimilar	&	KL = Similar	&	MN = Similar		1	5
JI = Dissimilar	&	KL = Similar	&	MN = Dissimilar		1	4
JI = Dissimilar	&	KL = Dissimilar	&	MN = Similar		1	1

Fig. 2. Photo pair HI is associated with vector "Min". Pairs JI, KL, and MN are the attributes in one of the reducts, followed by corresponding rules for classification.

After creating the various input files, we followed a standard procedure with RSES [2], as follows: Preprocessing: split the input table of 25 objects into 2 equal parts (1 for training and 1 for testing); Training: calculate up to 10 reducts from the training data using the genetic algorithm in RSES; Testing: generate rules from the reducts and test the results by classifying the testing data. Each input file was processed 10 times and the averages are reported in Table 5: average accuracy (including standard deviation), average reduct length, and average number of rules. All results had 100 percent coverage, which means that the classifier based on the reducts generated from an ensemble of reducts was able to recognize everything, which is valuable in itself.

To explore some of the data in Table 5 in more detail, each of Figure 1 (for the Max. vector) and Figure 2 (for the Min. vector) illustrate the attribute(s) associated with the vector, a reduct generated from the whole (unsplit) data table taken from the respective top bin , and the rules associated with that reduct.

4 Conclusions and Future Work

The computation of the D_{coeff} is an appealing approach to understanding the structure of results of card sorting exercises because it can be done very quickly. The limited experiment presented here has provided encouraging support for our hypothesis, but more work needs to be done. For example, for the same average

coefficient value, is it better to have fewer attributes with higher coefficient or more attributes with lower coefficient values?

Many of the vectors with highest average coefficients are close to each other. This leads to opportunities to analyze the structure of the decision classes (representing strategies) that are best-supported by the data. By the same token, the similarity of the photo pair attributes associated with the Max. and Min. vectors respectively are noticeably different - something that provides more support for this approach.

In hindsight, it is becoming clear that too many (356) photos were used in the original sorting study. The process outlined here has the potential to sharply reduce the number of photos considered. If this process can successfully determine important attributes (such as photo pairs AB and CD in Figure 1), it may be possible to effectively run card sorting studies with a large number of stimuli that could be reduced based on this kind of quantitative analysis.

It is not possible to assess how well 2 decision classes are formed without testing all potential decision classes. There are 16,777,215 ($2^{24} - 1$) ways to create 2 decision classes for 25 participants, and the inexpensive computation of D_{coeff} can facilitate their review.

Acknowledgements. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. Emad Almestadi acknowledges the Ministry of Higher Education in Saudi Arabia and the Saudi Arabian Cultural Bureau in Canada for their support. The comments of the anonymous reviewers were very helpful in improving the final version of this paper.

References

1. Bazan, J.G., Nguyen, H.S., Nguyen, S.H., Synak, P., Wróblewski, J.: Rough set algorithms in classification problem. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) Rough Set Methods and Applications. STUD FUZZ, vol. 56, pp. 49–88. Physica-Verlag HD (2000)
2. Bazan, J.G., Szczuka, M.: The rough set exploration system. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 37–56. Springer, Heidelberg (2005)
3. Choi, S.S., Cha, S.H., Tappert, C.C.: A survey of binary similarity and distance measures. Journal on Systemics, Cybernetics and Informatics 8(1), 43–48 (2010)
4. Hepting, D.H., Spring, R., Ślęzak, D.: A rough set exploration of facial similarity judgements. In: Peters, J.F., Skowron, A., Sakai, H., Chakraborty, M.K., Slezak, D., Hassanien, A.E., Zhu, W. (eds.) Transactions on Rough Sets XIV. LNCS, vol. 6600, pp. 81–99. Springer, Heidelberg (2011)
5. Janusz, A., Ślęzak, D.: Utilization of attribute clustering methods for scalable computation of reducts from high-dimensional data. In: 2012 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 295–302 (2012)
6. Pawlak, Z.: Rough set approach to knowledge-based decision support. European Journal of Operational Research 99(1), 48–57 (1997)
7. Rugg, G., McGeorge, P.: The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts. Expert Systems 22(3), 94–107 (2005)
8. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Slowinski, R. (ed.) Intelligent Decision Support: Handbook of Applications and Advances in Rough Set Theory, vol. 11, pp. 259–300. Kluwer Academic Publishers (1992)

An Unsupervised Deep-Learning Architecture That Can Reconstruct Paired Images

Ti Wang, Mohammed Shameer Iqbal, and Daniel L. Silver

Jodrey School of Computer Science, Acadia University
Wolfville, NS, Canada B4P 2R6
danny.silver@acadiau.ca

Abstract. This paper presents an unsupervised learning system that develops an associative memory structure that combines two or more channels of input/output such that input on one channel will correctly generate the associated response at the other channel and *vice versa*. A deep learning architecture is described that can reconstruct an image of a MNIST handwritten digit from another paired handwritten digit image. In this way, the system develops a kind of supervised classification model meant to simulate aspects of human associative memory. The system uses stacked layers of unsupervised Restricted Boltzmann Machines connected by a hybrid associative-supervised top layer to ensure the development of a set of high-level features that can reconstruct one image given another in either direction. Experimentation shows that the system reconstructs accurate matching paired-images that compares favourably to a back-propagation network solution.

1 Introduction

Humans learn knowledge by experiencing the world through their senses. Raw data is received at one or more sensory organs, such as the eyes and ears, and related signals are pass to the nervous system. The exact mechanism by which these experiences affect the structure of the human nervous system and how new memory is formed is not well understood [5]. This is a primary goal of research in neuroscience and artificial intelligence, particularly those working in the area of computational learning.

Deep learning architectures, or DLA, provide an exciting new substrate upon which to explore possible computational and representational models of how knowledge is acquired, consolidated and used [1]. Prior work has investigated the use of DLAs and unsupervised learning methods to develop models for a variety of purposes including auto-associative memory, pattern completion, and clustering as well as generalization and classification [3].

Our long-term research objective is to create a system that is capable of "showing us what it hears and telling us what it sees" using a DLA. This will require an architecture that can work with three sensory and motor modalities: audio, optical, and vocal. This program of study is meant to accomplish several objectives. Chief among these is the investigation of unsupervised learning methods that can create a model capable of generalization and classification from

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 388–396, 2013.

one input modality to another (eg. from optical to vocal). We are interested in how this can be done without resorting to any form of supervised learning. We are also interested in the abstract layers of features generated in a DLA for one modality channel and at the intersection of two or more channels – How do these features compare to what we know of the human nervous system? Finally, we are interested in knowledge transfer in a DLA using unsupervised methods for learning new tasks and new modalities.

In this paper we take a first step by examining a DLA that is capable of learning paired-associate images at two input channels. The DLA must reconstruct the matching image at channel A when it observes a paired image at channel B, and *vice versa*. By doing so the system uses unsupervised learning to develop an associative memory model that performs a form of classification from one channel to another. The system uses layers of Restricted Boltzmann Machine (RBM) machines stacked into a DLA. We will show that such a DLA can work quite well when assisted with supervised learning at only the highest level representation. Experimentation shows qualitatively and quantitatively that the system generates reasonably accurate matching images, as compared to a traditional Back-Propagation (BP) network solution.

2 Background

Artificial Neural Networks (ANN) are one of the most commonly used machine learning techniques. Although a variety of ANNs are used in modeling highly complex tasks like image recognition, many do not work in the same fashion as the human nervous system. For example, supervised BP ANNs are good for modeling complex mapping relations between input and output domains, but are not so good for recalling input patterns. Humans have the capability of recovering complete information, from partial information, using associative memory. When a child learns the characteristics of a cat, he or she learns both the appearance of the cat as well as the sound it makes. Later, on seeing a picture of a cat, the child can recall the sound a cat makes [5]. An associative ANN simulates aspects of how collections of neurons store and recall associative memories. Geoffrey Hinton, University of Toronto, advocates using Boltzmann Machine associative networks to simulating human brain structure [3]. After a Boltzmann Machine has been trained on a set of patterns, it has the ability to reconstruct one of those patterns from a partial or noisy version of the pattern.

Boltzmann Machines: A Boltzmann Machine (BM) is a stochastic neural network of binary neurons that is capable of reconstructing a stored pattern from a partial pattern [2]. A BM is made up of two layers of binary neurons, or units, that are either visible or hidden. All the neurons in the visible and hidden layers are inter-connected forming a complete graph. Given some input on its visible units, a BM will settle into an equilibrium state with energy $E = \sum_i E_i$; where $E_i = -\sum_{i \neq j} s_i s_j w_{ij} - b_i s_i$, where s_i and s_j are states of two neurons, i and j, w_{ij} is the weight of the connection between them and b_i is the bias weight for neuron i [2]. After being trained, the BM will settle into a memory

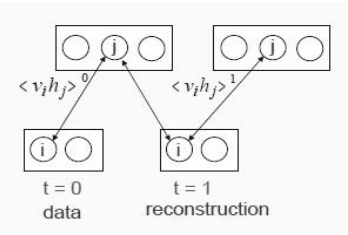


Fig. 1. RBM Training Process

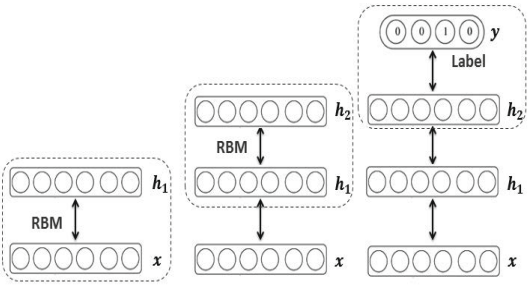


Fig. 2. Stacking Multi-level RBMs

state at equilibrium closest to the initial state of the neurons [2]. The activation function of a BM converts a weighted input and a temperature parameter, T, to a probability given by $p_{i=on} = \frac{1}{1+exp(-\frac{\Delta E_i}{T})}$ [2]. The neuron only comes on if its probability is greater than a random value. The energy E of the BM is affected by the global temperature T value that declines from a maximum value to 1 based on a predetermined schedule [2]. This technique helps the system from getting stuck in a local minima during the early stages of recall. As the temperature reduces to $T = 1$ the system moves towards a state of equilibrium, which will reconstruct the nearest stored pattern.

Learning is slow in BMs that have many hidden nodes. This is because the weight update equation requires sampling each neuron i for each training example, and then sampling the states of all other neurons j in order to compute E_i. The algorithm continues until the network reaches a state of equilibrium where its change in state is below a threshold.

Restricted Boltzmann Machines (RBM): An RBM is a variant of a BM that is meant to overcome the problem of long training times by limiting the number of connections in its network and using an approximate weight update algorithm. RBMs have both visible and hidden layers of neurons just like BMs, however all intra-layer connections are restricted [3]. When training data x_i is given to the visible neurons v_i, the RBM temporarily clamps their states and frees the states of hidden binary neurons h_j. Node h_j turns on with probability $p_j = \frac{1}{1+exp(-b_j-\sum_i w_{ij} v_i)}$. The visible units are then unclamped and node v_i turns on with probability $p_i = \frac{1}{1+exp(-b_i-\sum_j w_{ij} h_j)}$. The system computes the overall energy $E = -\sum_i b_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i h_j w_{ij}$ where b_i and b_j are the bias terms for their respective nodes [2]. The RBM computes the mean squared error (MSE) between the reconstructed input value x'_i and the original input value x_i and reduces it with a gradient-decent algorithm that changes the weights, w_{ij}. The state h_i of hidden neuron i keeps changing with i's probably p_i during training, and weight w_{ij} updates until either the global energy E or the probably p_i exceeds a threshold. At any point in time, with probability p_i, neuron i will reconstruct the input data x_i.

As shown in Figure 1, the weights are updated as per the following formula $\Delta w_{ij} = \epsilon(< v_i h_j >^0 - < v_i h_j >^1)$, which is an approximation of the gradient of the log likelihood [3]. This method of weight update is called contrastive divergence (CD). The CD algorithm is guaranteed not to get stuck in a local minima. The system is trained until the hidden layer is capable of reconstructing the original input pattern at the visible units to the desired level of accuracy. After training, the hidden layer weights of the RBM have learned the feature distribution of the input space, that is w_{ij}, gives the probability of feature h_j given input v_i.

Deep Learning Architectures: Most objects are made up of several other smaller parts or features. For example, a car is a combination of smaller features like wheels and a frame. Breaking it down further, a wheel is made up of smaller features like tires and rims. The higher-level abstraction is a car, whereas, the lower-level abstraction is a tire. Deep learning methods aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features [1,7].

One of the advantages of RBMs is that they can be stacked as layers to learn high level features of input data. As shown in Figure 4 the hidden layer of one RBM can be used as the input layer for a second RBM [1]. This second RBM layer will learn the feature distribution of the hidden layer of the first RBM. As layers are stacked the network learns increasing complex combinations of features from the original data.

These systems are capable of doing unsupervised *clustering* of unlabeled data based on a hierarchy of features. Hence, it is called *deep learning* or *deep feature learning*. Neuroscience studies have shown that the mammalian brain has a deep learning architecture with multiple levels of abstraction corresponding to different areas of the neocortex [6]. Many feel that RBM deep learning architectures develop a hierarchy of features in a fashion similar to the mammalian brain. Hinton has presented research on recognizing hand-writing images of digits, which simulates human vision, by using stacked RBMs [3].

3 Theory

The objective of this research is to develop a learning system that can memorize and recall knowledge using an associative memory network. The learning system should be able to recall the pattern from the associative network on one sensory modality given data on another sensory modality. The network will be trained such that when it is given an image, it will generate an associated image and in this way indicate the classification of the first image.

To achieve this goal, instead of using traditional labeled datasets, two or more unlabeled datasets are used to support unsupervised feature generation. The deep learning architecture (DLA) of the learning system is composed of two major parts, a hybrid associative-supervised memory network and two or more associative sensory channel networks (see Figure 3). The sensory channel networks are designed for the reconstruction of incoming sensory data. The hybrid

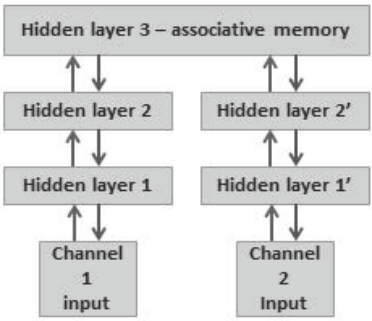


Fig. 3. Two channels DLA

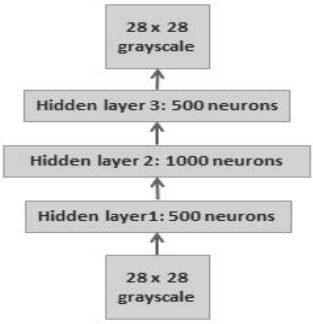


Fig. 4. BP ANN used in Experiment 1

associative-supervised memory network, which ties the sensory channel networks together, can be modeled with an RBM associative network [4].

The associative memory at the top of the DLA shown in Figure 3 simulates a human's long-term memory that combines separate channel features. The DLA will be given a variety of paired-associate handwritten digit images to learn. The challenge for our DLA at the top level is to create features of the digit images for one channel when presented with the only the features of the other channel [4]. To develop a more accurate model, we currently untie the associative memory weights and use the BP algorithm to fine-tune them using the posterior probabilities gathered from hidden layer 2 and 2'. When training to generate features of channel 2, the BP algorithm uses posteriors at hidden layer 2 as the inputs, and posteriors at hidden layer 2' as the supervised signal, and *vice versa*.

4 Experiment 1

Two empirical studies were carried out using two different data sets. The first experiment used the MNIST handwritten dataset. The second experiment used a synthetic dataset of handwritten digits. In both experiments, five pairs of odd and even digits were associated with each other 1-2, 3-4, 5-6, 7-8, and 9-0. A model using the DLA architecture described in Section 3 and two standard BP networks were trained and compared. One BP network is used for mapping from odd to even digits and another BP network is used for mapping from even to odd digits. Both methods were challenged to reconstruct the image of one digit from its paired-associate image.

Objective: The objective of this experiment is to compare unsupervised DLA with a supervised BP ANN approach to learning paired-associate images. As shown in Figure 3, each learning system is trained such that when a handwritten digit image is provided, the system will generate its paired digit image.

Material and Methods: This experiment uses a dataset of paired 28 x 28 gray-scale images of handwritten digits from MNIST database [3] as described above.

Table 1. Percent accuracy of test set reconstruction

Algorithm	1→2	2→1	3→4	4→3	5→6	6→5	7→8	8→7	9→0	0→9	Average
DLA	93.5	98.0	75.5	96.5	90.0	87.0	88.5	91.0	92.0	92.0	90.4
BP ANN	100.0	43.5	89.5	92.0	88.5	96.0	88.5	88.0	91.5	94.0	81.6

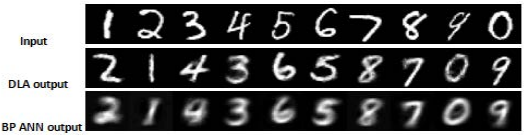


Fig. 5. Examples of reconstruction results from DLA and BP ANN

A dataset of 5000 examples is used to train the learning system. The training process stops when the maximum iteration (300) is reached or MSE exceeds the pre-set threshold. We use another set of 1000 examples as the validation set to monitor the BP fine-tuning training to avoid under-fitting and over-fitting. An independent set of 1000 examples is used as a test set. The odd digit image of a test example is used to test the reconstruction of its corresponding even digit image, and *vice versa*.

A deep learning architecture of RBMs is used to develop an unsupervised learning model for the problem. The architecture is in accord with Figure 3. A channel network is composed of two RBM layers, each of which contains 500 hidden neurons. Successively, hidden layers 1 and 1' and then layers 2 and 2' will develop more abstract features of the original images [3]. Hidden layers 1 and 2 will learn a generative DLA representation of the odd digits. Hidden layers 1' and 2' will learn a generative DLA representation of the even digits. The associative top layer contains 3000 neurons. It will bring together the features of layers 2 and 2' to create mapping functions that can reconstruct an image on one channel from the image on the other.

We developed two BP networks to learn the same paired-associate mapping. One network is trained to map odd digit images to even digits, the other *vice versa*. Both BP networks use the architecture shown in Figure 4. The BP network uses the same training set, validation set and testing set as the DLA.

The accuracy of reconstruction is measured by testing the output images using Hinton's DLA handwritten digits classification software. This software is known to classify MNIST dataset of handwritten digits with only 1.15% errors [3]. We passed the input images and the reconstructed images through Hinton's software to determine their accuracy. We note that the accuracy of Hinton's classification software is high because it was developed by using the BP algorithm to fine-tune all the weights of a DLA to classify an image. Our work is focused on generating paired images without little or no supervised learning.

Results and Discussion: Using Hinton's software, we tested reconstruction on the testing set. The results are shown in Table 1. On average, the DLA model generated images that were 90.4% accurate, and the BP ANN generated images

Fig. 6. The templates for each digit

Fig. 7. Examples of digits with 10% noise

that were 81.6% accurate. Figure 5 shows examples of reconstruction done by the DLA and BP ANN. One can see that the images generated by the DLA are clearer than those generated by the BP ANN. We conjecture that the DLA is able to better differentiate features from noise as compared to the BP network. We designed Experiment 2 to investigate this further.

5 Experiment 2

Objective: The objective of this experiment is to compare the DLA method to the BP ANN in overcoming noise injected into synthetic training examples. The DLA in this study uses only the unsupervised CD algorithm to train it's model.

Material and Methods: This experiment uses a synthetic dataset that contains five different categories of 10 x 5 paired images, similar to that used in Experiment 1 and shown in Figure 6. To create a variety of examples such as shown in Figure 7, 10% random noise was added to each template image to produce 20 instances of each digit, or 200 in total. The first 100 of these images are used as a training set while the remaining 100 is used as a test set.

A deep learning architecture of RBMs, in accord with Figure 3, is used to develop an unsupervised learning model. To achieve our goal of using a purely unsupervised DLA, we stack a 3-layer RBM to model the associative memory network instead of using a hybrid associative-supervised RBM. Each of these layers contains 100 hidden neurons. The training process stops when the maximum iteration (100) is reached. As in Experiment 1, we developed two BP networks to learn the same paired-associate mapping. Both BP networks used the architecture shown in Figure 4 with 40 neurons in the layer 1 and 3 and 20 neurons in layer 2. The BP network uses the same training set and testing set as the DLA, and 30 of the 100 examples from the training set is used as the validation set.

The accuracy of reconstruction was measured by comparing the similarity between reconstructed images and their corresponding target template images. We compute the pixel root mean square error (RMSE) between the generated image and its corresponding data template (without noise). The RMSE gives an average difference between corresponding pixels in these two images.

Results and Discussion: The RMSE of the reconstruction images is shown in Table 2. The DLA out-performs the BP network in generating the images in the presence of noise. Figure 8 and 9 show a set of example digit images reconstructed by the DLA.

Table 2. Percent accuracy of test set reconstruction

Algorithm	1→2	2→1	3→4	4→3	5→6	6→5	7→8	8→7	9→0	0→9	Average
DLA	95.30	93.52	94.15	93.65	94.01	94.99	94.86	87.96	94.5	94	93.7
BP ANN	74.61	78.33	73.59	82.06	76.73	77.07	70.31	77.98	79.45	71.18	75.75

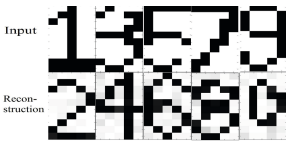


Fig. 8. Reconstruction of even digits

Fig. 9. Reconstruction of odd digits

6 Conclusion

We have presented work on an unsupervised learning system that is able to develop an associative memory structure that combines two or more channels of input or output. Our desire is to have the input on one channel correctly generate the associated response at the other channel and vice versa. Our long-term goal is to develop learning systems that are able to learn the relationships between different sensory input and/or motor output modalities similar to humans.

In this paper we present a deep learning architecture (DLA) that can reconstruct an image of a MNIST handwritten digit from another paired handwritten digit and *vice versa*. In this way, the system develops a kind of supervised classification model meant to simulate aspects of human associative memory. The system uses stacked layers of unsupervised Restricted Boltzmann Machines (RBM) connected by a hybrid associative-supervised top layer to ensure the development of a set of high-level features that can reconstruct one image when given another in either direction. Experimentation shows qualitatively (by viewing the generated images) and quantitatively (test set statistics) that the system reconstructs reasonably accurate matching images that compare favourably to a back-propagation network solution.

In future work, a full Boltzmann Machine will be used as the top-level associative memory replacing the BP fine-tuning of the current RBM top layer weights. In the long term, the DLA will be expanded to generate sound when provided an image or conversely generate an image when it hears a sound.

References

1. Bengio, Y.: Learning deep architectures for AI. Foundations and Trends in Machine Learning 2(1), 1–127 (2009)
2. Hinton, G.E., Sejnowski, T.J.: Parallel distributed processing: explorations in the microstructure of cognition. In: Learning and Relearning in Boltzmann Machines, chapter, vol. 1, pp. 282–317. MIT Press, Cambridge (1986)

3. Hinton, G.E., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Comput. 18(7), 1527–1554 (2006)
4. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: ICML 2011, pp. 689–696 (2011)
5. Rosenzweig, M.R.: Experience, memory, and the brain. American Psychologist 39(4) (April 1984)
6. Serre, T., Kreiman, G., Kouh, M., Cadieu, C., Knoblich, U., Poggio, T.: A quantitative theory of immediate visual recognition. In: Progress in Brain Research, pp. 33–56 (2007)
7. Srivastava, N., Salakhutdinov, R.: Multimodal learning with deep boltzmann machines. In: Advances in Neural Information Processing Systems, vol. 25, pp. 2231–2239 (2012)

Author Index